

Panic At Tortuga

Harrys KEDJNANE

Renaud-Dov DEVERS

Julien COHEN-SCALI

Paul DE LA PORTE DES VAUX

Janvier 2021



Introduction

Nous allons vous présenter notre projet de S2. Notre équipe a choisi de réaliser un jeu vidéo pour le projet de l'Info SUP.

Table des matières

1	Introduction	2
1.1	Présentation du jeu	2
1.2	L'équipe de LifeInvaders	2
1.3	Identité visuelle	3
2	Choix techniques	3
3	Répartition des tâches	4
4	Contrôles du personnage	5
4.1	Déplacement du joueur	5
4.2	Contrôle de la caméra	5
5	Gameplay	7
6	Organisation de partie	7
7	Système de progression	8
8	Heads Up Display (HUD)	8
9	Réalisation de l'intelligence artificielle	8
10	Réalisation du multijoueur	9
11	Création de la carte	10
11.1	Taille et topologie de la carte	10
11.2	Univers et design	10

1 Introduction

1.1 Présentation du jeu

Panic At Tortuga est un jeu multijoueur dans lequel les différents joueurs doivent s'entre-éliminer sur une île. Le jeu s'inspire de jeux connus : Le premier est le mod "Guess Who?"¹ de Garry's Mod où une première équipe doit rechercher et éliminer l'autre équipe qui s'est déguisée en NPC (Non-Player Character) et qui doit imiter les mouvements aléatoires du bot. Le deuxième jeu dont nous nous sommes inspirés est le mode multijoueur proposé dans quelques jeux Assassin's Creed. Dans ce mode, chacun des joueurs se voit attribuer une cible, ce qui fait que tous les joueurs sont à la fois des traqueurs et traqués.

Notre jeu se passera sur une île tropicale fictive dans les Caraïbes. L'objectif est de se cacher parmi la foule d'une petite île perdue au beau milieu de l'océan et d'éliminer votre cible sans être vu.

1.2 L'équipe de LifeInvaders

DEVERS Renaud-Dov (Chef de Projet)

J'adore l'informatique depuis des années, je suis tombé dans la marmite des vieux ordinateurs (Windows 2000 !) à disquette quand j'étais petit.

J'aime les travaux et les projets de groupe, en ayant trois à mon actif, dont le projet d'ISN et celui de SI. J'ai pu réaliser par exemple une application dynamique pour libraires avec une connexion à la database de Google Books. J'y ai appris le rôle d'un chef de groupe, de l'organisation à la responsabilité, je trouve les projets formateurs. Ce projet sera donc l'occasion de m'améliorer dans le travail de groupe, pour être prêt pour le monde professionnel.

KEDJNANE Harrys

Avant EPITA, je faisais une PACES et je n'avais jamais codé même si l'informatique est un sujet qui m'a toujours fortement intéressé. Depuis mon arrivé à cette école, j'ai découvert les joies de la programmation et ce projet me permettra d'approfondir ce sujet appliqué à un autre milieu qui m'intéresse tout autant : le jeu vidéo.

DE LA PORTE DES VAUX Paul

Ayant découvert le monde de l'informatique assez tard, je n'ai fait que des petits projets, comme des Hackintosh ou des projets python de cryptographie et de reconnaissance d'objets. La création d'un jeu en 3D m'enchantes donc, car elle va me permettre d'explorer de nouveaux horizons.

COHEN-SCALI Julien

Étant un gros joueur et passionné depuis mon enfance, ce projet est un moyen pour moi de m'exprimer grâce aux différents cours de programmation en C#. Par ailleurs, ayant fait un stage à l'ISART, le C# et Unity ne me sont pas inconnus. De plus, ce projet est un bon moyen de s'exercer à la Programmation Orientée Objet vue lors derniers cours de programmation, ce qui permettra à notre groupe d'améliorer notre

1. Lui même inspiré de jeux tels que Hide & Seek, Spy Party ou encore Prop Hunt

code et d'acquérir une méthode bien plus professionnelle, nous préparant à notre vie future. La collaboration autour du projet et l'importance de travailler en groupe grâce a git développe en chacun de nous un esprit d'équipe conséquent. Ce projet est donc, pour moi, un très bon entraînement pour le développement de projet dans ma vie futur. (mais aussi un très bon moyen de s'amuser).

1.3 Identité visuelle

Pour créer le logo de notre jeu, nous avons utilisé Photoshop pour faire un logo stylisé, dans les mêmes tons que propose notre jeu.



