

# Panic At Tortuga

## Rapport de soutenance n°1

Février 2021



**LIFE**  
**INVADERS**  
— — — — —  
**Production**

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Conception</b>	<b>2</b>
2.1	Réseau . . . . .	3
2.2	Intelligence Artificielle et NPC . . . . .	3
2.3	Mécaniques de jeu . . . . .	4
2.3.1	Environement . . . . .	4
2.3.2	Déplacement du personnage . . . . .	4
2.3.3	Système de verrouillage . . . . .	5
2.4	Menus : HUD/Menus . . . . .	5
2.5	Carte du jeu . . . . .	5
<b>3</b>	<b>Avances et retards</b>	<b>6</b>
3.1	Les réussites . . . . .	6
3.2	Les retards . . . . .	7
<b>4</b>	<b>Prévisions</b>	<b>7</b>
4.1	Map . . . . .	7
4.2	Interface . . . . .	7
4.3	Réseau . . . . .	7
4.4	I.A. . . . .	7
4.5	Mécaniques de jeu . . . . .	7
4.6	Autres . . . . .	7

# 1 Introduction

Panic At Tortuga est un jeu orienté multijoueur. Mais quel genre de est-ce ? Il s'agit du jeu s'inspirant de mécaniques de jeux divers, les principaux étant Assassin's Creed (son mode multi) et d'autres jeux comme "Guess Who ?".

Vous et d'autres joueurs êtes lâchés sur la petite île de Tortuga, une île où la population vit paisiblement. Mais parmi eux, déguisés, se cachent des pirates sanguinaires, et c'est à vous de les éliminer.

Le début de production de notre équipe de développement, LifeInvaders Production, composé de Paul, Harrys, Julien et Dov ; a pu commencer à développer le jeu depuis mi-décembre. Nous avons pu apprendre de nombreuses choses, se dépasser, sortir de notre zone de confort, mais surtout prendre du plaisir à créer notre jeu !

# 2 Conception

Le concept de notre jeu se base sur une idée simple : Vous avez une cible et êtes celle d'un autre joueur. Vous devez éliminer chaque cible attribué (en l'occurrence d'autres joueurs) et tenter de survivre dans ce petit archipel.

Nous nous sommes répartis les tâches rapidement les différentes tâches pour avoir les bases. Par exemple, le déplacement du personnage et de la caméra était primordial pour Harrys et Julien qui s'occupaient du multijoueur et nécessitait d'être livré rapidement. Pendant que Renaud-Dov réalisait l'implémentation de ces scripts, Julien et Harrys ont commencé à apprendre comment fonctionnait Photon. Paul s'occupait alors de la map.

Pour éviter de se perdre dans l'afflux de tâches à réaliser, nous décidé d'utiliser des outils de travail en équipe tel que des kanban (avec Github Projects).

