

# Testeur certifié Testeur de jeux vidéo (CT-GaMe) Syllabus

Version 1.0.1

---

International Software Testing Qualifications Board

---



---

Fourni par  
Russian Software Testing Qualifications Board (RSTQB)

---



## Avis de copyright

Avis de copyright © International Software Testing Qualifications Board (ci-après dénommé ISTQB®).

ISTQB® est une marque déposée de l'International Software Testing Qualifications Board.

Tous droits réservés.

Copyright © 2022, les auteurs Andrey Konushin (chair), Alexander Alexandrov, Evgeny Glushkin, Dmitriy Karasev, Elena Karaseva, Elizaveta Kruchinina, Vadim Lukovaty, Dmitry Melishev, Maxim Nikolaev, Alexander Prokhorov, Anton Savvateev, Ayrat Sayfullov, Pavel Sharikov, Kirill Shevelev, Artyom Stukalov, Nikita Sysuev, Tatiana Tepaeva, Alexander Torgovkin, Margarita Trofimova, Yaroslav Vereshchagin, Svetlana Yushina, Lyubov Zhuravleva.

Les auteurs transfèrent par la présente le droit d'auteur à l'ISTQB®. Les auteurs (en tant que détenteurs actuels des droits d'auteur) et l'ISTQB® (en tant que futur détenteur des droits d'auteur) ont accepté les conditions d'utilisation suivantes:

- Des extraits de ce document peuvent être copiés, à des fins non commerciales, à condition que la source soit mentionnée. Tout organisme de formation accrédité peut utiliser ce syllabus comme base d'un cours de formation si les auteurs et l'ISTQB® sont reconnus comme la source et les détenteurs des droits d'auteur du syllabus et à condition que toute publicité d'un tel cours de formation ne puisse mentionner le syllabus qu'après avoir reçu l'accréditation officielle du matériel de formation de la part d'un Membre reconnu par l'ISTQB®.
- Tout individu ou groupe d'individus peut utiliser ce syllabus comme base pour des articles et des livres, à condition que les auteurs et l'ISTQB® soient reconnus comme la source et les détenteurs des droits d'auteur du syllabus.
- Toute autre utilisation de ce syllabus est interdite sans l'accord préalable et écrit de l'ISTQB®.
- Tout Membre reconnu par l'ISTQB® peut traduire ce syllabus à condition de reproduire l'avis de copyright susmentionné dans la version traduite du syllabus.

La traduction française est la propriété du CFTL – Comité Français des Tests Logiciels. Elle a été réalisée par un groupe d'experts en tests logiciels : Eric Riou du Cosquer, Olivier Denoo et Bruno Legeard.

## Historique des modifications

Version	Date	Remarks
v1.0.1FR	16 août 2023	Version française
v1.0.1	21 octobre 2022	Version approuvée par l'Assemblée Générale de l'ISTQB®

## Table des matières

Avis de copyright .....	2
Historique des modifications .....	3
Table des matières .....	4
Remerciements .....	6
0.Introduction.....	7
0.1 Objectif de ce document .....	7
0.2 Testeur de jeux vidéo certifié .....	7
0.3 Parcours de carrière pour les testeurs.....	7
0.4 Objectifs métiers .....	8
0.5 Objectifs d'apprentissage examinables et niveaux de connaissance.....	8
0.6 L'examen de certification pour Testeur de jeux vidéo .....	8
0.7 Accréditation .....	9
0.8 Prise en compte des normes .....	9
0.9 Rester à jour.....	9
0.10 Niveau de détail .....	9
0.11 Organisation du syllabus.....	9
1.Spécificité des tests de jeux vidéo – 75 minutes (K2).....	11
1.1 Les principes de base du test de jeux vidéo .....	11
1.1.1 Les spécificités des tests de jeux vidéo .....	11
1.1.2 Risques liés aux jeux vidéo .....	11
1.1.3 Défauts liés aux tests de jeux vidéo .....	12
1.1.4 Comment les tests atténuent les risques liés au produit dans les jeux vidéo .....	13
1.1.5 La différence entre tester et "jouer" .....	14
1.2 Rôles habituels de l'équipe de développement de jeux vidéo .....	15
1.3 Activités de test tout au long du cycle de vie du développement logiciel des jeux vidéo .....	16
2. Tester les mécaniques des jeux vidéo - 180 minutes (K3) .....	18
2.1 Mécaniques des jeux vidéo.....	18
2.1.1 Types de mécaniques de jeux vidéo .....	18
2.1.2 Différence entre tester les mécaniques du gameplay et tester les mécaniques non liés au gameplay .....	19
2.1.3 2.1.3 Différence entre tester les principales mécaniques et les méta-mécaniques.....	19
2.1.4 Différence entre tester les mécaniques client, serveur et client-serveurs .....	20
2.1.5 Exemples de défauts dans les mécaniques des jeux et causes possibles de leur apparition .....	22
2.2 Approches de test des mécaniques de jeux vidéo .....	23
2.2.1 Procédures et approches pour tester les mécaniques des jeux vidéo tout au long du cycle de vie du développement logiciel du jeu vidéo .....	23
2.2.2 L'importance de tester les mécaniques des jeux vidéo.....	24
2.2.3 L'importance de la revue des mécaniques des jeux vidéo .....	24
2.2.4 Tester l'état du jeu vidéo après la reprise de la session et lorsque l'utilisateur est inactif .....	24
3. Tests du graphisme - 165 minutes (K3) .....	28
3.1 Principes et concepts du graphisme des jeux vidéo .....	28
3.1.1 Caractéristiques du contenu graphique des jeux vidéo .....	28
3.1.2 Types de défauts du contenu graphique .....	32
3.2 Approches de test du graphisme dans les jeux vidéo .....	36
3.2.1 Tests des aspects artistiques .....	36
3.2.2 Tests techniques .....	37
3.2.3 Tests de gameplay .....	39
3.3 Réalisation des tests du graphisme .....	39
3.3.1 Réalisation des tests du graphisme à différents stades de la production du jeu vidéo .....	39
3.3.2 Tests du graphisme pour l'exactitude historique .....	41
3.4 Outillage des tests du graphisme.....	42

4. Tests du son - 190 minutes (K3)	43
4.1 Caractéristiques du contenu sonore du jeu vidéo	43
4.1.1 Types de sons	44
4.1.2 Effets et technologies du son	45
4.1.3 Zone des sons	46
4.2 Types de défauts dans les contenus sonores	46
4.3 Approches de test du contenu sonore dans les jeux vidéo	48
4.3.1 Tests auditifs du contenu	48
4.3.2 Tester le mélange de musique et de sons du jeu	49
4.3.3 Tester la composition de la musique	49
4.4 Réalisation des tests sonores	50
4.4.1 Niveaux de test du contenu audio-musical au cours du cycle du développement logiciel des jeux vidéo	50
4.4.2 Intégrer les sons dans le jeu	50
4.4.3 Domaines de responsabilité des intervenants	50
4.4.4 Procédures et approches de la conduite des activités de test dans le cadre du test d'objets sonores	51
4.5 Outillage pour les tests du son	51
5. Tests des niveaux de jeu - 65 minutes (K2)	53
5.1 Principes et concepts de la conception des niveaux de jeu	53
5.1.1 Le terme "niveau" et sa spécificité en fonction du genre du projet de jeu	53
5.1.2 Comprendre les types de défauts dans la conception des niveaux	54
5.2 Étapes et réalisation des tests de niveau de jeu	56
5.2.1 Étapes de base de la conception et du test des niveaux de jeu	56
5.2.2 Domaines de responsabilité des intervenants	58
5.3 Outillage des tests des niveaux de jeu	59
6. Tests des contrôleurs de jeux vidéo - 95 minutes (K2)	60
6.1 Principes et concepts des contrôleurs de jeux vidéo	60
6.1.1 Types de manettes de jeux vidéo	60
6.1.2 Défauts liés aux spécificités des manettes de jeux	62
6.2 Approches pour tester les contrôleurs dans les jeux vidéo	63
6.3 Outillage pour tester les contrôleurs de jeux vidéo	64
7. Tests de localisation - 155 minutes (K3)	65
7.1 Principes et concepts des tests de localisation	65
7.1.1 Localisation et internationalisation	65
7.1.2 Différence entre la localisation d'un jeu vidéo et d'un logiciel applicatif	67
7.1.3 Les étapes des tests de localisation	68
7.2 Types de défauts de localisation et leurs causes	72
7.2.1 Causes possibles de défauts dans la localisation du jeu vidéo	72
7.2.2 Défauts et risques liés à la localisation	72
7.3 Approche de test de localisation et réalisation des tests	73
7.3.1 Différence entre les tests complets et partiels de localisation	73
7.3.2 Procédures et approches pour tester la localisation au cours du cycle de vie du développement logiciel d'un jeu vidéo	74
7.3.3 Types de tests de localisation	76
7.4 Outillage pour les tests de localisation	77
8. Références	78
8.1 Standards	78
8.2 Documents ISTQB	78
8.3 Livres	78
8.4 Liens (Web/Internet)	79
9. Annexe A - Objectifs d'apprentissage/Niveau de connaissance	80
10. Annexe B - Matrice de traçabilité des objectifs métiers avec les objectifs d'apprentissage	81
11. Annexe C - Notes de livraison ("Release Notes")	84
12. Annexe D - Termes spécifiques aux tests de jeux vidéo et autres termes	85

## Remerciements

Ce document a été produit par une équipe du Russian Software Testing Qualifications Board (RSTQB) :

Andrey Konushin (President of the Board)  
Margarita Trofimova (Vice-president of the Board)  
Alexander Alexandrov (Editor-in-Chief)  
Vadim Lukovaty  
Maksim Nikolaev  
Pavel Sharikov  
Alexander Torgovkin  
Svetlana Yushina

L'équipe des auteurs remercie l'équipe de réviseurs pour leurs suggestions et contributions. Le Russian Software Testing Qualifications Board tient à remercier la société LLC "Bytex" pour sa contribution à l'élaboration de ce syllabus.

En outre, l'équipe des auteurs tient à remercier les responsables et les membres des groupes de travail de l'ISTQB® pour les conseils qu'ils lui ont prodigués dès le début et qu'ils continuent de lui prodiguer: Galit Zucker (General Secretary), Graham Bath (chair, Specialist Working Group), Gary Mogyorodi (member, Glossary Working Group) and Klaus Skafte (chair, Exam Working Group).

Les personnes suivantes ont participé à la revue, au commentaire ou au vote de ce syllabus ou de ses prédécesseurs:

Ivan Pchelkin	Natalia Sterkhova	Irina Yakovleva
Matthias Hamburg	Chunhui Li	Blair Mo
Tal Pe'er	Wim Decoutere	Claude Zhang
Meile Posthuma	Nishan Portoyan	Francisca Cano Ortiz
Wojciech Becla	Jana Gierloff	Paul Weymouth
Nitzan Goldenberg	Jürgen Beniermann	Erwin Engelsma
Georg Haupt	Gary Mogyorodi	Péter Sótér
Bíró Ádám	Darvay Tamás Béla	Laura Albert
Gergely Ágneicz	Graham Bath	Mike Smith

Ce document a été officiellement approuvé par l'assemblée générale de l'ISTQB® le 21 octobre 2022.

## 0. Introduction

### 0.1 Objectif de ce document

Ce syllabus forme la base du module spécialisé en Test de jeux vidéo de la qualification internationale en tests de logiciels. L'International Software Testing Qualifications Board (ISTQB®) et le Comité Français des Tests Logiciels (ci-après appelé CFTL) fournissent ce syllabus comme suit :

1. aux Membres de l'ISTQB®, afin de traduire dans leur langue locale et d'accréditer les fournisseurs de formation. Les Membres peuvent adapter ce syllabus à leurs besoins linguistiques particuliers et ajouter des références pour l'adapter à leurs publications locales.
2. aux organismes de certification, afin de produire les questions d'examen dans leur langue locale en fonction des objectifs d'apprentissage pour ce syllabus.
3. aux organismes de formation, afin de produire le matériel de formation et déterminer les méthodes pédagogiques appropriées.
4. aux candidats à la certification, pour se préparer à l'examen de certification (dans le cadre d'un cours de formation ou indépendamment).
5. à la communauté internationale de l'ingénierie des logiciels et des systèmes, pour faire progresser les pratiques professionnelles en tests de logiciels et de systèmes, et comme base pour des livres articles.

L'ISTQB® et le CFTL peuvent permettre à d'autres entités d'utiliser ce syllabus à d'autres fins, à condition qu'elles demandent et obtiennent une autorisation écrite préalable.

### 0.2 Testeur de jeux vidéo certifié

La qualification Testeur certifié de jeux vidéo s'adresse à toute personne impliquée dans le test de logiciel qui souhaite élargir ses connaissances en matière de tests de jeux vidéo ou à toute personne qui souhaite entamer une carrière spécialisée dans le domaine des tests de jeux vidéo. La qualification s'adresse également à toute personne impliquée dans l'ingénierie logicielle des jeux qui souhaite acquérir une plus grande facilité de compréhension des tests de jeux vidéo.

Le syllabus prend en compte les principaux aspects suivants des tests de jeux vidéo :

- Les aspects techniques,
- Les aspects méthodologiques,
- Les aspects organisationnels.

### 0.3 Parcours de carrière pour les testeurs

S'appuyant sur le niveau Fondation, la certification en tests de jeux vidéo soutient la définition de chemins de carrière pour les testeurs professionnels. Une personne titulaire de la certification Test en Jeux vidéo aura élargi la compréhension générale des tests acquise au niveau Fondation pour lui permettre de travailler efficacement en tant que testeur professionnel dans le cadre d'un projet de jeu.

Les personnes titulaires d'une certification de Test en Jeux vidéo peuvent utiliser l'acronyme CT-GaMe.

Pour obtenir les dernières informations sur le programme de certification des testeurs de l'ISTQB® et du CFTL, veuillez consulter les sites [www.istqb.org](http://www.istqb.org) et [www.cftl.fr](http://www.cftl.fr).

## 0.4 Objectifs métiers

Cette section liste les objectifs métiers attendus d'une personne ayant obtenu la certification en test de jeux vidéo.

- GaMe-1 Comprendre les concepts de base des jeux vidéo et des tests de logiciels de jeux.
- GaMe-2 Déterminer les risques, les objectifs et les exigences du logiciel de jeu en fonction des besoins et des attentes des parties prenantes.
- GaMe-3 Concevoir, implémenter et réaliser des tests de base des logiciels de jeux vidéo.
- GaMe-4 Connaître les approches des tests de jeux vidéo et leur objectif.
- GaMe-5 Reconnaître les outils soutenant les tests de jeux vidéo.
- GaMe-6 Déterminer comment les activités de test s'alignent sur le cycle de vie du développement logiciel et réduire le coût du développement et de la publication des jeux vidéo.

## 0.5 Objectifs d'apprentissage examinables et niveaux de connaissance

Les objectifs d'apprentissage soutiennent les objectifs métier et sont utilisés pour créer les examens Testeur Certifié en test de jeux vidéo. Chaque objectif d'apprentissage se réfère à un processus cognitif (niveau K).

Un niveau K, ou niveau cognitif, est utilisé pour classer les objectifs d'apprentissage selon la taxonomie révisée de Bloom [Anderson01]. L'ISTQB® utilise cette taxonomie pour concevoir ses syllabus d'examen (voir l'annexe A pour plus de détails).

Ce syllabus prend en compte trois niveaux différents (K1 à K3) :

K-level	Mot-clé	Description
1	Se souvenir	Le candidat doit se souvenir ou reconnaître un terme ou un concept.
2	Comprendre	Le candidat doit choisir une explication pour une instruction en rapport avec le thème de la question.
3	Appliquer	Le candidat doit sélectionner l'application correcte d'un concept ou d'une technique et l'appliquer à un contexte donné.

En général, toutes les parties de ce programme de formation contiennent des objectifs d'apprentissage pour le niveau K1. En d'autres termes, un candidat à une certification doit reconnaître, mémoriser et reproduire un terme ou un concept. Les objectifs d'apprentissage pour les niveaux K2 et K3 sont indiqués au début des chapitres qui les contiennent.

## 0.6 L'examen de certification pour Testeur de jeux vidéo

L'examen de certification de Testeur certifié en test de jeux vidéo est basé sur ce syllabus. Les réponses aux questions de l'examen peuvent nécessiter l'utilisation d'exigences basées sur plus d'une section de ce syllabus. Toutes les sections du syllabus sont examinables, à l'exception de l'introduction et des annexes. Les standards et les livres sont inclus comme références, mais leur contenu n'est pas examinable, au-delà de ce qui est résumé dans le syllabus lui-même à partir de ces standards et livres.

Se référer à [ISTQB\_EXAM\_S&R] pour la structure et les règles de l'examen de testeur certifié en test de jeux vidéo.

La certification ISTQB® Testeur certifié niveau Fondation [ISTQB\_FL\_SYL] doit être obtenue avant de passer l'examen de certification de testeur de jeux vidéo. Cependant, il est fortement recommandé que les candidats aient également :



- au moins une expérience minimale dans le développement ou le test de jeux vidéo, par exemple une année d'expérience en tant que testeur d'acceptation utilisateur d'un logiciel de jeu.
- suivi un cours qui a été accrédité selon les standards de l'ISTQB® (par l'un des membres reconnus par l'ISTQB®).

## 0.7 Accréditation

Un Membre de l'ISTQB® peut accréditer les organismes de formation dont le matériel de cours suit ce syllabus. Les organismes de formation doivent obtenir les directives d'accréditation auprès du Membre de l'ISTQB® ou de l'organisme qui effectue l'accréditation. Un cours accrédité est reconnu comme étant conforme à ce syllabus et peut comporter un examen de l'ISTQB®.

Les directives d'accréditation pour ce syllabus suivent les directives générales d'accréditation publiées par le groupe de travail sur la gestion des processus et la conformité (Processes Management and Compliance).

## 0.8 Prise en compte des normes

Certaines normes sont référencées dans le syllabus de base (par exemple, les normes IEEE ou ISO). Ces références fournissent un cadre (comme les références à la norme ISO 25010 concernant les caractéristiques de qualité) ou une source d'informations supplémentaires si le lecteur le souhaite. Les documents de normalisation ne sont pas destinés à être examinés. Pour plus d'informations sur les normes, voir le chapitre 8.

## 0.9 Rester à jour

Certaines normes sont référencées dans le syllabus de base (par exemple, les normes IEEE ou ISO). Ces références fournissent un cadre (comme les références à la norme ISO 25010 concernant les caractéristiques de qualité) ou une source d'informations supplémentaires si le lecteur le souhaite. Les documents de normalisation ne sont pas destinés à être examinés.

## 0.10 Niveau de détail

Le niveau de détail de ce syllabus permet de proposer des formations et des examens cohérents à l'échelle internationale. Pour atteindre cet objectif, le syllabus se compose :

- Des objectifs pédagogiques généraux décrivant l'intention du testeur de jeux vidéo certifié.
- D'une liste de termes que les étudiants doivent être capables de se rappeler,
- Des objectifs d'apprentissage pour chaque domaine de connaissance, décrivant le résultat d'apprentissage cognitif à atteindre,
- D'une description des concepts clés, y compris des références à des sources telles que la littérature ou les standards acceptés.

Le contenu du syllabus n'est pas une description de l'ensemble du domaine de connaissance du test de jeux vidéo ; il reflète le niveau de détail à couvrir dans les cours de formation au test de jeux vidéo de la certification testeur certifié en jeux vidéo. Il se concentre sur les domaines et les techniques de test qui peuvent s'appliquer à la plupart des projets de jeux vidéo.

## 0.11 Organisation du syllabus

Le syllabus du spécialiste certifié en tests de jeux vidéo contient sept chapitres couvrant les connaissances nécessaires pour être un spécialiste en tests de jeux vidéo.

L'en-tête de chaque chapitre précise la durée minimale du chapitre ; le timing n'est pas indiqué au-dessous du niveau du chapitre. Pour les formations accréditées, le syllabus exige un minimum de 15 heures et 25 minutes d'enseignement, réparties sur les sept chapitres comme suit :

- Chapitre 1: 75 minutes Spécificité des tests de jeux vidéo
  - Le testeur apprend les principes de base liés aux tests de logiciels de jeux vidéo, les raisons pour lesquelles les tests de jeux sont exigés, les objectifs des tests et la différence entre les tests de jeux vidéo et le fait de jouer au jeu.
  - Le testeur comprend le processus de test de jeux vidéo, les principales activités et les produits d'activités.
- Chapitre 2: 180 minutes Tester les mécaniques des jeux vidéo
  - Le testeur apprend à faire la distinction entre les différentes mécaniques du jeu et ses approches de test.
  - Le testeur apprend les méthodes dynamiques et statiques utilisées pour tester les mécaniques des jeux vidéo.
- Chapitre 3: 165 minutes Tests du graphisme
  - Le testeur se familiarise avec les éléments et les types de contenu graphique du produit de jeux vidéo et avec les meilleures pratiques d'exécution des tests graphiques, y compris le recours à des outils appropriés.
  - Le testeur connaît les domaines de responsabilité des personnes impliquées.
- Chapitre 4: 190 minutes Tests du son
  - Le testeur apprend à connaître les types et les technologies du contenu sonore du jeu vidéo, les meilleures pratiques d'exécution des tests sonores, y compris le recours à des outils appropriés.
  - Le testeur apprend les domaines de responsabilité des personnes impliquées.
- Chapitre 5: 65 minutes Tests des niveaux de jeu
  - Le testeur apprend que la conception des niveaux à tester et les approches de test dépendent du genre de jeu et des outils utilisés.
  - Le testeur connaît les domaines de responsabilité des personnes impliquées..
- Chapitre 6: 95 minutes Tests des contrôleurs de jeux vidéo
  - Le testeur apprend à connaître les types de manettes de jeux, y compris les défauts qui y sont liés et leurs causes possibles.
  - Le testeur apprend les approches de test des contrôleurs et de leur support à l'aide d'outils.
- Chapitre 7: 155 minutes Tests de la localisation
  - Le testeur apprend les bases et les différences de la localisation et de l'internationalisation, ses risques, les approches de test et les outils utilisés.
  - Le testeur apprend les domaines de responsabilité des personnes impliquées.

## 1. Spécificité des tests de jeux vidéo – 75 minutes (K2)

### Mots-clés du domaine des tests logiciels

Tests fonctionnels, tests en situation de jeu

### Mots-clés spécifiques du domaine des jeux vidéo

Modèle 3D, étape de conception, concepteur de jeux vidéo, multiplateforme, étape de post-production, étape de pré-production, étape de production, jeu vidéo

### Objectifs d'apprentissage pour le chapitre 1

#### 1.1 Les principes de base du test de jeux vidéo

GaMe-1.1.1 (K1) Reconnaître les objectifs et les spécificités des tests de jeux vidéo

GaMe-1.1.2 (K2) Donnez des exemples de risques liés au produit dans le domaine des jeux vidéo

GaMe-1.1.3 (K2) Donnez des exemples de défauts spécifiques liés aux tests de jeux vidéo

GaMe-1.1.4 (K2) Résumez comment les risques des tests de jeux vidéo peuvent être atténués

GaMe-1.1.5 (K2) Comparer les activités de test de jeux vidéo avec celles de jeu

#### 1.2 Rôles habituels des équipes de développement de jeux vidéo

GaMe-1.2.1 (K1) Reconnaître les rôles et les tâches spécifiques au sein de l'équipe de développement de jeux vidéo

#### 1.3 Activités de test tout au long du cycle de vie du développement logiciel des jeux vidéo

GaMe-1.3.1 (K1) Rappeler les activités de test tout au long du cycle de vie du développement logiciel des jeux vidéo

### 1.1 Les principes de base du test de jeux vidéo

#### 1.1.1 Les spécificités des tests de jeux vidéo

Dans le domaine des tests de logiciels, il est d'usage de distinguer différentes spécialisations. Certains testeurs s'occupent de tester la performance des applications, d'autres se concentrent sur les tests d'applications mobiles, d'autres encore recherchent des vulnérabilités dans les systèmes de sécurité, et d'autres enfin recherchent des défauts dans les jeux vidéo.

Les jeux vidéo peuvent varier en termes de genre de jeu et être destinés à un seul ou à plusieurs joueurs. Cependant, les approches de tests des jeux peuvent être considérées comme indépendantes de ces caractéristiques. La section 2.1 de ce syllabus décrit en détail les approches de tests fonctionnels et non fonctionnels des mécaniques de jeux.

Il est important de noter que les termes "jeux", "jeu vidéo" et "jeu informatique" sont traités comme des synonymes dans ce syllabus. Ceci afin de rendre le texte plus lisible et dans le contexte du syllabus. Un jeu ne peut être considéré que comme un logiciel et ne peut être confondu avec des jeux de hasard ou des paris sportifs.

#### 1.1.2 Risques liés aux jeux vidéo

Le développement explosif de l'industrie des jeux vidéo a conduit à leur distribution sur de nombreuses plateformes différentes. Les jeux sont devenus plus grands, plus complexes et plus exigeants en termes de ressources techniques. Dans le même temps, le nombre de joueurs a augmenté de manière significative, ce qui a entraîné une sophistication et une exigence accrues quant à la qualité des jeux.

Les risques courants liés aux projets et aux produits logiciels s'appliquent aux jeux vidéo et à leur développement. Par conséquent, les jeux, comme les autres logiciels, sont soumis à des tests. Certains

risques sont spécifiques au domaine des jeux vidéo et devraient faire l'objet d'une plus grande attention. Ces risques sont notamment les suivants :

- Possibilité d'obtenir un avantage déloyal en trichant dans le jeu
- Dépendance du succès commercial découlant de l'opinion subjective des joueurs
- Problèmes liés aux plateformes multiples
- Complexité de la prévision de l'efficacité des mécaniques des jeux vidéo (voir section 2.1).

Les testeurs peuvent constater des défauts dans les domaines suivants :

- Dans l'architecture du logiciel de jeux au début du cycle de vie du développement
- Dans la conception sonore
- Dans le texte d'un jeu vidéo déjà prêt pour la release.

### 1.1.3 Défauts liés aux tests de jeux vidéo

Un certain nombre de défauts sont particulièrement fréquents dans les jeux vidéo. Il s'agit notamment des défauts suivants :

- des problèmes de graphisme et d'animation
- le non-respect des lois physiques de l'univers des jeux vidéo.
- des défauts dans l'intelligence artificielle du jeu vidéo.
- des inexactitudes dans la construction de l'intrigue des jeux vidéo.
- le fonctionnement incorrect de certains éléments d'interface (par exemple, personnages de jeux vidéo émergeant de l'air, n'attaquant pas les ennemis, et voitures en vol stationnaire...).

Bien que ces défauts puissent être faciles à repérer, il est souvent difficile d'identifier les étapes exactes pour les reproduire. Cette complexité rend les tests de jeux vidéo difficiles. Un bon testeur doit concevoir des tests conformes au scénario du jeu prévu, vérifier qu'il répond aux spécifications, analyser les écarts possibles par rapport au scénario conçu et faire un rapport sur les conséquences de ces écarts.

### Tests négatifs

Lorsque l'on teste des jeux, comme pour n'importe quel autre logiciel, il est important d'identifier les défauts évidents et de s'efforcer de constater les défauts implicites qui apparaissent à la suite d'actions non standard effectuées par l'utilisateur. Par exemple, au lieu de combattre des adversaires et de chercher la clé d'une porte verrouillée, l'utilisateur peut collecter des boîtes dans tout le niveau, les empiler en forme d'escalier et escalader la clôture, évitant ainsi un combat difficile, ce qui n'était pas l'intention du jeu.

D'autres défauts peuvent survenir si l'interaction avec d'autres objets est mal réglée, par exemple pour un modèle 3D d'un bâtiment utilisé dans le jeu. Ce type de défaut peut permettre à un personnage contrôlé par le joueur de passer à travers un mur, ce qui peut gâcher l'expérience utilisateur globale et donner aux joueurs des avantages indus. Ce problème est particulièrement grave dans les jeux vidéo multi-joueurs. En se cachant derrière un tel mur, le joueur sera invisible pour ses adversaires, mais il pourra leur tirer dessus.

Les risques de ce type soulignent l'importance des tests négatifs pour les projets de jeux vidéo. Il peut y avoir un grand nombre de joueurs qui ne veulent pas jouer comme le développeur l'a prévu, mais qui essaient de "casser" ou de contourner le système afin de gagner plus facilement et à moindre effort. Pour ce faire, il n'est pas toujours nécessaire d'utiliser des logiciels tiers ou des commandes spéciales. Il peut suffire de constater le "bon" défaut. De tels joueurs recherchent délibérément des défauts dans la logique du jeu et les utilisent pour obtenir des avantages personnels, comme l'accès à des ressources infinies ou à des armes puissantes. Dans les jeux vidéo monétisés, cela pourrait également entraîner une réduction des bénéfices du propriétaire du jeu.

---

## Dépendance à l'égard de l'opinion des joueurs

Les logiciels de jeux vidéo, plus que tout autre, dépendent du jugement subjectif et souvent émotionnel des joueurs finaux. Alors que pour les logiciels conçus pour résoudre des problèmes métier, la fonctionnalité est le facteur le plus important, pour les logiciels de jeux vidéo, le facteur le plus important est le niveau d'intérêt de l'utilisateur et ses impressions sur la session de jeu. Les joueurs peuvent être capables de "fermer les yeux" sur certains défauts fonctionnels, mais si le jeu s'avère ennuyeux, monotone ou utilise des graphismes dépassés, les utilisateurs peuvent tout simplement cesser de l'utiliser. Une partie importante du public des jeux vidéo prend ses décisions d'achat ou d'utilisation d'un produit en se fondant principalement sur les commentaires des critiques de jeux, des évaluateurs et d'autres leaders d'opinion.

## Multiplateforme

Une caractéristique importante des logiciels de jeux vidéo est qu'ils sont souvent utilisés sur plusieurs plateformes. Afin de toucher un public aussi large que possible, les studios/éditeurs de jeux vidéo les publient sur une grande variété de plateformes, y compris diverses configurations d'ordinateurs personnels (PC), le web, les appareils mobiles et les consoles.

Chaque mise à jour doit être testée pour chaque plateforme disponible. Cela entraîne des risques importants et augmente considérablement la durée des tests.

Le jeu peut très bien fonctionner sur un ordinateur de milieu de gamme, mais présenter une grande variété de défauts sur la dernière génération de consoles. Des défauts sont également possibles en raison d'une communication inefficace entre les différentes technologies ou d'exigences incomplètes pour le portage d'un jeu de la plateforme pour laquelle il a été développé à l'origine vers une autre plateforme appartenant à une organisation tierce (externalisée).

Pour atténuer ces risques, il est recommandé de consacrer plus de temps au développement et au test du logiciel de jeu, de tester les fonctionnalités et les performances du jeu sur de nombreuses configurations matérielles différentes, et d'effectuer des tests basés sur les caractéristiques spécifiques de chaque plateforme.

Tester les jeux vidéo sur des consoles de jeu est un aspect particulier à prendre en compte. Une console de jeu est un appareil conçu exclusivement pour les jeux vidéo et ne contient aucun autre logiciel, comme les appareils mobiles ou les ordinateurs.

Il y a très peu de différences entre les plateformes lorsqu'il s'agit de tester des jeux vidéo à l'aide de techniques de test boîte noire. Cependant, chaque fabricant de console de jeu peut avoir ses propres exigences spécifiques auxquelles un jeu vidéo doit se conformer avant d'être publié. Ces exigences sont des documents exclusifs fournis aux développeurs et aux éditeurs dans le cadre d'accords de confidentialité. Chaque liste d'exigences se compose de plusieurs catégories, et le jeu doit s'y conformer pour éviter d'être rejeté par le fabricant de la console [URL1], [URL2].

Par conséquent, les testeurs de jeux vidéo sur consoles doivent tester la conformité à ces exigences en plus des méthodes standard de test des logiciels.