FIGURE 1 – Logo de Panic At Tortuga

2 Choix techniques

Nous avons décidé de réaliser un jeu en 3D sur le moteur cross-platform de Unity Technologies. Ce moteur est complet et possède plein de fonctionnalités pour créer un environnement jeu fonctionnel.

Nous avons décidé d'acheter un asset contenant de multiples textures, prefabs, objets et personnages sur le thème des pirates et des Caraïbes.² Nous n'aurons donc pas, sauf quelques rares modèles, besoin de modéliser des mesh sur Blender.

Nous avons choisi d'intégrer l'outil PUN 2 de Photon pour la réalisation du multijoueur et l'interaction entre joueurs.

2. Voir Partie Coûts de production

3 Répartition des tâches

Tâche	Responsable	Suppléant
Carte		
Création de la carte	Paul	Renaud-Dov
Création du lobby	Renaud-Dov	Paul
Réseau		
Implémentation du multijoueur	Julien	Harrys
IA		
Réalisation des différents types d'IA	Renaud-Dov	Paul
Menus		
Interface	Julien	Harrys
HUD	Harrys	Julien
Game Core		
Contrôle du personnage	Renaud-Dov	Harrys
Animations	Renaud-Dov	
Objets dynamiques	Renaud-Dov	Harrys
Déroulé d'une partie	Harrys	Paul
Autre		
Système de progression	Harrys	
Réalisation et maintenance du site Web	Paul	-

Tâche	Soutenance		
	Soutenance 1	Soutenance 2	Soutenance 3
Mouvement	50%	70%	100%
Interfaces/HUD	40%	60%	100%
Cartes	40%	80%	100%
Réseau	20%	50%	100%
IA	30%	60%	100%
Mécaniques de jeu	50%	70%	100%
Progression	20%	60%	100%
Site internet	0%	100%	100%

4 Contrôles du personnage

4.1 Déplacement du joueur

Nous avons décidé de créer des contrôleurs avec une vue à la troisième personne. Notre jeu sera pour le moment, jouable uniquement au clavier-souris. Pour animer

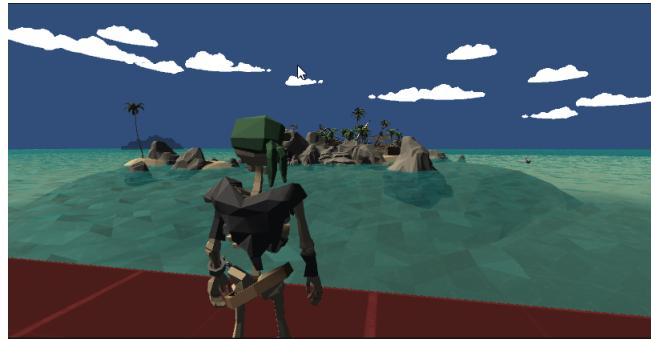
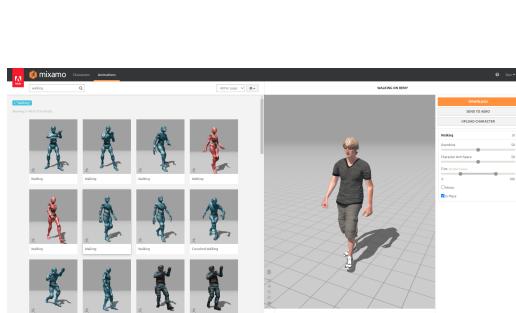


FIGURE 2 – Vue à la troisième personne

les mouvements des joueurs et des NPC, nous avons utilisé la banque d'animations Mixamo que propose Adobe. En plus d'avoir des animations détaillées, elles sont entièrement compatibles avec le moteur 3D.



(a) Banque d'animations Mixamo



(b) Animation de course sur un personnage

FIGURE 3 – Animations de Mixamo vers Unity

Nous avons également choisi d'intégrer des échelles pour permettre aux joueurs de grimper et pour pouvoir rajouter de la verticalité à notre carte³

4.2 Contrôle de la caméra

Pour le contrôle de la caméra, nous utiliserons une vue à la troisième personne avec possibilité de changer la caméra d'épaule. Nous avons déjà implémenté une caméra de base, mais qui pose deux problèmes :

- La caméra peut traverser les obstacles si le joueur à le dos collé au mur.

3. Voir la partie : Création de la carte



FIGURE 4 – Joueur grimpant sur l'échelle

- La caméra force la vision du joueur devant lui, ce qui signifie que quand quand le joueur tourne la tête, le personnage aussi, il ne peut donc regarder en arrière.

Pour régler ce problème, nous prendrons l'outil Cinemachine qui permet en plus de faire des travelings, transitions, possède des caméras adaptées.

