

# Rapport de soutenance

LA GRANDE EVASION

Groupe 9 ("Epi Games")

22 avril 2024



AKHDAR Wael (chef de groupe)

ZAHRA Youcef

ZIANI Jessim

OLLER Ilann

LASSIK Nabil

---

## Résumé

Notre entreprise “Epi Games” est composée de cinq élèves de première année à EPITA. Chaque membre mettra en œuvre ses compétences afin de concevoir un jeu vidéo d’escape game en trois dimensions, nommée LA GRANDE ÉVASION. Nous réalisons ce projet dans le but de mettre en œuvre nos connaissances acquises durant cette première année en école d’ingénieur, et de découvrir de nouveaux domaines tels que : l’Intelligence Artificielle, la modélisation 3D, la connectivité réseau, le développement Web.



FIGURE 1 – Logo Epi Games

---

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Retour sur le cahier des charges</b>	<b>5</b>
2.1	Scénario . . . . .	5
2.2	Tableau caractéristique du jeu . . . . .	6
2.3	Diagramme de Gantt . . . . .	7
<b>3</b>	<b>Avancement du jeu</b>	<b>8</b>
3.1	Design du jeu . . . . .	8
3.1.1	Environnement . . . . .	8
3.1.2	Personnage . . . . .	10
3.1.3	Garde . . . . .	10
3.2	Mécanique de jeu . . . . .	11
3.2.1	Mouvements du joueur . . . . .	11
3.2.2	Animations du joueur . . . . .	14
3.3	Intelligence Artificielle (IA) . . . . .	18
3.3.1	Navmesh . . . . .	18
3.3.2	Mouvement IA . . . . .	20
3.4	Site Web . . . . .	21
3.4.1	Architecture du site Web . . . . .	21
3.4.2	Sections du site Web . . . . .	21
<b>4</b>	<b>Prévisions de l'avancement</b>	<b>22</b>
4.1	Design du jeu . . . . .	22
4.1.1	Prison . . . . .	22
4.1.2	Personnage . . . . .	23
4.1.3	Garde . . . . .	24
4.1.4	Objets . . . . .	24
4.1.5	Menu du jeu . . . . .	24

---

4.2	Mécanique de jeu . . . . .	24
4.2.1	Inventaire . . . . .	25
4.3	Intelligence Artificielle (IA) . . . . .	25
4.3.1	Mouvement IA . . . . .	25
4.3.2	Animation IA . . . . .	25
4.4	Mode multijoueur . . . . .	26
4.5	Site Web . . . . .	26
<b>5</b>	<b>Conclusion</b>	<b>26</b>

---

## 1 Introduction

Epi Games est enchanté de vous présenter les débuts de notre jeu baptisé "LA GRANDE ÉVASION". Ce projet se présente comme un escape game en 3D se déroulant dans un environnement carcéral. L'objectif principal est de réussir à s'échapper. Pour y parvenir, la collaboration entre les joueurs est essentielle : ils devront résoudre une série d'énigmes et échapper à la surveillance du garde pour retrouver leurs libertés. Après avoir acquis une expertise approfondie de la plateforme de développement Unity, nous avons atteint un stade initial prometteur pour notre projet.



FIGURE 2 – Logo La Grande Evasion

---

## **2 Retour sur le cahier des charges**

### **2.1 Scénario**

Quatre prodiges se retrouvent capturés et enfermés par un scientifique dément, obsédé par l'idée d'extraire leur intelligence à des fins expérimentales. Leur unique but devient alors de s'évader dans un délai donné pour assurer leur survie. Avec ingéniosité et subtilité, les prisonniers explorent les moindres recoins de leur prison, exploitant chaque faille du système de sécurité pour concevoir des plans d'évasion astucieux. Chaque énigme résolue et chaque obstacle surmonté ne fait que renforcer leur résolution à retrouver leur liberté. Durant leur tentative d'évasion, le docteur patrouille activement dans la prison, traquant sans répit ceux qui osent s'échapper. S'il réussit à les attraper, il les ramène impitoyablement à leur cellule, renforçant ainsi son emprise implacable sur eux.