Il peut être nécessaire d'utiliser un équipement spécial pour détecter, identifier et comprendre les raisons pour lesquelles des défauts se manifestent. Cet équipement est essentiellement la même console, mais il fournit des modes supplémentaires qui aident à développer et tester les jeux vidéo. La console est enregistrée sur le site pour les développeurs et testeurs autorisés, après quoi elle est activée et l'accès aux modes spéciaux est accordé.

### 1.1.4 Comment les tests atténuent les risques liés au produit dans les jeux vidéo

La plupart des jeux vidéo, en particulier les jeux en monde ouvert, sont si complexes qu'il est impossible de tester toutes les combinaisons possibles d'objets, d'événements et de facteurs du jeu. Même la

réalisation de tests uniques de chaque combinaison pendant tout le cycle de vie du développement logiciel pourrait prendre beaucoup de temps, et les tests de confirmation après les modifications apportées à la version suivante du jeu nécessitent des efforts supplémentaires.

La gestion des risques permet d'allouer les ressources de manière efficace et d'évaluer les défauts possibles en fonction de leur probabilité d'occurrence et de leur impact sur le jeu.

Pour chaque cas de test, deux questions sont posées pour déterminer la priorité du test :

1. Quelle est la probabilité que les joueurs constatent un défaut ici ?
2. S'ils constatent un défaut, quelles seront les conséquences pour les joueurs et pour l'entreprise propriétaire du jeu ?

Par exemple, si un jeu a une structure linéaire et que l'utilisateur accède à son contenu étape par étape, les défauts qui apparaissent au cours de la première session de jeu auront très probablement une plus grande priorité (mais pas toujours une plus grande sévérité) à être corrigés que les défauts qui apparaissent à la cinquantième heure de jeu. Si un utilisateur fidèle a déjà passé beaucoup de temps dans le jeu, il est plus susceptible de se montrer plus clément face au défaut rencontré, et celui-ci n'aura pas beaucoup d'effet sur l'impression générale que l'utilisateur a du jeu. Il en va autrement si un tel défaut touche tous les nouveaux joueurs lors du tutoriel d'introduction. Dans ce cas, de nombreux joueurs risquent de rejeter le jeu vidéo.

Pour réduire les risques en testant, il est important d'identifier des groupes de défauts plutôt que de rechercher des défauts individuels dans le jeu. Cela permet aux développeurs de résoudre simultanément un problème dans le code ou l'architecture du jeu et de fournir une amélioration de la qualité du produit final.

Les groupes de défauts sont des regroupements de défauts dans des domaines critiques. Il existe des domaines dans les produits de jeux qui sont susceptibles de présenter des défauts, tels que les graphiques, le son, la localisation, l'interaction client-serveur et le matériel [Nystrom14].

Il est important de fidéliser les joueurs au jeu lors de la première prise en main ou lorsqu'ils suivent un tutoriel. À ce stade, un très grand nombre de fonctions du jeu sont disponibles pour le joueur afin qu'il puisse évaluer le produit. C'est pourquoi ce stade précoce est généralement déclaré comme étant d'une importance cruciale.

#### 1.1.5 La différence entre tester et "jouer"

On pourrait avoir l'impression que l'essence du métier de testeur de jeux vidéo consiste à jouer en ligne avec ses collègues toute la journée. Mais cette affirmation est loin de la vérité.

Le testeur parvient rarement à se contenter de jouer, sauf lorsqu'il se familiarise avec un nouveau projet ou lors de tests en situation de jeu. La majeure partie de son travail consiste à effectuer divers types de tests qui, à bien des égards, ont en commun avec les types de tests utilisés pour d'autres logiciels.

L'utilisateur moyen commence le jeu pour le terminer, battre ses adversaires ou passer un bon moment, alors que le testeur vérifie que le jeu répond aux exigences écrites dans le cahier des charges. En outre, les préférences personnelles du testeur peuvent ne pas correspondre au public cible du jeu lui-même. Néanmoins, le testeur parcourra le même niveau encore et encore pour tester, par exemple, que les objets du jeu s'affichent correctement. La monotonie ressentie par le testeur lors de ces actions répétées est compensée par la satisfaction qu'il éprouve à constater des défauts susceptibles de perturber le gameplay conçu par le développeur.

## Relation avec d'autres syllabi

Les approches discutées dans ce syllabus s'appliquent exclusivement aux tests de jeux vidéo. Elles s'appliquent à la fois aux tests de logiciels (par exemple, les tests de mécaniques de jeux) et aux tests de matériel (par exemple, les tests de contrôleurs).

Certains aspects du test d'un jeu ne sont pas si différents du test d'autres logiciels, en particulier lors de l'exécution de types de tests tels que les tests de régression, les tests de performance ou les tests d'utilisabilité. Les concepts de tests fondamentaux, les techniques de tests non fonctionnels et les caractéristiques des tests de logiciels sur les plateformes mobiles sont couverts en détail dans d'autres syllabi de l'ISTQB®.

## 1.2 Rôles habituels de l'équipe de développement de jeux vidéo

Dans la plupart des cas, une équipe de développement de jeux vidéo comprend (mais n'est pas limitée à) des rôles qui ne sont pas fondamentalement différents de ceux constatés dans d'autres domaines du développement de logiciels. Il s'agit notamment des B.O.s (business-owners) ou responsables métier, des gestionnaires de projet, des analystes, des développeurs, des concepteurs de jeux et des tests. Dans les petites équipes, une personne peut assumer plusieurs rôles en même temps. Voici un résumé des principales fonctions que peuvent remplir les rôles habituels dans un projet de logiciel de jeux vidéo. Les rôles supplémentaires qui peuvent être inclus dans une équipe pour des domaines particuliers du développement de jeux vidéo sont décrits dans les chapitres appropriés de ce syllabus.

Rôle	Responsabilités
Responsable métier	<ul style="list-style-type: none"> <li>Développe les aspects métier</li> <li>Définit la politique en matière de marketing</li> <li>Lance de nouveaux produits</li> <li>Peut participer à la constitution d'une équipe</li> </ul>
Chef de projet	<ul style="list-style-type: none"> <li>Définit et surveille l'exécution des tâches</li> <li>Suit les étapes du cycle de vie du développement logiciel</li> </ul>
Analyste	<ul style="list-style-type: none"> <li>Identifie les exigences du produit</li> <li>Travaille avec la spécification originale</li> <li>Met à jour les spécifications, les tient à jour</li> </ul>
Développeur	<ul style="list-style-type: none"> <li>Développe et débogue le code</li> <li>Corrige les défauts</li> <li>Développe des tests de composants</li> </ul>
Concepteur de jeux vidéo	<ul style="list-style-type: none"> <li>Conception des mécaniques de jeu, de l'équilibre, de l'intrigue, des niveaux et de l'économie du jeu.</li> <li>Création de la documentation des jeux vidéo</li> <li>Conçoit et développe le contenu des jeux vidéo</li> </ul>
Testeur	<ul style="list-style-type: none"> <li>Élabore un modèle de test et une stratégie de test</li> <li>Conçoit et prépare les données de test</li> <li>Révisé les spécifications.</li> <li>Effectue différents types de tests.</li> <li>Constata et analyse les défauts</li> </ul>

## 1.3 Activités de test tout au long du cycle de vie du développement logiciel des jeux vidéo

### Phase de conception

L'objectif principal de cette étape est de parvenir à une vision commune du futur produit entre l'équipe de développement et le client. Les idées sont élaborées sous la forme de documents qui décrivent brièvement la vision générale du jeu, son genre, son cadre, l'essence du gameplay, les mécaniques et ses principales caractéristiques.

Les artefacts susceptibles d'être révisés à ce stade sont les suivants :

- Document de vision,
- Document de conception,
- Restrictions techniques et de produits.

Le testware produit à ce stade comprend :

- La stratégie de test,
- L'ébauche du plan de test,
- L'estimation de haut niveau de l'effort de test,
- L'environnement de test.

### Phase de pré-production

L'objectif principal de cette étape est de créer un prototype d'un futur jeu qui est une première version de travail du produit. Le prototype implémente le gameplay principal en qualité d'ébauche. À ce stade, les principales mécaniques du jeu et les hypothèses initiales sont testés (voir le chapitre 2) et les capacités techniques de l'équipe sont évaluées. Les testeurs, les concepteurs de jeux et les autres membres de l'équipe de développement peuvent participer aux tests du prototype.

L'équipe de développement révisé les projets d'exigences pour établir la possibilité d'implémentation et l'équipe de test les révisé pour identifier l'interaction des mécaniques entre elles et la fonctionnalité du prototype assemblé. En outre, les testeurs participent à la création des documents relatifs au plan de test et au calendrier des tests.

Au fur et à mesure que la documentation du jeu est créée, le plan de test est rédigé et mis à jour, et la documentation est révisée. Ces mesures permettent d'atténuer les risques techniques du produit au stade de la pré-production.

Les éléments qui peuvent être révisés à ce stade sont les suivants :

- Spécifications des exigences,
- Prototype de jeu,
- Artefacts de test créés à l'étape précédente.

Le testware produit et mis à jour à ce stade comprend :

- Stratégie de test ,
- Plan de test,
- Calendrier des tests,
- Environnement de test.



---

## Phase de production

Au cours de la phase de production, le jeu vidéo est créé. C'est l'étape la plus longue et elle est généralement divisée en parties distinctes pour les versions Alpha et Beta du jeu.

À ce stade, les testeurs finalisent les listes de contrôle et les ensembles de conditions de test, et exécutent les tests fonctionnels et non fonctionnels des composants implémentés du produit. Les tests d'intégration des composants, les tests système et les tests d'acceptation sont effectués (voir [ISTQB\_FL\_SYL]). C'est à ce stade que le plus grand nombre de défauts est identifié, ce qui amène les testeurs à effectuer des tests de confirmation et des tests de régression.

Le testware produit ou mis à jour à ce stade comprend :

- Plan de tests ,
- Calendrier des tests,
- Cas de test, conditions de test, listes de contrôle, suite de tests, scripts,
- Rapports de défaut,
- Rapports de test.

## Phase de post-production

L'étape de post-production pour les tests de jeux vidéo comprend les activités de test courantes pendant la période de maintenance du logiciel. Il s'agit habituellement de tester les mises à jour et les tests de régression, comme décrit dans [ISTQB\_FL\_SYL].

## 2. Tester les mécaniques des jeux vidéo - 180 minutes (K3)

### Mots-clés du domaine des tests logiciels

Tests ad-hoc, tests fonctionnels

### Mots-clés spécifiques du domaine des jeux vidéo

Mécaniques du client, étape de conception, mécaniques principales, concepteur de jeux vidéo, mécaniques du jeu vidéo, méta-mécaniques, étape de pré-production, étape de production, mécaniques du serveur, état du jeu vidéo

### Objectifs d'apprentissage pour le chapitre 2

#### 2.1 Mécanique des jeux vidéo

GaMe-2.1.1 (K2) Classer les types de mécaniques de jeux vidéo

GaMe-2.1.2 (K2) Faire la différence entre tester les mécaniques de gameplay et les mécaniques non liées au gameplay.

GaMe-2.1.3 (K2) Différencier les tests des principales mécaniques et des méta-mécaniques.

GaMe-2.1.4 (K2) Différencier les tests des mécaniques client, serveur et client-serveur.

GaMe-2.1.5 (K2) Donnez des exemples de défauts dans les mécaniques des jeux vidéo

#### 2.2 Approches de test des mécaniques de jeux vidéo

GaMe-2.2.1 (K2) Résumer les principales approches et les principaux objets de test à différents stades de la création d'un jeu.

GaMe-2.2.2 (K2) Distinguer l'importance de tester les mécaniques des jeux vidéo

GaMe-2.2.3 (K2) Distinguer l'importance de réviser la documentation décrivant les mécaniques des jeux vidéo

GaMe-2.2.4 (K3) Appliquer les approches fondamentales pour tester les mécaniques des jeux vidéo

### 2.1 Mécaniques des jeux vidéo

#### 2.1.1 Types de mécaniques de jeux vidéo

Le chapitre 1 a mentionné que pour les jeux, l'une des choses les plus importantes est de susciter l'intérêt de l'utilisateur afin de fidéliser et d'accroître l'audience. Les joueurs évaluent les graphismes du jeu, le gameplay lui-même et les mécaniques qui la composent.

Les mécaniques des jeux vidéo sont au cœur de tout jeu. Les mécaniques des jeux vidéo sont une méthode d'interaction entre le jeu et l'utilisateur. Elle prend en compte l'impact et le retour d'information entre le jeu vidéo et l'utilisateur, ainsi que les modifications de l'état du jeu vidéo dans le cadre des exigences spécifiées. Les mécaniques peuvent être divisés en différents types, en fonction des différents aspects et objectifs du jeu.

Selon le type d'interaction du joueur avec les mécaniques :

- Les mécaniques de gameplay se rapportent au moment où l'utilisateur interagit consciemment avec le système de jeu, en fondant ses actions sur la disponibilité des informations. Il peut influencer le changement d'état du jeu vidéo, tout en voyant clairement le résultat de ses actions.
- Les mécaniques non liés au gameplay se rapportent au moment où l'utilisateur ne peut pas ou ne peut que partiellement influencer l'état de l'environnement de jeu vidéo par ses actions. Ils ne reconnaissent pas leur influence, car une partie de la mécanique des jeux vidéo leur est cachée.

En fonction de l'impact sur le gameplay principal :

- Les principales mécaniques constituent la base du jeu et définissent les actions que l'utilisateur répète tout au long du jeu. Elles visent à garantir à l'utilisateur une certaine expérience qui a été intégrée au jeu par ses créateurs.
- Les méta-mécaniques se situent en dehors du gameplay principal mais peuvent l'influencer. Elles visent à faire en sorte que le joueur fasse ce que la conception du jeu veut (par exemple, revenir dans le jeu, continuer à jouer ou faire des achats). Ces mécaniques peuvent être combinées avec les mécaniques principales pour rendre le jeu plus intéressant.

En fonction de l'architecture de la structure du jeu :

- La mécanique client concerne le traitement des actions de l'utilisateur et se produit exclusivement sur l'appareil de l'utilisateur.
- Les mécaniques de serveur se rapportent aux mécaniques qui sont traitées uniquement sur le serveur de jeux.
- Les mécaniques client-serveur concernent les mécaniques qui sont traitées à la fois sur l'appareil de l'utilisateur et sur le serveur. Dans ce cas, il y a un échange constant de données entre le serveur et le client.

2.1.2 Différence entre tester les mécaniques du gameplay et tester les mécaniques non liés au gameplay  
L'utilisateur n'interagit pas directement avec tous les mécaniques implémentées dans le jeu.

Les mécaniques de gameplay peuvent être décrits dans une spécification formelle ou informelle, mais les mécaniques non liées au gameplay doivent être décrites à l'utilisateur. Souvent, elles affectent directement la réalisation de l'objectif du gameplay. Par exemple, un personnage de jeu doit ramasser des pièces dans un temps limité. La fonctionnalité du mouvement de ramassage des pièces et l'augmentation du nombre de pièces sont des exemples de mécaniques du jeu.

Les mécaniques non liés au gameplay sont généralement cachées à l'utilisateur, bien qu'elles puissent être déclenchées par l'action de l'utilisateur lui-même. Par exemple, lorsque l'utilisateur effectue un achat de n'importe quel produit dans un jeu vidéo, un flux séquentiel de mécaniques non liés au jeu est lancé. Par exemple :

1. Une requête est adressée à la base de données pour obtenir l'état du compte de jeux vidéo de l'utilisateur.
2. Une transaction est effectuée pour retirer des fonds
3. L'achat est logué, c'est-à-dire que les informations relatives à l'action effectuée sont enregistrées dans un fichier texte spécial
4. Un courriel contenant les détails de l'achat est envoyé à l'utilisateur.

En outre, la date d'achat et le type de produit acheté peuvent être enregistrés, de sorte qu'un courrier automatique concernant une réduction pour un produit similaire sera envoyé. Les mécaniques non liées au gameplay doivent être spécifiées en tant qu'exigences formelles car elles ne sont pas évidentes.

Les tests des mécaniques liées au gameplay et non liées au gameplay sont des tests fonctionnels, tels que décrits dans [ISTQB\_FL\_SYL].

2.1.3 2.1.3 Différence entre tester les principales mécaniques et les méta-mécaniques

La différence d'approche pour tester les mécaniques principales et les méta-mécaniques dépend de leur impact.

La suppression d'une mécanique principale du jeu modifie complètement le gameplay. Par exemple, si le mouvement de la voiture est supprimé d'un simulateur de course, le jeu ne peut plus être qualifié de "course".

Des exemples de méta-mécaniques seraient le choix d'un modèle de voiture avec des paramètres spécifiques en fonction du type de piste de course, ou l'achat et l'équipement d'un personnage jouable avec une nouvelle armure avant d'entamer une bataille. La suppression d'un méta-mécanique du jeu change rarement l'essence du jeu tout entier.

Outre les exigences fonctionnelles, les mécaniques principales et les méta-mécaniques peuvent avoir d'autres exigences non fonctionnelles qui affectent directement la manière dont ils doivent être testés.

Par exemple, pour une méta-mécanique permettant de modifier l'apparence d'un personnage jouable, le plus important est qu'elle fonctionne correctement. Même si cette mécanique connaît un léger retard, de sorte que l'apparence change quelques secondes après que le joueur a appuyé sur le bouton, ni le testeur ni l'utilisateur ne considéreront qu'il s'agit d'un défaut. Cependant, pour la mécanique principale d'un personnage qui saute ou utilise une capacité lors d'un combat, un tel retard serait critique, car il pourrait empêcher le joueur de gagner. Par conséquent, les tests de performance sont également importants pour les mécaniques principales.

L'essence des principales mécaniques et des méta-mécaniques influe également sur le moment où elles sont testées. En règle générale, leur vérification commence à différents stades du développement des jeux vidéo. Les mécaniques principales, telles que le mouvement d'un personnage de jeu, la réception d'une récompense, etc. apparaissent déjà dans les premières versions du produit. Parallèlement, la vérification de l'exactitude fonctionnelle et du niveau d'intérêt du public cible commence.

Les méta-mécaniques sont généralement ajoutées au jeu et commencent à être testées plus tard, pendant la phase de production. L'évaluation de leur impact sur la durée moyenne d'une session de jeu, le nombre d'amis invités, le montant du paiement moyen, les produits de jeu les plus populaires, etc. a lieu au stade du test bêta. Parallèlement, les tests A/B sont souvent utilisés pour déterminer les mécaniques les plus efficaces.

#### 2.1.4 Différence entre tester les mécaniques client, serveur et client-serveurs

La nécessité de tester les mécaniques côté client, côté serveur et client-serveur dépend de l'architecture du jeu lui-même. Chacun de ces types de mécaniques nécessite une approche de test différente en fonction des spécificités de leur implémentation.

Les **mécaniques côté client** sont traitées sur les appareils des joueurs eux-mêmes, de sorte que l'utilisateur peut accéder à toutes les données stockées sur le client. Certains jeux peuvent être entièrement exécutés sur le client et n'utilisent pas du tout de serveur, c'est-à-dire que toutes les mécaniques sont du côté client. Ces jeux sont généralement des jeux à un seul joueur et ne nécessitent pas de connexion internet.

Le client du jeu contient le code qui est exécuté sur l'appareil du joueur et tous les paramètres, les formules de calcul des dégâts, les niveaux de jeu, les modèles de personnages, etc. Par conséquent, le joueur peut, s'il le souhaite, modifier toutes les données du client de jeu. En modifiant le client, l'utilisateur peut, par exemple, augmenter la vitesse de sa technique, la quantité de santé du personnage, la valeur de la récompense pour l'accomplissement de la tâche, etc.

De telles modifications du client peuvent donner au joueur un avantage dans le jeu, mais dans les jeux à un seul joueur, cela n'est généralement pas considéré comme un défaut majeur. Par exemple, dans un jeu vidéo à un seul joueur, les mécaniques d'un combat à l'épée avec des adversaires informatiques peuvent être implémentées. Dans ce cas, les paramètres de l'arme ne seront pas validés sur le serveur. Par conséquent, si un joueur particulier effectue une modification sur le client du jeu et augmente les dégâts de son épée, cela ne changera que son gameplay personnel. Cela n'affectera pas les intérêts des autres joueurs indépendants qui jouent au même jeu sur leurs appareils.

Les mécaniques du client peuvent également être présentes dans les jeux vidéo multijoueurs, mais elle n'affecteront toujours pas les autres joueurs. Par exemple, les mécaniques de visualisation d'une

cartographie de niveau, d'ouverture de l'inventaire d'un personnage, etc. sont traitées uniquement du côté client et vérifiées par des tests fonctionnels sans implication du serveur.

Le test de la mécanique client est généralement effectué à l'aide d'un test boîte noire. Ce faisant, le testeur peut être amené à vérifier l'interface utilisateur de l'application. Dans ce cas, l'interface utilisateur elle-même peut être utilisée pour tester le jeu et obtenir des résultats réels.

**Les mécaniques du serveur** sont implémentées exclusivement du côté du serveur. Grâce à cette caractéristique, la logique de la mécanique est protégée de l'intervention de l'utilisateur. Les joueurs ne peuvent pas influencer directement les processus qui s'exécutent sur le serveur distant, c'est pourquoi les développeurs y transfèrent les parties les plus importantes des calculs de la logique du jeu.

Tester les mécaniques du serveur est particulièrement important pour les jeux vidéo multijoueurs en ligne. Par exemple, lorsqu'un joueur commence une compétition en équipe en mode joueur contre joueur (JcJ), une sélection d'alliés et d'adversaires est effectuée sur le serveur avec des cotes de jeu et des niveaux de personnage approximativement égaux. Dans le cas contraire, la compétition pourrait s'avérer inintéressante et inconfortable pour tous les participants.

L'approche de test des fonctionnalités implémentées sur le serveur est différente de celle consistant à tester les mécaniques du client. Ici, le testeur n'a généralement pas besoin d'une interface utilisateur. Les tests sont effectués sur des consoles de serveur ou à l'aide d'outils spéciaux. Le testeur passe le plus clair de son temps à analyser les logs ou à interagir avec la base de données.

Pour la mécanique serveur, les types de tests les plus importants sont les tests fonctionnels, les tests de performance et les tests de sécurité.

Les **mécaniques client-serveur** impliquent à la fois le côté client et le côté serveur et combinent les caractéristiques des deux types de mécaniques mentionnées précédemment. Une partie de la logique est traitée sur les appareils des joueurs, l'autre sur le serveur distant. Les parties de ces mécaniques "communiquent" constamment entre elles, en échangeant des données. Dans un jeu vidéo multijoueur, le serveur reçoit des données du client, puis envoie le résultat des actions à tous les joueurs du serveur qui sont concernés par ces actions.

Les données qui arrivent au serveur subissent une validation obligatoire. Par exemple, si un utilisateur veut acheter quelque chose dans la boutique du jeu, le serveur doit vérifier qu'il dispose d'une quantité suffisante de monnaie du jeu disponible pour cet achat, en fonction de la progression du joueur. Même si le joueur tente de modifier la partie client de son appareil et d'envoyer de faux paramètres au serveur afin d'obtenir un avantage, le serveur les remplacera par des données correctes.

Lorsque les testeurs testent les mécaniques client-serveur, ils créent un ou plusieurs cas de test de manière à couvrir le plus grand nombre possible de mécaniques différentes.

Un exemple d'un tel test serait qu'un client se connecte à un serveur de jeux et tue un adversaire contrôlé par l'ordinateur. Malgré le petit nombre d'actions nécessaires qui composent ce test, presque toutes les mécaniques de l'interaction client-serveur sont utilisées ici:

- Envoi d'une requête du client au serveur chargé de l'autorisation,
- Envoi d'une requête du serveur chargé de l'autorisation à la base de données, l'exécution de l'autorisation et le rétablissement de l'état précédent du jeu vidéo,
- Sélection des cartographies déjà chargées sur le serveur ou création de nouvelles mécaniques de jeu sur le service,
- Connexion du client au serveur de mécaniques de jeu où se trouve la cartographie,
- Sélection des cartes déjà chargées sur le serveur ou création de nouvelles mécaniques de jeu sur le serveur,
- Connexion du client au serveur de mécaniques de jeu où se trouve la carte,
- Affichage d'une vue du personnage du jeu et des objets du jeu à proximité du personnage pendant que le serveur crée des objets et donne au client l'ordre d'afficher ceux-ci à l'utilisateur,

- Déplacement du personnage sur la carte tout en affichant de nouveaux objets de jeu,
- Constatation d'un ennemi informatique et de son attaquant et affichage de l'attaque sur le client.
- Réception de la commande "attaquant" du serveur,
- Vérification qu'une attaque est possible en s'assurant que le personnage et l'objet mobile (Mob) sont à une distance acceptable pour une attaque, et qu'il n'y a pas d'objets entre le personnage et le Mob empêchant l'exécution de l'attaque,
- Lorsqu'une attaque est possible, envoi par le serveur d'un message au Mob concernant la réception de dommages.
- Calcul du montant des dégâts reçus par le Mob sur le serveur, réduction de la valeur du paramètre responsable de la santé du Mob,
- Affichage des dégâts infligés au joueur par le Mob.

#### 2.1.5 Exemples de défauts dans les mécaniques des jeux et causes possibles de leur apparition

Les mécaniques du jeu impliquent des influences sur les objets du jeu et un retour d'information signalant le résultat de ces influences. L'ensemble de ces éléments crée le caractère unique du jeu, avec une dynamique de jeu appropriée. Le plus souvent, les jeux utilisent un large éventail d'influences et d'éléments de rétroaction.

Les types de défauts suivants sont associés aux mécaniques de jeu :

- Défauts fonctionnels au niveau de la mécanique,
- Défauts de reconnaissance de la pertinence du jeu,
- Défauts d'efficacité des mécaniques se manifestant dans un environnement de jeu spécifique.

Les défauts fonctionnels du jeu sont les plus mentionnés lorsque les joueurs parlent de défauts dans les mécaniques du jeu. Par exemple, les armes ne se rechargent pas, le butin n'est pas collecté, l'image ne se rapproche pas lorsque l'on utilise des jumelles. Ces défauts proviennent généralement d'erreurs commises par le développeur dans le code du jeu. Ils sont assez visibles et faciles à détecter et à corriger.

Les défauts de reconnaissance de pertinence du jeu sont liés à l'obtention de réponses de la part du jeu lors de l'utilisation d'une mécanique particulière. La situation est incorrecte lorsque le joueur ne comprend pas pourquoi le jeu est terminé. C'est pourquoi les mécaniques des jeux sont souvent accompagnées d'effets visuels et sonores. Il peut s'agir de l'effet visuel d'une explosion, du son d'un compteur de temps ou même d'un message texte. La présence même d'une réponse lors de l'utilisation d'une mécanique est testée et la justification de sa présence ou de son absence est évaluée. Le fonctionnement et l'exactitude de l'effet sont testés par d'autres types de tests (tests graphiques ou sonores)..

Le troisième type de défaut est lié à l'efficacité et au périmètre des mécaniques. Les mécaniques peuvent être efficaces en elles-mêmes, mais peuvent cesser de fonctionner dans le cadre du gameplay. Pour éviter une telle situation, les mécaniques sont testées en conjonction avec d'autres mécaniques, objets et autres contenus nécessaires au niveau du jeu. Les tests d'intégration sont donc effectués en même temps que les tests d'utilisabilité pour vérifier l'efficacité des mécaniques dans le cadre du jeu.

Par exemple, un concepteur de jeu peut ajouter au jeu deux adversaires ayant des comportements et des caractéristiques différents. Le premier est rapide et nerveux, mais inflige peu de dégâts, tandis que le second est lent et lourd, mais frappe avec précision et force. En termes de caractéristiques, les deux adversaires se valent. Ils ont leurs propres avantages et inconvénients. Chacun peut avoir son propre algorithme de comportement, et en théorie, ils devraient représenter à peu près le même degré de danger pour le joueur. Cependant, si une grande partie du niveau de jeu est constituée de diverses collines que le personnage du joueur peut escalader, la situation change. Se trouvant à une hauteur inaccessible à un ennemi lent, le joueur peut l'attaquer en toute sécurité. Dans ce cas, l'utilisateur aura des difficultés à affronter un autre ennemi, qui peut également utiliser les collines, mais qui se déplace à grande vitesse et

saute haut. La solution à ce problème peut être à la fois un changement dans la mécanique elle-même, et un changement dans l'emplacement des objets sur le niveau.

La complexité de ces tests est principalement liée à la difficulté de tester de telles situations et la façon dont la mécanique y fonctionnera. C'est pourquoi la recherche de problèmes d'interaction entre les mécaniques et les objets de l'environnement est souvent effectuée dans le cadre de tests ad-hoc au stade des tests bêta avec un grand groupe de joueurs.

## 2.2 Approches de test des mécaniques de jeux vidéo

### 2.2.1 Procédures et approches pour tester les mécaniques des jeux vidéo tout au long du cycle de vie du développement logiciel du jeu vidéo

À différents stades du développement d'un logiciel, le test des mécaniques des jeux vidéo peut être effectué à l'aide de diverses approches de test.

Au stade de la création d'un prototype de jeu, seuls les principales mécaniques sont implémentées. Dans ce cas, les principales tâches du testeur consistent à tester les mécaniques du jeu en fonction des caractéristiques suivantes :

- L'exactitude fonctionnelle,
- Le niveau d'intérêt pour les joueurs,
- L'attractivité pour les joueurs

Toutes les mécaniques et leurs règles d'exploitation sont revues afin de s'assurer qu'elles sont décrites de manière complète et sans ambiguïté dans le document de conception du jeu vidéo. Toutes les options autorisées pour l'interaction de l'utilisateur avec les mécaniques doivent être élaborées, et l'impact de chaque mécanique sur l'ensemble du gameplay doit être déterminé.

Au cours de la phase de production, les mécaniques implémentées sont soumises à des tests fonctionnels afin de vérifier qu'elles répondent aux exigences énoncées. En règle générale, les tests sont effectués à l'aide de conditions de test et de listes de contrôle. La séquence des tests doit viser une couverture maximale de tous les aspects de la mécanique. Les tests exploratoires et les tests ad-hoc sont plus efficaces pour tester l'interaction des différentes mécaniques entre elles. Cela est particulièrement évident dans les jeux multijoueurs à grande échelle, où le nombre de modes d'interaction entre les mécaniques et les divers composants du jeu devient trop important pour être pris en compte dans les conditions de test.

Dans le cadre des tests non fonctionnels, l'effet des mécaniques sur les performances globales du jeu est testé. L'utilisation des ressources (par exemple, RAM, CPU, etc.) et le résultat réel (par exemple, l'apparition de freezes dans le client du jeu, une diminution significative du taux de rafraîchissement, etc.) sont testés. La compatibilité des différents composants et logiciels est à tester, notamment avec les programmes antivirus.

Si le jeu exige la présence d'un serveur, les performances de ce dernier sont également testées. Les tests de performance sont planifiés et exécutés pendant la phase de production, lorsque les mécaniques prévues susceptibles d'affecter les performances sont implémentées. Les détails de la planification et de l'exécution des tests de performance sont décrits dans [ISTQB\_FL\_PT].

Une attention considérable est accordée aux méta-mécaniques lors de la réalisation de tests bêta avec un grand nombre de joueurs. La commodité et la clarté du jeu sont évaluées pour les joueurs. Comme pour les tests de logiciels autres que de jeux vidéo, l'objectif principal de ces tests n'est pas de constater des défauts fonctionnels évidents, qui devraient idéalement être identifiés lors des étapes précédentes, mais d'obtenir un retour d'information de la part des joueurs finaux. Toutefois, les joueurs finaux peuvent également participer et donner leur avis lors des tests alpha antérieurs, par exemple pour les grands jeux multijoueurs.

Après la release du jeu sur le marché, la tâche des testeurs est de traiter et de vérifier les défauts des mécaniques signalés par les joueurs, et de tester toute nouvelle mécanique ajoutée au jeu.

