

PROJET D'ANNÉE COMPTE RENDU FINAL

Boxe Anglaise - Hajime no Ippo Edition



Réalisé par :

Abdelghani AKNOUN

Zaid El Gaoual

Ahmed ELATRI

Sommaire

- Introduction.....3
- Description fonctionnelle détaillée.....4
- Description technique :.....5
- Organisation du projet6
- Bilan personnel.....7

Introduction

Ce projet de développement de jeu vidéo en 3D s'inscrit dans le cadre du concours "Games on Web 2024", organisé par CGI en partenariat avec plusieurs universités prestigieuses. Notre équipe, composée de Abdelghani AKNOUN, Zaid El Gaoual et Ahmed ELATRI, a entrepris de créer une démo de jeu mettant en scène la boxe anglaise dans le contexte de l'édition olympique.

L'objectif principal de notre jeu est de permettre au joueur de devenir le champion incontesté de la boxe anglaise. Pour ce faire, le joueur devra participer à des tournois, des combats individuels et des entraînements intensifs afin d'améliorer les compétences de son personnage et de remporter les ceintures prestigieuses.

Nous avons choisi de développer ce jeu en utilisant les langages HTML, CSS et Javascript, ainsi que la librairie Babylon.js pour la création d'environnements 3D immersifs et interactifs.

Dans ce rapport, nous présenterons en détail le scénario du jeu, les fonctionnalités principales, l'architecture technique de notre solution, ainsi que notre organisation et notre expérience personnelle au cours du développement de ce projet.

Description fonctionnelle détaillée :

Dans cette section, nous détaillerons les fonctionnalités principales, les mécaniques de jeu et les interactions avec l'environnement de notre jeu de boxe anglaise en 3D.

Animations et Mouvements du personnage :

Le jeu comprendra un ensemble complet d'animations de mouvement pour le personnage principal, visant à offrir une expérience de jeu fluide et immersive. Ces animations ont été soigneusement conçues pour refléter fidèlement les mouvements réels de la boxe anglaise et pour apporter une dimension réaliste aux combats.

Les principales animations incluses dans le jeu sont les suivantes :

- **Boxing Stance** : Animation représentant le personnage adoptant une posture de boxe prête à l'action, offrant une défense solide et la possibilité de contre-attaquer.
- **Jab Body** : Animation du personnage lançant un coup de poing rapide au corps de l'adversaire, idéal pour porter des coups rapides et infliger des dégâts modérés.
- **Left Hook** : Animation du personnage effectuant un crochet du gauche, une attaque puissante visant à contourner la garde de l'adversaire et à infliger des dégâts significatifs.
- **Right Pivot Right Upper (Body)** : Animation du personnage esquivant vers la droite tout en lançant un uppercut droit au corps de l'adversaire, une technique efficace pour briser la garde et porter un coup dévastateur.
- **Dodging Backwards** : Animation du personnage esquivant habilement en arrière pour éviter les attaques adverses tout en maintenant une position défensive solide.
- **Crouse** : Animation permettant au personnage de déclencher une séquence rapide de directes alternant entre gauche et droite, offrant une offensive dynamique tout en maintenant une posture basse pour une meilleure défense.

Les mouvements du personnage incluront des déplacements latéraux, des pas en avant et en arrière, ainsi que des pivots pour esquiver les attaques adverses.

Les commandes correspondantes sur le clavier sont :

Touche haut : Avancer vers l'avant.

Touche bas : Reculer vers l'arrière.

Touche gauche : Déplacer le personnage vers la gauche.

Touche droite : Déplacer le personnage vers la droite.

En combinant ces mouvements de base avec les touches d'action spécifiques, telles que la touche de frappe, le joueur pourra orchestrer des séquences de combat fluides et réalistes, tout en bénéficiant de la liberté de mouvement offerte par les animations du personnage.

Collisions et physique :

Les interactions entre les personnages et l'environnement seront gérées par le moteur physique de Babylon.js, assurant des collisions réalistes lors des combats.

Les personnages réagiront aux coups en fonction de la direction et de la force des impacts, ajoutant un aspect de réalisme aux combats.

