

# Rapport Soutenance n°1

## *Rio de Janeiro*

Semaine du 13 janvier 2025

---

### Informations Générales

- Nom du groupe: Booska D1
  - Promo : 2029
  - Classe : Prépa sup D1
- 

### Responsables du Projet

- Chef de Projet : Pages Virgile
  - Équipe de Développement :
    - Taland Alexandre
    - Nabi Chahine
    - Rischard Mattis
    - Onraed Antoine
- 

### Version du Document

- Version : 1.0
  - Dernière mise à jour : Semaine du 13 janvier 2025
- 

### Confidentialité

Ce document est confidentiel et destiné uniquement aux membres du projet et aux parties prenantes impliquées dans le développement de *Monkey Heist*. Toute divulgation non autorisée est interdite.

# Contents

<b>1 Nabi Chahine</b>	<b>4</b>
1.1 Gameplay : Développement et Implémentation . . . . .	4
1.1.1 Introduction . . . . .	4
1.1.2 Système de Mouvement . . . . .	4
1.1.3 Système de Tir . . . . .	4
1.1.4 Système de Mouvement de Caméra . . . . .	5
1.1.5 Système de Munitions . . . . .	5
1.1.6 Prochaine étapes . . . . .	6
1.1.7 Conclusion . . . . .	7
<b>2 Rischard Mattis</b>	<b>7</b>
2.1 Modélisation . . . . .	7
2.1.1 Introduction à la modélisation dans le projet . . . . .	7
2.1.2 Approche de modélisation . . . . .	7
2.1.3 Exemples d'assets modélisés . . . . .	7
2.1.4 Workflow et pipeline . . . . .	8
2.1.5 Planning et respect des objectifs . . . . .	9
2.2 Réseau . . . . .	9
2.2.1 Introduction au réseau dans le projet . . . . .	9
2.3 Architecture réseau . . . . .	9
2.3.1 Pourquoi un serveur externe ? . . . . .	9
2.3.2 Défis rencontrés et solutions . . . . .	10
2.3.3 Conclusion . . . . .	10
<b>3 Taland Alexandre</b>	<b>10</b>
3.1 Site web . . . . .	10
3.1.1 Introduction . . . . .	10
3.1.2 Objectifs et Organisation du Site . . . . .	10
3.1.3 Choix Techniques et Design . . . . .	11
3.1.4 Processus de Conception . . . . .	12
3.1.5 Pages détaillées . . . . .	12
3.1.6 Problème majeur rencontrer . . . . .	13
3.1.7 Conclusion . . . . .	14
3.2 IA . . . . .	14
3.2.1 Introduction . . . . .	14
3.2.2 Animation et Sons des Supporters . . . . .	14
3.2.3 Conception de l'IA du Speaker . . . . .	15
3.2.4 Perspectives et Prochaines Étapes . . . . .	16
<b>4 Pages Virgile</b>	<b>16</b>
4.1 Introduction . . . . .	16
4.2 L'implémentation du multijoueur dans notre jeu . . . . .	16
4.2.1 Création des Joueurs . . . . .	16
4.2.2 Synchronisation des Données . . . . .	17
4.2.3 Gestion des Interactions Multijoueurs . . . . .	17
4.2.4 Architecture Réseau . . . . .	17

4.2.5	Gestion de la Latence et des Problèmes Réseau . . . . .	17
4.3	Réalisation du Multijoueur . . . . .	17
4.3.1	Synchronisation des Animations . . . . .	18
4.3.2	Création d'un Lobby Multijoueur . . . . .	18
4.3.3	Synchronisation de l'Apparition des Armes . . . . .	18
4.3.4	Synchronisation des Événements et de l'IA . . . . .	18
4.3.5	Synchronisation des Caméras des Spectateurs . . . . .	19
4.3.6	Synchronisation de la Vie des Joueurs . . . . .	19
4.3.7	Gestion et Synchronisation du Score . . . . .	19
4.4	Réflexion sur le Marketing du Projet . . . . .	19
4.4.1	Identification du Public Cible . . . . .	19
4.4.2	Stratégies de Communication . . . . .	20
4.4.3	Modèle Économique . . . . .	20
4.4.4	Création de Hype Avant Lancement . . . . .	20
4.4.5	Prochaines Étapes . . . . .	20
<b>5</b>	<b>Onraed Antoine</b>	<b>21</b>
5.1	Introduction . . . . .	21
5.2	Interface Utilisateur (UI) . . . . .	21
5.2.1	HUD (Heads-Up Display) . . . . .	21
5.2.2	Menu Principal et Sous-Menus . . . . .	21
5.2.3	Écrans de Pause et d'Inventaire . . . . .	21
5.2.4	Fenêtres Contextuelles et Notifications . . . . .	21
5.3	Interfaces de Contrôle . . . . .	22
5.3.1	Agencement des Touches . . . . .	22
5.3.2	Personnalisation des Touches . . . . .	22
5.4	Interfaces Audio . . . . .	22
5.4.1	Bruitages et Effets Sonores . . . . .	22
5.4.2	Musique Dynamique . . . . .	22
5.5	Interfaces en Ligne . . . . .	22
5.6	Ce qui a été fait . . . . .	22
5.7	Prochaines Étapes . . . . .	23
5.8	Conclusion . . . . .	23
<b>6</b>	<b>Bilan</b>	<b>23</b>
<b>7</b>	<b>Pour la prochaine soutenance</b>	<b>23</b>
<b>8</b>	<b>Annexe</b>	<b>23</b>

# 1 Nabi Chahine

## 1.1 Gameplay : Développement et Implémentation

### 1.1.1 Introduction

Dans le cadre du développement de notre jeu, une attention particulière a été portée au gameplay afin de mettre en place plusieurs systèmes performants pour offrir une expérience immersive et réactive. Voici un détail des différents systèmes développés :

### 1.1.2 Système de Mouvement

Le système de mouvement est le pilier de l'interaction joueur-environnement. Nous avons adopté une approche basée sur un Character Controller afin d'assurer des mouvements fluides et précis. Les caractéristiques principales incluent :

- **Déplacement omnidirectionnel :**

- Les joueurs peuvent se déplacer dans toutes les directions en fonction des entrées clavier. Un système de configuration des touches clavier arrivera prochainement, si possible.

- **Vitesse adaptative :**

- Une option de sprint a été implémentée, permettant au joueur d'accélérer temporairement. Cela ajoute une stratégie supplémentaire au gameplay.

- **Gestion des collisions :**

- Le système d'interactions avec le terrain et les obstacles utilise un algorithme de détection avancé. Cela évite les blocages ou les comportements incohérents. L'avantage du **Character Controller** face à un **Rigidbody** est qu'il requiert moins de calculs, réduisant la surcharge pour l'appareil du joueur.

- **Sauts et gravitation :**

- Un système de saut énergique a été inclus, intégrant une gestion réaliste de la gravité.

### 1.1.3 Système de Tir

L'un des aspects centraux du gameplay est le système de tir. Il a été conçu pour être intuitif, réactif et personnalisable. Voici les fonctionnalités clés :

- **Tirs simples et automatiques :**

- Les joueurs peuvent choisir entre des armes à tir unique ou à rafale. Chaque arme possède ses propres propriétés, ce qui permet de les manipuler afin de concevoir toutes les armes possibles.

- **Détection de cibles :**

- Les tirs sont calculés à l'aide de Raycast, un système qui envoie un rayon invisible du joueur jusqu'à une potentielle cible. Ce système rend le jeu plus fluide et économise des ressources. Cependant, des armes à projectiles pourraient facilement être implémentées ultérieurement.