### 2.2.2 L'importance de tester les mécaniques des jeux vidéo

Ce sont les mécaniques du jeu qui constituent son épine dorsale. Cela constitue la base de l'expérience du joueur, ce qui est nécessaire pour maintenir son intérêt. Les mécaniques du jeu étant l'essence même de tout jeu, tester leur bonne exploitation est de la plus haute priorité pour un testeur, et les défauts constatés lors de ces tests sont généralement les plus critiques.

Un défaut dans les principales mécaniques, tel que le saut inopérant d'un personnage, peut entraîner l'impossibilité de surmonter un certain obstacle et, par conséquent, de terminer le jeu vidéo.

Mais même si la mécanique est rarement rencontrée dans le jeu ou est associée à des activités qui ne sont pas nécessaires pour passer le jeu vidéo (par exemple, la capacité du personnage à monter à cheval), son exploitation incorrecte aura un impact négatif sur la perception du jeu dans son ensemble.

### 2.2.3 L'importance de la revue des mécaniques des jeux vidéo

Une étape importante dans la création d'un jeu est l'élaboration de la documentation du jeu, qui décrit notamment les mécaniques du jeu.

Réviser la spécification des mécaniques du jeu dès le début de la phase de développement permet d'éviter de nombreux problèmes par la suite. Ces problèmes peuvent être liés au processus de développement lui-même, ainsi qu'à la réaction des joueurs aux mécaniques de jeu implémentés dans le produit final.

Au cours de la revue, le testeur doit prêter attention aux caractéristiques de qualité suivantes [ISO 25010] :

- L'exhaustivité de la description,
- La conformité avec la réalité,
- La structure et la facilité de navigation dans la documentation.

Problèmes dans le processus de développement des jeux vidéo que la revue peut éviter:

- La compréhension incorrecte de l'essence des mécaniques par le développeur et l'évaluation incorrecte de l'effort requis, en raison d'une description insuffisamment claire, détaillée et sans ambiguïté des mécaniques,
- L'incompatibilité des nouvelles mécaniques avec les mécaniques existantes,
- L'impossibilité d'implémenter des mécaniques dans un projet particulier.

Problèmes liés à la perception des mécaniques par les joueurs que la revue peut éviter :

- Inadéquation générale des mécaniques dans un jeu particulier.
- Mécaniques inintéressantes,
- Mécaniques déséquilibrées,
- Utilisation trop rare des mécaniques dans le jeu.

### 2.2.4 Tester l'état du jeu vidéo après la reprise de la session et lorsque l'utilisateur est inactif

#### État du jeu vidéo

L'état d'un jeu vidéo s'entend comme la valeur de tous les paramètres et variables qui décrivent tous les objets du jeu à un moment donné.

Plus le jeu est complexe et plus il offre de possibilités d'actions au joueur, plus chaque objet possède de paramètres.



Tous les paramètres peuvent être divisés en deux catégories : les paramètres visibles et les paramètres cachés.

Les informations relatives aux valeurs des paramètres visibles sont explicitement mises à la disposition du joueur. Cela peut concerner :

- Le niveau d'expérience en jeu de l'utilisateur pour la tâche accomplie,
- Les caractéristiques de l'arme sélectionnée,
- Le nombre d'objets autorisés dans l'inventaire,
- Le coût des objets dans la boutique du jeu.

Les paramètres cachés peuvent être utilisés par les développeurs pour influencer le cours du jeu à l'insu du joueur. Par exemple, en plus des indicateurs d'attaque et de défense connus du joueur, le personnage peut disposer d'un paramètre de précision, caché au joueur, qui détermine la probabilité qu'un tir atteigne la cible. Souvent, le joueur ignore l'existence de ces paramètres et ce qu'ils affectent exactement.

Les paramètres cachés peuvent être responsables de différents aspects du jeu :

- Le taux de chute, c'est-à-dire la chance d'obtenir un objet aléatoire en guise de récompense,
- La cadence de tir ou la vitesse de déplacement des équipements sur différents types de surfaces.

L'influence des choix du joueur sur le développement de l'intrigue du jeu. Un ensemble de paramètres cachés et explicites définit de manière unique le personnage du jeu contrôlé par le joueur et les autres objets.

### **Objectifs de test de l'état du jeu vidéo**

Un point important lorsque l'on teste les paramètres d'un jeu est de tester si le jeu utilise bien les valeurs réelles des paramètres.

Dans le cas de paramètres explicites, le test est effectué en comparant visuellement la valeur affichée avec celle réellement utilisée. Par exemple, si un personnage du jeu dont le paramètre d'attaque est égal à 1 inflige 10 unités de dégâts à l'ennemi, alors après que le joueur a augmenté le paramètre d'attaque à 2, la quantité de dégâts infligés au même ennemi devrait également augmenter et devenir, disons, 20 unités.

Un tel test est particulièrement important si le joueur dispose d'informations sur le nouveau montant des dégâts, sous la forme, par exemple, d'un indice de jeu : tel que "Chaque unité d'un paramètre d'attaque ajoute 10 unités de dégâts". Dans ce cas, le joueur pourra également comparer la valeur indiquée dans l'indice avec les dégâts réels.

Un joueur ordinaire se concentre généralement sur l'adéquation et la cohérence de la modification des dégâts, sans tenir compte des valeurs spécifiques des paramètres. Un tel joueur peut ne pas considérer comme un défaut le fait d'infliger un nombre incorrect de points de dégâts, mais si les dégâts ne changent pas ou diminuent, il le remarquera certainement.

Un testeur professionnel est guidé par les formules et les principes de calcul de la modification des paramètres spécifiés dans la documentation. Si cette information ne peut être obtenue à partir de la documentation, le testeur, comme le joueur, l'évalue de manière relative, en se basant sur le bon sens.

Pour tester les paramètres cachés, le testeur peut occasionnellement avoir besoin d'accéder à la base de données du jeu. Avec l'aide de celle-ci, le testeur peut comparer les valeurs réelles des paramètres avec les valeurs attendues.

En outre, si l'on suppose que l'utilisateur peut modifier les paramètres explicites de son personnage pendant le jeu, la possibilité même de modifier les paramètres et l'exactitude de leur affichage sont testées.

---

Tester l'état d'un jeu vidéo est étroitement lié à la sauvegarde et au chargement d'un jeu.

### **Sauvegarde et chargement**

De nombreux jeux sont impossibles à terminer en une seule session, ou ils peuvent en principe impliquer une jouabilité sans fin. C'est pourquoi les jeux utilisent souvent des fonctions de sauvegarde et de chargement de l'état du jeu vidéo, afin que le joueur n'ait pas à recommencer à chaque fois depuis le début.

D'autres raisons d'utiliser des sauvegardes sont d'encourager l'utilisateur à explorer l'univers du jeu, à trouver différentes solutions pour résoudre les problèmes proposés par le jeu, et même à gérer la complexité du jeu. Si le joueur a la possibilité de survivre devant un obstacle difficile, le coût de son erreur sera moins élevé que si une action incorrecte annule complètement la progression.

La **sauvegarde** désigne les informations relatives à la progression du jeu qui sont stockées sur le disque dur ou dans le cloud. La sauvegarde peut se présenter sous la forme d'un fichier qui contient des informations sur l'état actuel du jeu. La taille d'un tel fichier est plusieurs fois inférieure à celle du jeu lui-même et contient généralement les informations suivantes :

- Une liste d'objets pour lesquels des informations d'état doivent être stockées,
- Une liste de paramètres changeants pour chaque objet,
- Un code spécifique pour chaque paramètre, grâce auquel le jeu peut déterminer ce qui lui arrive en ce moment.

Presque toutes les variables peuvent agir comme des objets. Son état est exploité par le système du jeu et doit être mémorisé. Les objets variables comprennent, par exemple, le niveau de santé du personnage du jeu, le nombre de capacités apprises et les coordonnées du personnage sur la cartographie du jeu.

### **Types de sauvegardes et zones de test**

La méthode de sauvegarde, le format du fichier de sauvegarde et la liste des paramètres enregistrés peuvent varier d'un jeu à l'autre. Au cours de l'histoire des jeux vidéo, un grand nombre d'approches ont été inventées pour sauvegarder la progression. Les plus couramment utilisées sont décrites ci-dessous.

### **Utilisation des points de contrôle pour l'enregistrement**

Les sauvegardes se font automatiquement aux moments spécifiés par les développeurs. Le fait d'atteindre chaque point de contrôle suivant écrase les informations relatives à la progression actuelle du joueur. L'état du jeu vidéo sauvegardé aux points de contrôle n'est stocké que pour la session de jeu en cours et est supprimé lorsque le jeu se termine.

Parfois, le jeu vidéo n'indique pas explicitement que l'utilisateur a atteint le point de contrôle, et la progression est sauvegardée. Afin de tester l'exactitude de la sauvegarde et du chargement, le testeur a besoin d'une documentation fournissant la liste et l'emplacement exact de tous les points de contrôle existants. Lors de l'utilisation d'autres méthodes de sauvegarde, ces informations peuvent être absentes de la documentation ou être moins détaillées.

### **Les points de sauvegarde**

À certains endroits du jeu, les développeurs placent des points spéciaux où le joueur, s'il le souhaite, peut sauvegarder manuellement sa progression actuelle. Ces points sont généralement entourés de quelques objets de jeux vidéo, ce qui évite la création de gros fichiers de sauvegarde.

Le domaine de vérification de cette méthode est la possibilité de sauvegarder à chaque point, et la possibilité de sauvegarder et de charger à plusieurs reprises chaque état sauvegardé.

---

## Sauvegarde automatique

Il s'agit d'une combinaison de points de contrôle et de sauvegardes stationnaires. La progression du joueur est sauvegardée automatiquement tout au long du gameplay à certains points. Par exemple, de nombreux jeux disposent d'une fonctionnalité de sauvegarde automatique à la sortie. Au début d'un jeu, l'utilisateur peut commencer une nouvelle partie ou charger une partie précédemment sauvegardée.

Dans ce cas, la tâche du testeur consiste à vérifier que la sauvegarde a bien eu lieu et qu'il est possible de charger l'état du jeu vidéo atteint à tout moment.

## Sauvegarde manuelle ou libre

L'utilisateur peut sauvegarder à tout moment pendant le jeu en utilisant un élément spécial dans le menu du jeu. La sauvegarde rapide peut être implémentée lorsque l'enregistrement et le chargement s'effectuent en appuyant sur une seule touche ou en effectuant un seul clic.

Comme pour la sauvegarde automatique, la sauvegarde manuelle crée un fichier spécial qui contient des informations sur l'état du jeu vidéo. Dans ce cas, les tests incluent la vérification de l'exactitude du chargement des informations à partir du fichier, du nom correct du fichier, de l'emplacement du fichier dans la structure du système d'exploitation, de l'exploitation correcte dans la nouvelle version du jeu, et même de la possibilité d'utiliser le fichier de sauvegarde, par exemple, sur un autre ordinateur.

Il suffit de tester le chargement correct du niveau de jeux vidéo requis pour les jeux correspondant à la description suivante :

- Le jeu considère qu'il n'y a qu'un seul chemin correct et qu'une seule façon de passer (par exemple, une seule façon correcte de sortir d'un labyrinthe),
- Le personnage de jeu contrôlé par le joueur ne change pas d'un niveau à l'autre (par exemple, il n'augmente pas sa force d'impact, la portée de ses sauts, sa vitesse de déplacement à travers un niveau et d'autres caractéristiques qui pourraient augmenter la vitesse de passage d'un niveau).

Dans les jeux qui correspondent à la description suivante, les sauvegardes contiennent davantage d'informations uniques sur la session de jeu, le niveau, le personnage, etc. :

- Le personnage de jeu peut avoir de nombreux paramètres (par exemple, la vitesse de déplacement, la portée ou la hauteur du saut, la force d'impact, l'inventaire, l'apparence du modèle 3D et d'autres caractéristiques).
- Les paramètres décrits ci-dessus peuvent changer en fonction des actions et décisions du joueur dans le jeu (par exemple, le joueur trouve un artefact dans un labyrinthe et sa vitesse de déplacement est augmentée).

Pour ces jeux, les testeurs doivent tester l'exactitude du chargement du niveau requis du jeu et des informations uniques, mais aussi des informations uniques pour chaque joueur et chaque personnage du jeu. Lorsqu'il teste la sauvegarde, le testeur doit vérifier les éléments suivants :

- Le joueur se trouve au niveau de jeu auquel il a terminé la session de jeu précédente,
- Le personnage jouable possède un artefact,
- Le paramètre de vitesse du personnage de jeu est effectivement augmenté en raison de l'artefact.

## 3. Tests du graphisme - 165 minutes (K3)

### Mots-clés du domaine des tests logiciels

Tests en situation de jeu

### Mots-clés spécifiques du domaine des jeux vidéo

Modèle 3D, animation, collisions, niveau de jeu vidéo, hit box, niveau de détail (LoD), cartographie, ancrage, éclairage de scène, habillage, textures, effets visuels (VFX).

### Objectifs d'apprentissage pour le chapitre 3

#### 3.1 Principes et concepts du graphisme des jeux vidéo

GaMe-3.1.1 (K2) Expliquer les caractéristiques du contenu graphique d'un jeu vidéo

GaMe-3.1.2 (K2) Classer les types de défauts dans le contenu graphique

#### 3.2 Approches de test du graphisme dans les jeux vidéo

GaMe-3.2.1 (K2) Résumez les principales approches de test des aspects artistiques

GaMe-3.2.2 (K2) Résumer les principales approches de tests techniques

GaMe-3.2.3 (K2) Résumer les principales approches des tests du gameplay

#### 3.3 Réalisation des tests du graphisme

GaMe-3.3.1 (K3) Appliquer les approches fondamentales des tests de graphisme

GaMe-3.3.2 (K2) Expliquer l'importance de tester la validité historique des graphiques.

#### 3.4 Outillage des tests du graphisme

GaMe-3.4.1 (K2) Résumer l'utilisation des outils de tests du graphisme

## 3.1 Principes et concepts du graphisme des jeux vidéo

### 3.1.1 Caractéristiques du contenu graphique des jeux vidéo

Tout produit de jeu, qu'il s'agisse d'un jeu mobile ou d'un grand jeu à budget élevé, comporte un grand nombre d'éléments graphiques différents qui, ensemble, permettent à l'utilisateur de visualiser le déroulement du jeu.

Différents spécialistes remplissent certaines fonctions dans le processus de création d'objets graphiques pour les jeux vidéo :

Rôle	Responsabilités
Artiste	Crée divers contenus graphiques pour le jeu vidéo.
Modélisateur 3D	Crée les modèles 3D des objets du jeu.
Artiste de texture	Crée des textures pour les objets du jeu.
Spécialiste de l'animation	Travaille sur l'animation des objets du jeu.
Artiste technique	Effectue des tests techniques
Testeur	Teste les objets du graphisme dans le moteur du jeu vidéo.

Le testeur d'un jeu vidéo, quel que soit le budget du jeu et la quantité de contenu graphique qu'il contient, doit garder à l'esprit et tester les différents domaines qui sont partiellement, complètement ou séparément présents dans chaque jeu.

Les principales zones du jeu vidéo à tester sont les suivantes.

### **Niveaux (cartographies du jeu)**

Les niveaux sont une zone distincte du monde virtuel du jeu, qui représente généralement un lieu spécifique, par exemple un bâtiment ou une ville. Le terme provient des premiers jeux de rôle sur table (RPG), où il désignait les niveaux d'un donjon (c'est-à-dire l'environnement dans lequel se déroulait la majeure partie du jeu). Les joueurs commençaient dans les profondeurs du donjon (niveau 1) et devaient atteindre la surface en passant par tous les niveaux, qui devenaient de plus en plus difficiles, jusqu'à ce qu'ils atteignent le dernier (par exemple, le niveau 100), remportant ainsi le jeu. Aujourd'hui, les niveaux, sous une forme ou une autre, sont présents dans presque tous les jeux vidéo.

### **Modèles**

Un modèle est un objet quelconque en graphisme informatique. Selon les méthodes de création d'images, le graphisme peut être divisé en plusieurs catégories :

- Les graphismes en 2D,
- Les graphismes en 3D,
- Les graphismes en images de synthèse (CGI).

### **Graphisme en 2D**

Les graphismes informatiques 2D sont classés en fonction du type de présentation des informations graphiques et des algorithmes de traitement d'image qui en découlent. En général, les graphismes informatiques en 2D sont divisés en vecteurs et en trames, bien qu'il existe également un type de représentation d'image fractale.

### **Graphisme en 3D**

Les graphismes 3D exploitent des objets dans un espace 3D. Les graphismes informatiques en 3D sont largement utilisés dans les films et les jeux vidéo.

Les graphismes en 3D peuvent être :

- Polygonal,
- Voxels

Le graphisme polygonal est un ensemble de sommets, d'arêtes et de faces qui définissent la forme d'un objet polyédrique en infographie 3D et en modélisation volumétrique. Les faces sont généralement des triangles, des quadrilatères ou d'autres polygones convexes simples, ce qui simplifie le rendu.

Un voxel est un élément virtuel correspondant à un ensemble de six polygones rectangulaires. Tout ce qui se trouve dans le monde virtuel (c'est-à-dire les pixels, les polygones et les voxels) doit être projeté sur les pixels de l'écran physique. Les graphismes à voxels sont similaires aux graphismes à trame. L'objet consiste en un ensemble de formes 3D, le plus souvent des cubes.

### **Graphisme : images de synthèse (CGI)**

Les images de synthèse sont des images en 3D obtenues par ordinateur sur la base de calculs et utilisées dans les arts visuels, l'imprimerie, les effets spéciaux cinématographiques, à la télévision et dans les simulateurs. Les images en mouvement sont créées par l'animation par ordinateur, qui est un domaine plus restreint du graphisme en images de synthèse.

Toute image sur l'écran, en raison de sa planéité, devient une trame, puisque l'écran est une matrice composée de colonnes et de lignes. Les graphismes en 3D n'existent que dans notre imagination, puisque ce qu'un humain voit sur le moniteur est une projection d'une figure en 3D, et que le spectateur crée lui-même l'espace. Ainsi, la visualisation des graphismes ne peut être que matricielle et vectorielle, et la

---

méthode de rendu n'est que matricielle (c'est-à-dire un ensemble de pixels). La façon de définir l'image dépend du nombre de ces pixels.

### **Textures**

Une texture est une image bitmap appliquée à la surface d'un modèle polygonal pour lui donner de la couleur ou une illusion de relief. D'une manière générale, les textures peuvent être considérées comme un canevas à la surface d'une image sculpturale. L'utilisation de textures permet de reproduire de petits objets de surface pour lesquels la création avec des polygones serait excessivement gourmande en ressources. Par exemple, les cicatrices sur la peau, les plis dans les vêtements, les petites pierres et autres objets à la surface des murs et du sol.

La qualité d'une surface texturée est déterminée par les texels, qui représentent le nombre de pixels par unité de texture minimale.

L'objectif principal du test de texture est de tester la présence des textures sur les objets visibles, leur affichage correct et leur uniformité. Si les textures du jeu sont en haute résolution, il ne devrait pas y avoir de textures de mauvaise qualité.

### **Collisions**

La collision est un attribut responsable du passage des objets en collision les uns à travers les autres.

Les jeux vidéo, en particulier les jeux sur console, doivent répartir un grand nombre de leurs tâches entre des ressources matérielles limitées et un temps de jeu très restreint. Malgré ces limitations et l'utilisation d'algorithmes de détection des collisions relativement primitifs et imprécis, les développeurs de jeux vidéo ont réussi à créer des moyens visuellement crédibles et relativement réalistes de représenter les interactions des objets du jeu les uns avec les autres.

Dans la plupart des jeux vidéo, les principaux objets qui doivent éviter les collisions et les intrusions sont le paysage et l'environnement du niveau. Il s'agit de structures statiques, non interactives et indestructibles telles que les montagnes, les arbres, les bâtiments et les clôtures. Dans ce cas, le personnage est représenté par un seul point et la méthode de partitionnement binaire de l'espace est utilisée (c'est-à-dire la méthode de partitionnement récursif de l'espace euclidien en ensembles convexes et en hyperplans). Ainsi, les objets sont présentés sous la forme d'une structure de données utilisée pour effectuer avec efficacité des exploitations sur le graphisme informatique en 3D, notamment le tri des objets visuels par ordre de distance par rapport à l'observateur et la détection des collisions. On dispose ainsi d'un moyen viable, simple et efficace pour tester si un point représentant un personnage se trouve dans l'environnement (terrain) ou non. Les collisions entre les personnages et les autres objets dynamiques sont prises en compte et traitées séparément. De plus amples informations sur les collisions sont fournies à la section 3.1.2.

### **Animations**

Avant que le modèle ne soit animé, un "squelette" est créé dans celui-ci. Ce processus est appelé ancrage. Les modèles de créatures vivantes et tous les objets censés être animés peuvent avoir des os virtuels. Par exemple, la cape du protagoniste. Les os du modèle sont dépendants les uns des autres, de sorte que, par exemple, lorsque la main est déplacée, les os de la paume se déplacent également. Toute l'animation ultérieure dépend de la qualité de l'ancrage et de l'habillage.

L'animation est une technique permettant de créer l'illusion d'images en mouvement, c'est-à-dire le mouvement et/ou la modification de la forme des objets. Le morphing utilise une séquence d'images fixes (frames) qui se remplacent les unes les autres à une fréquence élevée (de 12 images par seconde pour l'animation dessinée à la main à 30 images par seconde pour l'animation par ordinateur).

## Effets

Les effets visuels peuvent être divisés en deux grands types de tâches :

- Les effets de gameplay (effets d'interaction),
- Les effets naturels (effets environnementaux).

Le principe appliqué dépend du projet spécifique.

Parmi les autres genres, les effets de gameplay sont plus importants dans les jeux vidéo de combat et les RPG. Par exemple, lorsque votre personnage jouable pousse le personnage jouable d'un autre joueur, l'animation d'interaction correcte doit être jouée pour chacun d'eux. Il existe d'autres genres de jeux, tels que les jeux de tir, et en particulier des jeux réalistes, pour lesquels les effets naturels sont tout aussi importants que le gameplay.. Les chutes d'eau, le brouillard et la pluie sont des exemples d'effets naturels.

## Éclairage de la scène

L'éclairage de la scène est nécessaire pour que le joueur puisse la voir. La lumière affecte les émotions. La connexion entre l'image et la réactivité émotionnelle fournit un autre outil puissant qui aide à travailler avec le personnage, la narration, le son et la mécanique des jeux. Dans ce cas, l'interaction de la lumière avec la surface lui permet d'influencer la luminosité, la couleur, le contraste, les ombres et d'autres effets.

En raison de ses caractéristiques structurelles, l'œil humain reconnaît les objets en trois dimensions dans un rayon de 110°, et les couleurs dans un rayon encore plus petit. Étant donné que la vision centrale (dans la plage mentionnée ci-dessus) est la première chose qu'un humain utilise, elle devrait pouvoir percevoir les éléments critiques que le joueur doit absolument voir comme le concepteur l'a prévu. La vision périphérique fournit un contexte et renforce la vision centrale. Dans un jeu vidéo, si les éléments qui relèvent de la vision périphérique ne fournissent pas le contexte nécessaire ou contredisent les éléments qui relèvent de la vision centrale, la connexion entre la conception et le joueur est rompue.

Les exemples suivants illustrent l'utilisation correcte de l'éclairage de scène :

- La lumière tombant dans le champ de vision central peut guider le joueur,
- L'éclairage peut changer de cadre. Par exemple, l'utilisation d'une lampe de poche en fait la principale source de lumière pour le joueur. La perspective modifiée capture la vue du joueur sur la zone éclairée et coupe tout le reste en raison du fort contraste,
- L'éclairage peut guider et aussi créer une atmosphère, comme un sentiment de peur. Le joueur est maintenu dans une tension constante lorsque, par exemple, quelque part dans l'obscurité se cache un terrible ennemi,
- La direction de la lumière peut soit faciliter la découverte d'éléments au niveau du jeu, soit la rendre plus compliquée,
- Un manque ou une surabondance d'éclairage peut pousser le joueur à utiliser des objets ou des actions spéciales.

L'éclairage des scènes est l'un des domaines de validation les plus importants, car il crée l'esthétique du jeu et influence l'expérience de jeu. Ainsi, de nombreuses techniques d'éclairage de scène utilisées dans les arts visuels, le cinéma et l'architecture sont utilisées dans les jeux vidéo pour compléter l'esthétique de l'espace virtuel et améliorer l'expérience du joueur. Cependant, les jeux vidéo sont très différents des films ou du théâtre ; leurs environnements sont dynamiques et imprévisibles. Outre l'éclairage statique des scènes, des sources lumineuses dynamiques sont utilisées. Elles ajoutent de l'interactivité et les émotions voulues.

## Exactitude historique

Tout élément, tel qu'un graphisme, une animation ou un effet, doit correspondre au style de jeu général. La description textuelle ou sonore d'un objet, d'un modèle ou d'un lieu doit également correspondre à la description du prototype historique.

Un contenu graphique riche et complexe est l'une des principales différences entre les jeux vidéo et tout autre type de logiciel. Cela modifie considérablement l'approche de test. Pour tester les graphismes des jeux, le testeur doit avoir des connaissances supplémentaires en physique et en optique, une facilité de compréhension des techniques de rendu des couleurs et une connaissance de l'histoire. Les tests du graphisme sont l'une des activités de test les plus critiques, car c'est là que se concentre la majeure partie des défauts, qui affectent directement la perception du jeu par l'utilisateur et son expérience de jeu.

### 3.1.2 Types de défauts du contenu graphique

Une catégorie courante de défauts recherchés lors des tests du graphisme est celle des défauts visuels. Les défauts graphiques comprennent le déchirement de l'image à l'écran, l'absence de textures et le recadrage inattendu de certaines zones de l'image.

Lors de la création de graphismes et d'animations pour les jeux vidéo mobiles, on utilise les mêmes moteurs que pour un PC. La différence est que ces moteurs sont adaptés à des plateformes spécifiques, au matériel qui y est utilisé et à leurs adaptabilités techniques [Gregory18]. Par conséquent, les défauts qui apparaissent dans ces jeux sont similaires.

#### Les manques au niveau de la texture

Parmi les défauts les plus fréquemment rencontrés lorsque l'on teste des textures, on peut citer :

- Manque de textures des objets graphiques,
- Perte de textures pendant le gameplay,
- Textures et bouchons temporaires.

Comme le montre l'expérience, les deux premiers groupes de défauts sont le plus souvent liés aux performances insuffisantes des processeurs graphiques ou à des pilotes de carte graphique obsolètes. En comparaison, des textures temporaires peuvent apparaître en raison d'un défaut de développement.

#### Niveau de détail (Level of Detail - LoD)

Dans un jeu vidéo moderne en 3D, de nombreux objets situés à différentes distances de la caméra (du point de vue du joueur) peuvent être affichés sur la scène en même temps. Afin de réduire la charge sur le système, les développeurs utilisent la technologie du niveau de détail (LoD).

Lorsque les développeurs créent un nouvel objet de jeu (par exemple, un arbre, un bâtiment ou une voiture), ils ajoutent plusieurs variantes de modèles d'objets au jeu. Il s'agit de modèles avec un nombre réduit de polygones et une géométrie simplifiée (low-poly model) et de modèles plus détaillés avec un grand nombre de polygones (high-poly model). En fonction de la distance à la caméra, des modèles avec un nombre différent de polygones sont utilisés pour afficher le même objet dans le jeu. À proximité de la caméra, lorsqu'il est exigé d'obtenir une qualité d'image maximale, on utilise des modèles à haute densité de polygones. Au fur et à mesure que la caméra s'éloigne, ils sont remplacés par des modèles moins détaillés comportant moins de polygones. À une distance suffisamment grande, le modèle n'apparaît plus que sous la forme d'une silhouette ou n'est pas rendu du tout. Cela permet de réduire le nombre de polygones traités et d'augmenter les performances du jeu.

Par exemple, pour une forêt à l'horizon, des textures basse résolution peuvent être utilisées avec seulement la couleur affichée, sans relief ni reflets. Si le joueur se rapproche, les arbres apparaîtront dans tous leurs détails.

La technologie LoD est également utilisée pour les animations, les squelettes et l'intelligence artificielle des bots, où un programme automatisé joue à un jeu donné pour le compte d'un joueur humain. Par exemple, dans les jeux de tir avec un grand nombre d'ennemis dans le cadre, les bots auront différents niveaux de détail. Les bots en arrière-plan seront représentés de manière simplifiée avec des textures peu détaillées,



tandis que les adversaires proches seront soigneusement dessinés avec des animations élaborées et sembleront suffisamment intelligents pour combattre le joueur.

Les développeurs modifient également le niveau de détail en fonction de la fréquence d'images actuelle, de la vitesse de déplacement de l'objet à l'écran et du nombre total d'objets visibles simultanément.

Les défauts découlant de l'utilisation de LoD sont principalement liés aux faibles performances du système. Les développeurs essaient de faire en sorte que l'augmentation des détails de l'objet ou le remplacement du modèle se fasse en douceur et de manière imperceptible pour le joueur, mais ce n'est pas toujours le cas. Dans les jeux vidéo fonctionnant sur du matériel peu performant, les modèles à faible densité ne sont pas toujours remplacés par des modèles à forte densité assez rapidement. Par conséquent, le joueur qui s'approche d'un objet verra d'abord un modèle "laid", puis il sera remplacé par un modèle de meilleure qualité.

### **Collisions**

La collision est la manière dont un objet dans le jeu gagne en taille et réagit aux collisions avec d'autres objets. Un maillage de collision ou collisionneur est créé pour que le modèle d'objet fini interagisse avec son environnement et les autres objets. Il s'agit d'une forme simplifiée invisible d'un objet qui lui est liée pour calculer les collisions avec d'autres objets. Le collisionneur peut être appelé le modèle physique de l'objet. Il reste invisible pour le joueur, mais toutes les collisions du collisionneur sont gérées par le moteur du jeu vidéo [Buttfield19].

La forme du collisionneur doit correspondre approximativement au maillage du modèle, mais une approximation assez grossière est souvent suffisante. Pour la plupart des objets statiques de l'environnement, tels que les bâtiments, les rochers et les voitures accidentées, les développeurs utilisent des collisionneurs comportant un nombre de polygones réduit. Cela ne se distingue pas dans le gameplay, mais améliore l'efficacité du traitement de tous les objets visibles.

Dans ce cas, la taille du collisionneur doit correspondre au modèle visible, surtout si le joueur peut interagir directement avec cet objet.

Si le collisionneur de l'objet est beaucoup plus petit que son modèle visuel, le personnage pourra "tomber dans la texture" ou la traverser. À proprement parler, l'expression "tomber dans les textures" (par exemple, lorsqu'un personnage peut traverser un autre objet) est incorrecte, mais elle décrit parfaitement ce que le joueur voit lorsque son personnage est partiellement ou complètement immergé dans un autre objet.

De telles défaillances sont particulièrement critiques dans les jeux vidéo multijoueurs, car elles peuvent donner au joueur un avantage injuste. Par exemple, un char enfoncé dans une pierre sera pratiquement invisible pour les autres joueurs. Dans le même temps, une telle pierre ne protégera pas des tirs ennemis. C'est pourquoi les testeurs doivent accorder une attention particulière à ce type d'objets sur les cartographies en mode joueur contre joueur (player versus player - PvP).

Néanmoins, certains objets dans le jeu peuvent très bien être décoratifs et ne présenter aucune collision. Par exemple, un char peut facilement traverser un buisson, et le joueur n'a pas besoin d'être obligé de contourner ou de sauter par-dessus chaque petit caillou ou boîte de conserve.