En combinant des animations authentiques de boxe avec des mécaniques de jeu engageantes et des interactions réalistes avec l'environnement, notre jeu vise à offrir une expérience immersive et captivante aux joueurs passionnés de boxe anglaise.

Description technique :

Dans cette section, nous allons aborder l'architecture du jeu, l'implémentation des fonctionnalités principales ainsi que l'utilisation des technologies telles que Babylon.js, HTML et JavaScript.

Scènes et modèles 3D :

La scène principale du jeu, gérée par la classe RingScene, comprend l'arène de combat avec le ring de boxe, ainsi que les éléments environnementaux essentiels. Cette scène est créée à l'aide de logiciels de modélisation Blender. En plus de la scène, des éléments comme la caméra et la lumière sont également intégrés pour offrir une expérience immersive.

La caméra, une instance de ArcRotateCamera dans notre cas, est positionnée pour fournir une vue dynamique de l'arène de combat, permettant au joueur de contrôler l'angle et le zoom de la caméra pendant le jeu. Elle est associée à la scène pour garantir un suivi précis des mouvements du joueur.

Quant à la lumière, une HemisphericLight, elle est utilisée pour éclairer la scène de manière réaliste, en fournissant une lumière ambiante qui simule l'éclairage naturel. Cela ajoute de la profondeur et de la dimension à l'environnement du jeu, améliorant ainsi l'esthétique visuelle globale.

En outre, le personnage principal du jeu, avec ses animations de mouvements de boxe, d'esquives et d'autres actions, est créé et intégré à la scène. Ces modèles et animations sont exportés dans un format .glb qui est compatible avec Babylon.js pour garantir une intégration fluide dans le jeu et une expérience de gameplay réaliste.

Implémentation des fonctionnalités principales :

Les joueurs disposent d'un contrôle total sur les mouvements de leur personnage grâce à l'utilisation du clavier. Cette liberté de mouvement inclut la possibilité d'avancer, de reculer, de se déplacer latéralement et d'exécuter diverses actions telles que les coups de poing et les esquives. Ces contrôles sont intégrés de manière fluide dans le gameplay pour offrir une expérience immersive.

La classe Player assure la gestion des déplacements du personnage, couvrant une gamme de mouvements comprenant les déplacements latéraux, les pas en avant et en arrière, ainsi que les pivots pour esquiver les attaques adverses. En utilisant les entrées du joueur, cette classe garantit des déplacements réactifs et naturels du personnage dans l'environnement du jeu, renforçant ainsi l'authenticité de l'expérience de jeu.

Quant aux animations du personnage, comme les coups de poing, les esquives et les mouvements de base, elles sont également prises en charge par la classe Player. Cette dernière charge les animations à partir des modèles 3D exportés depuis Blender, ce qui permet de les déclencher de manière contextuelle en réponse aux actions du joueur. Ainsi, les mouvements du personnage sont non seulement fluides et réalistes, mais également visuellement captivants, contribuant ainsi à une immersion totale dans le monde du jeu.

Utilisation de Babylon.js :

Le moteur de jeu Babylon.js est utilisé pour la création et le rendu des scènes 3D, la gestion des animations, des caméras et des lumières. Il offre une API riche et des fonctionnalités puissantes pour le développement de jeux en ligne.

HTML est utilisé pour définir la structure de la page web du jeu, tandis que JavaScript est utilisé pour la logique du jeu, y compris la gestion des entrées du joueur, les interactions avec les scènes 3D et la communication avec le serveur si nécessaire.

Organisation du projet :

L'organisation du projet s'est déroulée de manière collaborative, avec une répartition équilibrée des tâches entre les membres de l'équipe. Plutôt que d'attribuer des responsabilités spécifiques à chaque membre, nous avons opté pour une approche où nous travaillions ensemble sur toutes les phases du développement. Cette méthode nous a permis de partager nos connaissances et nos compétences, tout en favorisant une meilleure compréhension globale du projet. Au début du projet, nous avons établi un plan détaillé, mais nous avons dû être flexibles. Parfois, des ajustements étaient nécessaires en raison de problèmes techniques ou de nouveaux objectifs.