- **Effets visuels et sonores :**

- Chaque tir est accompagné d'effets immersifs, tels que des flashes lumineux, des traces de balles et des sons dédiés. Tous ces effets sont personnalisables.

- **Gestion des impacts :**

- Un système d'effets d'impact sera développé pour la prochaine soutenance. Il affichera différentes animations et interactions en fonction de la surface touchée (murs, ennemis, objets).

#### 1.1.4 Système de Mouvement de Caméra

Pour maximiser l'immersion, un système de mouvement de caméra fluide et adaptatif a été mis en place :

- **Vue subjective (FPS) :**

- La caméra suit les mouvements de la souris avec précision, simulant la perspective du joueur. Le système inclut une sensibilité ajustable pour convenir à tous les styles de joueurs.

- **Gestion des limites de rotation :**

- Un système limite les angles verticaux pour éviter les mouvements excessifs ou incohérents.

- **Effets dynamiques :**

- Ces effets devraient être implémentés pour la prochaine soutenance. La caméra inclura des effets comme des légers tremblements lors des tirs ou des sprints, renforçant l'immersion.

#### 1.1.5 Système de Munitions

La gestion des munitions est cruciale pour ajouter une couche de stratégie au gameplay. Voici les fonctionnalités incluses :

- **Vue subjective (FPS) :**

- La caméra suit les mouvements de la souris avec précision, simulant la perspective du joueur. Le système inclut une sensibilité ajustable pour convenir à tous les styles de joueurs.

- **Gestion des limites de rotation :**

- Un système limite les angles verticaux pour éviter les mouvements excessifs ou incohérents.

- **Effets dynamiques :**

- Ces effets devraient être implémentés pour la prochaine soutenance. La caméra inclura des effets tels que de légers tremblements lors des tirs ou des sprints, renforçant l'immersion.

### 1.1.6 Prochaine étapes

Nous allons désormais vous présenter ce que notre équipe souhaite avoir fait pour la prochaine soutenance :

**Système d'ouverture de Coffre-Fort** Pour diversifier les objectifs du joueur, un système d'ouverture de coffre permettra de donner un nouvel objectif.

- **Gestion de l'interaction :**

Les joueurs peuvent commencer l'ouverture du coffre adverse à l'aide d'une touche sur laquelle ils doivent rester appuyés pendant une certaine durée.

- **Gestion du coffre :**

Pour se protéger, un joueur peut réparer son coffre, augmentant ainsi le temps que l'adversaire devra passer pour l'ouvrir.

**Système de visée** Le système d'arme étant fait, ajouter une fonctionnalité telle que la visée peut fortement augmenter la diversité de nos armes.

- **Gestion de l'événement :**

Le joueur doit pouvoir changer de vue lors du clic droit de la souris.

- **Gestion de l'animation :**

Chaque visée devra être accompagnée d'une animation.

**Système de barre de vie** Ce système de barre de vie améliora énormément l'affichage du joueur

**Gestion de l'événement** Le joueur doit pouvoir changer de vue lors du clic droit de la souris.

**Système de minimap** Ce système de minimap affichera en temps réel la position du joueur sur la carte

- **Liaison caméra-joueur et minimap :**

- Le joueur pourra apercevoir son adversaire sur la carte uniquement s'il a une ligne de vue directe sur lui.

- **Rotation de la minimap :**

- La minimap pourra être soit fixe, soit tourner en fonction de la rotation du joueur, selon les préférences de celui-ci.

### 1.1.7 Conclusion

Ces différents systèmes ont été conçus avec soin pour garantir une expérience de jeu fluide et immersive. Ensemble, ils constituent une base solide pour le gameplay, permettant aux joueurs de profiter d'une action dynamique et engageante. Les prochaines étapes incluent des ajouts complémentaires primordiaux afin d'offrir un meilleur gameplay.

## 2 Rischard Mattis

### 2.1 Modélisation

#### 2.1.1 Introduction à la modélisation dans le projet

La modélisation 3D a joué un rôle central dans le développement de ce jeu vidéo, car elle définit l'apparence visuelle et l'ambiance générale. L'objectif principal était de produire des modèles immersifs, réalistes et bien intégrés dans l'univers du jeu, tout en tenant compte des contraintes techniques imposées par le moteur Unity.

Blender a été choisi comme outil principal pour la création des assets 3D. Ce logiciel open source est reconnu pour sa puissance, sa flexibilité, et sa communauté active, qui a permis un accès rapide à des ressources et des tutoriels. Après la modélisation, Unity a servi de plateforme pour intégrer ces modèles et les tester directement dans le moteur de jeu, permettant une itération rapide entre la création et l'intégration.

#### 2.1.2 Approche de modélisation

Contrairement à une approche low-poly souvent utilisée pour les jeux nécessitant une optimisation avancée des performances, nous avons opté pour une modélisation plus détaillée. L'objectif était de capturer un maximum de réalisme et de détails visuels afin de créer une immersion visuelle forte.

\*Niveau de détail élevé Les modèles ont été conçus pour inclure des détails subtils comme les plis des vêtements, les textures de surface et les éléments architecturaux complexes.

- **Textures** : Bien que nous n'ayons pas utilisé de PBR (Physically Based Rendering), les textures ont été soigneusement créées pour correspondre au style artistique du jeu. Cela inclut des matériaux texturés manuellement dans Blender ou à partir d'outils externes.
- **Gestion des polygones** : Même sans implémentation de LOD (Levels of Detail), nous avons veillé à équilibrer la complexité des modèles pour éviter une surcharge sur le moteur. Par exemple, les assets visibles de loin ont été modélisés avec moins de détails comme, par exemple, le stade qui est là pour rajouter une ambiance compétitive.

#### 2.1.3 Exemples d'assets modélisés

**Personnage principal** Le personnage principal est représenté avec un design soigné, portant un costume distinctif qui reflète son rôle dans l'histoire et dans cet univers décalé. Les détails du costume, tels que les plis, les textures spécifiques ou les accessoires, ont été modélisés pour renforcer son caractère et son charisme (cf. annexe : Figure 1).

Les animations telles que les mouvements corporels ont également influencé le design des modèles pour garantir une bonne articulation au niveau des squelettes.

**PNJ spectateur** Un personnage non-joueur (PNJ), de style "Mii", a été conçu pour représenter les spectateurs contrôlés par une intelligence artificielle (cf. annexe : Figure 2). Ce design simplifié et caricatural a été choisi pour apporter une touche de légèreté visuelle et contrastante par rapport au style plus réaliste du personnage principal.

Le PNJ interagit de manière autonome avec son environnement grâce à un système d'IA, qui régit ses comportements tels que marcher, regarder des objets spécifiques, ou réagir aux événements en jeu.

**Description du terrain de jeu principal** Le jeu se déroule dans un terrain compact situé au centre d'un grand stade. Cette configuration crée une ambiance immersive où les joueurs évoluent dans un espace restreint, tout en étant entourés d'une structure massive qui renforce le sentiment d'enfermement et de spectacle (cf. annexe : Figure ??).

- **Conteneurs** : Placés stratégiquement, ces conteneurs servent à la fois de couvert et d'obstacles pour le gameplay. Ils sont modélisés avec des détails tels que des portes, des textures de métal usé et des graffitis pour accentuer leur réalisme.
- **Petits murs** : Ces murs délimitent certaines zones du terrain et offrent des possibilités tactiques aux joueurs, comme se cacher ou se protéger des attaques adverses.