Les conventions des jeux vidéo s'appliquent ici. De la même manière, on peut envoyer un personnage muni d'une arme longue dans un couloir étroit et bas pour repousser des ennemis attaquant de différentes directions. Contrairement à la réalité, le personnage pourra tourner librement dans la direction souhaitée et ne remarquera même pas qu'en tournant, la lance passe à travers les murs. De plus, les longs cheveux du personnage peuvent passer à travers les épaulettes massives de l'armure. Cependant, la présence d'une clôture basse, qui empêche soudainement le joueur de grimper, ennuiera le joueur et réduira son plaisir.

Si la taille du collisionneur de l'objet est plus grande que le modèle visible, une situation peut se présenter lorsque le personnage se soutient contre un mur invisible ou se tient en l'air sur un soutien minuscule. De telles défaillances sont souvent exploitées pour le déroulement rapide et imprévu de jeux vidéo. Les "speedrunners" recherchent spécifiquement des endroits où les développeurs ont été paresseux ou ont oublié de redimensionner les collisionneurs des objets environnants. Cela permet parfois au joueur de contourner une partie importante du lieu et de réduire le temps de passage.

### **Boîtes à outils**

Si l'on suppose qu'un personnage ou un objet de jeu peut subir des dégâts, par exemple, d'une balle / d'un projectile ennemi, il faut alors, pendant le combat, calculer la position de chaque point sur la surface de cet objet et déterminer s'il a été touché ou non.

Pour des objets de forme complexe, cela est très coûteux et pas toujours justifié.

Pour simplifier ces calculs, on utilise le concept de hitbox. Par exemple, dans les jeux vidéo de plates-formes ou de combat en 2D, elle peut se présenter sous la forme d'un ou plusieurs rectangles. Leur position les uns par rapport aux autres est facile à calculer.

Pour un objet en 3D, la forme de hitbox la plus simple est une sphère. Dans ce cas, il suffit de connaître son centre et son rayon pour effectuer les calculs. Ainsi, à tout moment, il est facile de déterminer si une balle ou un autre objet se trouve à l'intérieur de cette sphère. Cependant, une hitbox à tuyaux parallèles se révèle être une forme plus adaptée car elle contient moins d'espace vide par rapport à une sphère, et il suffit de connaître le point central de l'objet et les trois dimensions : longueur, largeur et hauteur pour effectuer les calculs.

Pour plus d'authenticité, l'objet peut être décrit par de multiples hitboxes, séparément pour chaque partie du modèle : corps, (par exemple, bras, jambes, tête, queue, et même armes).

Cependant, les parties du modèle qui peuvent être endommagées et celles qui peuvent causer des dommages ne coïncident pas toujours. C'est pourquoi, dans certains jeux vidéo, la hitbox et la hurtbox sont attribuées séparément pour un même objet de manière à définir les zones qui infligent des dégâts (par exemple, les armes, les poings, etc.) et les zones sur lesquelles les dégâts sont infligés (par exemple, la tête, les membres). Pour déterminer si un personnage doit subir des dégâts, on teste si la hitbox de l'arme et la hurtbox du corps du personnage se chevauchent.

La tâche principale du testeur lorsqu'il travaille avec des hitboxes sera de tester si la zone visible de l'impact correspond au résultat réel de l'attaque.

Même dans les jeux vidéo de grande envergure, il y a des moments où le joueur voit clairement que son arme a touché l'ennemi, mais n'a pas causé de dégâts. Au contraire, si la hitbox est trop grande, l'épée de l'ennemi peut blesser le personnage alors qu'elle passe visuellement à côté de lui.

### **Autres défauts**

Lorsque l'on travaille sur un projet d'envergure, chaque étape de la production du modèle passe par de nombreuses approbations et confirmations. Elle est testée par les artistes, les modélistes et les texturistes, entre autres. Même dans ce cas, les testeurs peuvent encore détecter des défauts visuels apparus au cours du processus de production.

### **Incompatibilité avec le schéma historique**

Les joueurs peuvent accepter des armures irréalistes avec des épaulettes géantes et l'apparence invraisemblable d'un engin volant dans des jeux vidéo de fantasy ou de science-fiction. Cependant, si le jeu se veut réaliste, il y aura certainement des joueurs méticuleux qui indiqueront que les hélices de certains modèles d'avions étaient situées complètement différemment, et qu'il y avait beaucoup plus de canons sur certains cuirassés.

C'est pourquoi il est bon que le testeur prenne connaissance d'au moins quelques photographies de l'objet testé, si possible, afin d'imaginer à quoi il ressemblait dans la réalité.

### **Objets suspendus dans les airs**

Un grand nombre de spécialistes peuvent travailler en même temps sur un lieu. Certains ajoutent ou enlèvent des objets ; d'autres modifient la géométrie du terrain (cartographie) ou fixent les tuiles (textures appliquées au terrain). C'est pourquoi des défauts apparaissent lorsqu'un objet existant précédemment est noyé dans un terrain ou suspendu dans les airs, parce que son ancien "soutien" a été supprimé.

### **Jonction visible entre les textures**

De tels défauts sont le plus souvent constatés sur le terrain des grandes cartographies et aux endroits où plusieurs textures sont mélangées. Par exemple, une "couture" peut être visible aux transitions entre une surface d'herbe et du sable ou de la pierre. Pour plus de détails, voir le chapitre 3.1.

### **Destruction d'objets**

Habituellement, le modèle possède plusieurs états afin d'ajouter du réalisme aux différentes situations, comme l'état d'origine, l'état endommagé et l'état détruit. Si le jeu suppose une destruction réaliste des objets, un modèle de collision spécial est créé pour eux, éventuellement avec son propre modèle de collision. À un certain moment, le modèle original est remplacé par un modèle détruit.

Il est important de veiller à ce que l'effet de la destruction corresponde au modèle lui-même. Le camion détruit doit être différent du modèle original, et les morceaux d'un bâtiment en briques cassées ne doivent pas être en bois.

### **Défauts de luminosité**

Les défauts de luminosité sont les suivants :

- Les défauts de l'éclairage global,
- Les défauts des sources lumineuses ponctuelles,
- Les défauts des sources lumineuses des projecteurs,
- Les défauts des sources ayant une surface,
- Les défauts des sources lumineuses directionnelles,
- Les défauts des sources lumineuses émettrices,
- Les défauts des sources de lumière diffuse.

Les moteurs de jeux vidéo modernes comprennent des capacités très avancées pour créer différents types de sources lumineuses en parfaite conformité avec les lois de l'optique. Cependant, il existe encore des jeux vidéo où l'éclairage n'est pas idéal, par exemple en raison de paramètres d'éclairage incorrects ou d'un manque d'expérience du développeur. La prise en compte de la composition visuelle (position de la lumière, ses angles, ses couleurs, son champ de vision et son mouvement) a un impact important sur la façon dont les joueurs perçoivent l'environnement du jeu, et crée l'atmosphère émotionnelle adéquate pour engager le joueur. Les concepteurs doivent y veiller en créant une intégrité visuelle. [Lee16], Tavakkoli18], Romero19].

### **Défauts d'animation**

Ce défaut résulte de l'accrochage du squelette au modèle, appelé habillage. Plus le modèle a d'os, plus l'animation peut être réaliste. Différents objets peuvent être animés, par exemple les cheveux du protagoniste.

Les os d'un modèle dépendent les uns des autres. Ainsi, lorsqu'une main est déplacée, les os de la paume bougeront également. Toute l'animation ultérieure dépend du réalisme avec lequel tout a été fait et mis en place lors de l'ancrage et de l'habillage.

Si la conception a commis des erreurs, cela peut entraîner l'allongement des bras/jambes des personnages dans diverses animations, et des parties de modèles qui sont "détachées" du reste des éléments. Souvent, ces défaillances apparaissent lors de collisions avec d'autres modèles ou dans diverses animations du modèle.

### Défaillances dans le domaine des effets visuels (VFX)

Les jeux vidéo utilisent souvent divers effets liés aux actions des personnages, aux événements et aux phénomènes naturels, tels que les explosions, les étincelles, la fumée, la disparition et l'apparition.

Lorsque l'on travaille avec des animations, les effets visuels sont généralement attachés aux squelettes, aux assistants (c'est-à-dire aux objets utilitaires utilisés pour créer et animer des modèles) et aux objets de la scène. La création d'effets visuels est un processus itératif au cours duquel les équipes d'animation et les artistes des effets visuels interagissent. Ces derniers peuvent être invités à apporter des modifications ou des ajouts à l'animation, par exemple,

- ajouter davantage d'images clés pour que la traînée de l'épée soit plus fluide,
- faire pivoter l'assistant pour utiliser sa rotation comme direction pour les éclaboussures de sang,
- modifier le changement de caméra pour éviter l'aspect peu esthétique du maillage de particules.

Tout d'abord, il est important que ces effets soient synchronisés avec les événements qui les produisent. Par exemple, le feu et la fumée provenant de la bouche du canon doivent être synchronisés avec le moment du tir et la balle qui sort du canon. Dans le cas contraire, le réalisme de l'événement ne sera pas respecté. De telles occurrences sont traditionnellement considérées comme une défaillance.

Un autre problème à l'origine des défaillances VFX est la non-conformité aux conditions techniques liées au maintien de la fréquence d'image (NDT : frame rate) et à l'optimisation des ressources matérielles utilisées pour afficher correctement l'effet. Quelle que soit l'ampleur de l'effet, il est peu probable que le joueur soit heureux de voir à l'écran des nuages de poussière qui se tortillent ou des explosions figées dans le temps. Par conséquent, les effets spéciaux dans les jeux vidéo sont soumis à des limites techniques strictes, dont le but est de maintenir un certain nombre stable d'images affichées.

Lorsqu'il teste les effets spéciaux, le testeur doit s'assurer que les capacités matérielles de l'appareil de jeu sont utilisées de manière optimale et que les effets sont synchronisés avec les événements qui les déclenchent.

Il est souhaitable de trouver le plus grand nombre possible de défauts au stade de la production du modèle. Par exemple, le dépassement du nombre de polygones du modèle ou les défauts de texturage, qui augmentent considérablement la charge du matériel utilisé pour les calculs du modèle. Cela entraîne inévitablement de longs retards dans le jeu en raison de la surcharge du processeur de l'adaptateur vidéo.

Pour tester cela, le développeur doit avoir reçu des exigences prédéterminées et fixes pour des aspects tels que le LoD et les modèles de collision.

Les testeurs, en règle générale, constatent les manifestations de ces défauts, y compris les pannes et défaillances qui en découlent. Si un objet a, par exemple, un modèle de collision avec trop de polygones, le défaut ne sera pas détecté tant que le taux de rafraîchissement est élevé lors de l'interaction avec cet objet.

## 3.2 Approches de test du graphisme dans les jeux vidéo

### 3.2.1 Tests des aspects artistiques

Lors du processus de création d'un objet graphique, celui-ci passe par de nombreuses étapes et revues différentes. Certaines personnes créent la conception et d'autres l'intègrent dans l'application. Par conséquent, le résultat peut ne pas correspondre tout à fait à l'intention de l'auteur de la conception. Un

testeur peut tester la conformité de la conception avec les standards et remarquer les défauts, mais il y a des défauts que l'artiste remarquera plus facilement. Par exemple, une transparence insuffisante de l'arrière-plan ou le fait que l'application ne se comporte pas comme prévu lors de la mise à l'échelle.

À l'inverse, l'artiste ne peut pas tout prévoir et ne remarquera le défaut que dans l'application en cours d'exécution, lorsqu'il est déjà devenu une défaillance. C'est pourquoi il est bon de confier l'application au concepteur pour qu'il la révise après l'implémentation, afin d'aider à éliminer les défauts critiques. Dans ce processus, le testeur prépare l'environnement nécessaire, crée des comptes de test, décrit les défauts après les tests et est alors en mesure de les tester ou de les envoyer au créateur pour revue.

Dans le cas de projets comportant une grande quantité de contenu, des spécialistes sont souvent impliqués dans les tests du graphisme. Il s'agit principalement d'artistes qui comprennent parfaitement le processus de création des modèles. Ils confirment chacune des étapes suivantes de la création du modèle :

- Création de géométrie,
- Création de textures,
- Création de modèles de collision.

L'objectif de ces travaux est d'améliorer la qualité des modèles avant leur exportation vers le moteur de jeux vidéo. Par exemple, ils testent des aspects tels que le polygonage (c'est-à-dire le nombre de polygones) de la géométrie, la justesse de son emplacement, la justesse du déroulement du modèle, la présence de polygones allongés et de coutures dans le modèle.

Le test artistique est une tâche vaste et complexe qui peut être réalisée à différents stades de la production, en commençant par la création de la géométrie et des textures les plus simples, et en terminant par l'exportation du modèle vers le moteur et son placement sur la carte. Pour effectuer des tests artistiques, le testeur n'a souvent pas besoin d'outils spéciaux ni d'éditeurs de contenu. Il peut le faire directement en exécutant le jeu. Toutefois, la disponibilité d'outils et d'éditeurs de contenu permet de constater les défauts beaucoup plus rapidement et avec beaucoup plus d'efficacité.

Les tests artistiques sont réalisés par :

Rôle	Responsabilités pour les tests des aspects artistiques
Artistes	Lors de la revue d'objets. Lors de la visualisation et de l'évaluation d'objets
Modélisateurs 3D	Lors de la revue d'objets
Superviseurs	Lors de la visualisation et de l'évaluation des objets
Testeurs	Après l'exportation finale des modèles vers le moteur
Joueurs	Lors de la participation à des tests en situation de jeu

### 3.2.2 Tests techniques

Les tests techniques impliquent un ensemble de tâches liées aux paramètres techniques du graphisme, telles que :

- La conformité avec les limites du nombre de polygones du modèle, qui ont été mentionnées précédemment.
- Les formats de texture,

- Les distances de basculement LoD, pour lesquelles il suffit de comparer les ressources disponibles avec la spécification.

Les tests techniques proprement dits sont réalisés par des testeurs ainsi que par des artistes techniques. Il comprend les procédures décrites ci-dessous :

### **Recherche de fichiers inutilisés et temporaires dans les ressources de contenu graphique**

Lorsque l'on travaille avec des artistes, on peut accidentellement ajouter au référentiel des ressources qui ne seront pas utilisées par la suite. Par exemple, un développeur peut charger dans le référentiel toutes les ressources, y compris celles qui ne sont pas utilisées. Dans ce cas, l'espace occupé sur le disque dur par le client du jeu augmente, ce qui réduit les performances et augmente finalement la quantité totale de correctifs livrés au joueur (par exemple, la mise à jour des fichiers individuels du jeu vidéo). Pour éviter cela, une recherche des ressources inutilisées est effectuée.

### **Recherche de la présence de tous les enregistrements de contenu requis dans les ressources du client**

Dans les ressources du client du jeu, le contenu graphique n'existe pas indépendamment des autres ressources. Les références aux textures sont inscrites dans les modèles, les références aux modèles sont inscrites sur les cartographies. Même les plus petits défauts dans ces enregistrements peuvent entraîner des conséquences inattendues.

### **Test des formats de texture**

Pour chaque type de texture, les artistes techniques fixent leurs propres exigences. Cela s'explique par la nécessité de trouver un équilibre entre la performance du contenu et le composant visuel. Le format de la texture implique le type de compression et sa taille.

### **Test des logs des clients lors des tests de contenu**

Parfois, le travail des sous-systèmes clients n'est pas très visible. Par exemple, les post-effets sont des effets qui se superposent à l'image, comme une lentille lorsqu'on regarde le soleil. Une texture dédiée est créée pour chaque post-effet, en fonction de la superposition sur l'écran. Lorsque l'on regarde le soleil avec une lentille, par exemple, on verra la lentille et les rayons du soleil, que la texture soit affectée au post-effet ou non. Dans ce cas, le joueur ne verra tout simplement pas le défaut, bien qu'il existe. Pour constater de tels défauts, il faut examiner les logs du client car ils mentionneront certainement que le rendu a essayé d'utiliser une texture qui n'existe pas dans les ressources.

### **Vérification de la conformité avec les limites du nombre de textures, de modèles**

Chaque type de contenu a ses propres limites, qui sont généralement générales pour un groupe spécifique. Par exemple, pour les bâtiments indestructibles à plusieurs étages, il est permis d'utiliser jusqu'à 15 000 polygones, mais pour les bâtiments à un étage, seulement 10 000. Étant donné que les modifications de contenu sont permanentes et que le nombre de téléchargements dans le référentiel est important, il est nécessaire, après chaque nouvelle exportation du modèle, de tester la conformité aux limites et aux exigences afin d'éviter des situations liées à un manque de ressources matérielles nécessaires pour tester les modèles du processeur vidéo.

Malgré la facilité de ces tests, il est essentiel de comprendre l'importance des tests techniques. Le graphisme peut avoir un impact considérable sur les performances globales du système. C'est pourquoi les tests techniques comprennent des tests de performance, afin de s'assurer que les performances et la taille du jeu restent au niveau souhaité. En augmentant le nombre de textures et de modèles, la taille du contenu augmente également. Cette inadéquation des formats de textures détériore l'indicateur du taux de rafraîchissement, ce qui entraîne une augmentation de la mémoire consommée par le client. Les tests techniques permettent de s'assurer que le contenu artistique sera livré au joueur dans toute son étendue et son périmètre.

### 3.2.3 Tests de gameplay

Le test de gameplay est une approche de test visant les facteurs qui affectent le gameplay. Il peut s'agir à la fois d'une évaluation de la conformité du modèle de collision avec le modèle visuel, et d'une vérification de la conformité des réglages des objets du gameplay avec les exigences du mode de jeu.

Les tests de gameplay des objets comprennent :

- Tester la conformité des points de vie (PV). Il s'agit d'une valeur dans les jeux de rôle et les jeux vidéo qui détermine la quantité maximale de dégâts pouvant être appliquée à un objet pour le détruire. Malgré le fait que les PV soient attribués aux objets automatiquement, en tenant compte du matériau, de la taille et du type de l'objet, des défauts peuvent toujours survenir. C'est pourquoi l'objet du PV est testé manuellement en fonction des caractéristiques de l'objet.
- Tester que le modèle de collision correspond au modèle visuel. Comme indiqué précédemment, le modèle de collision est considérablement simplifié par rapport au modèle visuel. Cela peut entraîner des défauts lorsque le joueur est visuellement caché derrière le modèle, mais qu'il peut en fait être endommagé par un ennemi.

Les tests du graphisme sont un processus complexe dont les types se chevauchent. L'approche de test décrite ci-dessus permet de s'assurer que le contenu graphique du jeu est testé de manière approfondie à différentes étapes afin de détecter les défauts de graphisme.

## 3.3 Réalisation des tests du graphisme

### 3.3.1 Réalisation des tests du graphisme à différents stades de la production du jeu vidéo

#### Création d'une boîte grise

Il s'agit d'un modèle simplifié d'un objet du graphisme, qui est en fait une maquette utilisée pour tester le gameplay. À ce stade, les défauts découverts sont principalement liés à la commodité avec laquelle les artistes peuvent travailler avec ce modèle.

#### Création d'une géométrie visible

L'étape suivante de la modélisation et de la cartographie comprend la cartographie et la création d'une projection en 3D sur un objet physique de l'environnement. Cela prend en compte sa géométrie et son emplacement dans l'espace. La géométrie visible et le dépliage UV du modèle sont créés et la correspondance entre les coordonnées sur la surface de l'objet 3D (X, Y, Z) et les coordonnées sur la texture (U, V) est vérifiée. Étant donné que les inspections sont souvent effectuées à l'aide d'éditeurs de graphisme 3D, les défauts à ce stade peuvent également être détectés par les artistes et les modélisateurs 3D. En règle générale, ces défauts comprennent :

- La géométrie dupliquée. La duplication de polygones peut dégrader considérablement les performances.
- Des détails excessifs. Par exemple, un modèle hors de la visibilité du joueur présente des biseaux, ce qui signifie que lors du traitement de telles "décorations", des ressources matérielles supplémentaires seront allouées, ce que le joueur ne remarquera tout simplement pas.
- Des détails insuffisants. Ce type de défaut est l'exact opposé du défaut d'excès de détails. Si le modèle est clairement visible par le joueur, il doit être suffisamment détaillé pour ne pas avoir l'air anguleux.
- Des parties de la géométrie qui ne sont pas rendues. Par exemple, le fait qu'il manque des parties nécessaires à un modèle peut le rendre irréaliste.

## Texturer le modèle

À ce stade, les textures sont acceptées par l'artiste ou le directeur artistique. Dans ce cas, l'application incorrecte des textures, l'absence de coutures et de tiraillements (c'est-à-dire la topologie incorrecte des polygones) peuvent ne pas être découvertes. Le test ne porte que sur la palette générale des couleurs et le niveau de contamination (par exemple, présence de salissures, insertions de couleurs inutiles, pixels aléatoires de couleurs différentes sur le modèle). Les défauts habituels constatés à ce stade sont les cas où la cartographie automatique et la texturation subséquente sont effectuées de telle sorte que des textures de couleurs différentes sont appliquées aux parties du modèle qui ont un bord commun, ce qui donne lieu à des coutures perceptibles.

## Revue du modèle de LoD et de collision de l'objet

À ce stade, les artistes et les concepteurs de niveaux participent à l'évaluation de la qualité des modèles. Les concepteurs de niveaux évaluent les modèles de collision en fonction de leur impact sur le gameplay. Par exemple, ils déterminent s'il est nécessaire d'utiliser un modèle détaillé à un endroit précis ou s'il est possible de le simplifier. Les défauts habituels dans ce cas sont les modèles qui sont soit trop détaillés (par exemple, plus que le nombre autorisé de polygones dans le modèle de collision), soit pas assez détaillés (par exemple, contiennent un modèle de collision insuffisamment détaillé).

## Exportation d'un modèle vers un moteur de jeux vidéo

Il s'agit de vérifier l'exactitude des paramètres relatifs à la distance à laquelle il est possible de passer d'un LoD à l'autre et au nombre de points de vie de l'objet.

Les testeurs effectuent les tests directement dans le moteur du jeu vidéo, à la fois dans les éditeurs de contenu et directement dans l'application du jeu. Parallèlement, la plupart des tests artistiques se réduisent à une évaluation visuelle de l'objet, et un certain nombre de tests reproduisent les tests des artistes et des modélistes. Les tests abordent les points suivants :

- La visibilité lors de commutation des LoDs des objets, en tenant compte de l'emplacement et de la situation dans laquelle l'objet sera utilisé,
- Les lacunes et coutures dans la géométrie de l'objet qui sont perceptibles par le joueur,
- La visibilité des inscriptions sur les différents objets graphiques,
- L'effet de destruction du modèle (par exemple, en tenant compte de la couleur et de la taille.),
- Les défauts visuels dans l'animation des objets.

Lors des tests du graphisme, les joueurs peuvent directement prendre part, qui évaluent visuellement le contenu et peuvent trouver d'autres défauts graphiques.

Les tests du graphisme sont souvent effectués lors des tests des cartographies du graphisme. Dans ce cas, il y a plusieurs étapes supplémentaires.

## Placement d'objets sur la cartographie

C'est l'une des étapes les plus importantes pour tester. Le placement des objets est effectué par les concepteurs de niveaux. Au cours de ce processus, un défaut habituel apparaît : les objets "suspendus" et encadrés sous la carte (map). Cela se produit souvent lorsque de nombreux spécialistes travaillent en parallèle sur la carte et sont susceptibles de modifier la géométrie du terrain, d'exposer des objets et d'éditer la carte elle-même.

## Création et placement d'effets

À ce stade, les artistes des effets travaillent à la création de nouveaux effets spéciaux. L'observer dans diverses situations leur permet de détecter les défauts de rendu artistiques et techniques. Un exemple de test est le test visuel d'un effet dans un moteur. Les artistes d'effets regardent chaque effet dans le client



du jeu pour identifier les paramètres d'effet incorrects (par exemple, un temps de décroissance insuffisant de l'effet).

Les tests du graphisme sont souvent effectués lors de tests en situation de jeu. Il s'agit de tous les tests au cours desquels la carte est testée dans des situations de jeux vidéo. Les objectifs de ces tests sont vastes et s'appliquent à n'importe quelle étape de la production (par exemple, faire une évaluation du gameplay sur la carte). Plus près de la fin de la production, il peut s'agir d'une évaluation artistique et de gameplay de la carte du point de vue des joueurs. Outre la collecte de statistiques dont les concepteurs de jeux ont besoin pour ajuster l'équilibre de la carte, ces tests révèlent divers problèmes artistiques et techniques, allant des objets suspendus aux défauts de rendu du graphisme (par exemple, image figée, défauts et défaillances). Dans les tests purement artistiques, le test en situation de jeu révèle des problèmes perceptibles du point de vue du joueur.

### Tester la carte

Il s'agit de la clôture des tests, lorsque la carte est complètement prête et qu'il n'y a plus de travail sur elle (à de rares exceptions près). Les testeurs procèdent à une validation complète de la carte sur la base d'une liste de contrôle de validation de celle-ci, en recherchant les objets suspendus et en testant les animations. Les testeurs se concentrent sur ces tests, ce qui leur permet de détecter beaucoup plus de défauts (par exemple, des objets "suspendus") que les artistes. Lors des tests du graphisme, les artistes inspectent visuellement les cartes, tandis que les testeurs utilisent un ensemble d'outils spécialisés pour détecter des problèmes plus importants dans les objets du graphisme.

#### 3.3.2 Tests du graphisme pour l'exactitude historique

Le test de l'exactitude historique est un autre domaine du test du graphisme. L'exactitude historique prend en compte à la fois l'exactitude et l'authenticité historiques.

L'authenticité historique influence la qualité d'un jeu, dans lequel un épisode spécifique de l'histoire est décrit à l'aide d'images caractéristiques. Celles-ci ne font pas référence à des événements spécifiques, à des personnalités, etc.

L'exactitude historique influence la qualité d'un jeu, dans lequel une période spécifique de l'histoire est décrite. Tous les événements clés sont décrits dans leur dynamique, y compris toutes les personnalités concernées et leurs actions, à l'aide d'images caractéristiques de la période. Les images étrangères à la période sont fondamentalement exclues.

En règle générale, dans les jeux, il est impossible de maintenir l'authenticité historique, car l'accent est mis sur l'aspect addictif. Il est difficile d'impliquer les joueurs dans un jeu dont l'issue est prédéterminée dès le départ.

Cependant, l'exactitude historique est un domaine des jeux vidéo qui nécessite une attention particulière, surtout lorsque l'intrigue du jeu est liée à une époque historique spécifique.

Les aspects habituels à prendre en compte lors des tests du graphisme pour vérifier l'exactitude historique sont :

- La ressemblance des personnages avec des schémas historiques,
- L'exactitude lors de la reproduction d'objets architecturaux,
- L'exactitude dans la reproduction d'armes, d'équipements, de véhicules, de vêtements d'une époque donnée et de leurs caractéristiques,
- L'exactitude de la recreation de la vie quotidienne,
- L'exactitude des événements et des dates historiques.

Lorsqu'il teste l'exactitude historique, le testeur doit disposer d'un large éventail de connaissances et d'horizons. Très souvent, les développeurs font appel à des consultants historiens pour tester leurs produits afin de créer le jeu le plus réaliste possible.

### 3.4 Outillage des tests du graphisme

Lors des tests des jeux, les testeurs de graphisme utilisent un large éventail d'outils, allant des éditeurs de contenu du jeu aux outils automatisés et aux scripts de test. En règle générale, les jeux sont créés à l'aide de moteurs de jeux vidéo qui contiennent plusieurs outils intégrés, notamment pour tester le graphisme : un éditeur d'objets, un éditeur de monde et un éditeur d'éclairage.

Tous ces outils permettent de personnaliser le contenu existant pour que le joueur puisse interagir avec lui. Les éditeurs de modèles permettent de lier des effets aux modèles tels que divers déclencheurs, sons et autres événements qui se produisent lors de l'interaction avec le moteur. L'éditeur de monde permet de créer une cartographie, d'y placer des objets et de personnaliser les mécaniques de jeu de la carte. C'est dans ces éditeurs que les testeurs effectuent la plupart des tests. Pour ce faire, ils utilisent les mêmes outils que les développeurs. Ces éditeurs fournissent également une boîte à outils étendue qui permet de mettre en place le contenu et de le tester.

Des outils spécialisés sont habituellement développés par les fabricants de cartes graphiques. Ces outils permettent la capture d'images dans le jeu et l'analyse détaillée de toute application utilisant divers ensembles d'API utilisés pour créer des graphismes 2D et 3D. Ils permettent également d'évaluer les éléments suivants :

- La façon dont la trame du jeu est préparée (par exemple, la géométrie, les textures, (draw calls)),
- Où les problèmes de performance se posent,
- La géométrie de l'exécution,
- Le débogage pour les modeleurs 3D, les graphistes et les animateurs.

Plus de détails sur l'utilisation de chacun de ces outils sont décrits dans la spécification spécifique du logiciel.

Les tests du graphisme comprennent des scripts automatisés qui assurent le passage automatique du personnage le long d'un itinéraire donné dans le jeu. Ils sont particulièrement utilisés dans les tests de performance et de compatibilité

## 4. Tests du son - 190 minutes (K3)

### Mots-clés spécifiques du domaine des jeux vidéo

Ambiance, effet binaural, distorsion, bruit de décalage (Foley), occlusion, réverbération, saut, discontinuités sonores, effets sonores, zones sonores, volume.

### Objectifs d'apprentissage pour le chapitre 4

#### 4.1 Caractéristiques du contenu sonore du jeu vidéo

GaMe-4.1.1 (K1) Rappeler les caractéristiques du contenu sonore d'un jeu vidéo

#### 4.2 Types de défauts dans les contenus sonores

GaMe-4.2.1 (K1) Rappeler les types de défauts dans le contenu sonore

GaMe-4.2.2 (K2) Classer les défauts dans le contenu sonore

#### 4.3 Approches de test du contenu sonore dans les jeux vidéo

GaMe-4.3.1 (K2) Résumer les principales approches des tests du contenu sonore

GaMe-4.3.2 (K2) Résumer les principales approches pour tester le mélange de musique et de sons.

GaMe-4.3.3 (K2) Résumer les principales approches pour tester la composition musicale.

#### 4.4 Réalisation des tests sonores

GaMe-4.4.1 (K2) Expliquer les niveaux de tests des contenus audio-musicaux

GaMe-4.4.2 (K1) Rappeler les caractéristiques de l'intégration des sons dans le client

GaMe-4.4.3 (K1) Rappeler les domaines de responsabilité des tests effectués sur le son

GaMe-4.4.4 (K3) Appliquer les approches de test du son

#### 4.5 Outillage pour les tests du son

GaMe-4.5.1 (K2) Résumer l'utilisation d'outils de test pour le son

### 4.1 Caractéristiques du contenu sonore du jeu vidéo

Le son est un élément important de tout jeu vidéo moderne. Les sons et la musique créent l'ambiance adéquate, avertissent du danger, transmettent les émotions des personnages, complètent et créent une image complète de l'univers du jeu.

Une personne confrontée à un film ou à un jeu vidéo dépourvu de tout son est susceptible d'éprouver une légère dissonance. Cela se produit parce que votre cerveau essaie d'activer votre ouïe, mais il n'y parvient pas et signale plutôt une erreur de perception sur ce qui se passe devant vos yeux.

Des sons de mauvaise qualité, irréalistes, inappropriés ou fonctionnant mal peuvent irriter le joueur et perturber le gameplay.

Le processus de création d'une bande sonore dans un jeu vidéo est assez long et fastidieux, car il est nécessaire de prendre en compte la complexité de l'organisation du contenu du jeu vidéo :

- Les sons environnementaux,
- Les voix des personnages,
- La bande sonore d'arrière-plan,
- Les effets sonores,
- Les divers objets sonores,
- Les fichiers vocaux.

Cela complique la tâche pour les tests du son.

#### 4.1.1 Types de sons

La conception sonore des jeux vidéo peut varier d'un jeu vidéo à l'autre et être créée à des fins différentes. Examinez les types de sons qu'il est d'usage de distinguer, les fonctions qu'ils remplissent et les raisons pour lesquelles ils sont créés.