Pour faciliter notre progression et notre apprentissage, nous avons suivi des tutoriels sur Babylon.js, organisant des réunions de formation chaque mardi et samedi pendant deux heures. Ces sessions nous ont permis d'explorer les fonctionnalités de Babylon.js, de résoudre les problèmes rencontrés et de discuter des meilleures pratiques de développement. En travaillant de cette manière, nous avons pu développer nos compétences en matière de développement de jeux en 3D tout en avançant sur notre projet.

En ce qui concerne le planning de développement, nous avons adopté une approche flexible, axée sur des itérations régulières et des ajustements en fonction des progrès réalisés. Cela nous a permis de rester réactifs aux changements et d'ajuster nos priorités en fonction des défis rencontrés.

En ce qui concerne les outils de gestion de projet, nous avons utilisé Discord pour organiser nos tâches, suivre notre progression et collaborer efficacement.

Avec le début de notre stage récemment, notre disponibilité pour le projet de jeu de boxe était très limitée. Nous avons dû jongler entre nos emplois du temps chargés et le développement du jeu, ce qui a rendu difficile la réalisation de toutes les tâches prévues. Malgré notre engagement, nous avons été contraints de faire face à des retards et de prioriser nos responsabilités professionnelles et académiques. En conséquence, certaines fonctionnalités n'ont pas été achevées dans les délais souhaités. Bien que nous soyons déçus de ne pas avoir pu tout terminer, cette expérience nous a appris à gérer efficacement notre temps et nos priorités.

Bilan personnel :

Zaid EL GAOUAL :

Dans le cadre de mon projet sur l'intégration et l'amélioration des animations et des comportements des personnages non joueurs (NPC) dans un jeu développé avec Babylon.js, j'ai accompli de nombreux progrès tout en faisant face à divers défis. Ce bilan personnel détaille les réussites et les obstacles que j'ai rencontrés, y compris mes tentatives et échecs avec l'intégration de l'intelligence artificielle (IA).

Ce que j'ai accompli

1. Gestion des animations : J'ai mis en place un système efficace pour contrôler les animations des personnages, telles que marcher, reculer et frapper. Utilisant la méthode `playAnimation`, j'ai réussi à créer des transitions fluides entre les différents états, ce qui a amélioré l'immersion et la réactivité du jeu.

2. Contrôle des mouvements : J'ai développé des commandes permettant aux caractères de se déplacer en fonction de leurs interactions avec le joueur. Les méthodes `move` et `update` que j'ai élaborées gèrent ces mouvements de manière efficace, fournissant une base solide pour des comportements plus complexes.

3. Orientation des personnages : J'ai essayé de faire en sorte que les caractères soient toujours orientés vers un point central, tel que le milieu de l'arène, pour augmenter le réalisme de leur comportement. Bien que mes premières tentatives aient été infructueuses, elles ont démontré ma compréhension des exigences d'un gameplay réaliste.

Les erreurs et défis rencontrés

1. Problèmes d'extraction et d'accès aux fichiers : J'ai eu des difficultés techniques avec l'extraction et l'analyse des fichiers sources, ce qui a retardé l'accès au code nécessaire pour faire des ajustements.

2. Synchronisation des animations : Les animations ne se terminaient pas toujours où elles étaient censées, affectant le positionnement et l'orientation des personnages. Cela nécessite une gestion plus rigoureuse des événements de fin d'animation.

3. Problèmes de physique : L'intégration de la physique pour gérer les collisions et les interactions avec l'environnement a été problématique, les réactions physiques étant parfois imprévisibles.

4. Échec de l'implémentation de l'IA : Le défi le plus significatif a été ma tentative infructueuse d'intégrer une IA sophistiquée pour les NPC. Malgré mes efforts, je n'ai pas réussi à mettre en œuvre des comportements intelligents et autonomes pour les NPC, principalement en raison de mes lacunes en matière de compétences techniques dans ce domaine spécifique.

Leçons apprises

1. Importance des fondamentaux : J'ai appris l'importance de maîtriser les bases de la programmation des jeux et de la gestion des animations avant de tenter de mettre en œuvre des systèmes complexes comme l'IA.

2. Tests itératifs : Cette expérience a renforcé l'importance des tests itératifs pour identifier rapidement les problèmes et ajuster les comportements en conséquence.