**Grand stade comme décor principal** Le stade qui entoure le terrain de jeu est conçu pour donner une échelle impressionnante à l'environnement de jeu. Bien qu'il soit en arrière-plan, il joue un rôle esthétique majeur en suggérant un lieu de compétition intense ou de spectacle. Les gradins sont représentés avec un souci de simplification pour éviter de surcharger les ressources. Cependant, des détails visuels comme des écrans géants renforcent l'immersion.

**Objets interactifs** Les objets avec lesquels les joueurs peuvent interagir, tels que des armes, des grenades, et des bananes représentant du bouclier (appelé aussi "chill"), ont été modélisés pour inclure des détails fonctionnels (cf. annexe : Figures ??, ?? et ??).

#### 2.1.4 Workflow et pipeline

Le processus de création des modèles a suivi un pipeline bien défini :

1. **Création des maillages** : Les modèles de base ont été créés dans Blender, en utilisant des techniques comme le sculpting pour les détails organiques comme le corps du personnage principal et le modeling classique pour les formes géométriques.
2. **Exportation vers Unity** : Les modèles ont été exportés au format FBX, un standard bien supporté par Unity, en veillant à maintenir l'échelle et les propriétés des matériaux.
3. **Intégration et tests** : Une fois dans Unity, les modèles ont été intégrés dans les scènes, puis testés pour détecter tout problème potentiel.

### 2.1.5 Planning et respect des objectifs

#### Objectifs initiaux

- **Modélisation du personnage principal** : Terminé. Ce personnage est central au gameplay et a été modélisé avec un souci du détail pour refléter son importance dans l'univers du jeu.
- **Création des armes** : Terminé. Elles ont été conçues pour être fonctionnelles et visuellement distinctes, renforçant leur rôle dans les mécaniques du jeu.
- **Conception de l'environnement** : Terminé. Le terrain de jeu, les conteneurs, les murs, et le stade ont été modélisés pour créer une ambiance compétitive et immersive.

**Respect des délais** À ce jour, nous sommes dans les temps par rapport au planning initial. Toutes les étapes prévues ont été accomplies.

## 2.2 Réseau

### 2.2.1 Introduction au réseau dans le projet

Le réseau joue un rôle essentiel dans le projet, car il permet de connecter plusieurs joueurs dans un environnement multijoueur compétitif. Contrairement à une approche peer-to-peer (P2P), nous avons opté pour une architecture basée sur un serveur externe. Cette décision a été motivée par des besoins en stabilité, en synchronisation précise des joueurs et en contrôle centralisé des interactions.

## 2.3 Architecture réseau

L'architecture réseau repose sur un serveur centralisé qui gère la communication entre les clients (les joueurs). Voici les principales caractéristiques de cette architecture :

- **Serveur externe** : Le serveur agit comme une autorité centrale qui synchronise les actions de chaque joueur. Par exemple, il vérifie les collisions, gère les états des objets interactifs et synchronise les positions des joueurs en temps réel.
- **Protocole de communication** : Une communication bidirectionnelle est établie entre le serveur et les clients à l'aide de [KCP]. Cela permet une faible latence et une transmission fiable des données critiques, comme les entrées des joueurs ou les événements de jeu.

### 2.3.1 Pourquoi un serveur externe ?

Nous avons choisi un serveur externe plutôt qu'une approche peer-to-peer (P2P) pour plusieurs raisons :

- **Synchronisation centralisée** : Le serveur garantit que tous les joueurs partagent le même état du jeu, éliminant les incohérences dues à des désynchronisations.
- **Sécurité** : Une architecture centralisée est plus résistante aux manipulations ou à la triche, car le serveur peut valider toutes les actions des joueurs avant de les appliquer.

- **Évolutivité** : Un serveur centralisé permet d'ajouter plus facilement de nouveaux joueurs ou fonctionnalités à l'avenir, sans dépendre directement des capacités des clients.

### 2.3.2 Défis rencontrés et solutions

#### Latence et performance

- **Défi** : La latence est un problème clé dans tout jeu multijoueur. Une mauvaise connexion peut entraîner des retards ou des décalages entre les actions des joueurs et ce qui est visible à l'écran.
- **Solution** : Nous avons implémenté un système de compensation de latence pour anticiper les mouvements des joueurs en cas de décalages mineurs. De plus, l'utilisation d'un protocole UDP pour les données sensibles a permis de minimiser les délais de transmission.

#### Gestion des états des objets

- **Défi** : La synchronisation des objets interactifs, comme les armes, les bananes ou le flag, nécessitait une gestion précise pour éviter les conflits d'état.
- **Solution** : Le serveur maintient une liste d'états des objets en temps réel, et chaque interaction est validée avant d'être diffusée aux autres joueurs.

### 2.3.3 Conclusion

L'utilisation d'un serveur externe a permis de garantir une expérience multijoueur fluide et équitable. Bien que cette approche nécessite une infrastructure plus complexe que le peer-to-peer, elle offre des avantages significatifs en termes de stabilité, de sécurité et de synchronisation, ce qui est essentiel pour un jeu compétitif comme le nôtre.

## 3 Taland Alexandre

### 3.1 Site web

#### 3.1.1 Introduction

Le développement du site web de *Monkey Heist* a été une étape essentielle pour présenter le jeu, le rendre accessible au public et fournir des informations sur l'équipe ainsi que sur son développement. Le site a intégré un design réactif (*responsive design*) pour s'adapter à divers dispositifs, allant des ordinateurs de bureau aux smartphones. J'ai pris en charge cette tâche de manière intégrale, depuis la conception jusqu'à la réalisation.

#### 3.1.2 Objectifs et Organisation du Site

J'ai structuré le site en cinq pages principales :

- **Accueil** :
  - Introduction au jeu.

- Présentation des principales caractéristiques et du contenu du jeu.
- **Téléchargement :**
  - Accès aux fichiers nécessaires pour installer le jeu.
- **Histoire du jeu :**
  - Présentation du storytelling du jeu.
- **About :**
  - Présentation de l'équipe de développement.
  - Chronologie de la réalisation.
  - Problèmes rencontrés au cours du projet.
- **Ressources :**
  - Accès aux différents rapports de soutenance et au cahier des charges.
  - Liens vers les logiciels utilisés.
  - Références aux sons, images, bibliothèques et applets utilisés dans le projet.

### 3.1.3 Choix Techniques et Design

**Langages Utilisés** J'ai choisi une combinaison de HTML, CSS, et JavaScript pour la conception du site :

- **HTML :**
  - Utilisé pour définir la structure et le contenu du site web.
- **CSS :**
  - Employé pour le style et les effets visuels, comme les dégradés, les survols interactifs et les transitions.
- **JavaScript :**
  - Utilisé pour les interactions dynamiques et les comportements interactifs, tels que les redirections et l'affichage des chronologies.

**Palettre de Couleurs** La palette de couleurs est dominée par le noir, le gris, et le blanc. Ces teintes créent une esthétique sobre et moderne, en accord avec le thème du jeu qui s'inscrit dans un univers de braquage et d'adrénaline.

**Effets Visuels** J'ai intégré plusieurs effets pour enrichir l'expérience utilisateur :

- **Dégradés de couleur :**

- Utilisés pour créer des transitions fluides et esthétiques, apportant une touche moderne au design.