##### **La musique**

Le thème musical principal du jeu, sa carte de visite, peut apparaître même dans le jeu le plus simple. Les mélodies des vieux jeux vidéo 8 bits sont depuis longtemps des classiques.

Les principales fonctions de la musique :

- Met l'accent sur le côté épique du moment,
- Renforce la dynamique de l'événement,
- Augmente l'atmosphère,
- Met le joueur dans l'ambiance et explique l'état émotionnel du personnage,
- Forme inconsciemment une certaine réactivité émotionnelle chez le joueur en raison de la mélodie répétitive.

##### **Les effets sonores**

Si le monde du jeu est rempli d'objets avec lesquels le héros peut interagir, chacun d'entre eux doit être associé à des sons réalistes. Cela rend le monde du jeu vivant et crédible.

L'ouverture des portes, les coups, le tir et le rechargement des armes, l'utilisation de la trousse de secours, l'explosion des bidons d'essence, le bruit des vitres qui se brisent - tout cela est nécessairement exprimé et joué au bon moment.

Même dans les petits jeux, les développeurs ajoutent des effets sonores d'interaction avec les éléments d'interface (boutons, interrupteurs, éléments de menu). Cela permet au joueur d'avoir un retour d'information et de comprendre que ses actions sont perçues et analysées.

##### **Sons des personnages (Foley)**

Les sons émis par les personnages dans le jeu comprennent le bruissement des vêtements, la respiration, les exclamations, les pas, les gémissements, etc. Le terme est nommé d'après Jack Foley, l'un des pionniers des effets sonores. Dans certains jeux, comme les jeux de tir, ces sons peuvent indiquer que le personnage est blessé, ce qui est encore plus clair pour le joueur que les chiffres de dégâts affichés dans les fenêtres.

##### **La parole**

Bien que les paroles soient reproduites par les personnages, il est préférable de les mettre en évidence séparément. Dans les jeux vidéo actuels, chaque personnage non joueur (non-player character - NPC) a une voix unique, enregistrée par un acteur professionnel. Les développeurs consacrent beaucoup d'argent et d'efforts à l'interprétation des voix, car des répliques enregistrées avec qualité peuvent transmettre correctement les émotions des personnages et créer un lien émotionnel entre eux et le joueur.

##### **Sons ambiants**

Les sons d'ambiance font référence aux sons caractéristiques d'un lieu particulier. Comme le thème musical, l'ambiance met l'accent sur l'atmosphère et crée l'ambiance nécessaire au joueur. Le son ambiant n'implique aucun objet d'où il provient. Il s'agit d'un son environnemental qui accompagne simplement un certain lieu, une certaine situation ou une certaine étape du cycle de jeu.

Les sons d'ambiance sont généralement joués en arrière-plan et ne dépendent pas des actions du personnage. Des exemples de tels sons sont :

- Bruissement des feuilles et chant des oiseaux dans la forêt,
- Éclaboussement des vagues sur la mer,
- Conversation indistincte des visiteurs d'un saloon,
- Sirènes de police et bruit des voitures qui passent, etc.

Tout cela aide le joueur à naviguer dans l'espace et à mieux percevoir le monde environnant du jeu.

#### 4.1.2 Effets et technologies du son

Les bandes sonores sont souvent traitées par des éditeurs de son pour créer un effet d'authenticité, en tenant compte des capacités de l'équipement et de la technologie audio modernes.

Pour immerger l'utilisateur dans le monde virtuel du jeu, l'utilisation de sons réalistes peut ne pas suffire. Pour renforcer le réalisme de ce qui se passe, les développeurs hyperbolisent souvent certains sons. Cette technique est souvent utilisée, par exemple, dans les films comportant des scènes de combat, où les coups et même les mouvements des combattants sont beaucoup plus forts que dans la réalité.

Pour obtenir cet effet, diverses techniques et technologies sont utilisées.

#### **Occlusion**

L'une des techniques les plus courantes est l'effet d'occlusion, lorsque le son passe à travers une barrière (par exemple, le mur d'un bâtiment). Pour plus de réalisme, les spécialistes du son, dans de telles situations, ne se contentent pas de baisser le volume, mais ajoutent un effet spécial.

Le son principal et les sons réfléchis sont réalisés différemment lorsqu'il y a un mur vide entre la source sonore et l'auditeur, lorsque les objets se trouvent dans des pièces différentes mais en visibilité directe (porte), et lorsque les objets se trouvent dans la même pièce mais ne peuvent pas se voir (colonne entre eux).

#### **Réverbération**

L'effet de réverbération est utilisé pour transmettre le volume et la profondeur de l'espace (à partir de la réverbération).

Dans ce cas, le son de la source dans un espace clos est réfléchi par les murs et provoque de nombreux échos qui s'estompent progressivement. Cet effet fonctionne bien si le personnage se trouve dans une grotte, une petite pièce ou toute autre pièce présentant des surfaces sonores bien réfléchissantes.

Avec des moteurs sonores avancés, on peut facilement créer des échos et ajouter du volume aux bruits de tirs, d'explosions, de voix, de bruits de pas, etc.

#### **Effet binaural**

L'effet binaural (du latin bini - deux, paire et auris - oreille) est basé sur le fait qu'une personne écoute avec deux oreilles simultanément et que lorsqu'elle tourne la tête sur le côté, le son atteint une oreille avant l'autre. Grâce à cela, une personne peut déterminer dans quelle direction se trouve la source du son, et même la distance qui la sépare d'elle.

Des microphones spéciaux sont utilisés pour enregistrer le son avec cet effet, et pour sa perception, il est recommandé d'utiliser un casque. Cela est nécessaire pour que les sons puissent arriver dans chaque oreille à partir d'une direction spécifique. L'environnement sonore rend l'immersion plus profonde et peut aider le joueur. Grâce à l'audio binaural, le joueur sait de quel côté vient le son (par exemple, l'endroit où l'ennemi apparaîtra dans un instant).

### 4.1.3 Zone des sons

Dans certains projets, des zones sonores sont utilisées pour distribuer l'environnement sonore à différents endroits.

Dans les jeux vidéo, par exemple, lorsqu'un personnage d'un jeu de rôle entre dans un village et que les sons caractéristiques apparaissent, ou dans un jeu de stratégie, lorsque le joueur approche la caméra de sa base et entend les sons de la collecte des ressources.

Les zones sonores doivent comporter des transitions. Près de la transition, le son d'une zone doit s'atténuer légèrement et le son de l'autre zone doit apparaître.

Un exemple de l'effet du zonage sonore dans la vie réelle peut être observé en ouvrant et en fermant les fenêtres de votre pièce. Lorsque vous ouvrez la fenêtre, les bruits de la rue sont clairement audibles, alors que lorsque vous la fermez, vous ne les entendez pas du tout ou ils sont extrêmement étouffés.

## 4.2 Types de défauts dans les contenus sonores

Il existe trois catégories de défauts de contenu sonore :

- Absence de son,
- Son incorrect,
- Lecture incorrecte du son.

### **Effet sonore manquant**

Aucun son n'est joué lorsqu'il devrait l'être, pour une certaine action du personnage ou à un certain moment.

### **Lecture d'un son incorrect**

Réglage incorrect du son de l'objet ou de l'environnement, par exemple, un coup de canon ressemble à un coup de pistolet, un bateau à moteur ressemble à un avion, ou les personnages (dans le jeu et hors du jeu) disent les répliques de quelqu'un d'autre.

L'effet sonore viole l'authenticité historique, par exemple, les personnages du jeu utilisent un vocabulaire moderne, des citations de films et de musique, le char AMX-30 sonne comme un char Leclerc. Rare, mais peut être significatif si le réalisme et l'authenticité sont déclarés comme une valeur clé du jeu testé.

### **Lecture incorrecte du son correct**

### **Chute du son**

Lorsqu'un tel défaut se produit, certains des sons reproduits peuvent disparaître, comme lorsque vous êtes au téléphone avec une mauvaise connectivité. De ce fait, le lecteur peut, par exemple, se méprendre sur le sens d'une phrase dans un dialogue. Si c'est le seul son qui est actuellement joué, les testeurs constateront facilement le problème. Mais si plusieurs sons sont joués en même temps et que l'un d'entre eux disparaît, il est alors beaucoup plus difficile de détecter un tel défaut.

### **Sautillement**

Le saut dans le son de la musique ou des effets audio peut être lié à plus qu'un simple dommage au fichier son lui-même. Les sauts sont souvent dus à des problèmes de performance. Dans ce cas, ils se produisent lorsque la fréquence d'images échoue.

### **Distorsion**

En raison de problèmes de performance, certaines phrases peuvent sembler déformées et inaudibles.

### **Retard de lecture**

Dans les jeux vidéo, il y a des moments où la conception sonore est en retard par rapport à l'animation, à l'objet ou à la situation de jeu affichée.

### **Exemples de défauts dans le contenu sonore et leurs causes possibles**

1. Absence de son de l'objet/environnement.

Défaut : Lorsqu'il se déplace sur une surface métallique, il n'y a pas de bruit de pas, le personnage se déplace silencieusement.

Cause : Le développeur a oublié de paramétrer ou de connecter le son sur cette surface.

2. Le son de l'objet ou de l'environnement est trop fort/silencieux.

Défaut : Un certain objet dans le lieu brûle, mais le bruit du feu est trop fort, ou au contraire très silencieux, ce qui semble irréaliste.

Cause : Le volume de l'effet est mal réglé.

3. Le son de l'objet ou de l'environnement est mal réglé.

Défaut : Lors de la conduite d'un bateau à moteur sur l'eau, le jeu émet le son d'une voiture de course.

Cause : Un fichier son incorrect pour l'objet a été écrit dans le code.

4. Positionnement incorrect du son en fonction de sa source.

Défaut : Le jeu présente un forgeron qui frappe continuellement son épée avec un marteau, mais le son de la frappe est joué dans un autre coin de la pièce.

Cause : Dans l'éditeur de sons, l'effet sonore est réglé sur une position incorrecte.

5. La zone de couverture sonore est mal configurée.

Défaut : Le son d'un coup de feu n'est entendu que dans un rayon de quelques mètres. Si le joueur s'éloigne, le son n'est pas entendu du tout, ce qui n'est pas réaliste.

Cause : Le rayon du son est mal configuré.

6. Distorsion du son.

Défaut : Les joueurs constatent que le son commence à grésiller et à siffler.

Cause : Un fichier audio corrompu est utilisé.

7. Répétition continue du son.

Défaut : Le son est joué en boucle et en continu.

Cause : Un pointeur indiquant à quel moment la boucle sonore doit s'arrêter n'est pas réglé correctement.

8. Distorsion du son sous forme de "bégaiement" de la lecture.

Défaut : Une distorsion de la conception sonore du mouvement se produit pendant le mouvement d'un des personnages du jeu.

Cause : Un problème peut survenir en raison de problèmes liés à l'assemblage du son ou en raison de l'instabilité générale du client du jeu vidéo.

9. Décalage temporel de la lecture audio.

Défaut : Lors d'une scène, un personnage commence à parler, mais sa réplique est reproduite avec un peu de retard par rapport au moment où ses lèvres commencent à bouger.

Cause : l'animation du mouvement des lèvres du personnage est désynchronisée par rapport au début du fichier sonore correspondant.

### 4.3 Approches de test du contenu sonore dans les jeux vidéo

La conception sonore du jeu peut être représentée sous la forme d'effets, de voix de personnages ou de musique. Les tests du contenu audio des jeux peuvent être effectués de manière plus formelle ou informelle. Les approches formelles des tests audio consistent à tester la présence, l'exactitude, le volume et l'intégrité de tous les fichiers audio. Le test de sons informel est une sorte d'évaluation du réalisme ou de la maintenabilité de l'atmosphère du jeu. Par exemple, une musique "lourde" accompagnant le lieu dans les jeux du genre horreur pour accentuer la situation et maintenir périodiquement le joueur dans un état de tension.

Le testeur dérive des cas de test à partir de la spécification d'un objet sonore particulier révisée précédemment. À partir de cette spécification, le testeur doit comprendre sans ambiguïté à quels endroits, dans quelles scènes de jeux et dans quelles situations le son doit être joué. Dans certains cas, cela peut nécessiter une consultation avec un concepteur audio pour connaître les détails spécifiques du contenu sonore créé. Dans d'autres cas, le testeur peut utiliser l'éditeur de ressources du jeu vidéo pour tester la synchronisation des fichiers sonores avec les actions du joueur ou l'environnement du jeu.

Tester la conception sonore du jeu par un testeur peut comporter plusieurs étapes.

#### 4.3.1 Tests auditifs du contenu

Le testeur écoute les sons et teste les paramètres audio du contenu créé à l'aide de l'éditeur de sons ou de l'éditeur de ressources de jeux vidéo. Tous les fichiers sonores doivent être testés.

Il s'agit de l'une des étapes les plus approfondies. Les caractéristiques sonores suivantes sont vérifiées par rapport aux exigences :

- Être adapté à l'objet et au décor. Par exemple, si le personnage porte une armure de fer, les sons du métal doivent dominer ceux du tissu lorsqu'il se déplace. L'explosion faiblement audible d'un bâton de dynamite ou d'un rocher tombé, le piétinement évident du trottoir lors d'un déplacement sur le sable, peuvent perturber l'immersion dans le monde virtuel.
- Les voix des acteurs. Les voix des acteurs pour les phrases entièrement doublées doivent correspondre aux personnages qui les prononcent et au cadre du jeu. Généralement, la voix doit correspondre au sexe et à l'âge de l'acteur. C'est fait pour maintenir l'atmosphère générale, par exemple, le personnage sudiste a un accent du sud, et non un accent irlandais moderne.
- Non gênant. Des échantillons de plusieurs sons réunis peuvent parfois devenir désagréables pour l'oreille humaine.
- Niveau de volume. Le volume de chaque fichier sonore doit être le même lors de l'utilisation de plusieurs fichiers au sein d'un même jeu.
- Effets sonores. Les effets appliqués à la fois aux sons et à la voix off doivent également être utilisés correctement. Si le lieu, ou la scène, est rempli de divers détails, ces derniers sont mis en valeur par la voix off. Dans un grand bureau, une personne peut entendre le ronronnement des ordinateurs, le bourdonnement des imprimantes, le bruit des climatiseurs, la pression sur les claviers et les conversations des travailleurs.

Le doublage soigné de tous les objets contribue à la création d'une image réelle. Par exemple, lorsque le joueur se retrouve dans le village, il n'entendra pas nécessairement le meuglement des vaches, mais il s'attendra très probablement aux sons caractéristiques d'une scierie si celle-ci apparaît devant lui. Un petit ruisseau dans les bois peut très bien n'avoir aucun son associé, mais si le héros se trouve sur place à côté



d'une cascade, l'absence du bruit de l'eau ne peut s'expliquer que par un défaut (ou un son désactivé dans les paramètres).

#### 4.3.2 Tester le mélange de musique et de sons du jeu

La taille du projet et les compétences professionnelles de l'équipe de développement varient d'un projet à l'autre, ce qui ne permet pas de distinguer clairement les fonctions exercées par les rôles. Il peut arriver qu'il ne soit pas nécessaire de faire appel à un acteur vocal, que le travail d'un bruiteur (spécialiste de la création de bruits) soit confié à un concepteur audio, voire qu'un spécialiste crée ou achète lui-même des échantillons prêts à l'emploi, les associe au client et teste le son obtenu.

Le rôle d'un concepteur audio est de créer des sons et d'enregistrer des fichiers audio. Ces fichiers audio doivent être testés pour différentes utilisations. Le système sonore d'une salle d'arcade est très différent de celui d'un téléphone portable - et l'audio doit être testé. Les jeux vidéo doivent être joués correctement à la fois sur les petits haut-parleurs intégrés des téléviseurs et sur les systèmes de home cinéma.

Même un son de qualité et apparemment approprié peut être source de confusion et d'irritation pour le joueur si le mixage final n'est pas incorrect. Par exemple, le son de l'action dure beaucoup plus longtemps que l'action elle-même, ou son volume n'est pas adapté.

Le testeur teste l'image sonore finale et l'exactitude des modifications apportées au son dans le client, à l'aide de l'éditeur de ressources intégré au moteur du jeu vidéo ou en se basant sur sa propre expérience ou sur la documentation fournie.

Étant donné que durant le jeu, le joueur entend un grand nombre de sons provenant de différents objets et de la musique jouée en arrière-plan, il est important que tous les composants audio soient en harmonie les uns avec les autres. Le testeur doit évaluer le composant sonore du projet et identifier les écarts éventuels, les points irréalistes concernant le volume, ainsi que la qualité et le naturel du son en se basant sur le bon sens. Il est également important d'auditer la précision de l'emplacement de la source sonore par rapport aux objets eux-mêmes. Par exemple, dans le jeu, lorsqu'un personnage ouvre une porte, le joueur entend le son de l'ouverture de la porte non pas depuis la porte, mais de manière incorrecte depuis le dos du personnage. Il est également important que le testeur soit en mesure d'évaluer correctement le moment où les sons sont joués par rapport à ce qui se passe à l'écran.

#### 4.3.3 Tester la composition de la musique

La composition ou le mixage est l'étape qui consiste à créer un enregistrement final à partir de pistes enregistrées individuellement. C'est l'étape qui suit l'enregistrement audio et qui consiste à sélectionner et à éditer (parfois à restaurer) les pistes enregistrées originales, à les combiner en un seul projet et à les traiter avec des effets. L'édition est souvent une étape de travail indépendante.

Le mixage dans les jeux vidéo qui utilisent de la musique électronique est l'étape qui suit la création du contenu audio. L'étape de l'enregistrement sonore d'un projet électronique est le plus souvent absente. La frontière entre la création et le mixage de la musique électronique est floue ; le projet arrive au mixage déjà partiellement mixé, car de nombreux synthétiseurs virtuels disposent déjà d'un traitement par gabarit des différents sons.

En conséquence, un projet multicanal est converti en phonogramme monophonique, stéréophonique ou multicanal, qui prend généralement sa forme définitive au cours d'un processus appelé "mastering".

Une fois le mastering clôturé, le test de la musique masterisée commence. Le testeur doit tester le composant musical pour détecter les défauts, les discontinuités sonores et les interférences et faire un rapport des défaillances constatées.

## 4.4 Réalisation des tests sonores

### 4.4.1 Niveaux de test du contenu audio-musical au cours du cycle du développement logiciel des jeux vidéo

Souvent, les tests sonores commencent tard dans le développement. Alors qu'un développeur crée le contenu d'un jeu (modèles d'objets/personnages, cartographies, objets, etc.), le composant sonore n'est élaboré qu'à un stade ultérieur. Au stade du développement, un objet peut ne pas avoir de son, mais un objet lui-même a déjà un modèle fini avec une animation personnalisée intégrée dans le client et prête à être testée.

Les échantillons sonores doivent être testés au préalable par les spécialistes qui ont créé le son (c'est-à-dire les concepteurs audio, les artistes Foley, les spécialistes des effets sonores). Ils peuvent participer aux tests d'un son particulier et l'envoyer ensuite au service de développement pour qu'il soit intégré au client.

### 4.4.2 Intégrer les sons dans le jeu

Après intégration de la conception sonore dans le jeu, le testeur évalue les caractéristiques suivantes des sons :

- Possibilité d'activer et de désactiver les sons et/ou la musique.
- Possibilité de modifier le volume.
- Conformité à un format unique des noms de fichiers audio et leur répartition correcte dans les dossiers du projet.
- Localisation des sources sonores, des zones de distance de propagation du son sur la cartographie du jeu.
- Performances du système avec les sons intégrés.
- Compatibilité des sons sur différents systèmes audio : casque, haut-parleurs, enceintes de moniteur.

### 4.4.3 Domaines de responsabilité des intervenants

Le processus de création de la musique et de la conception sonore du projet fait intervenir plusieurs types de spécialistes différents, remplissant certaines fonctions :

Rôle	Responsabilités
Compositeur	Participe à la création de sons et de musiques pour le jeu vidéo.
Ingénieur audio	Participe à l'exploitation des équipements d'enregistrement, d'amplification du son et de diffusion.
Ingénieur du son	Spécialiste qui possède une expertise maximale en matière de son et qui détermine l'image sonore finale de chaque objet, de chaque scène de jeu et du jeu vidéo dans son ensemble.
Concepteur audio	Crée du contenu à partir des fichiers audio originaux avec des sons enregistrés (échantillons) et l'ajuste dans l'éditeur de son.
Artiste de doublage	Rend les sons qui existent dans la réalité plus vivants et plus riches, et est responsable de la création d'effets sonores qui n'existent pas dans la réalité. En combinant les sons d'objets ordinaires, ils enregistrent des échantillons et créent à partir d'eux les effets, par exemple, d'un bourdonnement d'épée laser, d'un grognement de zombie ou d'un claquement d'ouverture de portail.

Rôle	Responsibilités
Acteur vocal	Joue avec sa voix pour donner vie à un personnage qui aura des répliques audibles dans le jeu.
Développeur	Intègre le contenu audio créé et personnalisé dans le code client du jeu.
Concepteur de niveau	Distribue les sources sonores par niveau et lie ensuite les fichiers sonores aux sources sonores.
Concepteur narratif	Spécialiste responsable de l'histoire et de la réactivité dans le développement des jeux vidéo. Il/elle travaille sur le son et la musique joués pendant les moments liés au développement du scénario (scènes, intros, dialogues, moments importants du jeu, etc.)
Concepteur de jeux vidéo	Participe à l'élaboration de la documentation du jeu vidéo (par exemple, un document de conception du jeu vidéo). Ce document décrit les règles et les caractéristiques du jeu dans un langage simple. Ainsi, avant même que le moteur ne soit développé, le concepteur de jeux développe une vision holistique du jeu. Au cours du processus de développement, il participe également en partie au rôle de "consultant", en approuvant la conformité des propositions/modifications à l'idée principale du projet. Il participe également personnellement au test de jeux vidéo.
Testeur	Réalise les tests de l'image sonore finie et l'exactitude des nouvelles modifications sonores apportées par le client.

#### 4.4.4 Procédures et approches de la conduite des activités de test dans le cadre du test d'objets sonores

Lors du test des objets de jeux vidéo, le testeur doit implémenter les procédures et les actions nécessaires pour obtenir des informations complètes sur la manière dont le son est correctement configuré dans une version particulière du jeu. La liste des actions dépend directement des objets qui doivent être testés. Par exemple, pour le test final de la conception sonore des armes ajoutées dans le client, le testeur effectue un certain nombre d'actions : ajout d'armes au cas où elles ne seraient pas initialement disponibles dans le client, chargement de nouveaux contenus sur la cartographie de test et toute la liste des actions nécessaires pour tester au maximum tous les sons attachés à l'arme (tir, rechargement, rechargement à nouveau, clic pour changer de mode de tir, etc.)

Il est important de comprendre que la liste des procédures et démarches pour chaque objet est différente. Par exemple, si c'est un objet de l'environnement qui est testé et non l'objet avec lequel le personnage interagit, alors le testeur teste l'inclusion de la conception sonore elle-même - par exemple, le murmure d'un ruisseau ou d'une rivière lorsque le personnage s'en approche. Le testeur doit savoir où trouver les informations sur tous les sons disponibles pour un objet particulier. Il doit également tirer parti de l'éditeur de ressources et de l'éditeur de sons intégrés pour tester l'intégrité du contenu sonore et le synchroniser avec les actions du personnage et l'environnement du jeu.

## 4.5 Outillage pour les tests du son

### Editeur audio du moteur de jeux vidéo

Le testeur n'a pas toujours accès à l'éditeur audio du moteur de jeux vidéo. C'est pourquoi, lors des tests, il utilise un ensemble standard de logiciels pour la validation du contenu. Ce type d'éditeur est utilisé par les testeurs travaillant au sein de l'équipe de développement (s'il y en a une), ou par des spécialistes travaillant directement dans l'éditeur audio (par exemple, les compositeurs, les concepteurs audio, etc.).

Les concepteurs audio doivent participer aux tests. Ils testent les ressources, notamment en équilibrant les niveaux, en réglant les égaliseurs et en mixant/alignant toutes les ressources ensemble et individuellement. Dès que les ressources sont ajoutées à la version de travail du jeu, les concepteurs audio doivent pouvoir les récupérer facilement et les modifier si nécessaire.

Pour faciliter la maintenance du contenu sonore, tous les échantillons sonores doivent se trouver sur des pistes différentes, être classés dans des dossiers, nommés, horodatés et dotés d'une version. Il est également efficace de convertir les pistes audio au format numérique compressé, afin de ne pas alourdir les ressources matérielles de l'adaptateur de son.

### **Éditeur de cartographie/lieux**

Parfois, le testeur a accès au logiciel de développement, le plus souvent à l'éditeur de cartes/lieux. Cela permet au testeur d'étendre ses capacités lorsqu'il teste le contenu audio et tout le contenu qui se trouve dans le client. Les développeurs de jeux ajoutent des fonctionnalités aux éditeurs de cartes qui soutiennent le gameplay "simulé" directement sur la carte du niveau. Ainsi, le testeur peut effectuer les tests nécessaires sans passer de l'éditeur de cartes au client de jeu lui-même, ce qui représente un gain de temps considérable.

La plupart des effets sonores et des sources sonores disponibles sur la carte sont arrangés et ajustés à l'aide de l'éditeur de niveaux. Dans l'éditeur de cartes, le testeur peut voir l'emplacement des sources sonores sur les lieux du client, liées à différents objets dans le lieu, ou aux personnages. En conséquence, lorsque le spécialiste teste la source sonore à l'aide de l'éditeur, il peut tester le son joué et la distance de propagation du son, soit sous la forme d'une sphère, soit sous la forme de "zones" où le testeur ou l'utilisateur final commence à entendre le son/la musique joué(e), provenant d'une source sonore liée, lorsqu'il/elle franchit les limites de ces zones.

Si la propagation du son est implémentée sous forme de "zones", le testeur teste la lecture du son provenant de la source lorsqu'il franchit la limite de la "zone". Il est également utile de tester l'augmentation du volume à l'approche de la source sonore afin de rendre le son lié plus réaliste. Le test de la distance de propagation du son sous la forme d'une sphère ne diffère que par la forme de la zone de propagation. L'éditeur permet également de tester la liaison, l'emplacement et, respectivement, la configuration de la source sonore de chaque client.

Grâce à ces informations, le testeur peut évaluer le réalisme du contenu sonore, la distance de propagation du son, la justesse des sons (sur l'emplacement, les objets, les personnages) et indiquer au développeur les défauts en cas de détection. Par exemple, créer une description d'un défaut concernant une distance mal définie sur une source sonore particulière. Ou le volume d'un son particulier (se référant à une source sonore particulière) est trop fort par rapport à l'arrière-plan des sources "voisines". Dans la description des défauts, le testeur, par l'intermédiaire de l'éditeur, peut faciliter le travail des développeurs, en leur permettant de trouver plus facilement les zones problématiques en signalant un problème particulier.

## 5. Tests des niveaux de jeu - 65 minutes (K2)

### Mots-clés du domaine des tests logiciels

Tests en situation de jeu

### Mots-clés spécifiques du domaine des jeux vidéo

Code couleur, niveau de jeu vidéo, lieux inaccessibles, éditeur de niveau, profil de niveau, narration, géométrie structurelle, déclencheur.

### Objectifs d'apprentissage pour le chapitre 5

#### 5.1 Principes et concepts de la conception des niveaux de jeu

GaMe-5.1.1 (K1) Rappeler les composants des niveaux de jeu

GaMe-5.1.2 (K2) Classer les défauts habituels dans les niveaux de jeu

#### 5.2 Étapes et réalisation des tests de niveau de jeu

GaMe-5.2.1 (K2) Résumez les tests effectués aux différents stades de la création des niveaux de jeu.

GaMe-5.2.2 (K2) Comparer les domaines de responsabilité des spécialistes qui participent aux tests des niveaux de jeu

#### 5.3 Outillage des tests des niveaux de jeu

GaMe-5.3.1 (K2) Résumer l'utilisation des outils permettant de tester les niveaux de jeu

## 5.1 Principes et concepts de la conception des niveaux de jeu

### 5.1.1 Le terme "niveau" et sa spécificité en fonction du genre du projet de jeu

Le niveau de jeu est une zone distincte du monde virtuel du jeu, dans laquelle le joueur doit accomplir une certaine tâche : trouver un trésor, vaincre tous les adversaires ou simplement atteindre la sortie. La tâche générale requise pour terminer un niveau est inextricablement liée à l'emplacement et se compose souvent de nombreuses petites tâches et quêtes nécessaires pour terminer un niveau ou de quêtes secondaires, qui ne sont pas nécessaires pour terminer un niveau. Dans de nombreux moteurs de jeux vidéo, les niveaux de jeu sont séparés par un écran de chargement.

La conception des niveaux est l'étape du développement des jeux vidéo associée à la création des cartographies, des lieux, des tâches, des missions et d'autres environnements pour les niveaux du jeu. La conception des niveaux comprend l'apparence des objets et des mécaniques du jeu, les obstacles sur le chemin du joueur, le scénario du jeu et d'autres éléments qui créent collectivement l'expérience de jeu prévue.

Pour créer des niveaux, un logiciel spécial est généralement utilisé - un éditeur de niveaux.

Les niveaux de jeu sont constitués de nombreux composants reliés les uns aux autres pour former un tout. Les principaux éléments sont les suivants :

#### Géométrie structurelle

La géométrie structurelle, ou géométrie de niveau de base, est l'un des éléments les plus importants et les plus formateurs de tout espace de jeu. C'est la géométrie structurelle qui définit le relief du terrain et la surface sur laquelle les personnages du jeu peuvent se déplacer.

La géométrie structurelle limite également l'espace de niveau disponible pour le déplacement des personnages. Par exemple, il peut y avoir des montagnes infranchissables ou d'autres caractéristiques du terrain aux limites d'un niveau.

Si un niveau comporte des points d'entrée et de sortie clairement indiqués, la géométrie structurelle peut fournir des indications sur l'endroit où le joueur doit se rendre pour atteindre la fin du niveau.

Selon l'endroit où se déroule le jeu, divers éléments peuvent faire office d'objets formant le niveau. Par exemple, dans la ville : les bâtiments, les rues, les ponts et les passages souterrains ; à l'extérieur, les éléments du paysage naturel : les champs, les collines, les rochers, les ravins et les grottes.

### **L'environnement du jeu vidéo**

Les objets de l'environnement du jeu sont utilisés par le concepteur du niveau pour créer une image plus crédible de l'espace de jeu requis.

Par exemple, une rue de village est composée de maisons, de dépendances et de routes, et le concepteur la remplit donc d'objets de différentes tailles. On distingue les grands objets tels que les arbres, les clôtures, les charrettes, les objets de taille moyenne tels que les puits, les tonneaux, les bancs, ainsi que les petits objets tels que les plantes du jardin, les pierres, etc.

### **L'éclairage**

L'utilisation de différents types de sources lumineuses permet d'obtenir l'atmosphère souhaitée au niveau du jeu. Elle permet par exemple d'orienter le joueur dans la bonne direction, d'attirer son attention ou de le cacher des adversaires.

### **Accompagnement sonore**

Bande sonore d'arrière-plan ou effets sonores joués pour diverses actions ou événements. Pour en savoir plus, voir le chapitre "4. Test du son".

### **Fonctionnalité du jeu**

La fonctionnalité de jeu comprend un système de réglages permettant d'organiser le gameplay à un niveau spécifique. On distingue les types de paramètres suivants :

Paramètres liés au passage du niveau. Il s'agit notamment :

- des points d'apparition des personnages et de leurs adversaires ;
- des conditions de victoire et de défaite ;
- des points de sauvegarde automatique du jeu ;
- du système de réglages de déclenchement qui lancent certains événements du jeu, ou scripts. Par exemple, toutes les deux minutes, un avion survole le champ de bataille, ou lorsqu'un personnage entre dans un bâtiment, une vidéo de l'histoire est jouée, ou encore lorsqu'un garde franchit la frontière d'une zone protégée, l'alarme est déclenchée.