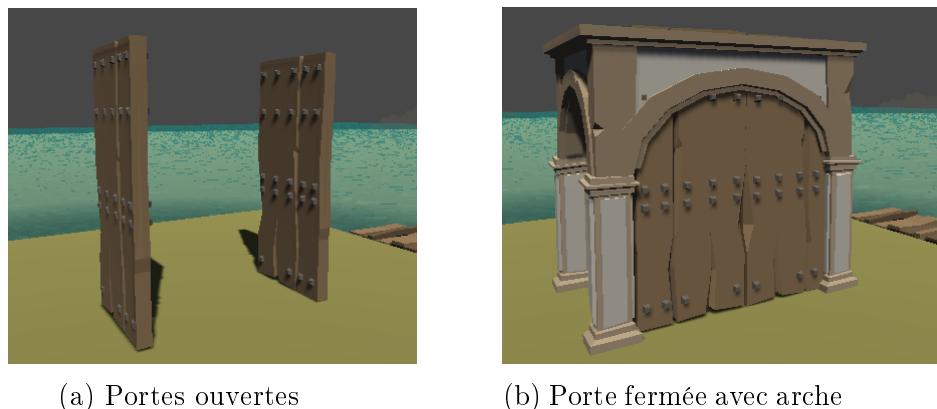
5 Gameplay

Chaque joueur se voit attribuer une cible et est ciblé par un autre joueur. Nous intégrons donc un système de sélection de cibles, et un moyen pour le joueur de sélectionner un personnage (joueur ou non) à proximité de lui.

Le but est de tuer le plus de cibles joueurs dans le temps imparti. Chaque assassinat vous rapporte X points.

À l'aide de la classe de personnage, nos personnages **pourraient** avoir des pouvoirs spécifiques, tel que se déguiser en quelqu'un d'autre pendant une durée limitée, pouvoir tuer à moyenne distance, ralentir et affaiblir un joueur (l'empêcher de courir, d'utiliser des capacités par exemple), avoir une boussole plus précise...

Nous avons également rajouté des objets pouvant être utilisés par les joueurs (Echelles⁴, Portes) et comptons en rajouter d'autres.



(a) Portes ouvertes

(b) Porte fermée avec arche

FIGURE 5 – Exemple des portes que nous avons réalisé

6 Organisation de partie

L'organisation de partie concerne la création des scripts et ressources permettant à chaque partie multijoueur de se dérouler suivant un schéma prédéterminé. Cela comprend le début et la fin de manche (choix du gagnant, présentation du classement de fin de partie par exemple), le placement des éléments dynamiques de la scène, les timers, le comptage des points, l'assignation des cibles...

Pour offrir la meilleure expérience de jeu possible, cette organisation doit permettre à chaque manche d'être unique et imprévisible. Pour cela, on introduit de la complexité dans les différentes mécaniques de jeu. Par exemple, on peut attribuer plus de points au premier joueur ayant abattu sa cible ou à un joueur ayant survécu un certain temps sans se faire tuer.

4. Voir Figure 4

7 Système de progression

Un des points qui fait du jeu vidéo le passe-temps favori de nombreuses personnes, c'est le sentiment de progression. Le joueur veut sentir que ses efforts sont, d'une façon ou d'un autre, récompensés autrement que par un simple écran de victoire. C'est pourquoi nous voulons intégrer un système de progression qui peut se traduire par un ensemble d'éléments évoluant au fil des parties et du temps passé à jouer.

Un exemple très courant et très apprécié des joueurs, ce sont les *achievements*, ou succès, qui sont débloqués quand le joueur réalise une action spécifique (par exemple, après avoir gagné x parties, ou après avoir tués y personnes). Un tel système doit aussi s'accompagner d'un système de sauvegarde d'informations, car il faut que les données concernant la progression du joueur soient persistantes, même au redémarrage, pour pouvoir suivre l'évolution au fil de plusieurs sessions de jeu.

8 Heads Up Display (HUD)

L'HUD est l'interface donnant au joueur les informations dont il a besoin en cours de partie, et qui se superpose au jeu lui-même. Pour notre projet, quelques propositions concernant l'affichage sont :

- Une boussole pointant vers la cible,
- Une photo de la cible,
- Le timer,
- certaines statistiques comme le nombre de kills,
- La disponibilité de certaines capacités(capacités de classe, objets...)

Cette interface ne doit pas être trop chargée cependant. En effet elle doit être discrète pour ne pas distraire le joueur pendant sa partie.