- **Effets de survol :**

- Les boutons et images changent d'apparence lorsqu'on passe la souris dessus, attirant l'attention sur les éléments interactifs.
  - Sur certaines images, un texte explicatif apparaît lors du survol, fournissant des informations supplémentaires.

- **Redirections fluides :**

- Les liens entre les pages sont optimisés pour garantir une navigation intuitive et agréable.

- **Système de chronologie :**

- En cliquant sur un cercle représentant une étape, une zone de texte s'affiche pour décrire cette étape, rendant la navigation interactive et informative.

**Adaption** Début de réflexion sur l'adaptation du site web pour les écrans de téléphone. Les pages s'affichent correctement sur mobile, mais la barre de navigation n'est pas encore optimisée.

### 3.1.4 Processus de Conception

**Structure HTML** Chaque page a été organisée avec une structure logique : Une barre de navigation présente sur toutes les pages, permettant de passer facilement de l'une à l'autre. Une mise en page adaptée pour un affichage clair sur ordinateur et, à terme, sur mobile. Un contenu divisé en sections bien définies pour faciliter la lecture.

**Mise en Style avec CSS** La feuille de style CSS contient : Les définitions des couleurs, des polices, et des marges. Les animations et transitions pour les boutons, images et textes. Une conception responsive en cours pour adapter le site aux différents écrans.

**Intégration JavaScript** J'ai utilisé JavaScript pour : Les interactions dynamiques, comme l'affichage des zones de texte dans la chronologie.

### 3.1.5 Pages détaillées

**Page d'Accueil** La page d'accueil sert de vitrine au jeu. Elle comprend : Un titre principal et un bouton d'appel à l'action (Call to Action) pour télécharger le jeu ou en savoir plus.(à noter qu'il y a eu un bug de dernière minute, cela ne s'affiche plus) Une image ou vidéo du jeu mise en avant.( en attente du trailer) Des zones décrivant les caractéristiques du jeu vidéo.

**Page de Téléchargement** Cette page fournit les fichiers d’installation et les informations de téléchargement. Le bouton de téléchargement sont stylisés avec des effets de changement de couleur pour indiquer leur interactivité.

**Page d’Histoire** Cette page présente : Le storytelling de l’univers du jeu avec des photos.

**Page de About** Cette page présente : L’équipe, avec des photos et des descriptions. Les principaux problèmes rencontrés et les solutions apportées. La chronologie de réalisation du projet sur l’année, divisée en 4 étapes. La zone de chronologie utilise des cercles interactifs pour présenter les différentes étapes du projet. Chaque clic affiche une zone de texte contenant des détails sur une étape précise. Les zones de texte restent affichées sur la page tant qu’on ne clique pas à nouveau sur le cercle correspondant pour les enlever. Il est possible d’afficher les 4 étapes en même temps ou uniquement celles souhaitées.

**Page des Ressources** La page ressources inclut : Des liens vers des documents officiels, tels que le cahier des charges et le rapport de soutenance. Des liens vers les logiciels utilisés. Des fichiers contenant les images, sons, bibliothèques et applets utilisés pour le site (on en a pas encore utilisé) et le jeu.

### 3.1.6 Problème majeur rencontrer

Le développement du site web pour Monkey Heisty n’a pas été sans défis. Voici trois problèmes majeurs rencontrés et les solutions apportées :

**Conception de l’effet d’apparition du texte** Lors de la mise en place de l’effet d’apparition du texte (dans la page ”ABOUT” en survolant les images de l’équipe), le comportement initial causait un bug majeur : toutes les images de la page disparaissaient ou devenaient non interactives. Solution : Après avoir identifié que le problème venait de conflits dans les animations CSS, j’ai isolé les effets d’apparition dans un conteneur dédié. J’ai également optimisé les événements déclencheurs pour éviter toute interférence avec les autres éléments interactifs.

**Difficulté pour l’effet de chronologie** La chronologie, avec ses cercles interactifs affichant des textes au clic, a posé des défis techniques. Au début, les zones de texte ne s’affichaient pas correctement, prenaient toute la page et on ne pouvait pas enlever le texte. Solution : J’ai restructuré le système en utilisant des classes CSS spécifiques combinées avec une fonction JavaScript. Cela a permis une gestion fluide des interactions et un rendu visuel propre. (À noter qu’il faut encore rétrécir la zone, car il reste une partie vide.)

**Effacement des boutons de téléchargement et du titre** Le dernier problème en date : les boutons de téléchargement et le titre principal de la page Accueil disparaissaient de manière aléatoire, surtout lors des tests de navigation rapide entre les pages. Ils ne s’affichent plus du tout. Il semble y avoir un bug avec l’image de la page d’accueil qui s’afficherait en grand. Solution : En cours de résolution. Il est nécessaire de revoir les fonctions associées et, potentiellement, de refaire celle-ci.

### 3.1.7 Conclusion

Le site est actuellement dans une version brute. Les prochaines étapes incluent :

- Finaliser la conception responsive pour les appareils mobiles.
- Ajouter des améliorations graphiques et d'accessibilité.
- Tester le site sur différents navigateurs et résoudre d'éventuels bugs.
- Ajouter des mots-clés pour retrouver le site sur le web.
- Regardez le côté sécurité du site.

## 3.2 IA

### 3.2.1 Introduction

Le développement de l'IA pour "Monkey Heist" a été conçu pour améliorer l'immersion des joueurs à travers des comportements dynamiques et adaptatifs des personnages non-joueurs (PNJ). Les deux principaux aspects couverts sont les supporters, qui ajoutent de l'ambiance et de l'émotion au jeu, et le speaker, qui fournit des commentaires contextuels en temps réel. L'objectif global est d'offrir une expérience riche et réactive à chaque moment clé de la partie. Après avoir mené des recherches approfondies, effectué des tests et exploré différentes solutions, j'ai désormais un plan clair pour l'implémentation de ces systèmes d'IA dans le jeu final. Voici un récapitulatif de ce travail préparatoire.

### 3.2.2 Animation et Sons des Supporters

**Objectif et Conception** L'idée principale est de créer des supporters vivants et réactifs, qui applaudissent, sautent, ou réagissent en fonction des événements dans le jeu. Cela inclut :

- **Animations variées :**
  - Applaudir, sauter, lever les bras, effectuer une ola .
- **Sons immersifs :**
  - Cris, applaudissements, sifflets.

### Configuration des Animations

#### • Préparation des animations :

- Après avoir testé plusieurs animations, j'ai sélectionné des animations de base, comme évoqué ci-dessus, dans Mixamo.
- Un Animator Controller sera créé pour gérer les transitions entre ces animations.
- Un prefab supporter sera configuré avec un composant Animator lié à ce controller.

#### • Système d'événements aléatoires :

- Un script sera écrit pour déclencher aléatoirement des animations et des sons des supporters.

## Intégration des Sons

- **Bibliothèque audio :**

- J'ai collecté des fichiers audio ( cris , applaudissements , sifflets ) et les ai organisés dans un dossier.
- Une AudioSource sera ajoutée au prefab pour jouer ces sons.

- **Spatialisation audio :**

- Pour une immersion accrue, je vais régler le Spatial Blend en 3D, simulant la propagation sonore dans le stade.