Mise en place des objets avec lesquels le joueur peut interagir, par exemple les portes, les pickups, les objets destructibles, les pièges, etc.

Paramètres de la structure physique du niveau qui empêchent le personnage de sortir du niveau ou de rester coincé dans sa géométrie.

En règle générale, les niveaux de jeu sont cachés aux joueurs et ne sont visibles que dans l'éditeur de niveau.

#### **5.1.2 Comprendre les types de défauts dans la conception des niveaux**

Un certain nombre de défauts sont très fréquents lors de la création de niveaux. Le suivi de ces défauts peut s'avérer difficile, de sorte qu'ils peuvent rester inaperçus au niveau pendant un certain temps.

Certains outils dédiés à la création de cartes disposent d'une fonctionnalité intégrée permettant de détecter de tels défauts sur les cartes. Les concepteurs de niveaux utilisent souvent ces outils lors des dernières

étapes de la création d'une carte ou d'un niveau. Mais dans la plupart des cas, la meilleure façon de constater les défauts d'une carte est de la faire tester par des joueurs expérimentés, capables de détecter le problème et de le rapporter.

Une part importante des défauts constatés aux niveaux est liée à l'apparence et à l'emplacement des objets du jeu, à leur éclairage, au modèle de collision, etc. Une description détaillée de ces défauts est donnée dans le chapitre "3. Tests du graphisme".

Cependant, les défauts suivants sont directement liés à la conception des niveaux.

### **Géométrie**

Les défauts de géométrie sont des situations où le personnage du joueur est partiellement coincé dans certaines parties du niveau / de la carte. En règle générale, ces défauts sont constatés sur les bords de la carte, les corniches, les pentes. Le joueur guide son personnage vers un objet ou marche à proximité d'un objet et se retrouve coincé. Souvent, à l'aide de sauts, de flexions et d'autres mouvements, le personnage parvient à se sortir d'une telle situation, mais dans certains cas, cela ne sert à rien. Dans ces cas-là, le joueur doit recommencer le niveau/la carte, revenir au dernier point de sauvegarde de la progression ou utiliser une combinaison spéciale de boutons/touches pour se déplacer vers l'endroit le plus proche sans défaut de collision.

Par exemple, un personnage de jeu peut se retrouver coincé dans la géométrie de la carte, incapable d'en sortir, ou dans des zones dont l'accès devrait lui être interdit. Il est particulièrement important d'identifier de tels défauts dans les jeux vidéo multijoueurs, où les défauts de niveau peuvent placer un joueur ou une équipe dans une meilleure position.

Une autre situation fréquente est la mort en étant bloqué, c'est-à-dire que le personnage tombe visuellement sous la surface du niveau/de la carte et chute. En fin de compte, soit le personnage meurt et le niveau doit être recommencé, soit il tombe sans fin dans le vide et le jeu doit être redémarré.

### **Endroits inaccessibles**

Dans les jeux vidéo, en particulier sur les cartes des jeux multijoueurs, il peut y avoir des endroits intentionnels dans lesquels le joueur obtient une meilleure position que les autres joueurs. Par exemple, un mitrailleur situé sur une colline disposera d'un vaste secteur de tir. Bien que le joueur qui est le premier à prendre une telle position gagne un certain avantage sur ses adversaires, il s'agit d'une intention du développeur, d'un élément de conception du niveau, et non d'un défaut.

Mais il y a des situations où un tel endroit est créé par accident en raison d'une erreur de la conception du niveau. Selon le concept du développeur, cette zone devrait être inaccessible aux joueurs. Cependant, un certain personnage du jeu, grâce à ses caractéristiques uniques (par exemple, des sauts plus élevés que les autres), peut s'y retrouver. De cette manière, le joueur peut obtenir un avantage indu dans le jeu par rapport à ses rivaux. Par exemple, les autres joueurs ne pourront pas le détecter, lui apporter des dégâts, etc.

### **Gameplay compliqué**

Problèmes liés à l'absence de pointeurs d'objectifs ou à la non-évidence des actions requises pour terminer le niveau. Par exemple, le joueur a vaincu tous les adversaires dans le territoire accessible, mais ne comprend pas ce qu'il faut encore faire pour aller plus loin.

### **Niveaux d'équilibre du jeu**

De nombreux défauts apparaissent lors de la définition de l'équilibre du niveau. Par exemple, des adversaires, gérés par le jeu, trop compliqués au stade actuel, empêchent complètement la possibilité de passer ce niveau. Dans d'autres jeux, plusieurs équipes tentent d'atteindre le point souhaité de la carte

avant les adversaires. Ici, le défaut dans l'équilibre du niveau pourrait être des conditions de départ inégales pour les équipes.

### **Restrictions et conventions inappropriées**

Ces problèmes n'affectent pas le gameplay et n'apportent pas d'avantage de jeu, mais ils dégradent la perception et l'immersion dans le jeu.

Par exemple, le joueur voit que le chemin du personnage est bloqué par une clôture à hauteur de taille, mais qu'il est impossible de la franchir. Comme prévu par le concepteur du niveau, cet endroit est une zone inaccessible au joueur, et c'est ainsi que cela doit être. Mais du point de vue du joueur, cette situation ressemble à un défaut qui gâche l'immersion dans le jeu. De même, le joueur considérera comme un défaut une situation où le personnage est enfermé derrière des barreaux et ne peut pas en sortir, alors qu'il voit que la distance entre les barreaux est suffisamment grande.

### **Récit**

Niveaux de jeu associés à une rupture du style général de la narration, de l'intrigue du jeu. L'impact de ces défauts est minime sur le gameplay lui-même, mais ils affectent négativement l'immersion de l'utilisateur et la perception générale du jeu. Par exemple, si un personnage se retrouve dans une ville qui, selon l'intrigue, a été soumise à un bombardement nucléaire, l'apparition de bâtiments intacts dans celle-ci provoquera une dissonance.

### **Modifier le code couleur des objets**

Les objets rencontrés par l'utilisateur au cours du jeu ont des propriétés différentes. Certains sont des décorations, d'autres permettent au joueur d'interagir. L'attribution d'un code couleur aux objets ayant des propriétés différentes est un élément important de la conception des niveaux. Il est donc souhaitable qu'une fois adopté, le code couleur ne change pas au cours du jeu.

Par exemple, un joueur a découvert un tonneau rouge dans le niveau, qui a explosé à la suite d'un tir. Si un joueur rencontre à nouveau un tel tonneau, il s'attendra à ce qu'il ait le même comportement. Par conséquent, tous les tonneaux qui explosent à la suite d'un tir doivent toujours être de la même couleur (généralement rouge).

Les boîtes que le joueur peut casser, les portes qu'il peut ouvrir, les rochers et les rebords sur lesquels il peut s'agripper et grimper, etc. doivent avoir une forme et/ou une couleur différente des autres objets.

Sinon, le joueur sera obligé de passer plus de temps à comprendre ce qu'il doit faire ensuite.

## **5.2 Étapes et réalisation des tests de niveau de jeu**

### **5.2.1 Étapes de base de la conception et du test des niveaux de jeu**

Le niveau de jeu est une entité qui combine les mécaniques de jeu, les objets visuels, la bande sonore, l'intelligence artificielle et d'autres composants. Les approches de test pour chacun de ces composants sont examinées en détail dans les sections correspondantes de ce syllabus.

Le test des niveaux de jeu implique de tester tous ces composants ensemble et est similaire au test des systèmes, lorsque tous les composants sont considérés et testés comme un tout.

Le test des niveaux de jeux commence dès les premières étapes de leur création. Les objets du test, la nature des tests et la procédure de leur implémentation, ainsi que les domaines de responsabilité des différents spécialistes dépendent du stade actuel de développement du niveau de jeu.

Considérons les principales étapes de la conception et des tests des niveaux de jeu en prenant l'exemple d'un jeu en 3D.



---

## Prototypage de niveaux de jeu (Block Out/Boîte grise du concepteur)

À ce stade, sur la base d'un croquis préalablement élaboré, un modèle 3D du futur niveau du jeu est créé à partir d'objets spéciaux. Seuls les éléments ayant une incidence directe sur le gameplay sont utilisés : cubes, sphères, cylindres et plans gris, qui forment schématiquement la disposition du niveau de jeu.

Ces formes géométriques monochromes et non détaillées permettent de modifier rapidement la structure du niveau, par exemple en supprimant ou en ajoutant de grands morceaux d'emplacement et, si nécessaire, en effaçant tout proprement et en recommençant. Parfois, à ce stade, toutes les mécaniques de jeux ne sont pas prêtes à être utilisées, de sorte que le concepteur doit les implémenter progressivement, en adaptant le niveau aux conditions de développement en constante évolution.

L'agencement est créé en tenant compte de toutes les proportions et de toutes les échelles. Pour ce faire, le concepteur de niveau, en collaboration avec le concepteur de jeux vidéo, forme un ensemble de métriques - les tailles et les paramètres que possède le personnage du jeu. Ainsi, le concepteur du jeu fournit des informations sur la distance et la hauteur de saut du personnage, sur les positions dans lesquelles le personnage peut se trouver (par exemple, debout à pleine hauteur, accroupi, couché) et sur la manière dont cela modifie la taille de la figure, sur l'angle d'inclinaison maximal de la surface le long de laquelle le personnage peut se déplacer sans glisser. Ces métriques affectent directement les dimensions des objets et des éléments de l'environnement du niveau.

Une fois que le concepteur du niveau a décidé de la taille du niveau et défini l'échelle correcte pour les objets, d'autres fonctionnalités sont configurées pour soutenir le gameplay à ce niveau. Il s'agit notamment des points d'apparition des personnages, des points d'exploitation (par exemple, les déclencheurs) des événements du jeu, etc.

À ce stade, le développement des niveaux de jeu se fait en général en plusieurs itérations, au cours desquelles les tests permettent d'identifier et d'éliminer les éléments problématiques.

### Tests du prototype de niveau de jeu

Les exigences spécifiées n'étant pas finalisées au stade du prototypage, les tests exploratoires et autres techniques de test basés sur l'expérience sont considérés comme les plus efficaces.

Les premiers tests en situation de jeu ont lieu sur la carte créée pour tester le gameplay. Les participants aux tests en situation de jeu peuvent être des testeurs de l'équipe de développement du logiciel de jeu ou des testeurs externalisés. En recevant les commentaires des participants aux tests, le développeur peut tirer des conclusions sur les solutions réussies et celles qui ne le sont pas et, si nécessaire, modifier, déplacer ou supprimer des objets sur la carte.

Par ailleurs, en effectuant des tests en situation de jeu sur le prototype, on teste les mécaniques du jeu et la manière dont elles sont combinées avec la disposition et la taille des objets sur le modèle.

Par exemple, les abris situés sur le niveau sont suffisamment grands pour que le personnage puisse s'y cacher des ennemis, la largeur de la fissure dans la surface permet au personnage de sauter par-dessus, ou le personnage peut escalader tous les rebords nécessaires et ne se cogne pas la tête sur des plafonds trop bas.

Par ailleurs, il convient de tester la complexité souhaitée du jeu, créée par l'agencement du niveau, et la bonne sensation que procure le gameplay en général. Par exemple, si l'on suppose qu'à un moment donné du jeu, il y aura une fusillade avec de nombreux adversaires, le niveau devra comporter un nombre suffisant d'abris. S'il s'agit de se battre contre un seul ennemi tenace (le boss), l'espace du niveau doit être limité et débarrassé des objets inutiles.

---

## Prototypage de la géométrie (Block Out/White Box artistique)

Des modèles 3D temporaires créés par l'artiste en charge de l'environnement remplacent les objets à partir desquels le concepteur de niveau a réalisé le prototype, et des objets supplémentaires sont ajoutés. La tâche principale consiste à élaborer le composant visuel du niveau sans modifier le gameplay lui-même.

Cependant, un tel remplacement peut entraîner des défauts dans le niveau qui compliquent le gameplay. Par exemple, un modèle d'arbre temporaire chevauche la sortie du niveau. Dans ce cas, le niveau repasse entre les mains du concepteur du niveau. Celui-ci a désormais l'exigence de tester à nouveau si les mécaniques fonctionnent correctement, si le joueur peut aller partout et s'il voit tout ce dont il a besoin.

### Tester le prototype de géométrie

À ce stade, les tests sont effectués pour s'assurer qu'après l'habillage artistique du niveau, il n'y a pas de défauts qui entravent le gameplay.

Des exemples de tels défauts peuvent être un modèle 3D temporaire d'une pierre bloquant le passage du niveau, ou une couronne d'arbre bloquant la vue du joueur.

Pour identifier de tels problèmes, on a de nouveau recours à des tests en jeu, où l'on teste les mécaniques du jeu pour valider que la taille et l'emplacement des modèles d'objets correspondent aux métriques requises et à la justesse du gameplay.

Parallèlement à cela, un test de l'apparence des objets, de l'éclairage, de l'environnement, etc. peut être effectué (plus de détails sont donnés dans le chapitre "3. Test du graphisme").

### La réalisation de la version finale

À ce stade, le concepteur de niveaux intègre les logiciels de toute l'équipe de développement en un seul ensemble. À partir de tous les contenus créés, il rassemble un seul espace de jeu. À ce stade, des détails décoratifs sont ajoutés au niveau, les objets deviennent similaires aux objets réels en termes d'apparence, de fonction et d'emplacement. Par exemple, des papiers, de la papeterie, des écrans d'ordinateur apparaissent sur le bureau, que le joueur peut briser et jeter hors de la table.

### Tests de la version finale

À ce stade, des tests de performance et de compatibilité sont effectués. Les performances du niveau sont testées sur différents appareils, le nombre d'images par seconde est mesuré, la qualité d'affichage des modèles visuels des objets à différentes distances, l'éclairage et les ombres sont testés.

Dans le cadre des tests en situation de jeu, les testeurs testent la présence de modèles de collision pour tous les objets du jeu, la correspondance entre les modèles physiques et visuels des objets, etc.

Par exemple, le test des modèles de collision se déroule comme suit : le testeur amène le personnage du jeu vers l'objet du jeu, par exemple vers une grosse pierre, et essaie de "marcher sur la pierre" de différents côtés. Si le modèle de collision avec les pierres est correctement configuré, le corps du personnage ne fera qu'effleurer visuellement la pierre, mais ne la traversera pas et n'y pénétrera pas complètement.

#### 5.2.2 Domaines de responsabilité des intervenants

Divers spécialistes travaillent à la création d'un niveau de jeu ou d'une cartographie, et souvent ces tâches sont effectuées en parallèle, ce qui entraîne l'apparition de divers défauts.

Considérons les tâches qui relèvent de la responsabilité de ces spécialistes du point de vue de l'obtention d'un produit de qualité :

Rôle	Responsabilité
Concepteur de niveaux de jeu	Responsable de la géométrie et de la forme du niveau, de l'emplacement des objets et des points de déclenchement, de la taille des abris, etc. Tout cela doit soutenir le gameplay principal et favoriser l'engagement de l'utilisateur dans le jeu
Artiste	D'un point de vue artistique, tous les objets de la carte doivent avoir des textures, des éclairages et d'autres effets visuels corrects. Cela s'applique à la fois à l'apparence (placement correct des zones sombres et claires) et au réalisme de l'image. Par exemple, une cabane dans les bois faite de blocs de béton aurait l'air étrange et déplacée.
Testeur	En règle générale, les testeurs travaillent avec les versions finales des niveaux et des cartes, lorsque tous les objets ont déjà été placés et configurés. La tâche principale d'un testeur est de tester si un joueur peut jouer sur une carte donnée sans rencontrer de défauts techniques et artistiques.  L'interaction avec tous les objets de la carte est testée, à savoir : <ul style="list-style-type: none"><li>- Les objets ont un modèle physique (modèle de collision),</li><li>- Absence d'obstacle invisible,</li><li>- Impossibilité de sortir de la carte.</li></ul>

### 5.3 Outillage des tests des niveaux de jeu

Divers outils sont utilisés pour faciliter le test de niveaux ou de cartes, notamment des éditeurs 3D, des éditeurs de niveaux et des moteurs de jeux vidéo.

Toutefois, les testeurs utilisent les mêmes outils que les développeurs qui ont créé le niveau.

Le moteur de jeux vidéo dispose généralement de fonctionnalités qui permettent de personnaliser la carte et les objets qui s'y trouvent pour divers tests et la recherche de défauts. Le moteur développé en interne peut être modifié à la demande de l'équipe de testeurs afin d'accroître l'efficacité des tests.

Par exemple, à la suite de changements dans la géométrie de la surface de niveau, certains objets peuvent être suspendus dans l'air ou s'enfoncer sous la surface. De tels défauts peuvent être découverts par une inspection visuelle de tous les objets ou, si l'éditeur de cartographie le permet, un algorithme spécial peut simplifier le test.

En outre, en désactivant l'affichage des textures des différents objets dans l'éditeur, le testeur peut constater la présence d'endroits infranchissables sur la carte, ou inversement, d'un chemin viable non prévu par le concepteur de la carte. Ceci est particulièrement important pour les jeux vidéo multijoueurs, car dans certains cas, cela peut donner à certains joueurs un avantage non désiré sur les autres.

## 6. Tests des contrôleurs de jeux vidéo - 95 minutes (K2)

### Mots-clés du domaine des tests logiciels

Conformité, tests d'ergonomie, tests fonctionnels

### Video Game Specific Keywords

Accéléromètre, contrôleur de jeu vidéo, manette de jeu, gyroscope, volant de course, écran tactile, trackball

### Objectifs d'apprentissage pour le chapitre 6

#### 6.1 Principes et concepts des contrôleurs de jeux vidéo

GaMe-6.1.1 (K2) Classer les périphériques d'entrée habituels et les périphériques spécialisés

GaMe-6.1.2 (K2) Donner des exemples de différents dispositifs d'entrée en fonction de leur application

GaMe-6.1.3 (K1) Rappeler les différents types de manettes de jeux vidéo

GaMe-6.1.4 (K2) Classer les défauts d'un jeu vidéo liés aux spécificités des contrôleurs de jeu, et les causes possibles de leur apparition

#### 6.2 Approches pour tester les contrôleurs dans les jeux vidéo

GaMe-6.2.1 (K2) Donnez des exemples de conditions de test à couvrir pour tester les contrôleurs de jeux vidéo

GaMe-6.2.2 (K2) Classer les tâches des spécialistes UX, des testeurs et des concepteurs de jeux lors des tests de jeux vidéo

#### 6.3 Outillage pour tester les contrôleurs de jeux vidéo

GaMe-6.3.1 (K2) Résumer l'utilisation des outils permettant de tester le comportement des contrôleurs de jeux vidéo

## 6.1 Principes et concepts des contrôleurs de jeux vidéo

### 6.1.1 Types de manettes de jeux vidéo

Un contrôleur de jeu est un dispositif d'entrée utilisé dans les jeux vidéo sur console et sur ordinateur. La manette est généralement connectée à une console de jeux ou à un ordinateur personnel. À l'aide de la manette, le joueur contrôle les mouvements et les actions des éléments du jeu. Dans ce cas, le type d'éléments dépend du jeu lui-même, mais il s'agit le plus souvent de l'un des personnages du jeu.

### Dispositifs d'entrée habituels

Les appareils de jeux, tels que les téléphones, les PC, les consoles et les machines à sous, sont censés permettre l'utilisation de l'un des dispositifs suivants.

Dispositif d'entrée	Description
Manette de jeu	Il s'agit du principal périphérique d'entrée des consoles de jeux vidéo
Clavier et souris	Comme ces dispositifs sont devenus des périphériques d'entrée courants pour les ordinateurs personnels, ils sont également couramment utilisés pour les jeux vidéo. Certaines consoles de jeux permettent également de connecter une souris et un clavier et de les contrôler dans les jeux.
Volant de course	Il s'agit d'une manette dotée d'un bouton de rotation et d'un ou plusieurs boutons d'action

Dispositif d'entrée	Description
Boule de commande (NDT: trackball)	Ce dispositif ressemble à une demi-boule qui dépasse de la base ; on fait tourner la boule en passant la paume de la main dessus.
Écran tactile	Il est utilisé dans les téléphones, les PDA, les consoles portables et les machines à sous modernes. Certains jeux comportent des éléments de gameplay spécifiquement destinés à l'écran tactile

Les **dispositifs spécialisés** sont des dispositifs axés sur certains types de jeux vidéo.

Dispositif spécialisé	Description
Manette de jeu	À l'origine, il s'agissait d'un appareil de jeux vidéo universel. On a constaté qu'un clavier/souris ou un game pad était préférable dans les jeux vidéo rapides, et le joystick est devenu un dispositif spécialisé pour les jeux du genre simulateur de vol. Toutefois, par habitude, les manettes de jeu sont souvent appelées "joysticks".
Manette de jeu de course	Utilisé pour simplifier le jeu de course sur la console. Il s'agit en fait d'une manette de jeu classique, qui intègre un axe de direction supplémentaire (et parfois des boutons analogiques d'accélération et de freinage). Un tel dispositif est beaucoup moins cher qu'un volant de course à part entière.
Volant	Pour les simulateurs de vol civils (les simulateurs de vol militaires utilisent un joystick).
Pédales	Pour les simulateurs de voiture et de vol dont la conception est fondamentalement différente.
Accélérateur informatique	Également connu sous le nom de levier de commande du moteur. Utilisé pour les simulateurs de vol.
Lever de vitesse	Pour les simulateurs de conduite.
Pistolet à faisceau lumineux	Pour tirer sur des objets à l'écran.
Tablette graphique	Utilisé pour contrôler le curseur à la place de la souris.
Contrôleurs de rythme	Utilisé dans les jeux vidéo musicaux pour simuler des instruments de musique tels que des guitares, des tambours ou la console d'un DJ.
Tapis de danse	Une piste de danse (comme sur les machines à sous). Il s'agit d'un tapis avec plusieurs boutons sur lequel on peut marcher avec les pieds. Le gameplay de ces jeux consiste à marcher sur la séquence de boutons nécessaire pour ressembler à une danse.
Clavier de jeu vidéo	Il s'agit d'un clavier spécialisé dont les boutons sont placés en fonction des spécificités d'un jeu particulier, et de boutons supplémentaires permettant de créer des macros.

Dispositif spécialisé	Description
Contrôleur de canne à pêche	Pour les jeux de simulateurs de pêche.
Microphone	Un microphone ou un casque est utilisé comme dispositif d'entrée supplémentaire, avec lequel des commandes sont données au jeu vidéo de manière à ce que les personnages et les joueurs puissent communiquer.
Panneau de commande de train	Un simulateur d'un panneau de commande pour le contrôle du matériel roulant d'un chemin de fer de traction / d'un tramway.

### Technologies de capture du mouvement

- Depuis le début des années 2000, les systèmes de suivi de la tête sont utilisés pour jouer sur des simulateurs de vol et pour les personnes à mobilité réduite,
- Des dispositifs de télécommande qui suivent sa position dans l'espace à l'aide de capteurs IR et d'accéléromètres,
- Des caméras qui suivent les mouvements du contrôleur dans l'espace 3D et reconnaissent les images,
- Les dispositifs qui permettent d'utiliser des commandes verbales, des postures corporelles et des objets ou images affichés.

#### 6.1.2 Défauts liés aux spécificités des manettes de jeux

Les causes des défauts associés aux contrôleurs peuvent être différentes. L'apparition de défauts peut être causée par le logiciel lui-même, le mariage des composants du contrôleur, et même la défaillance du développeur à respecter les instructions d'utilisation du contrôleur fournies par le fabricant :

- Pilote du contrôleur obsolète,
- Incompatibilité du modèle de contrôleur avec l'application,
- Défaut d'un appareil individuel ou de l'ensemble du lot,
- Incohérence du gameplay avec les instructions.

Le défaut le plus courant est le manque de remplacement ou l'absence totale d'infobulle lors du changement de manette en cours de jeu. Les combinaisons de touches peuvent varier d'un jeu à l'autre. Elles peuvent également être réaffectées par le joueur à sa propre discrétion.

Une version obsolète des pilotes ou leur absence peut entraîner le fait que le contrôleur ne fonctionne pas comme prévu. Si les défauts des logiciels peuvent être éliminés par une mise à jour, les dysfonctionnements techniques et les lacunes des contrôleurs ne peuvent être corrigés que par la release de leur nouvelle révision.

Cependant, une imprécision dans la lecture du mouvement d'un volant de course ou de tout autre contrôleur peut provenir d'un défaut matériel ou d'un défaut de calcul logiciel. Pour les jeux où la précision de la lecture des signaux de commande du contrôleur de jeu est critique, le document de conception du jeu vidéo doit contenir les valeurs requises en degrés d'inclinaison.

De même, lorsqu'un jeu vidéo est publié sur une plateforme populaire, le propriétaire de la plateforme peut fournir des exigences concernant les images du jeu de sa manette. Les images « in-game » sont les images du contrôleur utilisées dans le jeu vidéo. Par exemple, il peut s'agir d'une représentation schématisée d'une manette de jeu dans le menu des paramètres d'assignation des touches ou dans les conseils de gameplay. Les images peuvent ne pas seulement se trouver à l'intérieur du jeu ; les contrôleurs peuvent également être affichés sur l'emballage d'un logiciel ou sur une couverture numérique. Cette exigence s'applique

généralement aux éditeurs connus et simultanément aux fabricants de consoles et de contrôleurs. Les sociétés, dans leur documentation de test pour les développeurs de jeux, peuvent fournir des exigences sur la façon dont les contrôleurs doivent être représentés dans une application, y compris les contours et les marques déposées.

De même, pour les jeux vidéo où l'accéléromètre et le gyroscope de la manette sont utilisés, le propriétaire de la plateforme impose des exigences de sécurité. L'exemple le plus simple serait la nécessité de faire des mouvements larges avec des contrôleurs pour implémenter le gameplay. Les capteurs de mouvement qu'elle contient permettent de l'utiliser comme une commande dans l'espace 3D. Dans ce cas, il est exigé d'indiquer, avant de commencer le jeu, que l'utilisateur doit mettre les sangles de maintien qui sont attachées aux contrôleurs. Sinon, en faisant des mouvements brusques avec les manettes, celles-ci peuvent glisser des mains de l'utilisateur et endommager l'équipement environnant ou, pire encore, devenir un risque pour la santé.

## 6.2 Approches pour tester les contrôleurs dans les jeux vidéo

Les tests utilisant des manettes de jeux commencent généralement au moment où les fonctionnalités du jeu sont prêtes pour l'utilisation d'une manette standard : pour les PC, il s'agit d'un clavier / d'une souris, pour les consoles, d'une manette de jeu.

Ce type de test peut comprendre les éléments suivants :

- connexion / déconnexion du contrôleur du PC / de la console,
- faible niveau de batterie du contrôleur,
- prise en charge des jeux pour certains fabricants de contrôleurs,
- prise en charge de certaines API permettant à une application de recevoir des données d'un contrôleur,
- utilisation d'un et / ou de plusieurs contrôleurs,
- utilisation non triviale du contrôleur (tests négatifs),
- vibrations (présence et degré de leur intensité).

### Tests fonctionnels

Si le logiciel exige l'utilisation d'un contrôleur comme périphérique d'entrée, toutes les fonctionnalités de l'application doivent interagir correctement avec l'appareil connecté. L'utilisateur doit pouvoir contrôler à la fois les éléments de l'interface utilisateur et directement le gameplay sans changer de manette. Les éléments de contrôle attribués : boutons, sticks, tours de volant, commandes vocales ou mouvements dans l'espace doivent correspondre aux actions du personnage du jeu. Un testeur doit également s'assurer que lorsque la manette est déconnectée de l'appareil, l'application se met en pause, si possible. Si l'implémentation du gameplay s'effectue en temps réel et ne peut pas être arrêtée, alors la déconnexion de la manette de jeu ne doit pas entraîner la déconnexion du joueur du serveur, et doit également être accompagnée d'un message d'information.

### Tests de sécurité

Une manette peut contenir une vulnérabilité technologique et ouvrir l'accès au piratage de la console. Des tests de sécurité doivent être effectués pour atténuer les risques de sécurité : obtenir un accès non autorisé au mode développeur, éviter de bloquer l'appareil sur le réseau en utilisant le mode vol, etc. [ISTQB\_AL\_SEC]. Les tests de sécurité des manettes de jeux vidéo combinent les approches consistant à découvrir les défauts du matériel qui influencent la fonctionnalité du logiciel.

### Test d'ergonomie

Les boutons des manettes des marques les plus populaires ont une fonction bien établie. Les boutons X, A et , Les boutons X des manettes de jeu sont très souvent utilisés dans les applications comme boutons

pour "accepter", "accepter la sélection" ou "interagir" avec un élément interactif, et les boutons B et ○ comme "annuler", "refuser" ou "retourner".

Les combinaisons de boutons sont définies et contrôlées par le logiciel. Lorsqu'il teste les combinaisons de boutons, le testeur doit veiller à ce qu'elles conviennent anatomiquement à différents groupes de joueurs (en fonction du genre, de l'âge, du handicap, etc.) : le joueur doit avoir la possibilité d'atteindre plusieurs boutons simultanément ou successivement.

### Test de conformité des jeux vidéo

Les rôles de spécialiste UI / UX et de concepteur de jeu sont obligatoires lors de la conception de jeux vidéo. Leur tâche consiste à développer une interface qui sera agréable à l'œil, pratique sur le plan fonctionnel et conforme aux conventions généralement acceptées, ou aux standards du genre. Le but des tests sera à la fois de tester l'opérabilité de l'interaction des entrées envoyées par la manette et des actions effectuées dans l'interface, ainsi que de tester l'UX (expérience utilisateur) dans son ensemble.

Le testeur doit s'assurer que le développeur utilise des dispositions traditionnelles pour les contrôleurs populaires, par exemple, comme indiqué précédemment, les boutons X, A et □, X sont utilisés comme boutons " d'accord ", et les boutons B et ○ sont utilisés comme boutons " d'annulation ". Pour le clavier, les boutons W, A, S et D sont utilisés comme boutons directionnels, Ctrl ou C pour s'accroupir ou ramper, et la barre d'espace pour sauter. La fonction traditionnelle du bouton gauche de la souris serait un tir ou une attaque pour les jeux de tir à la première personne, ou un choix pour les stratégies. Le bouton droit de la souris est généralement réservé à la visée dans les jeux de tir à la première personne ou au déplacement des troupes dans les stratégies.

Les testeurs doivent également s'assurer que l'utilisation d'une manette, quelle qu'elle soit, ne donne pas au joueur un avantage significatif sur les autres. Les manettes de jeu étant nettement inférieures au couple clavier/souris en termes de rapidité et de précision de visée, elles sont souvent ajoutées à la visée automatique d'une cible hostile et à la poursuite qui s'ensuit. Le viseur "colle" simplement à l'adversaire. Dans ce cas, il est de la responsabilité du testeur de surveiller à quelle distance le réticule commence à suivre l'adversaire et s'il cible sa tête, qui est souvent une zone vulnérable, infligeant des dégâts supplémentaires lorsqu'il est touché.

## 6.3 Outillage pour tester les contrôleurs de jeux vidéo

Plusieurs outils logiciels peuvent être utilisés pour apporter un soutien aux tests des contrôleurs de jeux vidéo par l'enregistrement vidéo ou la représentation schématique des contrôleurs.

Les outils de capture/rejeux permettent d'enregistrer des vidéos à partir de l'écran d'un PC. Ils sont utilisés pour le rapport de défaut afin d'afficher la nature du défaut et les étapes à suivre pour le reproduire.

Un autre type d'outil est utilisé pour afficher une représentation schématique de la manette de jeu et des commandes sur vidéo. Ces outils simplifient la compréhension des activités du contrôleur pour découvrir et analyser le défaut.

Il existe des services qui soutiennent le test de l'entrée du contrôleur et du degré de déviation des sticks de la manette de jeu, ainsi que les performances de vibration du contrôleur (voir [URL3]). Ces services disposent de tous les outils de test nécessaires pour la manette de jeu, ce qui permet d'auditer l'état de santé de la manette.