9 Réalisation de l'intelligence artificielle

Plus spécifique aux mécaniques de l'intelligence artificielle et des NPC, trois types de personnages doivent être implémentés à la carte :

- Des NPC qui restent bloqués sur place et qui discutent en groupe
- Des IA qui se déplacent de manière aléatoire (ou qui patrouillent)
- Des NPC qui effectuent des animations précises (juste pour le décor plus réaliste et travaillé)

Pour créer l'IA, nous utiliserons le système de Nav Mesh intégré à Unity, zone où les NPC peuvent naviguer pour aller d'un point A vers un point B. On appelle cela du pathfinding, c'est-à-dire que l'IA va calculer le meilleur chemin possible pour atteindre une position. Nous avons déjà réalisé un début d'IA qui selon une liste de positions, se déplace d'un point à un autre avant de revenir à sa position initiale.

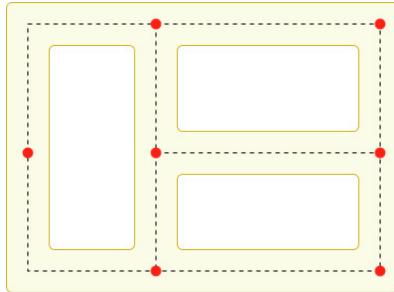


FIGURE 6 – Schéma d'un mouvement de patrouille

10 Réalisation du multijoueur

Afin de développer le cœur de notre jeu, le système permettant de connecter les joueurs ensemble, nous avons utilisé la framework Photon. Plus précisément Photon 2.

Selon Wikipédia, "un framework désigne un ensemble de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel".

Photon a donc intégré à notre projet :

- Une gestion du nom en jeu (nickname).
- Une géolocalisation des joueurs afin d'optimiser les temps de latence et de les connecter au serveur le plus proche.
- La possibilité de rapprocher des joueurs du monde entier grâce à 4 serveurs à travers pouvant accueillir jusqu'à 20 joueurs chacun.

Par soucis de simplicité, nous résumerons photon en deux parties :

- La partie "**Client Side**"
- La partie "**Server side**"

Photon fonctionne, majoritairement, grâce à deux scripts, le **launcher.cs** ainsi que le **GameManager.cs**. Le launcher.cs gère la partie Client Side tandis que le GameManager.cs gère la partie Server side.

La partie Client side, est étroitement liée avec l'UI (User Interface) puisqu'elle gère toutes les fonctions permettant l'instanciation (la création d'un objet dans une scène) du joueur dans la salle d'attente, puis dans la partie. La partie Server side permet à l'administrateur d'avoir des logs des joueurs entrants en sortants de la partie.

Ce script permet aussi à l'UI de fonctionner, grâce aux méthodes Disconnect() ou LoadArena(), qui n'ont d'effets que si le joueur est connecté à l'un des serveurs photon, d'où leur intégration dans GameManager.cs.

La plus grande difficulté a été de développer une version stable d'un multi-joueur, pour ensuite l'adapter à notre projet. Il a fallu réadapter les components de notre Player et créer une variante pour chacun des modèles personnalisables possible.

11 Création de la carte

11.1 Taille et topologie de la carte

La carte est un élément crucial du jeu. En effet, toutes les mécaniques (saut, échelles, course) deviennent inutiles si elles sont implémentées sur une carte inadéquate (avoir la possibilité de grimper sur une carte plate est aussi inutile que frustrant). A l'inverse, une carte adaptée aux possibilités de déplacement du joueur donne à ce dernier l'impression de quelque chose de fini.

Une carte spacieuse et assez verticale (accidentée et/ou avec des maisons escaladables) se présente donc comme la meilleure option, car elle est ainsi le moins restrictive possible.

11.2 Univers et design

Le jeu se déroule dans un univers pirate, aux Caraïbes. Les bâtiments doivent donc être d'inspiration hispanique, mais paraître peu entretenus, signe d'une longue occupation pirate. La carte est une île, ce qui permet de justifier la limite de carte facilement (océan). En revanche, certaines incohérences (skin de squelette, peut-être des armes futuristes) permettent d'inscrire le jeu dans un registre informel, et de rappeler que ce n'est qu'un jeu.

Tous ces éléments nous ont fait opter pour l'achat d'un asset graphique chez Synty Studios (POLYGON - Pirate Pack). Ce dernier possède de nombreux avantages, comme sa grande diversité de prefabs et textures, ou encore son aspect "Low Poly", c'est-à-dire en polygones, qui donne un aspect moderne au jeu tout en diminuant la complexité des mesh (réduisant ainsi la puissance de calcul nécessaire au jeu).



FIGURE 7 – Image de présentation de l'asset Polygon Pirate