### 3.2.3 Conception de l'IA du Speaker

**Idée Initiale** Le speaker devra servir de narrateur en réagissant aux actions dans le jeu, comme un commentateur sportif. Les dialogues seraient joués en fonction des événements tels que :

- **Le début de la partie**
- **Le vol de l'objet**
- **Les courses-poursuites ou combats**
- **La victoire ou la défaite**

**Préparation des Dialogues** J'ai créé une liste de dialogues divisés par thèmes :

- **Introduction :** Mesdames et messieurs, bienvenue au stade !
- **Vol :** Incroyable ! L'objet vient d'être subtilisé !
- **Course-poursuite :** Le voleur est rapide, mais les chasseurs ne sont pas loin !
- **Combat :** C'est un affrontement direct ! Les coups pleuvent !
- **Victoire :** Et c'est terminé ! Quelle performance incroyable !
- **Défaite :** Les espoirs s'effondrent... Le voleur a été arrêté !

Ces dialogues vont être enregistrer sous forme d'AudioClips et regroupés dans des tableaux pour chaque thème.

## Scripting du Speaker

- **Configuration AudioSource :**

- Il faudra ajouter une AudioSource au prefab Speaker pour jouer les dialogues.

- **Script de gestion des dialogues :**

- Un script sera écrit pour sélectionner et jouer un dialogue en fonction d'un thème donné (vol, combat, etc.).

### 3.2.4 Perspectives et Prochaines Étapes

Pour la prochaine fois, il faudra :

- Eventuellement ajouter des animations supplémentaires pour enrichir l'expérience.
- Eventuellement ajouter davantage de sons pour éviter les répétitions et diversifier l'ambiance.
- Eventuellement enrichir les dialogues afin de couvrir une plus grande variété d'événements. et ajouter une fonction aléatoire.
- Appliquer le plan pour intégrer l'IA dans le jeu.
- Effectuer les effets par rapport au activation des animations, sons et dialogue qui colle au moment du gameplay.

## 4 Pages Virgile

### 4.1 Introduction

La partie multijoueur implique plusieurs défis techniques, notamment la gestion des connexions réseau, la synchronisation des actions des joueurs (comme les déplacements, les tirs, et la gestion des points de vie), ainsi que la fluidité des interactions malgré les contraintes liées à la latence. Cette partie détaille les étapes nécessaires pour concevoir et implémenter cette fonctionnalité, en s'appuyant sur un framework réseau adapté et des bonnes pratiques d'optimisation. Cette implémentation ne se limite pas à une simple extension du jeu, mais constitue un pilier essentiel à la viabilité du jeu. Notre processus d'implémentation permettra de garantir une expérience de jeu immersive et fluide.

### 4.2 L'implémentation du multijoueur dans notre jeu

#### Étapes d'Implémentation

Pour gérer les communications entre les joueurs, nous avons choisi d'utiliser **Mirror** qui nous permettra de créer un hôte, pour gérer tous les joueurs et synchroniser les données essentielles comme les déplacements, les tirs, les dégâts, etc...

##### 4.2.1 Création des Joueurs

Chaque joueur est représenté par un **objet réseau** (un prefab) qui est instancié lorsque le joueur rejoint une partie. Ce prefab inclut :

- **Apparence et Modèle 3D** : Pour représenter visuellement le joueur.
- **Composants Réseau** : Des éléments qui synchronisent la position, la rotation, et d'autres données entre les clients.
- **Composants Locaux** : Une caméra pour le joueur et des contrôles pour le déplacement et le tir.

Lorsque le joueur rejoint une partie, son objet est créé sur le réseau pour être visible par tous les autres participants.

#### 4.2.2 Synchronisation des Données

Pour garantir que tous les joueurs voient la même scène en temps réel :

- Les **positions et rotations** des joueurs seront mises à jour en continu. Cela sera fait avec des composants de synchronisation fournis par **Mirror**.
- Les **actions spécifiques** (comme tirer) sont communiquées à tous les joueurs à l'aide de méthodes réseau comme des appels distants.
- Les **variables critiques** (comme la santé ou le score) sont gérées de manière centralisée, sur un hôte de partie, pour éviter les désynchronisations.

#### 4.2.3 Gestion des Interactions Multijoueurs

Dans notre jeu, plusieurs actions nécessitent une gestion spécifique :

- **Déplacements** : Les mouvements d'un joueur doivent être transmis aux autres via le réseau. Pour lisser les mouvements et compenser la latence, nous utiliserons l'interpolation.
- **Tirs** : Lorsqu'un joueur tire, cette action est communiquée aux autres pour afficher les impacts ou les projectiles.
- **Santé** : Si un joueur subit des dégâts, l'information est centralisée et propagée aux autres pour mettre à jour les indicateurs de santé.

#### 4.2.4 Architecture Réseau

Actuellement, La mise en œuvre du multijoueur dans notre jeu repose sur une architecture en **Peer-to-peer**(Un des joueurs agit comme hôte, et les autres joueurs échangent directement des informations avec lui.).

#### 4.2.5 Gestion de la Latence et des Problèmes Réseau

Dans un environnement multijoueur, la latence peut affecter l'expérience :

- Utilisation de **techniques d'interpolation/extrapolation** sera importante pour rendre les mouvements fluides même avec des retards réseau.
- Limitation de la fréquence des mises à jour réseau pour réduire la charge (par exemple, envoyer des données de position toutes les 50 ms au lieu de 10 ms).

### 4.3 Réalisation du Multijoueur

Pour la mise en œuvre du multijoueur, un framework open-source comme **Mirror** simplifie la synchronisation des objets et des événements dans Unity. Ce choix permet une mise en œuvre rapide, adaptée aux besoins de notre projet, tout en minimisant les coûts d'infrastructure.

Cette section décrit les différentes étapes et fonctionnalités spécifiques nécessaires à la réalisation du multijoueur.

#### 4.3.1 Synchronisation des Animations

Les animations des joueurs doivent être visibles en temps réel pour tous les participants.

- Les transitions entre les états d'animation, comme courir, sauter ou tirer, sont déclenchées via des variables synchronisées, appelées `SyncVars` dans Mirror.
- Les modifications d'état sont transmises par des commandes (`Commands`) du client vers l'hôte, et des RPC (`ClientRpc`) pour les autres clients.
- Une attention particulière est accordée à l'interpolation des animations pour garantir leur fluidité malgré la latence réseau.

#### 4.3.2 Création d'un Lobby Multijoueur

Un lobby est mis en place pour permettre aux joueurs de se connecter à une session avant de lancer la partie.

- L'hôte peut créer une salle où les autres joueurs se connectent via leur adresse IP ou un système de matchmaking local.
- Les informations des joueurs (nom, avatar, équipe) sont synchronisées pour apparaître sur l'interface utilisateur du lobby.
- Une fois que tous les joueurs sont prêts, l'hôte peut déclencher le démarrage de la partie, ce qui instancie les objets nécessaires sur le réseau.

#### 4.3.3 Synchronisation de l'Apparition des Armes

Les armes doivent être visibles et utilisables pour tous les joueurs.

- Lorsqu'un joueur sélectionne ou change une arme, cette action est synchronisée via des `Commands` pour garantir que l'arme soit affichée sur tous les clients.
- Les projectiles tirés sont des objets réseau (`NetworkObjects`) instanciés par l'hôte et synchronisés avec leur direction et position pour apparaître correctement.
- Des événements de son et de particules sont également synchronisés pour enrichir l'expérience visuelle et auditive.

#### 4.3.4 Synchronisation des Événements et de l'IA

Le jeu peut inclure des événements dynamiques ou des entités contrôlées par une intelligence artificielle (IA).