Des outils spéciaux permettant d'afficher les frappes sur le clavier sont également disponibles et peuvent s'avérer utiles lors de l'exécution des tests.



## 7. Tests de localisation - 155 minutes (K3)

### Mots-clés du domaine des tests logiciels

Conformité, internationalisation, localisation

### Mots-clés spécifiques du domaine des jeux vidéo

Adaptabilité culturelle, Exactitude historique, Localité

### Objectifs d'apprentissage pour le chapitre 7

#### 7.1 Principes et concepts des tests de localisation

GaMe-7.1.1 (K1) Rappeler les principaux objectifs de l'internationalisation et de la localisation

GaMe-7.1.2 (K1) Connaître les étapes des tests de localisation

GaMe-7.1.3 (K2) Comparer les capacités d'internationalisation et de localisation

#### 7.2 Types de défauts de localisation et leurs causes

GaMe-7.2.1 (K2) Classer les défauts de localisation et leurs causes

#### 7.3 Approche de test de localisation et réalisation des tests

GaMe-7.3.1 (K1) Connaître les tests complets et partiels de localisation

GaMe-7.3.2 (K3) Classer les types de test de localisation

GaMe-7.3.3 (K2) Résumer les tâches de test pour un rédacteur, un éditeur, un traducteur et un testeur de localisation

#### 7.4 Outillage pour les tests de localisation

GaMe-7.4.1 (K2) Résumer l'utilisation des outils permettant de tester la localisation des jeux vidéo.

### 7.1 Principes et concepts des tests de localisation

#### 7.1.1 Localisation et internationalisation

Le processus de localisation de tout produit, qu'il s'agisse d'un lecteur vidéo, d'un jeu ou d'un système d'exploitation, comporte des étapes, des approches et des techniques similaires et se distingue par les procédures internes uniques adoptées par l'entreprise de développement de logiciels [Chandler11].

Le développement d'un jeu est une idée qui est implémentée par l'écriture d'un code, complétée par le contenu nécessaire et qui est mise à disposition des joueurs. Un stade précoce important du développement consiste à déterminer le marché pour lequel le jeu sera livré :

- Cette étape n'est pas toujours prise en compte, ce qui peut avoir une incidence négative sur les ventes du jeu à l'avenir.
- La présence d'un contrôle de version local du jeu est obligatoire pour que sa promotion sur le marché national soit réussie.
- La plupart des jeux sont développés en tenant compte de la nécessité d'adapter le produit au marché cible.
- L'adaptation du jeu au marché cible peut multiplier ses ventes.

Étant donné que la sortie du jeu sur d'autres marchés peut intervenir après la fin du développement de la version locale du jeu (par exemple, l'entreprise a de l'argent pour lancer le jeu sur un autre marché ou des partenaires sont apparus qui sont prêts à promouvoir le produit dans certains pays), alors lors du développement du jeu, l'équipe doit prendre en compte le fait que le jeu peut être adapté ultérieurement à certains autres marchés. Dans le cas contraire, les adaptations ultérieures risquent d'entraîner des coûts importants en termes de finances et de ressources.

## L'internationalisation

Pour éviter les risques et les difficultés liés à l'adaptabilité à une région spécifique, un processus appelé internationalisation est utilisé. L'internationalisation est une adaptation d'un produit en vue d'une utilisation potentielle presque partout, tandis que la localisation est une modification en vue d'une utilisation dans une région spécifique.

Contrairement à la localisation, l'internationalisation d'un logiciel est un ensemble d'activités réalisées pendant la phase de développement afin de faciliter la traduction et la localisation ultérieures du logiciel.

L'internationalisation est réalisée au cours des premières étapes du développement et, dans la plupart des cas, elle est effectuée sans l'intervention de linguistes-traducteurs et relève de la responsabilité du développeur du logiciel. La localisation fait appel à des traducteurs spécialisés (dans certains cas des locuteurs natifs) qui disposent d'un grand nombre de connaissances supplémentaires.

L'internationalisation comprend :

- La création et le développement du logiciel de manière à ce qu'il n'y ait pas d'obstacles à la localisation et à l'utilisation internationale. L'utilisation éventuelle d'Unicode ou la fourniture d'une approche de codage des caractères (si nécessaire).
- La création de possibilités d'utiliser des éléments qui ne peuvent pas être appliqués avant le processus de localisation. Par exemple, ajouter un cadre pour le texte bidirectionnel au langage de balisage DTD (définition du type de document). Ou ajouter au CSS (Cascade Style Sheet) une base pour le texte vertical ou les caractères typographiques non latins.
- La possibilité d'inclure les références régionales, linguistiques ou culturelles. Cela comprend généralement l'introduction de données localisées prédéfinies ou d'une traduction créée antérieurement et stockée dans un logiciel spécialisé. Exemples : formats de date et d'heure, calendriers locaux, formats et systèmes de nombres (chiffres romains et arabes), sélection et présentation de listes, utilisation de noms de personnes et de formulaires de contact.
- L'extraction d'éléments localisés du code ou du contenu afin que les versions localisées puissent être téléchargées ultérieurement ou sélectionnées en fonction des préférences de l'utilisateur. En règle générale, cela est implémenté sous la forme de packs linguistiques chargeables.

Cette liste n'inclut pas nécessairement la localisation du contenu, du programme ou du produit dans une autre langue. Il s'agit de techniques et d'approches du développement de logiciels qui permettent une migration aisée vers la localisation à l'avenir.

## La localisation

La localisation est le processus d'adaptation d'un logiciel à la culture d'un pays. Un cas particulier de localisation est la traduction de l'interface utilisateur, de la documentation et des fichiers logiciels connexes [Retsker81].

La localisation de jeux vidéo comprend :

- Traduction des textes des dialogues et sous-titres des jeux vidéo, des infobulles, des descriptions et messages, des noms de personnages, des noms d'objets,
- Traduction des écrans de veille, de l'interface et des éléments de menu,
- Redessiner les textures et les graphismes,
- Sélection des acteurs, doublage et enregistrement des fichiers sonores,
- Intégration des matériaux localisés dans le jeu,
- Traduction et adaptabilité du site web du projet et impression,

- Traduction du matériel publicitaire (actualités, communiqués de presse et matériel de marketing),
- Soutien au jeu après la sortie de la version localisée (mises à jour, nouvelles et correctifs).

Ainsi, tester la localisation consiste à vérifier dans quelle mesure un produit de jeu est adapté à un public cible spécifique en fonction de ses caractéristiques et références culturelles, linguistiques, religieuses, politiques et autres. Habituellement, on considère ici les aspects culturels et linguistiques, notamment la traduction de l'interface utilisateur, de la documentation et des fichiers dans une autre langue, ainsi que les formats des devises, des chiffres, des heures, des numéros de téléphone, etc.

Le test de localisation consiste à tester le contenu d'une application de jeux vidéo par rapport aux exigences linguistiques et culturelles, ainsi qu'aux spécificités d'un pays ou d'une région en particulier. Ce type de test permet de constater les défauts de localisation ou les défaillances de traduction d'une version localisée avant que le produit final ne parvienne à l'utilisateur. L'objectif des tests de localisation est de constater et de corriger les défauts dans les différentes versions localisées d'un produit pour différents marchés et contextes régionaux (locales).

Il est important de noter que la localisation ne se limite pas à la traduction en plusieurs langues, et que les tests de localisation et les tests linguistiques ne sont pas la même chose. Les tests linguistiques consistent principalement à tester les défauts orthographiques, grammaticaux et stylistiques.

#### 7.1.2 Différence entre la localisation d'un jeu vidéo et d'un logiciel applicatif

La différence entre tester la localisation de jeux vidéo et de logiciels applicatifs repose sur la facilité de compréhension que ces deux types de produits diffèrent considérablement l'un de l'autre dans divers aspects pris en compte lors de la localisation.

Les principales différences sont :

- L'adaptabilité du contenu graphique au public cible.
- La localisation du contenu audio
- La localisation et l'adaptabilité du contenu textuel
- La conformité avec les caractéristiques du genre et de la littérature.

En règle générale, une grande partie du contenu graphique peut nécessiter une adaptabilité pour le public cible. Ce type de graphisme comprend les signes, les personnages, les objets du jeu, les cartes de niveaux, les symboles et attirails, les éléments du jeu, les économiseurs d'écran publicitaires, etc.

Le contenu audio peut également nécessiter une localisation minutieuse. Souvent, le contenu audio d'un jeu vidéo doit être reproduit par les acteurs dans la langue du public cible, et le contenu lui-même doit être adapté.

Les styles mixtes de contenu textuel nécessitent une traduction et une adaptabilité appropriées. Les styles suivants peuvent être utilisés dans un jeu :

- Un style scientifique (description des mécaniques, instructions),
- Un style journalistique (journaux, magazines, articles),
- Un style artistique (journaux personnels, livres),
- Le vocabulaire du métier (dossier, documents divers),
- Le vocabulaire de la vie quotidienne (journaux, magazines, articles),
- Le langage familier de tous les jours (dialogues des personnages).

Par exemple, dans le genre de l'aventure (quête), il y aura du quotidien scientifique, journalistique, familier, puisque le joueur doit réparer ou assembler un appareil quelconque selon les instructions, trouver les informations nécessaires dans les journaux et interroger des personnes.

Il existe une exigence d'adaptabilité obligatoire du contenu d'un jeu vidéo, prenant en compte des facteurs tels que : exactitude historique, caractéristiques religieuses, culturelles, politiques et idéologiques du public cible, etc. Dans certains cas, la violation de cette exigence peut entraîner une perception ambiguë du produit par le public cible et aboutir à l'interdiction de son utilisation sur le territoire d'un pays ou d'une région en particulier.

Il est nécessaire de se conformer aux particularités du genre et de la littérature lors de la localisation d'un jeu vidéo. Par exemple, dans un jeu de rôle multijoueurs, il est nécessaire, lors de la traduction, de tenir compte des noms des peuples des joueurs, des noms d'objets, etc. acceptés dans le genre. Parfois, l'adaptabilité du contenu est exigée en utilisant des concepts stylistiquement proches et liés au genre.

Des connaissances et des informations supplémentaires sont requises pour l'implémentation d'une traduction de qualité d'un jeu vidéo. Par exemple, diverses références à d'autres produits de jeux vidéo, à des œuvres médiatiques ou à des événements de la réalité qui peuvent être contenus dans le jeu.

Voici quelques-unes des approches de la localisation des produits de jeux vidéo qui sous-tendent les tests de localisation.

### 7.1.3 Les étapes des tests de localisation

Les tests de localisation comprennent le test de l'exactitude du contenu traduit, des divers éléments d'interface, des défauts et des messages du système, ainsi que le test des sections "Foire aux questions" et "Aide".

#### **Tests de traduction**

Les tests de traduction ont pour but de tester l'interface multilingue du jeu pour détecter les défauts de traduction, l'exactitude des adresses postales, des noms et prénoms, des devises, des formats de date et d'heure, etc.

Parfois, au cours du développement, il est nécessaire d'adapter l'apparence du nombre et la valeur numérique aux standards nationaux (par exemple, les unités de mesure). Lors des tests, le testeur doit toujours se souvenir des systèmes de mesure adoptés dans tel ou tel pays afin d'informer clairement l'utilisateur sur la vitesse, la longueur, le poids, la température, etc.

Le message "Vous vous déplacez à 62 miles par heure" peut être difficile à comprendre pour un joueur originaire d'un pays utilisant les unités métriques. Dans ce cas, il ne suffit pas de changer la valeur numérique de la vitesse, mais il est également nécessaire d'adapter l'unité de mesure.

Ce qui précède s'applique également à la monnaie utilisée dans les jeux vidéo pour effectuer des achats. L'exemple le plus simple de localisation des prix est la conversion automatique dans la devise de la région où le produit a été activé. Outre la conversion au taux de change, le symbole de la devise correspondante doit également être présent. Toutefois, lors du développement d'une application, le testeur doit également tenir compte du fait que les services de distribution de logiciels peuvent fixer des prix régionaux.

Dans divers jeux vidéo, les prix de la boutique « in-app » sont convertis en devises et varient selon les régions.

Lors des tests des jeux vidéo, il convient de prêter attention aux graphismes. Ils doivent correspondre aux réalités du pays pour lequel le jeu est publié. Par exemple, les panneaux de signalisation peuvent être différents d'un pays à l'autre. En outre, des images et des scènes de fêtes locales sont ajoutées. Dans les pays musulmans, les graphismes sont radicalement revus - toutes les images de personnes et d'animaux sont supprimées et des arabesques (ornements orientaux médiévaux complexes composés d'éléments géométriques et végétaux) sont ajoutées.

La célébration du Nouvel An chinois est particulièrement populaire. Dans de nombreux jeux vidéo, l'interface est décorée de symboles nationaux : lanternes, dragons et feux d'artifice. De plus, le jeu peut être enrichi d'un contenu limité (disponible pour une durée limitée) : des habillages de personnages stylisés de héros chinois de mythes, des animations pour lancer des pétards et des fusées, chevaucher des cochons et des dragons.

Dans les jeux, les blagues doivent souvent être adaptées, et parfois même l'intrigue doit être ajustée pour s'aligner sur la mentalité du pays dans lequel le jeu vidéo sera vendu.

La localisation ne se limite pas au travail sur le texte - elle comprend également la réconciliation des caractéristiques culturelles et des aspects moraux et éthiques.

Traditionnellement, il existe certains sujets et certaines orientations en matière de tests de localisation auxquels les testeurs doivent prêter la plus grande attention :

- La religion, y compris l'occultisme et le satanisme ;
- Le sexe, les personnages habillés de manière provocante et le langage vulgaire ;
- Les préjugés et les stéréotypes associés à la culture et aux personnes elles-mêmes ;
- Les guerres, les conflits militaires, le terrorisme ;
- La politique, y compris le point de vue spécifique des différents pays sur l'histoire.

Par exemple, si un jeu vidéo contient la présence d'un passage du Coran lors du doublage, la plupart des pays islamiques interdiront très probablement ce jeu vidéo.

Dans les conditions de contrôle strict du politiquement correct dans le monde moderne, son respect doit être pris au sérieux.

Outre les difficultés liées au composant technique du jeu ou à une préparation insuffisante de la localisation du jeu, un traducteur de jeux vidéo peut également être confronté à un certain nombre de difficultés liées non pas au jeu qu'il traduit, mais à la nécessité de disposer de connaissances supplémentaires pour réaliser une traduction de qualité.

Par connaissances supplémentaires, on entend également le contexte d'autres produits vidéo, des œuvres médiatiques ou des événements de la réalité qui peuvent être contenus dans le jeu. Ces références doivent être traduites correctement, conformément à la manière dont elles ont été traduites précédemment. Le traducteur doit donc être au courant des derniers événements qui se déroulent dans le monde, ainsi que de divers films et jeux vidéo populaires.

### Tests de conformité

Le système de restrictions d'âge pour les jeux et les applications prend en compte les particularités de la législation et de la culture des différents pays. Cela permet aux développeurs de définir plus précisément les restrictions de contenu et de distribuer les applications au public auquel elles sont destinées.

Lorsque l'on teste la localisation, il convient de prendre en compte les exigences légales soutenues dans les différentes régions. Par exemple, en France, l'âge de la majorité est fixé à 18 ans, aux États-Unis, cet âge varie de 18 à 21 ans, et au Japon, les jeunes sont considérés comme des adultes à partir de 20 ans. Les jeux vidéo dont le contenu est différent doivent, dans la plupart des cas, suivre le système de restriction d'âge en vigueur dans les différents pays.

Un même jeu peut recevoir des classifications d'âge différentes selon les pays.

ESRB (USA)	PEGI (EU)	RARS (Russia)	ACB (Australia)	USK (Germany)
				
				
				
				
				
				

Classement des jeux vidéo dans différents pays  
<https://www.kaspersky.com/blog/gaming-age-ratings/11647/>

Lorsque l'on teste la localisation pour vérifier la conformité avec la législation et les exigences légales, il est nécessaire de prendre en compte de nombreuses circonstances, parfois associées non pas tant à la langue cible qu'aux caractéristiques juridiques et culturelles du pays pour le marché dans lequel le jeu vidéo est proposé.

En outre, il convient de prêter attention au fait que le jeu prend en charge la monnaie et les opérations qui y sont liées pour chaque pays. Par exemple, si dans un certain pays les opérations de change sont interdites à tout le monde sauf à la Banque de ce pays, essayer de lancer un jeu sur ce marché conduira certainement les éditeurs de jeux à de sérieux problèmes.

Indirectement, les transactions de devises incluent l'achat de « loot boxes » - littéralement "boîtes à butin", qui, en règle générale, contiennent divers tenues / apperences de personnages, des articles consommables, des améliorations ou des montures.

En décembre 2016, le ministère de la Culture de la République Populaire de Chine a annoncé la promulgation d'une loi exigeant qu'un éditeur de jeux vidéo en ligne publie la probabilité d'acquisition de tous les objets et services virtuels à partir de mai 2017.

En Australie, les jeux de type "loot box" sont soumis à des restrictions en matière de jeux d'argent s'ils peuvent être joués contre de l'argent ou tout autre objet de valeur. Des questions subsistent même dans le cas où la valeur d'un objet qui n'existe que dans le jeu ne peut être déterminée qu'en lien avec le prestige de cet objet.

### Couleurs et symboles

Les couleurs et les symboles spécifiques, qui peuvent avoir des significations différentes selon les pays, doivent également être pris en compte lorsque l'on teste la localisation.

Par exemple, pour les habitants de la Chine, le rouge est un symbole d'endurance et de foi ; en Inde, il symbolise la pureté. L'Europe, en revanche, voit dans cette couleur le péché et le sacrifice. Pour les Sud-Africains, c'est la couleur du deuil. Aux États-Unis et au Japon, le rouge symbolise le danger et la menace terroriste, tandis que les Égyptiens l'associent au deuil. Par conséquent, lorsqu'il teste un projet de jeux vidéo, le testeur doit prêter attention aux combinaisons de couleurs, car elles peuvent être importantes.

En outre, lors des tests, le testeur doit prêter attention au soutien des dispositions de clavier régionales et des touches de raccourci.

Dans le cas où l'application utilise des intégrations avec des ressources tierces (par exemple, stockage dans le cloud ou réseau social), il est nécessaire de prendre en compte leur disponibilité pour les régions.

Ainsi, les processus de test pour la localisation et l'internationalisation seront différents :

Internationalisation	Localisation
Encodages UTF	Traduction
Formats de données	Exigences légales
Sens du texte	Devises et opérations de change
	Couleurs et symboles
	Disposition du clavier et touches de raccourci
	Intégration avec des ressources tierces

## 7.2 Types de défauts de localisation et leurs causes

### 7.2.1 Causes possibles de défauts dans la localisation du jeu vidéo

Même les grandes entreprises ne sont pas à l'abri des failles "culturelles". Lors de la sortie de toute application, le testeur doit se familiariser avec les précédents et les réglementations existantes. Dans certains cas, une phrase ou une blague peut entraîner l'interdiction totale de la vente du jeu dans certains pays. Les causes suivantes d'éventuels défauts de localisation (détails ou contenus interdits dans une certaine région, par exemple) doivent être traitées :

- Les images,
- Le son et les bandes sonores,
- Des scènes réalistes ou historiques,
- Phrases ou citations.

### 7.2.2 Défauts et risques liés à la localisation

Une mauvaise connaissance ou un manque de connaissance de la langue cible ou du produit lui-même peut compliquer considérablement les tests de localisation, en particulier si le produit est un représentant du genre simulation : médical, technique ou sportif [URL4].

Il faut également garder à l'esprit que tester la localisation peut être un processus assez long, car il faut du temps pour étudier les caractéristiques des différentes régions.

Les principaux défauts de localisation peuvent être regroupés comme suit :

#### Aspects techniques

- Mojibake ("texte brouillé") / Garbage characters ("caractères brouillons"). Un texte déformé apparaît à la suite du décodage d'un texte utilisant un encodage de caractères mal choisi. Le résultat est le remplacement systématique des caractères par des caractères sans aucun rapport, souvent issus d'un système d'écriture différent. En règle générale, cela conduit à des textes illisibles.
- La phrase localisée ne rentre pas dans les limites fixées par l'interface (coupée, défilée (NDT : qui nécessite un déroulement dans le fenêtre)). Un terme traduit d'une langue à l'autre peut utiliser un nombre de caractères différent.
- Décalage. Après la localisation d'une application, la disposition des éléments de l'interface utilisateur peut nécessiter une reconfiguration afin de conserver l'alignement d'origine.
- Absence de traduction des lignes dans la langue localisée. Les testeurs doivent prêter attention aux textes des boîtes de dialogue, aux images ou aux captures d'écran des documents ou de l'interface utilisateur. Tous ces contenus doivent être localisés pour répondre aux attentes des utilisateurs.
- Le chevauchement . Cela se produit lorsque certains contrôles d'une fenêtre ou d'une boîte de dialogue se superposent à d'autres.
- Textes manquants. Lors de la traduction ou de la création d'applications, des textes peuvent occasionnellement être perdus.
- Police / taille incorrectes. Les pays utilisent des polices et des tailles par défaut différentes. Par exemple, les pays asiatiques utilisent généralement une taille de police 9 par défaut, tandis que les États-Unis utilisent une taille de police 8. Ce problème n'interfère généralement avec aucune fonctionnalité, mais il affecte grandement l'expérience utilisateur, rendant le texte difficile à lire.
- Touches de raccourci incorrectes. Dans certains cas, le raccourci clavier n'est pas disponible en raison de l'absence de la lettre dans la langue ou le clavier localisé.
- Variables dans les gabarits de texte. Les différents types de langues (analytiques, synthétiques) utilisent différentes formes de déclinaisons, de cas et de pluriels, elles peuvent changer ou non les terminaisons et les sons, les variables peuvent donc affecter les formes des mots. L'une des



erreurs les plus courantes commises par les développeurs consiste à diviser en parties les phrases contenant des variables. La structure grammaticale de la langue cible est presque la pierre angulaire de la localisation. En effet, lorsqu'ils ajoutent des variables, les développeurs ne tiennent souvent pas compte du fait que, dans un certain nombre de langues, leur signification peut modifier la construction d'une phrase.

- Désynchronisation de la séquence sonore de l'original et de la traduction. Lors de la traduction, les phrases des personnages peuvent être plus longues ou, à l'inverse, plus courtes que dans l'original en raison d'un nombre différent de syllabes. Les clips vidéo des jeux vidéo sont souvent préenregistrés, ce qui oblige les localisateurs à adopter une approche plus prudente de la traduction des indices : il est important de maintenir un équilibre entre la précision de la traduction et le nombre de syllabes, tout en tenant compte de la synchronisation du son avec les mouvements des lèvres des personnages (lip sync).

### Défauts de traduction

- Traduction ou translittération incorrectes de noms propres, de dates, de valeurs numériques, de noms d'événements historiques, de jours fériés, de réalités, de jargon, de blasphèmes, de vocabulaire familier, d'abréviations, de noms et de surnoms " évocateurs ", etc.
- Termes contradictoires. Un même terme doit avoir une traduction cohérente dans toute l'application. Il arrive souvent que les traductions soient incompatibles entre elles.

### Adaptabilité culturelle du contenu

- Humour . L'attitude à l'égard des objets de plaisanterie, d'humour et de satire, la considération de l'humour et sa licéité par rapport à des objets individuels, etc.
- Religion. L'attitude du public cible à l'égard des questions religieuses, y compris les sites religieux, les rituels et les cérémonies. Attitude du public cible à l'égard des cultes marginaux (satanisme, paganisme).
- Exactitude historique et perception des événements. L'interprétation d'événements historiques, de conflits militaires, de découvertes dans le domaine de la science, de l'apparence et des caractéristiques d'objets existants spécifiques, des particularités du mode de vie des peuples, etc.
- Particularités de la culture nationale et de la vision du monde (y compris les attitudes à l'égard des enfants, du sexe, de la violence, des autres cultures, etc.). Stéréotypes nationaux, cuisine nationale, vêtements, mode de vie des peuples. Attitude envers les enfants, attitude envers les minorités sexuelles, violence envers les animaux, etc.
- Restrictions légales. Prise en compte des classements par âge, de la législation en matière de protection de l'enfance, des croyances religieuses, de la propagande de la violence, de la drogue, des relations sexuelles, de la race, du terrorisme, des discours contre le gouvernement et le système étatique, de l'interdiction d'utiliser des symboles spécifiques, etc.
- Localisation excessive. Tout ne doit pas être localisé et certains éléments doivent conserver leur aspect original sans traduction, par exemple : les marques, le logo, les abréviations, les noms de produits....

## 7.3 Approche de test de localisation et réalisation des tests

### 7.3.1 Différence entre les tests complets et partiels de localisation

#### Test complet de la localisation

Cela implique un test complet de la localisation pour détecter les défauts. Ce type de test n'est efficace que lors du développement d'une nouvelle localisation pour un client, à condition que toutes les chaînes aient été traduites et n'aient pas été testées auparavant. C'est le plus coûteux, car il implique tous les types de tests.

## Test partiel de la localisation

Ce test est utilisé lorsque du texte est modifié dans des localisations qui ont été testées précédemment. La validation permet d'identifier les lignes qui ont été modifiées dans le cadre d'une mise à jour majeure qui a affecté l'histoire principale et sa localisation de référence. Seul le texte modifié est testé, qui doit correspondre au texte précédemment présenté dans le jeu. Cette approche de test est considérée comme optimale en termes d'efficacité tout en minimisant les coûts.

### 7.3.2 Procédures et approches pour tester la localisation au cours du cycle de vie du développement logiciel d'un jeu vidéo

Les tests de localisation permettent de vérifier la traduction, les fichiers de soutien, la justification et l'adaptabilité correctes des éléments d'interface, ainsi que les règles d'écriture du texte.

L'objectif des tests de localisation est de s'assurer que le jeu vidéo supporte une interface et des fonctionnalités multilingues, et qu'il n'y a pas de problèmes de localisation (traduction dans une autre langue, format des dates et des nombres, adresses postales, ordre des noms et des prénoms, devises, etc.)

Le processus des tests de localisation comprend :

- La détermination et l'étude de la liste des langues soutenues,
- Les tests de l'exactitude de la traduction, y compris des éléments de l'interface utilisateur, des messages du système et des défauts,
- La vérification de la traduction de la section "Aide" et de la documentation d'accompagnement (le cas échéant).
- Les tests de localisation consistant à comparer les chaînes de caractères traduites par l'équipe de localisation aux chaînes de caractères de la localisation de référence afin de constater :
  - Les défauts de grammaire, de ponctuation, de syntaxe,
  - Des violations des exigences régionales de la localisation testée (format de l'heure, de la date, des mesures, conformité légale, etc.)
- L'absence des données techniques nécessaires (variables, sections utilisées pour la mise en forme du texte, etc.)
- Les violations du style artistique et du contexte de la localisation de référence,
- Les défauts d'affichage du texte dans les interfaces du jeu vidéo (lignes trop longues, violation du format d'affichage des polices de caractères, etc.)

Tous les tests susmentionnés doivent être effectués par l'équipe chargée de tester le jeu. Cependant, il n'est pas toujours possible de tester la grammaire, la ponctuation, la syntaxe ou le style artistique et le contexte de la localisation de référence, car il est assez difficile de trouver des testeurs connaissant chacune des langues dans lesquelles le jeu est traduit. L'exécution de ces tests incombe aux localisateurs. Les testeurs vérifient les violations des exigences régionales de la localisation testée, contrôlent l'absence des données techniques nécessaires et identifient les défauts d'affichage du texte dans les interfaces du jeu. (Pour plus de détails, voir la section "7.3.3. Types de tests de localisation").

### L'étape préliminaire des tests de localisation

Cette étape comprend :

- La fourniture aux testeurs de toute la documentation nécessaire sur le produit,
- La création d'un glossaire et d'une mémoire de traduction pour aider les testeurs à interpréter correctement les termes utilisés,

- La fourniture d'une version antérieure du produit, s'il a déjà été localisé auparavant, à des fins d'évaluation,
- La sélection et la configuration d'outils de gestion des défauts - un document ou une plateforme où seront enregistrés tous les défauts constatés lors des tests de localisation.

### Tester les caractéristiques régionales et culturelles

C'est l'une des étapes les plus importantes des tests de localisation. Des captures d'écran ou une version localisée du jeu seront utilisées. Les éléments suivants doivent être testés :

- Le format de la date et de l'heure pour la région sélectionnée,
- Les formats des numéros de téléphone et des adresses,
- Les schémas de couleurs,
- La conformité des noms de produits avec les standards régionaux,
- Le format des devises,
- Les unités.

### Tests linguistiques

Les fonctionnalités linguistiques sont testées. Le testeur doit s'assurer que :

- La même terminologie est utilisée,
- Il n'y a pas de défauts grammaticaux,
- Il n'y a pas de défaut d'orthographe,
- Les règles de ponctuation sont respectées,
- Le sens correct du texte est utilisé (de droite à gauche ou de gauche à droite),
- Les noms corrects des marques, des villes, des lieux, des positions, etc. sont définis.

### Interface utilisateur (ou apparence)

Il est important de s'assurer de ce qui suit :

- Toutes les légendes des images sont localisées,
- La mise en page de la version localisée est la même que celle de l'original,
- Les sauts de ligne sur les pages / écrans sont placés conformément aux règles de la langue cible,
- Les conversations, les pop-ups et les notifications s'affichent correctement,
- La longueur des lignes ne dépasse pas les limites existantes et le texte s'affiche correctement (parfois, le texte de la traduction est plus long que l'original et ne tient pas sur les boutons).

### Fonctionnalité

Il est nécessaire de tester si l'application localisée fonctionne correctement, l'attention est donc attirée sur :

- La fonctionnalité d'un produit localisé,
- Les fonctions permettant de saisir des informations,
- La prise en charge des caractères spéciaux pour les différentes locales et langues,
- La prise en charge des raccourcis clavier,
- La prise en charge pour diverses polices de caractères,
- La prise en charge de divers séparateurs de format.

### Les tests de vidéos et d'échelles incluent des tests :

- La correspondance de la longueur de l'échelle et de l'élément du jeu auquel elle appartient,
- La correspondance de l'échelle avec les personnages par genre,
- La pureté du son (c'est-à-dire l'absence d'interférences, ainsi que la synchronisation de l'échelle de l'original et de la traduction et de leur intensité sonore l'une par rapport à l'autre),
- La fiabilité de la bande sonore, y compris l'aspect historique,
- Les caractéristiques stylistiques et culturelles du son (accents, caractéristiques de la parole).

#### 7.3.3 Types de tests de localisation

##### Localisation de la boîte

Si un jeu est distribué et vendu sur un support physique, ce qui est écrit sur l'emballage est localisé. S'il n'est pas vendu sur un support physique, mais sur une plateforme, sa page dans le magasin est traduite : description et captures d'écran. La localisation de la boîte se limite à cela.

##### Localisation de l'interface

La description et la boîte seront traduites dans le jeu, l'interface, la page d'aide, les libellés des boutons, etc. Un type de localisation inhabituel se présente dans une situation où l'inscription sur le bouton "Play" est faite dans une langue alors que l'intrigue est entièrement dans une autre langue.

##### Localisation du texte

En général, tous les textes du jeu doivent être traduits en sous-titres. Cela signifie qu'un utilisateur peut écouter et essayer de comprendre, par exemple, l'argot afro-américain dans le jeu, tout en voyant les sous-titres dans une autre langue.

##### Localisation avec interprétation vocale

Les paroles et les dialogues sont traduits et prononcés par les acteurs. Si la localisation avec l'interprétation vocale est de bonne qualité, elle n'est pas perçue comme quelque chose d'étranger.

##### Localisation du graphisme

Tout jeu contient un certain type de moteur, de conception, d'objets graphiques, de textures - tout ce qui n'est pas du texte, par exemple une inscription sur une clôture dans un jeu. La localisation graphique implique que toutes les inscriptions à l'intérieur doivent être traduites. Il peut s'agir, par exemple, de journaux, d'enseignes de magasins et de certaines notes.