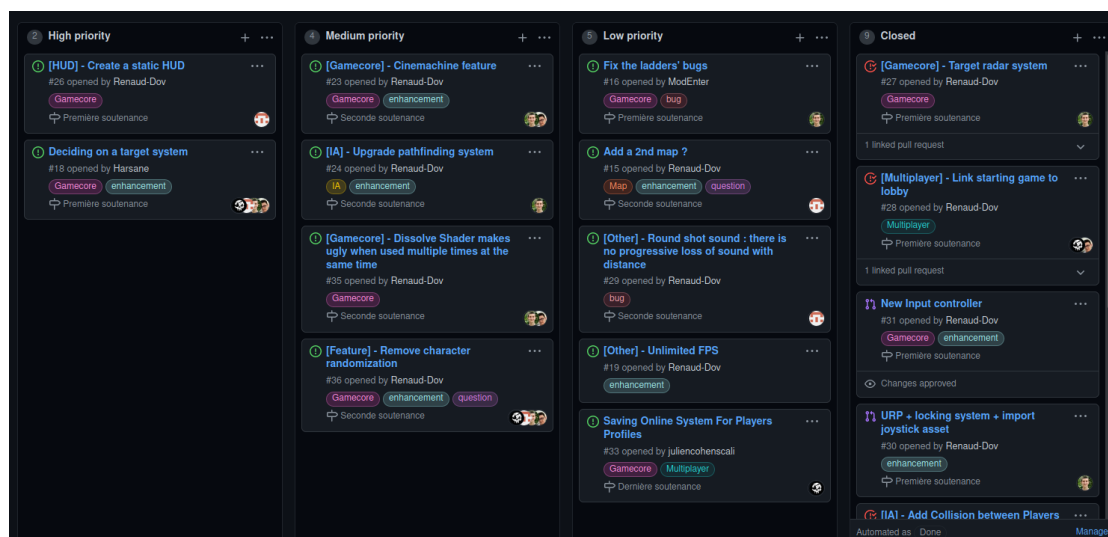


FIGURE 1 – Kanban pour la priorisation des tâches sur Github

## 2.1 Réseau

## 2.2 Intelligence Artificielle et NPC

Pour rendre notre jeu plus vivant et permettre aux joueurs de se mêler dans la foule, il nous fallait créer des IA se déplaçant dans la ville. Pour cela, nous avons décidé d'utiliser des NavMesh pour créer des zones où les NPC peuvent se balader d'un point A vers un point B. Pour ne pas rendre leur comportement linéaire et sans saveur, ils se déplacent de manière aléatoire vers un point quelconque défini dans une liste de coordonnées. Aucun NPC n'aura donc le même trajet.

L'outil de navigation est assez poussé, et permet de définir les dimensions des personnages, ainsi que la hauteur de laquelle ils peuvent sauter et les pentes qu'ils peuvent emprunter. Ici, la pente maximale est de  $33.4^\circ$ , mais si ce chiffre avait été plus élevé, les pentes sur la capture d'écran seraient devenues bleues, et les personnages auraient pu monter sans utiliser les escaliers (ce qui n'est évidemment pas le but).

Pour rendre ces IA plus humaines, les joueurs peuvent les bousculer et les étourdir en courant vers eux. Ils reprenent leur trajet au bout de quelques secondes. Il faudra ainsi, dans les prochaines versions, améliorer la navigation afin que le chemin emprunté ne soit pas toujours le plus court.

Plusieurs problèmes se posent actuellement avec l'utilisation des NavMesh :

- L'IA calcule le chemin le plus court, et n'essaie pas de se déplacer sur le côté si d'autres IA marchent dans le sens contraire. Par exemple, si l'on met deux escaliers côte à côte, toutes les IA vont prendre l'escalier le plus proche et donc se bloquer.



FIGURE 2 – Problème rencontré lorsque de nombreuses IA vont au même endroit

## 2.3 Mécaniques de jeu

### 2.3.1 Environnement

Nous nous sommes beaucoup inspiré du mode multijoueur des premiers Assassin's Creed. Nous n'avons pu récréer un système de grimpe car trop ardu à réaliser de nos propres mains. En revanche, nous avons réutilisé quelques idées de gameplay pour l'environnement, comme les échelles et les portes.



FIGURE 3 – Joueur grimpant sur l'échelle

L'échelle permet de créer un peu de verticalité dans nos niveaux. Elles ne peuvent être utilisées que par les joueurs, prendre de la hauteur permet d'emprunter des raccourcis, mais retire la discrétion (car personne de civilisé devrait être sur les toits !)

Les portes peuvent vous permettre de barrer le passage pour échapper à son tueur. Elles se ferment quand le joueur passe dessus en courant et se réouvrent au bout de quelques secondes.

### 2.3.2 Déplacement du personnage

Sans déplacement du personnage, le jeu serait ennuyeux si on ne pouvait inter-agir avec. Nous avons donc réalisé un script de déplacement qui permet au joueur de marcher, courir, sauter...