- L'hôte est responsable de gérer les événements globaux (comme un changement de phase de jeu ou une alerte) et de les synchroniser avec tous les joueurs.
- Les comportements de l'IA sont exécutés sur l'hôte, qui envoie ensuite les mises à jour (position, état) aux clients pour garantir leur cohérence.
- Cette approche réduit la charge réseau sur les clients et garantit une logique centralisée pour l'IA.

#### 4.3.5 Synchronisation des Caméras des Spectateurs

Pour les joueurs éliminés ou inactifs, une caméra spectateur est mise en place.

- Les caméras des spectateurs peuvent suivre un joueur actif.
- La position et l'état de ces caméras sont synchronisés pour éviter les conflits si plusieurs joueurs les utilisent.
- Des transitions fluides entre les caméras sont gérées localement pour une meilleure expérience utilisateur.

#### 4.3.6 Synchronisation de la Vie des Joueurs

La gestion de la vie est critique pour la fluidité et l'immersion dans notre jeu.

- Chaque joueur possède une variable de vie synchronisée via `SyncVars`.
- Lorsqu'un joueur subit des dégâts, l'information est envoyée à l'hôte, qui met à jour la variable de vie et notifie les autres clients.
- Des événements visuels (comme les barres de santé les éclaboussures de sang) sont synchronisés pour refléter les changements.

#### 4.3.7 Gestion et Synchronisation du Score

Le score global et individuel est une métrique essentielle à synchroniser.

- Le score est maintenu sur l'hôte en tant que `SyncVar` ou `NetworkList`, et est mis à jour lorsqu'un joueur effectue une action (comme éliminer un adversaire ou capturer un objectif).
- Les scores individuels sont affichés sur l'interface utilisateur de chaque joueur, et les modifications sont propagées en temps réel.
- Les événements de fin de partie, (comme l'annonce de l'équipe vainqueur par exemple), sont déclenchés par l'hôte et transmis aux clients.

### 4.4 Réflexion sur le Marketing du Projet

Bien que le marketing ne fasse pas encore partie intégrante de nos travaux actuels, une réflexion préliminaire sur cette étape clé du projet a été menée afin d'assurer une transition fluide vers la commercialisation. L'objectif principal du marketing sera de maximiser la visibilité de notre jeu auprès de notre public cible et de créer une communauté engagée autour de celui-ci.

#### 4.4.1 Identification du Public Cible

Nous avons identifié un public cible principalement constitué de :

- **Amateurs de FPS compétitifs** : Des joueurs appréciant les mécaniques dynamiques et les parties multijoueurs intenses.

- **Joueurs occasionnels** : Un segment attiré par une prise en main simple mais immersive, favorisée par notre gameplay accessible.
- **Jeunes adultes** : Principalement âgés de 16 à 30 ans, adeptes de jeux en ligne et présents sur les réseaux sociaux et plateformes de streaming.

#### 4.4.2 Stratégies de Communication

Pour atteindre ce public, plusieurs canaux de communication ont été envisagés :

- **Réseaux sociaux** : Créer des comptes officiels sur des plateformes comme Twitter, Instagram, et TikTok pour partager des teasers, des visuels, et des annonces de mises à jour.
- **Streaming et plateformes vidéo** : Collaborer avec des créateurs de contenu sur Twitch et YouTube pour promouvoir le jeu via des démonstrations en direct et des vidéos sponsorisées.
- **Communautés en ligne** : Participer à des forums et des groupes Discord dédiés aux jeux vidéo pour interagir directement avec les joueurs.

#### 4.4.3 Modèle Économique

Nous avons réfléchi à plusieurs approches potentielles pour monétiser le jeu :

- **Modèle freemium** : Offrir le jeu gratuitement avec des options d'achats intégrés, comme des skins ou des personnalisations cosmétiques.
- **Événements et DLC** : Générer des revenus supplémentaires via des extensions ou des événements temporaires.

#### 4.4.4 Crédit de Hype Avant Lancement

Pour susciter l'intérêt avant le lancement, nous avons envisagé des actions telles que :

- **Démo jouable** : Proposer une version limitée du jeu pour engager les joueurs et recueillir des retours afin d'améliorer ou d'optimiser certaines fonctionnalités.
- **Campagnes de pré-lancement** : Utiliser un compte à rebours et des annonces progressives pour maintenir l'enthousiasme.

#### 4.4.5 Prochaines Étapes

Notre réflexion autour du marketing reste au stade conceptuel, et l'objectif sera de travailler ultérieurement sur un plan marketing détaillé. Cette étape inclura des études de marché plus approfondies, le développement de visuels publicitaires, et la définition d'un budget dédié à la promotion.

## 5 Onraed Antoine

### 5.1 Introduction

Les interfaces ont joué un rôle essentiel dans le développement de ce jeu vidéo, car elles définissent la manière dont le joueur interagit avec l'univers du jeu et perçoit les informations essentielles. L'objectif principal était de concevoir des interfaces intuitives, immersives et adaptées à l'expérience de jeu, tout en respectant les contraintes ergonomiques et techniques imposées par le moteur Unity.

L'interface utilisateur (UI) a été pensée pour offrir une navigation fluide et une lisibilité optimale, en intégrant des éléments comme les menus, les indicateurs de statut et le HUD. L'aspect audio a également été travaillé pour renforcer l'immersion, en combinant des effets sonores dynamiques. Enfin, les interfaces de contrôle ont été optimisées pour assurer une prise en main naturelle au clavier.

Unity a servi de base pour la mise en place et le test des différentes interfaces, permettant une intégration rapide et une itération efficace entre conception visuelle, feedback utilisateur et ajustements techniques.

### 5.2 Interface Utilisateur (UI)

L'interface utilisateur joue un rôle fondamental dans l'expérience de jeu en facilitant l'interaction entre le joueur et l'environnement virtuel. Elle regroupe l'ensemble des éléments visuels conçus pour transmettre des informations essentielles tout en assurant une navigation fluide et intuitive.

#### 5.2.1 HUD (Heads-Up Display)

Le HUD constitue l'un des éléments les plus visibles de cette interface. Il permet d'afficher en permanence ou de manière ponctuelle des indicateurs cruciaux tels que la barre de vie, la jauge de ressources, la minimap ou encore divers compteurs liés au gameplay. Son intégration est pensée pour fournir ces informations sans nuire à la lisibilité de l'action à l'écran.

#### 5.2.2 Menu Principal et Sous-Menus

Ces éléments offrent une organisation structurée des options du jeu. Ils permettent d'accéder aux paramètres ou encore de naviguer entre différentes fonctionnalités. L'ergonomie de ces menus est un aspect essentiel pour garantir une prise en main rapide et efficace.

#### 5.2.3 Écrans de Pause et d'Inventaire

Ces écrans constituent un espace où le joueur peut gérer son équipement ou bien consulter les statistiques de ses armes. Ces interfaces sont conçues pour offrir une vue claire et organisée des ressources disponibles, favorisant ainsi une gestion optimisée de la progression.

#### 5.2.4 Fenêtres Contextuelles et Notifications

Elles apportent des informations ponctuelles en fonction de la situation de jeu. Elles incluent les messages d'alerte et les stades de la partie, assurant ainsi une transmission efficace des données essentielles sans interrompre le rythme de l'expérience.

## 5.3 Interfaces de Contrôle

L'interface de contrôle joue un rôle essentiel dans l'expérience de jeu, en assurant une interaction fluide et réactive entre le joueur et l'environnement virtuel. Lorsqu'elle repose sur l'utilisation du clavier, elle doit être conçue pour offrir un confort optimal, une accessibilité intuitive et une réactivité adaptée aux exigences du gameplay.