Souvent, les actions du jeu se déroulent à certains endroits. En même temps, l'approche de la localisation peut être différente. Si un objet (par exemple, les notes d'un journal) contribue à l'intrigue, il doit être traduit, sinon un point important sera perdu. Dans le même temps, les inscriptions sur les murs dans la langue du lieu où se déroule le jeu n'ont pas besoin d'être traduites s'il s'agit simplement d'un arrangement.

##### Localisation approfondie - Adaptabilité culturelle

Il s'agit d'une adaptation culturelle, lorsque le jeu est entièrement refait. Il ne reste que le code source et les mécaniques. Les développeurs peuvent refaire les textures, l'intrigue, les dialogues et les modèles de personnages et faire un jeu complètement différent sur le moteur du jeu vidéo. Cela se fait assez rarement, mais cette méthode de localisation existe toujours. Cela se fait dans les cas où un jeu sans une telle adaptabilité ne peut pas être utilisé par le public et vendu sur un marché particulier.

### 7.3.4 Domaines de responsabilité des intervenants

Rôle	Responsabilités
Rédacteur / Concepteur narratif	<ul style="list-style-type: none"><li>Élaboration d'une stratégie et d'un plan de localisation.</li><li>Définition de la terminologie, traduction des concepts spécifiques au genre du jeu et des termes difficiles à traduire liés au projet.</li><li>Explication du contexte aux rédacteurs et aux traducteurs.</li></ul>
Traducteur	<ul style="list-style-type: none"><li>Traduction du texte</li></ul>
Rédacteur en chef	<ul style="list-style-type: none"><li>Relecture du matériel traduit.</li><li>Test de conformité du texte aux caractéristiques culturelles ainsi qu'aux aspects moraux et éthiques de la langue dans laquelle la localisation est effectuée.</li><li>Contrôle de l'uniformité du style de la traduction.</li></ul>
Testeur de localisation	<ul style="list-style-type: none"><li>Tester l'exactitude de la traduction (contexte et sens des remarques, cohérence du texte, conformité avec le style de jeu).</li><li>Test de la conformité technique de la traduction (apparence, interfaces, fonctionnalité des polices, séparateurs, caractères spéciaux, etc.)</li><li>Contrôle de l'intégration technique de la traduction (avec les applications internes et externes).</li></ul>

## 7.4 Outillage pour les tests de localisation

Les outils de test de localisation peuvent être utilisés à différents stades du test et pour répondre à différents problèmes. Ces outils comprennent :

- Outils de comparaison visuelle des chaînes de caractères. Ces outils permettent de comparer les chaînes de la localisation de référence et de la localisation cible.
- Outils de comparaison automatique de chaînes. Les outils de ce type comprennent tous les scripts auxiliaires, programmes, utilitaires, grâce auxquels il est possible d'obtenir des données sur la présence de défauts de toute nature dans la localisation testée.
- Outils permettant de tester la structure des fichiers de la localisation pour détecter les fichiers manquants ou redondants.
- Voici quelques exemples d'outils automatisés permettant de tester la localisation :
  - des outils qui comparent les captures d'écran de deux localisations (référence et vérifiée) et génèrent des défaillances en cas de divergence importante,
  - des outils qui déterminent la présence de changements dans la localisation cible en fonction de la présence de changements dans la référence,
  - des outils qui comparent les variables et les valeurs numériques dans chaque paire de chaînes de caractères référence - et localisation cible.

## 8. Références

### 8.1 Standards

[ISO25000] ISO/IEC 25000:2014, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE

[ISO25010] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models

### 8.2 Documents ISTQB

[ISTQB\_AL\_SEC] ISTQB Advanced Level Security Testing Syllabus, Version 2016

[ISTQB\_ALTA\_SYL] ISTQB Advanced Level Test Analyst Syllabus, Version 3.1.2

[ISTQB\_ALTTA\_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 4.0

[ISTQB\_ALTM\_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 2012

[ISTQB\_CTFL\_MAT] ISTQB Mobile Application Tester, Version 2019

[ISTQB\_EXAM\_S&R] ISTQB Exam Structure and Rules, Game Testing, Version 1.0

[ISTQB\_FL\_AT] ISTQB Foundation Level Agile Tester Syllabus, Version 2014

[ISTQB\_FL\_PT] ISTQB Foundation Level Performance Testing Syllabus, Version 2018

[ISTQB\_FL\_SYL] ISTQB Foundation Level (Core) Syllabus, Version 2018

[ISTQB\_GLOSSARY] ISTQB Glossary of Terms used in Software Testing, <https://glossary.istqb.org/>

[ISTQB\_UT\_SYL] ISTQB Foundation Level Usability Testing Syllabus, Version 2018

### 8.3 Livres

[Nystrom14] Nystrom, R. (2014). Game programming patterns. Genever Benning., ISBN: 978-0990582908. URL: <https://www.gameprogrammingpatterns.com/>

[Gregory18] Gregory, J. (2018). Game engine architecture. AK Peters/CRC Press., ISBN: 978-1466560017. URL: <https://www.gameenginebook.com/>

[Buttfield19] Buttfield-Addison, P., Manning, J., & Nugent, T. (2019). Unity game development cookbook: essentials for every game. O'Reilly Media., ISBN: 9781491999158

[Lee16] Lee, J. (2016). Learning unreal engine game development. Packt Publishing Ltd., ISBN: 9781784398156

[Tavakkoli18] Tavakkoli, A. (2018). Game Development and Simulation with Unreal Technology. CRC Press., ISBN-13: 978-1498706247

[Romero19] Romero, M., & Sewell, B. (2019). Blueprints Visual Scripting for Unreal Engine: The faster way to build games using UE4 Blueprints. Packt Publishing Ltd., ISBN: 9781789347067

---

[Chandler11] Chandler, H. (2011). The Game Localization Handbook 2nd Edition, Jones & Bartlett Learning, ISBN: 0763795933

[Retsker81] Retsker, Ya. I. (1981). Textbook for translation from English into Russian. M.: Prosveschenie. (In Russian).

## 8.4 Liens (Web/Internet)

Note : Toutes les références sont à jour au 28 avril 2022.

- [URL1] <https://research.ncl.ac.uk/game/mastersdegree/workshops/technicalrequirementschecklists/Technical%20Requirements%20Checklist%20Workshop.pdf>
- [URL2] <https://docs.microsoft.com/en-us/gaming/gdk/content/gc/live/get-started/live-xbl-overview>
- [URL3] <https://gamepad-tester.com/>
- [URL4] <https://igda-website.s3.us-east-2.amazonaws.com/wp-content/uploads/2021/04/09142137/Best-Practices-for-Game-Localization-v22.pdf>

## 9. Annexe A - Objectifs d'apprentissage/Niveau de connaissance

Les objectifs d'apprentissage suivants sont définis comme s'appliquant à ce syllabus. Chaque sujet du syllabus sera examiné en fonction de l'objectif d'apprentissage qui lui est associé. Les objectifs d'apprentissage commencent par un verbe d'action correspondant à leur niveau cognitif de connaissance, comme indiqué ci-dessous.

**Niveau 1 : Se souvenir (K1)** - le candidat se souviendra, reconnaîtra et se rappellera d'un terme ou d'un concept.

**Verbes d'action** : identifier, rappeler, se souvenir, reconnaître.

**Exemples** :

- "Identifier les objectifs de test typiques".
- "Rappeler les concepts de la pyramide des tests".
- "Reconnaître comment un testeur ajoute de la valeur à la planification des itérations et des releases".

**Niveau 2 : Comprendre (K2)** - le candidat peut sélectionner les raisons ou les explications des affirmations liées au sujet, et peut résumer, comparer, classer et donner des exemples pour le concept de test.

**Verbes d'action** : classer, comparer, contraster, différencier, distinguer, exemplifier, expliquer, donner des exemples, interpréter, résumer.

**Exemples** :

- "Classer les différentes options pour la rédaction des critères d'acceptation".
- " Comparer les différents rôles dans les tests " (rechercher les similitudes, les différences ou les deux).
- " Faire la distinction entre les risques projet et les risques produit" (permet de différencier les concepts).
- " Donner un exemple de l'objectif et du contenu d'un plan de test".
- "Expliquer l'impact du contexte sur le processus de test".
- " Résumer les activités du processus de revue".

**Niveau 3 : Appliquer (K3)** - le candidat peut exécuter une procédure lorsqu'il est confronté à une tâche familière, ou sélectionner la procédure correcte et l'appliquer dans un contexte donné.

**Verbes d'action** : appliquer, mettre en œuvre, préparer, utiliser.

**Exemples** :

- "Appliquer la priorisation des cas de test" (doit faire référence à une procédure, une technique, un processus, un algorithme, etc.).
- "Préparer un rapport de défaut".
- "Utiliser l'analyse des valeurs limites pour dériver les cas de test".

**Références** pour les niveaux cognitifs des objectifs d'apprentissage:

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon



## 10. Annexe B - Matrice de traçabilité des objectifs métiers avec les objectifs d'apprentissage

Cette section présente la traçabilité entre les objectifs métiers et les objectifs d'apprentissage du Syllabus Testeur Certifié en tests de jeux vidéo.

Objectif Métier: Testeur Certifié en tests de jeux vidéo		GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-1	Comprendre les concepts de base des jeux vidéo et des tests de logiciels de jeu	17					
GaMe-2	Déterminer les risques, les objectifs et les exigences du logiciel de jeu en fonction des besoins et des attentes des parties prenantes		12				
GaMe-3	Concevoir, implémenter et réaliser des tests de base des logiciels de jeux vidéo			6			
GaMe-4	Connaître les approches des tests de jeux vidéo et leur objectif				9		
GaMe-5	Reconnaître les outils soutenant les tests de jeux vidéo					7	
GaMe-6	Déterminer comment les activités de test s'alignent sur le cycle de vie du développement logiciel et réduire le coût du développement et de la publication des jeux vidéo						6

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
1. 1.Spécificité des tests de jeux vidéo								
GaMe-1.1.1	K1	Reconnaître les objectifs et les spécificités des tests de jeux vidéo	x					
GaMe-1.1.2	K2	Donner des exemples de risques liés au produit dans le domaine des jeux vidéo	x	x				
GaMe-1.1.3	K2	Donner des exemples de défauts spécifiques liés aux tests de jeux vidéo	x					
GaMe-1.1.4	K2	Résumez comment les risques des tests de jeux vidéo peuvent être atténués		x				
GaMe-1.1.5	K2	Comparer les activités de test de jeux vidéo avec celles de jeu	x					
GaMe-1.2.1	K1	Reconnaître les rôles et les tâches spécifiques au sein de l'équipe de développement de jeux vidéo	x					
GaMe-1.3.1	K1	Rappeler les activités de test tout au long du cycle de vie du développement logiciel des jeux vidéo					x	
2. 2. Tester les mécaniques des jeux vidéo								
GaMe-2.1.1	K2	Classer les types de mécaniques de jeux vidéo	x					
GaMe-2.1.2	K2	Faire la différence entre tester les mécaniques de gameplay et les mécaniques non liées au gameplay			x			

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-2.1.3	K2	Différencier les tests des principales mécaniques et des méta-mécaniques			x			
GaMe-2.1.4	K2	Différencier les tests des mécaniques client, serveur et client-serveur			x			
GaMe-2.1.5	K2	Donner des exemples de défauts dans les mécaniques des jeux vidéo		x				
GaMe-2.2.1	K2	Résumer les principales approches et les principaux objets de test à différents stades de la création d'un jeu				x	x	
GaMe-2.2.2	K2	Distinguer l'importance de tester les mécaniques des jeux vidéo	x					
GaMe-2.2.3	K2	Distinguer l'importance de réviser la documentation décrivant les mécaniques des jeux vidéo	x					
GaMe-2.2.4	K3	Appliquer les approches fondamentales pour tester les mécaniques des jeux vidéo				x		
3. 3. Tests du graphisme								
GaMe-3.1.1	K2	Expliquer les caractéristiques du contenu graphique d'un jeu vidéo	x					
GaMe-3.1.2	K2	Classer les types de défauts dans le contenu graphique		x				
GaMe-3.2.1	K2	Résumez les principales approches de test des aspects artistiques				x		
GaMe-3.2.2	K2	Résumer les principales approches de tests techniques				x		
GaMe-3.2.3	K2	Résumer les principales approches des tests du gameplay				x		
GaMe-3.3.1	K3	Appliquer les approches fondamentales des tests de graphisme			x			
GaMe-3.3.2	K2	Expliquer l'importance de tester la validité historique des graphiques		x				
GaMe-3.4.1	K2	Résumer l'utilisation des outils de tests du graphisme					x	
4. 4. Tests du son								
GaMe-4.1.1	K1	Rappeler les caractéristiques du contenu sonore d'un jeu vidéo	x					
GaMe-4.2.1	K1	Rappeler les types de défauts dans le contenu sonore		x				
GaMe-4.2.2	K2	Classer les défauts dans le contenu sonore		x				
GaMe-4.3.1	K2	Résumer les principales approches des tests du contenu sonore				x		
GaMe-4.3.2	K2	Résumer les principales approches pour tester le mélange de musique et de sons				x		
GaMe-4.3.3	K2	Résumer les principales approches pour tester la composition musicale				x		
GaMe-4.4.1	K2	Expliquer les niveaux de tests des contenus audio-musicaux	x					
GaMe-4.4.2	K1	Rappeler les caractéristiques de l'intégration des sons dans le client	x					
GaMe-4.4.3	K1	Rappeler les domaines de responsabilité des tests effectués sur le son						x
GaMe-4.4.4	K3	Appliquer les approches de test du son			x			
GaMe-4.5.1	K2	Résumer l'utilisation d'outils de test pour le son					x	
5. Tests des niveaux de jeu								
GaMe-5.1.1	K1	Rappeler les composants des niveaux de jeu	x					

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-5.1.2	K2	Classer les défauts habituels dans les niveaux de jeu		x				
GaMe-5.2.1	K2	Résumer les tests effectués aux différents stades de la création des niveaux de jeu				x		x
GaMe-5.2.2	K2	Comparer les domaines de responsabilité des spécialistes qui participent aux tests des niveaux de jeu						x
GaMe-5.3.1	K2	Résumer l'utilisation des outils permettant de tester les niveaux de jeu					x	
6. Game Controllers Testing								
GaMe-6.1.1	K2	Classer les périphériques d'entrée habituels et les périphériques spécialisés	x					
GaMe-6.1.2	K2	Donner des exemples de différents dispositifs d'entrée en fonction de leur application		x				
GaMe-6.1.3	K1	Rappeler les différents types de manettes de jeux vidéo	x					
GaMe-6.1.4	K2	Classer les défauts d'un jeu vidéo liés aux spécificités des contrôleurs de jeu, et les causes possibles de leur apparition		x				
GaMe-6.2.1	K2	Donner des exemples de conditions de test à couvrir pour tester les contrôleurs de jeux vidéo		x				
GaMe-6.2.2	K2	Classer les tâches des spécialistes UX, des testeurs et des concepteurs de jeux lors des tests de jeux vidéo						x
GaMe-6.3.1	K2	Résumer l'utilisation des outils permettant de tester le comportement des contrôleurs de jeux vidéo					x	
7. 7. Tests de localisation								
GaMe-7.1.1	K1	Rappeler les principaux objectifs de l'internationalisation et de la localisation	x					
GaMe-7.1.2	K1	Connaître les étapes des tests de localisation			x			
GaMe-7.1.3	K2	Comparer les capacités d'internationalisation et de localisation		x				
GaMe-7.2.1	K2	Classer les défauts de localisation et leurs causes		x				
GaMe-7.3.1	K1	Connaître les tests complets et partiels de localisation				x		
GaMe-7.3.2	K3	Classer les types de test de localisation						x
GaMe-7.3.3	K2	Résumer les tâches de test pour un rédacteur, un éditeur, un traducteur et un testeur de localisation						x
GaMe-7.4.1	K2	Résumer l'utilisation des outils permettant de tester la localisation des jeux vidéo					x	

## **11. Annexe C - Notes de livraison ("Release Notes")**

Il s'agit de la première release du syllabus Testeur certifié de jeux vidéo. Le développement de ce module a commencé en 2020 en réponse au marché croissant du logiciel de divertissement interactif pour lequel le test en pratique est un domaine hautement technique mais où aucun standard n'existait pour le test de jeux vidéo.

L'équipe de développement de Testeur certifié de jeux vidéo évalue à plus de 400 000 le nombre de spécialistes dans le domaine du développement de jeux vidéo.

Le module Testeur certifié de jeux vidéo permet d'acquérir une certification internationalement reconnue afin d'être reconnu par ses collègues, ses employeurs et ses clients.

## 12. Annexe D - Termes spécifiques aux tests de jeux vidéo et autres termes

Terme en français	Terme en anglais	Définition
Modèle 3D	3D model	Modèle 3D d'un objet utilisant des formes mathématiques visuelles.
accéléromètre	accelerometer	Un capteur qui permet au logiciel de déterminer et de transmettre des informations sur la façon dont la manette du jeu vidéo est située dans l'espace à un moment donné.
ambiance	ambient	Le son de l'environnement qui accompagne un certain lieu, une certaine situation ou une certaine étape d'un jeu vidéo.
animation	animation	Technique permettant de créer l'illusion d'images en mouvement à l'aide d'une séquence d'images fixes et se remplaçant les unes les autres à une fréquence élevée.
son d'ambiance	background sound	Musique d'ambiance du jeu vidéo pour renforcer l'atmosphère d'un jeu vidéo.
effet binaural	binaural effect	Illusion auditive observée lorsque des stimuli oscillatoires sont délivrés à deux fréquences adjacentes à chaque oreille en même temps.
boss	boss	Un adversaire contrôlé par ordinateur qui est beaucoup plus difficile à vaincre qu'un ennemi normal dans un jeu vidéo.
mécanique du client	client mechanics	Mécaniques des jeux vidéo qui fonctionnent du côté du client du jeu vidéo. Voir également Mécanique des jeux vidéo.
collision	collision	Technologie responsable de l'intersection des objets des jeux vidéo.
code couleur	color coding	Le processus de marquage des objets avec des couleurs différentes comme moyen d'identification.
image générée par ordinateur (CGI)	computer-generated imagery (CGI)	L'application du graphisme informatique en 3D pour créer des effets spéciaux.
étape du concept	concept stage	Étape de planification initiale d'un projet de développement de jeux vidéo, qui se concentre sur la création de concepts principaux et la rédaction de documents de conception initiaux qui décrivent le futur jeu vidéo.
mécanique principale	core mechanics	Mécaniques des jeux vidéo qui définissent l'expérience de jeu prévue pour le joueur. Voir également Mécaniques des jeux vidéo.
adaptation culturelle	cultural adaptation	Processus d'adaptation des conditions de l'environnement des jeux vidéo pour créer les caractéristiques d'une culture donnée.
déformation	distortion	Changement de forme de l'onde sonore au cours du traitement. D'après The Oxford Companion to the English Language.
combat	fighting	Genre de jeux vidéo simulant le combat au corps à corps d'un petit nombre de personnages dans un espace et un temps limités.

Terme en français	Terme en anglais	Définition
Tir à la première personne (FPS)	first person shooter	Sous-genre de jeux vidéo de tir centré sur les armes à feu et autres combats à base d'armes dans une perspective à la première personne, le joueur vivant l'action à travers les yeux du protagoniste.
taux de rafraîchissement	frame rate	Le nombre d'images modifiées par unité de temps.
manette de jeu	gamepad	Type de manette de jeu à deux mains utilisée avec les consoles de jeux vidéo. Voir également contrôleur de jeux vidéo.
gameplay	gameplay	Les règles qui décrivent l'interaction entre le monde du jeu vidéo et le joueur.
personnage de jeu vidéo	game character	Une personne ou toute autre entité agissant dans un jeu vidéo.
boîte grise	grey box	Étape du développement des niveaux de jeux vidéo au cours de laquelle une "ébauche" de mise en page 3D est créée à partir de formes unicolores non définies afin de tester le processus de jeu.
gyroscope	gyroscope	Capteur qui mesure l'orientation et la vitesse angulaire d'un corps dans son cadre de repos.
piratage	hacking	Manière malhonnête d'obtenir un avantage dans un jeu vidéo.
aide	help	Documentation qui explique les fonctionnalités d'un programme et aide l'utilisateur à en comprendre les facilités.
exactitude historique	historical accuracy	Conformité du contenu d'un jeu vidéo au contenu correspondant dans la vie réelle.
hit box. Synonyme : zone affectée	hit box. Synonym: affected area	Modèle d'objet 3D simplifié pour la détection et la gestion des collisions d'objets dans un jeu vidéo.
lieu inaccessible	inaccessible place	Un emplacement dans le monde des jeux vidéo qui est mis à la disposition d'un joueur de manière incorrecte.
décalage	lag	Retard dans les performances d'un jeu vidéo dû à une connexion Internet lente, à des fuites de mémoire ou à une utilisation sévère du CPU/RAM/HDD (par exemple, mise à jour des fenêtres/antivirus en arrière-plan).
conception du niveau	level design	Étape du développement d'un jeu vidéo qui consiste à créer des niveaux de jeu et des missions.
éditeur de niveau	level editor	Logiciel utilisé pour créer et modifier des lieux et des environnements dans un jeu vidéo.
Niveau de détail (LoD)	level of detail (LoD)	Technologie dans laquelle une copie simplifiée d'un objet est créée pour afficher le même objet vu à une distance différente.
prototype de niveau	level prototype	Un premier échantillon d'un niveau de jeu vidéo qui est créé comme preuve de concept. Voir également niveau de jeu vidéo.
local	locale	Un ensemble d'options qui définissent la langue, la région et toute variante spéciale de préférences dans l'interface utilisateur d'un jeu vidéo.
boîte à butin	loot box	Une récompense sous forme d'objet(s) virtuel(s) pour avoir terminé un match, gagné un niveau d'expérience, ou

Terme en français	Terme en anglais	Définition
		tout autre accomplissement dans le jeu. d'après Wikipédia
cartographie	mapping	Discipline du développement de jeux vidéo permettant de créer un niveau dans un jeu vidéo. Voir aussi niveau de jeux vidéo.
méta-mécaniques	meta mechanics	Mécanique de jeu vidéo définissant la manière dont un joueur est censé utiliser un jeu vidéo. Voir aussi les mécaniques des jeux vidéo.
objet mobile (Mob)	mobile object (Mob)	Un ennemi qui rôde dans une zone spécifique d'un jeu vidéo.
transformation	morphing	Effet visuel qui donne l'impression d'une transformation transparente d'un objet en un autre.
multiplateforme	multiplatform	La capacité d'utiliser un logiciel sur différentes plates-formes.
narratif	narrative	Une histoire véhiculée par les mécaniques d'un jeu vidéo. Voir également conception des jeux vidéo.
personnage non joueur (PNJ)	non-player character (NPC)	Personnage contrôlé par ordinateur. Voir aussi Mobbing.
occlusion	occlusion	Processus qui imite la façon dont les sons sont perçus dans des environnements "fermés".
plateforme	platformer	Genre de jeux vidéo consistant à sauter sur des plates-formes, à monter des escaliers et à collecter les objets nécessaires pour vaincre les ennemis ou terminer un niveau.
personnage-joueur	player character	Dans les jeux vidéo, personnage contrôlé par le joueur, habituellement protagoniste de l'intrigue du jeu. Voir également personnage de jeu vidéo.
joueur contre joueur (JcJ)	player versus player (PvP)	Mode de jeu vidéo dans lequel un joueur interagit avec un ou plusieurs autres joueurs.
polygone	polygon	Ensemble de sommets, d'arêtes et de faces qui définissent la forme d'un objet polyédrique en graphisme informatique 3D et en modélisation volumique.
phase de post-production	post-production stage	Processus de maintenabilité et de distribution d'un jeu vidéo, et de commercialisation après sa release.
phase de pré-production	pre-production stage	Processus de planification et de documentation des éléments d'un jeu vidéo.
phase de production	production stage	Processus de développement d'un jeu vidéo.
éditeur	publisher	Entreprise qui finance, distribue et commercialise un jeu vidéo. d'après Wikipédia
quête	quest	Un genre de jeux vidéo qui comprend des activités basées sur des objectifs, soit pour une histoire interactive, soit pour l'avancement du niveau du personnage. d'après Wikipédia
volant de course	racing wheel	Type de manette de jeu vidéo à deux mains utilisée pour imiter un volant, des pédales et un levier de vitesse.
raster	raster	Un type de graphisme qui représente une grille de pixels visualisable sur un écran.

Terme en français	Terme en anglais	Définition
réverbération	reverb	Un effet d'écho produit électroniquement. D'après Merriam-Webster
ancrage	rigging	Le processus de construction d'un modèle de squelette dans un jeu vidéo pour l'animer par la suite.
jeux vidéo (RPG)	role-playing game (RPG)	Genre de jeux vidéo dans lequel le joueur endosse le rôle de personnages dans un cadre fictif.
éclairage des scènes	scene lighting	La quantité, la taille, la couleur et la dureté de la lumière entourant un personnage pour correspondre à la scène d'un jeu vidéo.
mécaniques du serveur	server mechanics	Mécaniques de jeux vidéo qui fonctionnent sur le serveur de jeux vidéo. Voir également Mécaniques des jeux vidéo.
réglage	setting	L'environnement du jeu vidéo dans lequel se déroule l'action.
habillage	skinning	Le modèle de processus de liaison d'un personnage 3D à un squelette de modèle dans un jeu vidéo. Voir aussi ancrage.
saut de scène	skipping	Un arrière-plan audio défectueux qui donne l'impression que des parties de la bande sonore sont sautées.
discontinuité du son	sound discontinuity	Une absence prolongée et inattendue de son ou une coupure brutale du son.
effet sonore	sound effect	Un son autre que la parole ou la musique créé artificiellement pour un objet de jeu vidéo.
zone sonore	sound zone	Une technologie qui permet de créer des zones dans le jeu avec des propriétés sonores réelles.
speedrunner (coureur de vitesse)	speedrunner	Tentative de terminer un jeu vidéo le plus rapidement possible. d'après Wikipédia
stick	stick	Contrôleur de jeux vidéo dont la mobilité est limitée à deux degrés de liberté. Voir aussi contrôleur de jeux vidéo.
magasin	store	Un service qui assure la distribution numérique de jeux vidéo et d'autres contenus de jeux vidéo.
géométrie structurelle	structural geometry	Technologie qui fournit le relief du terrain et la surface sur laquelle les personnages de jeux vidéo peuvent se déplacer.
terrain	terrain	Ensemble d'éléments liés à la surface de jeu sur laquelle se déplace un personnage de jeu vidéo.
texture	texture	Image matricielle superposée à la surface d'un modèle polygonal pour lui donner de la couleur ou imiter le relief.
tuile	tile	Petit objet ou fragment à partir duquel un niveau ou une autre grande image est construit dans un jeu vidéo.
écran tactile	touchscreen	Un contrôleur de jeux vidéo qui repose sur le toucher physique d'un écran.
Trackball / boule de comande	trackball	Un contrôleur de jeu dans lequel une boule en rotation libre est utilisée comme dispositif d'entrée pour un jeu vidéo. Voir également contrôleur de jeux vidéo.
déclencheur	trigger	Une ou plusieurs actions qui déclenchent un événement.
jeux vidéo	video game	Un jeu vidéo dans lequel un joueur contrôle des images sur un écran vidéo. Merriam-Webster



Terme en français	Terme en anglais	Définition
console de jeux vidéo	video game console	Un dispositif électronique conçu pour les jeux vidéo.
contrôleur de jeux vidéo	video game controller	Un appareil utilisé pour fournir des données à un jeu vidéo.
conception de jeux vidéo	video game design	Le processus de création de la forme et du contenu du gameplay du jeu vidéo en cours de développement.
document de conception de jeux vidéo	video game design document	Documentation contenant une description détaillée du jeu vidéo en cours de développement.
concepteur de jeux vidéo	video game designer	Personne responsable de l'élaboration des règles et du contenu de gameplay. Voir également conception de jeux vidéo.
moteur de jeux vidéo	video game engine	Le logiciel dans lequel tous les autres composants d'un jeu vidéo sont construits.
environnement de jeux vidéo	video game environment	Le contenu du jeu : le gameplay, les niveaux de jeu, les objets du jeu et, dans certains cas, l'histoire du jeu.
niveaux de jeu vidéo	video game level	Un lieu distinct dans le monde virtuel d'un jeu vidéo.
mécaniques de jeux vidéo	video game mechanics	Les actions au sein d'un jeu vidéo définies par certaines règles.
objet de jeu vidéo	video game object	Tout objet avec lequel un joueur peut interagir dans un jeu vidéo.
ressource de jeu vidéo	video game resource	Entité ou caractéristique d'un jeu vidéo importante pour le joueur afin d'atteindre les objectifs du jeu vidéo.
état d'un jeu vidéo	video game state	La valeur de tous les paramètres et variables qui décrivent tous les objets d'un jeu vidéo à un moment donné.
effet visuel (VFX)	visual effect (VFX)	La création ou la manipulation de toute image à l'écran qui n'existe pas physiquement dans la vie réelle.
volume	volume	Le degré d'intensité d'un son. Merriam-Webster
boîte blanche	white box	Étape du développement des niveaux d'un jeu vidéo au cours de laquelle un plan en 3D est créé à partir de modèles temporaires afin de tester la géométrie des niveaux.

## 13. Annexe E – Index

modèle 3D, 13, 27, 28, 55, 56, 57  
accéléromètre, 60, 61  
tests ad-hoc, 22, 23  
ambiance, 43  
animation, 13, 28, 29, 30, 31, 32, 35, 40, 45, 47, 49, 67  
effet binaural, 44  
mécaniques du client, 19, 20, 21  
collisions, 30, 33  
codage des couleurs, 55  
conformité, 14, 24, 36, 37, 38, 49, 50, 66, 69, 73, 74, 76, 86  
Mécaniques principaux, 18, 19, 20, 22, 23  
adaptabilité culturelle, 75  
distorsion, 46  
tests fonctionnels, 15, 17, 19, 21, 23, 62  
niveaux de jeu, 20, 22, 26, 27, 31, 52, 53, 55, 56, 57  
manette de jeu, 59, 60, 61, 62, 63  
gyroscope, 61  
Exactitude historique, 40, 41, 66, 72  
internationalisation, 65, 70  
bruit de décalage (Foley), 48  
éditeur de niveau, 51, 52, 53, 58  
niveau de détail (LoD), 32, 36, 37, 39, 40  
locale, 66, 74  
localisation, 15, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 75, 76  
cartographie, 39  
méta-mécaniques, 18, 19, 20, 23  
multiplateforme, 13  
narration, 31, 50, 55, 75  
occlusion, 44  
tests en situation de jeu, 15, 37, 40, 56, 57  
phase de post-production, 17  
phase de pré-production, 17  
phase de production, 17, 20, 23, 36  
roue de course, 59, 60, 61, 62  
réverbération, 44  
ancrage, 30, 35  
éclairage de scène, 31  
mécaniques du serveur, 19, 20, 21  
habillage, 30, 35  
saut, 45  
discontinuités sonores, 48  
effets sonores, 22, 42, 43, 47, 49, 51, 53  
zones sonores, 45  
géométrie structurelle, 52  
textures, 28, 30, 32, 33, 34, 36, 37, 38, 39, 41, 57, 58, 65, 75  
écran tactile, 59  
trackball, 59  
déclencheur, 41, 53, 56, 57  
jeux vidéo, 12, 13, 14  
contrôleur de jeux vidéo, 59, 61, 62, 63  
conception de jeux vidéo, 15, 16, 17, 18, 22, 40, 50, 56, 62  
mécaniques des jeux vidéo, 18  
état des jeux vidéo, 18, 21, 24, 25, 26  
effet visuel (VFX), 22, 30, 35, 57  
volume, 44, 46, 47, 48, 49, 51