3. Besoin de formation continue : Mon échec à implémenter l'IA a souligné le besoin de poursuivre mon éducation et ma formation, en particulier dans des domaines spécialisés tels que l'IA pour les jeux.

Conclusion

Mon parcours dans le développement de ce jeu a été à la fois instructif et difficile. Chaque étape m'a offert des opportunités d'apprentissage et m'a poussé à étendre mes compétences techniques. Alors que je continue de développer mes compétences en programmation et en conception de jeux, je reste déterminé à surmonter les défis futurs, en particulier dans l'intégration réussie de l'IA dans mes projets de jeux.

Abdelghani AKNOUN :

Dans ce projet, j'ai principalement travaillé sur la modélisation 3D et l'intégration des actifs dans le jeu, ainsi que sur les animations. Cette expérience a été extrêmement enrichissante à plusieurs égards. Tout d'abord, elle m'a permis de développer mes compétences en modélisation et en animation, en particulier dans des logiciels Blender. J'ai appris à créer des modèles de personnages, des environnements et même des éléments complexes comme le ring de boxe.

Une leçon importante que j'ai retenue de ce projet est l'importance de la planification et de l'organisation dès le début du processus de développement. Il est essentiel d'avoir une vision claire des objectifs du projet et de définir des priorités pour chaque étape du

développement. De plus, il est crucial d'établir une communication efficace au sein de l'équipe pour résoudre rapidement les problèmes et éviter les retards.

Pour les projets futurs, je recommanderais de consacrer plus de temps à la planification et à la documentation des processus. Cela peut aider à éviter les erreurs et les malentendus tout au long du développement. De plus, je suggérerais d'investir dans des outils de gestion de projet efficaces pour faciliter la collaboration et le suivi des tâches. Enfin, il est important de continuer à apprendre et à se former régulièrement pour rester à jour avec les nouvelles technologies et les meilleures pratiques de l'industrie.

Ahmed ELATRI

J'ai eu l'opportunité de participer à un projet unique au cours de cette année académique. Notre équipe s'est engagée dans la création d'un jeu de boxe en 3D utilisant la bibliothèque Babylon.js, et j'ai eu le privilège de me concentrer sur le développement des personnages et de leurs mouvements, en utilisant Blender comme principal outil de travail pour les animations.

Mon rôle dans ce projet a été double : d'une part, conceptualiser et modéliser les combattants, et d'autre part, créer les animations qui leur donnent vie. Cette expérience a été une opportunité d'appliquer mes connaissances tout en développant de nouvelles compétences techniques.

La phase initiale du projet a été consacrée à la conception de la scène et du personnage. En collaboration avec mes coéquipiers, nous avons défini les caractéristiques distinctives du combattant, en tenant compte à la fois des contraintes techniques et des exigences du gameplay. Cette étape a été cruciale et à la fois compliquée pour établir une base solide sur laquelle nous pourrions construire le reste du jeu.

Après, j'ai plongé dans le processus d'animation. Utilisant Blender, j'ai créé une variété de mouvements, des mouvements de base tels que les coups de poing et les uppercuts et les jabs aux séquences plus complexes de combinaisons de coups. L'objectif était de rendre les mouvements à la fois réalistes et fluides, ce qui a nécessité un équilibre entre la créativité et les contraintes techniques.

Travailler avec Babylon.js a également été une expérience enrichissante. Cette bibliothèque a offert des fonctionnalités puissantes pour la création d'environnements 3D et l'intégration des modèles de personnages et des animations. J'ai utilisé Babylon.js pour mettre en place les scènes de combat, gérer les collisions et intégrer les animations de manière interactive.

Néanmoins, ce projet n'a pas été sans ses défis. La gestion du temps et des ressources a été un défi constant. L'animation des personnages, en particulier, a exigé une attention minutieuse aux détails et une patience infinie pour obtenir les résultats désirés. De plus, la coordination avec les autres membres de l'équipe pour assurer la cohérence et l'harmonie visuelle dans le jeu a été parfois complexe.

Pour surmonter ces défis, j'ai utilisé les ressources mises à disposition, notamment les vidéos pédagogiques du professeur Olivier Arguimbau. Ses explications détaillées sur les techniques d'animation et les bonnes pratiques de modélisation m'ont été extrêmement