### 5.3.1 Agencement des Touches

Il est pensé pour équilibrer ergonomie et efficacité. Les commandes de déplacement sont généralement assignées aux touches les plus accessibles, permettant une prise en main rapide et naturelle.

### 5.3.2 Personnalisation des Touches

Notre jeu offrira la possibilité de reconfigurer les commandes afin de s'adapter aux préférences individuelles et aux besoins spécifiques de chaque joueur.

## 5.4 Interfaces Audio

L'interface audio occupe une place centrale dans l'expérience de jeu, en enrichissant l'immersion et en renforçant l'impact des actions du joueur.

### 5.4.1 Bruitages et Effets Sonores

Ils joueront un rôle fondamental dans la perception de l'environnement virtuel. Leur spatialisation permettra également de renforcer la perception de l'espace.

### 5.4.2 Musique Dynamique

Elle accompagnera l'évolution du jeu en s'adaptant aux différentes situations rencontrées.

## 5.5 Interfaces en Ligne

L'interface en ligne constitue un élément clé de l'expérience multijoueur, en assurant une connexion fluide entre les joueurs.

## 5.6 Ce qui a été fait

Les interfaces ont été entamées le 06/01 et intégrées dans les différents systèmes comme suit :

- **Interface utilisateur** : La barre de vie a été mise en place.
- **Interface de contrôle** : Des touches prédéfinies ont été configurées.
- **Interface audio** : Des effets sonores ont été intégrés.
- **Interface en ligne** : Le mode multijoueur est opérationnel.

## 5.7 Prochaines Étapes

Pour la prochaine soutenance, notre équipe vise à accomplir les objectifs suivants :

- Terminer le HUD.
- Créer les menus et fenêtres contextuelles.
- Optimiser les touches de contrôle.
- Améliorer l'ambiance sonore.

## 5.8 Conclusion

En conclusion, nous avons déjà implanté les bases essentielles de l'interface, avec un HUD fonctionnel, des contrôles ergonomiques et des éléments sonores de base. À venir, nous finaliserons les menus, implémenterons des fenêtres contextuelles et offrirons plus de personnalisation des contrôles.

## 6 Bilan

En résumé, "Monkey Heist" a réalisé des progrès significatifs. Plusieurs aspects essentiels ont été achevés et intégrés, tels que la modélisation des personnages, des armes et des cartes, ainsi que l'implémentation du gameplay de base (déplacements des personnages, système d'armement, et cartes). Le début de la barre de vie et les premiers effets sonores ont aussi été ajoutés. Le site web est désormais dans sa première version et les fonctionnalités multijoueurs et en ligne sont pleinement opérationnelles. Par ailleurs, le système d'IA et d'interface est en phase de développement. Et la phase de réflexion sur le côté marketing a commencé.

## 7 Pour la prochaine soutenance

Pour la prochaine soutenance, plusieurs tâches essentielles demeurent à accomplir pour finaliser le projet :

Tout d'abord, en matière de modélisation, il reste à peaufiner les derniers détails et ajustements des personnages ainsi que des objets. Du côté du gameplay, l'implémentation de l'IA, des interfaces utilisateurs, ainsi que des effets sonores et visuels doivent être finalisées. L'interface et la musique nécessitent également d'être complétées pour garantir une expérience cohérente.

En ce qui concerne le site internet, la version 2 doit être développée, avec des optimisations et des améliorations pour offrir une navigation plus fluide. Pour les aspects réseaux et multijoueur, il faudra continuer le développement en intégrant les ajouts du jeu. L'IA également doit être achevée, notamment pour soutenir les interactions des spectateurs et du speaker.

Enfin, une continuité sur la réflexion sur des stratégies marketing est à envisager pour promouvoir efficacement le jeu.