Pour le côté technique, nous avons dans un premier temps utilisé l'Input System de base proposé par Unity. Mais plusieurs problèmes se sont posés :

- On ne peut paramétrer en profondeur les keymaps



FIGURE 4 – Porte fermée après le passage d'un joueur

- Paramétrer d'autres périphériques autre que le clavier/souris est compliqué. Il faudrait avoir des scripts différents pour chaque périphérique, et donc des prefabs différents

C'est pour cette raison que nous avons migré vers le New Input System. Celui ci est ergonomique, multiplateform et permet un mix de différents périphériques à fois, comme la manette par exemple. Il a fallu editer certains bout de code pour les rendre compatible ; heureusement, la migration ne fut pas très longue après avoir compris son fonctionnement.

Il est donc actuellement possible de jouer au clavier/souris ou à la manette (pas les deux à fois).

### 2.3.3 Système de verrouillage

Le système qui attribue des cibles à chaque joueur n'est pas entièrement implémenté.

Mais le joueur peut vouloir tuer une cible, qu'elle soit la bonne ou non. Pour cela, il passe en mode verrouillage, et les personnages pointés par le viseur surbrillent. Pour les sélectionner, un coup de molette suffit, et le contour devient alors jaune, pour indiquer que la cible est verrouillée. Il lui suffit alors d'être à moins d'1 mètre pour l'éliminer.

## 2.4 Menus : HUD/Menus

## 2.5 Carte du jeu

Pour la partie graphique, nous n'avons pas choisi de réaliser nous-même les tex-



(a) Personnage au centre de l'écran en surbrillance



(b) Personnage sélectionné et verrouillé

FIGURE 5 – Système de verrouillage

tures et autres meshes pour notre jeu. Nous avons plutôt opté pour un asset payant, nous permettant de nous focaliser plus sur les mécaniques de jeu et l'implémentation du multijoueur.

### 3 Avances et retards

Partie \ Tâche	Prévu	Réalisé
Mouvement	50%	80%
Interface/HUD	40%	40%
Cartes	40%	40%
Réseau	20%	50%
IA	30%	40%
Mécaniques de jeu	50%	40%
Progression	20%	20%

TABLE 1 – Tableau des avances et retards dans les différentes parties

#### 3.1 Les réussites

La création du système de mouvement n'a pas été finalement très compliqué. Il manque un peu de finesse, et d'une physique de saut plus réaliste. Il reste cependant complet.

L'utilisation du pathfinding pour les mouvements des IA a été beaucoup plus simple à implémenter que prévu.

## 3.2 Les retards

Certaines mécaniques de jeu sont encore en développement et ne sont pas finalisées.

## 4 Prévisions

### 4.1 Map

### 4.2 Interface

### 4.3 Réseau

### 4.4 I.A.

Pour la seconde période, notre objectif serait de pouvoir avoir une IA plus autonome et intelligente, qui n'emprunte plus forcément le chemin le plus court. Egalement, les joueurs peuvent bousculer les NPC qui s'arrêtent alors de marcher, mais ils ne peuvent jamais les pousser.

### 4.5 Mécaniques de jeu

Nous prévoyons d'intégrer pour la deuxième soutenance des pouvoirs comme un écran de fumée ou encore des couteaux, pour rendre le jeu plus intéressant et dynamique.

Nous comptons également finir l'implémentation du système d'attribution des cibles.

### 4.6 Autres

Pour tester le multijoueur avec des invités, nous avons réalisé un début de launcher qui télécharge les dernières mises à jour. Nous aimerions continuer à le développer pour avoir le launcher qui se mette à jour automatiquement sans à avoir à le réinstaller.



## Table des figures

1	Kanban pour la priorisation des tâches sur Github . . . . .	2
2	Problème rencontré lorsque de nombreuses IA vont au même endroit .	3
3	Joueur grim pant sur l'échelle . . . . .	4
4	Porte fermée après le passage d'un joueur . . . . .	5
5	Système de verrouillage . . . . .	6

## Liste des tableaux

1	Tableau des avances et retards dans les différentes parties . . . . .	6
---	---	---