## 8 Annexe

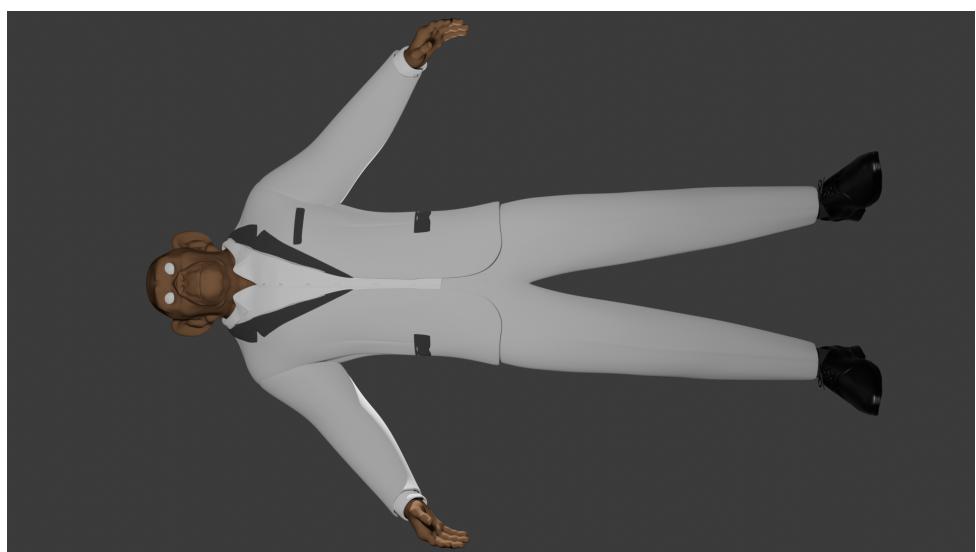


Figure 1: Personnage principal

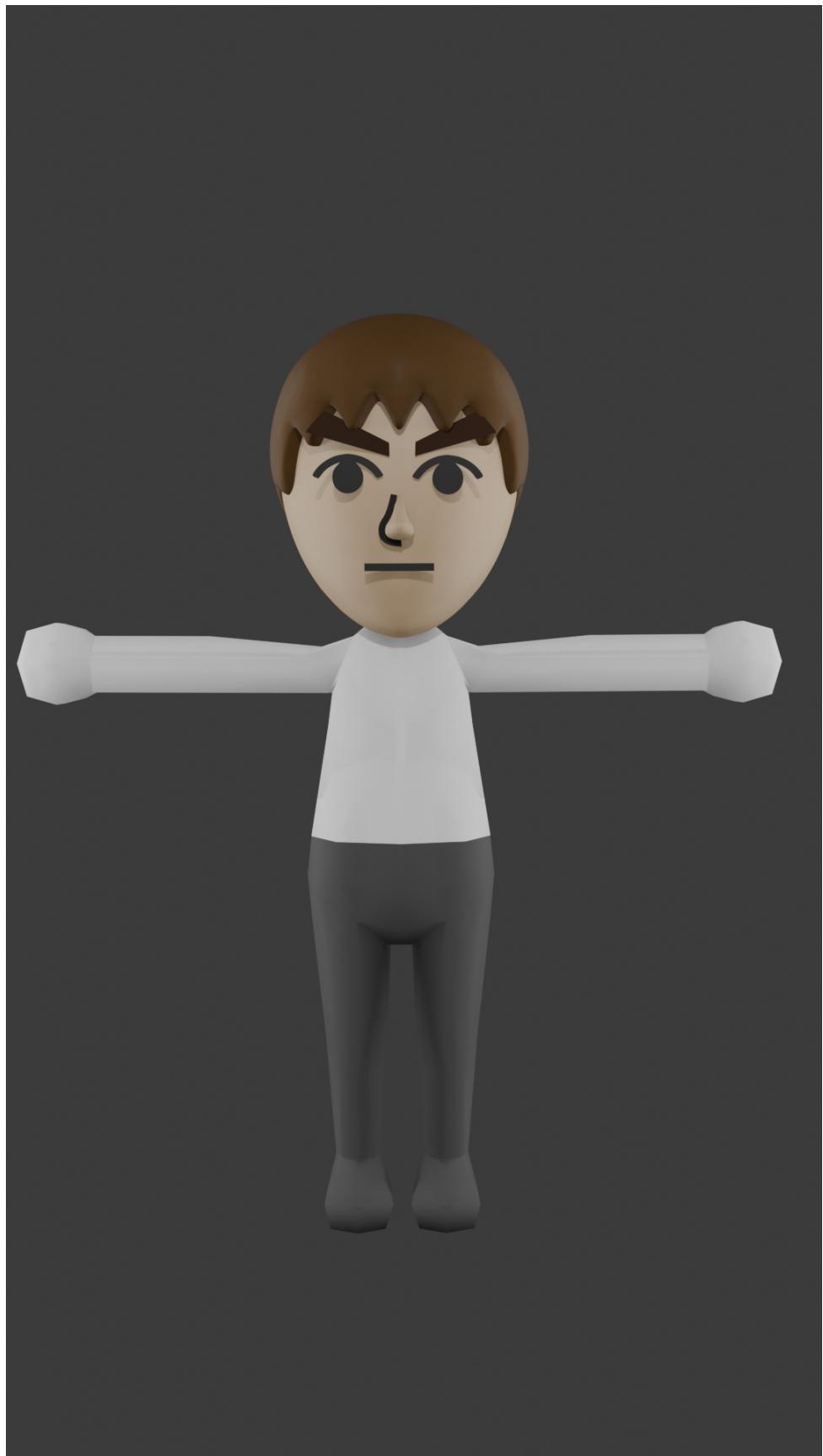


Figure 2: PNJ Mii

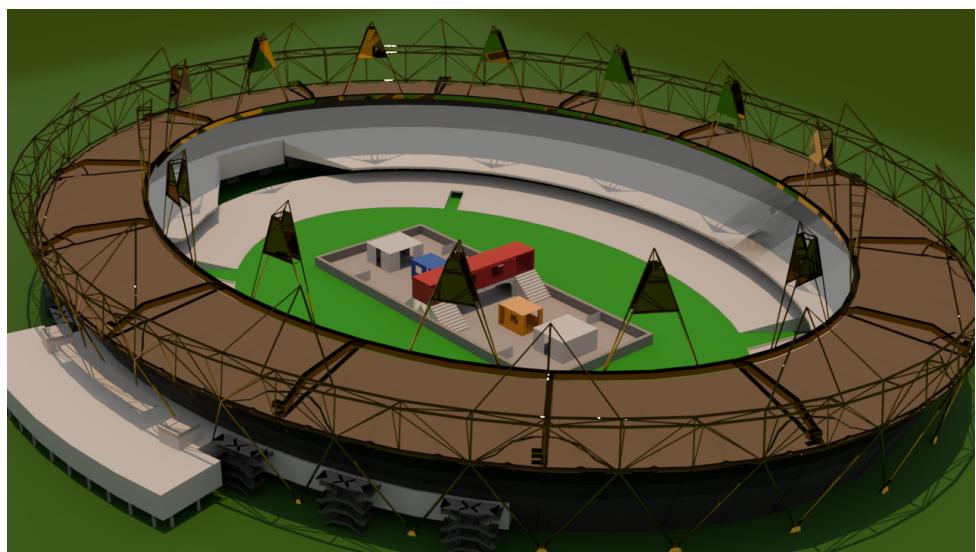


Figure 3: Terrain de jeu

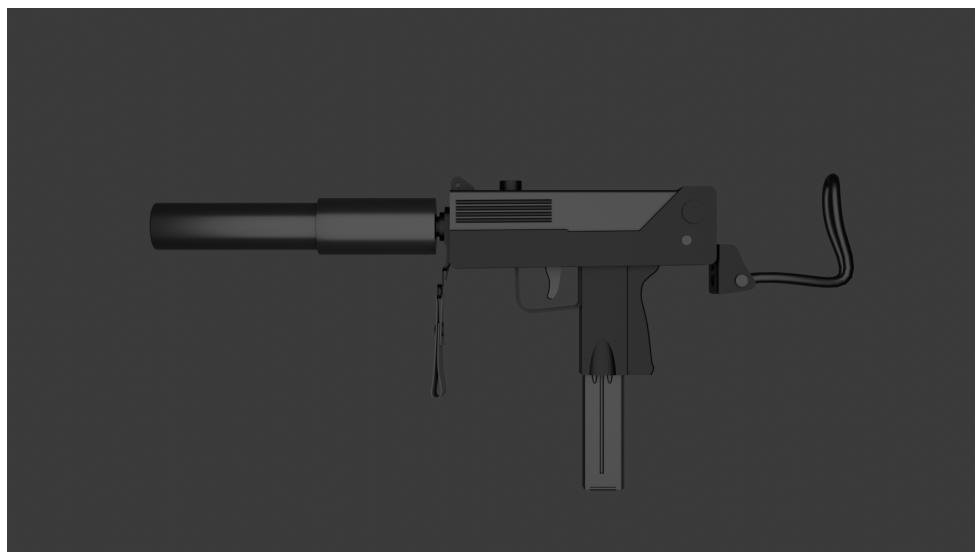


Figure 4: Arme 1



Figure 5: Arme 2

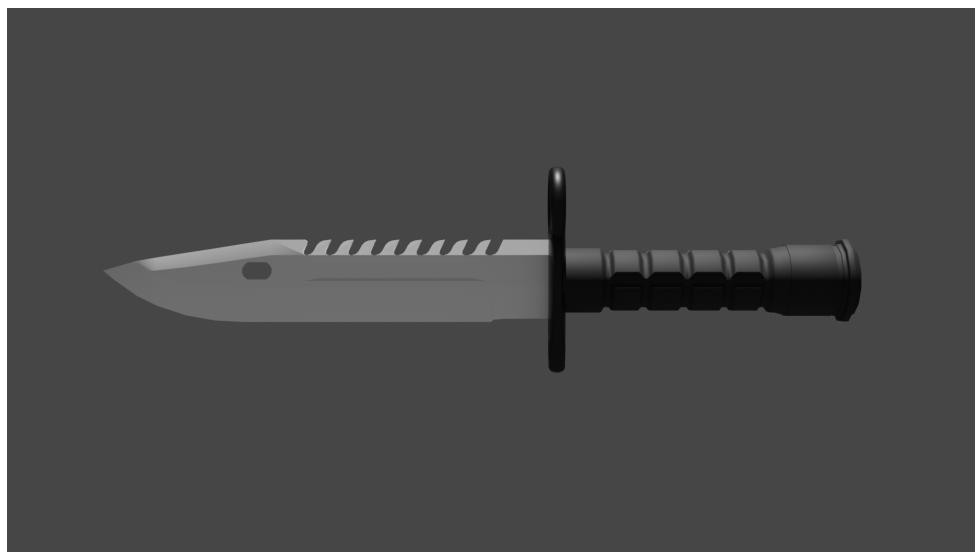


Figure 6: Arme 3



Figure 7: Arme 4



Figure 8: Grenade 1



Figure 9: Grenade 2



Figure 10: Banane