## 2.2 Tableau caractéristique du jeu

Type de jeu :						
Action/Aventure	Battle Royale	Beat them all	Combat	Simulation		
FPS	MMORPG	MOBA	Party Games	Survival Horror		
Plateforme	Puzzles	Reflexion	Rogue Like	TPS		
RPG	RTS	Sandbox	Shoot them up	Course		
Autre :						
<b>Caractéristiques générales du jeu :</b>						
IA : Errer	Attaquer	S'échapper	"Path Finder"			
Multijoueurs : Coopé	Battle (2-4)	Massif				
Réseau : P2P	Lan	Online				
<b>Caractéristiques graphiques :</b>						
Dimension : 2D	3D	Autres :				
Particularités : Stéreoscopie	AR	VR				
graphiques : Perso	Custom	Existant				
Précisions :						
<b>Caractéristiques sonores :</b>						
Musique : Perso	Custom	Existant				
FX : Perso	Custom	Existant				
Précisions :						
<b>Autres caractéristiques :</b>						
Site Web : Perso	Custom	Préfabriqué				

FIGURE 3 – Ancien cahier des charges technique

Type de jeu :						
Action/Aventure	Battle Royale	Beat them all	Combat	Simulation		
FPS	MMORPG	MOBA	Party Games	Survival Horror		
Plateforme	Puzzles	Reflexion	Rogue Like	TPS		
RPG	RTS	Sandbox	Shoot them up	Course		
Autre :						
<b>Caractéristiques générales du jeu :</b>						
IA : Errer	Attaquer	S'échapper	"Path Finder"			
Multijoueurs : Coopé	Battle (2-4)	Massif				
Réseau : P2P	Lan	Online				
<b>Caractéristiques graphiques :</b>						
Dimension : 2D	3D	Autres :				
Particularités : Stéreoscopie	AR	VR				
graphiques : Perso	Custom	Existant				
Précisions :						
<b>Caractéristiques sonores :</b>						
Musique : Perso	Custom	Existant				
FX : Perso	Custom	Existant				
Précisions :						
<b>Autres caractéristiques :</b>						
Site Web : Perso	Custom	Préfabriqué				

FIGURE 4 – Nouveau cahier des charges technique

La mise en place du mode multijoueur est un élément clé de notre projet, et après mûre réflexion, nous avons décidé d'opter pour un mode multijoueur en ligne plutôt qu'en local (LAN). Cette décision découle de notre volonté de rendre l'expérience de jeu plus accessible et conviviale. Le choix du mode multijoueur en ligne permettra aux joueurs de se connecter et de jouer ensemble sur différents appareils, éliminant ainsi les contraintes liées à la proximité physique requise pour le mode LAN.

## 2.3 Diagramme de Gantt

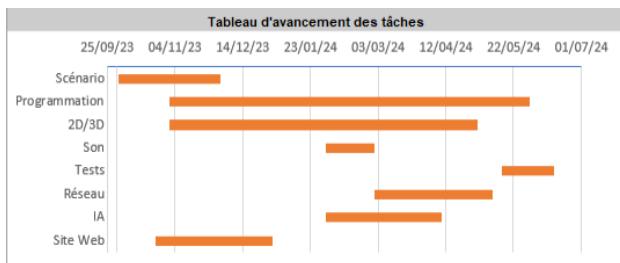


FIGURE 5 – Ancien diagramme de Gantt

Nous avons réalisé que les tests sont une étape cruciale qui s'étend tout au long du processus de création du jeu, ce qui nous a amenés à prolonger cette phase. De plus comme énoncé précédemment nous avons remis en question le type réseau utilisé et donc retardé sa conception, de ce fait nous avons donc prioriser l'avancement de L'IA afin de prévenir tout retard éventuel. Le scénario et le site web ont été effectués dans les temps. Concernant les aspects sonores et visuels du jeu, nous avons décidé de les reporter temporairement afin de concentrer nos efforts sur le développement de l'intelligence artificielle. Cela ne signifie pas un abandon complet du son, mais plutôt une réattribution des ressources à un stade ultérieur, une fois que l'IA sera plus avancée. Nous croyons fermement que cette approche nous permettra de garantir la fonctionnalité du jeu, même si cela implique de sacrifier certains détails esthétiques pour le moment.



FIGURE 6 – Nouveau diagramme de Gantt

---

## **3 Avancement du jeu**

### **3.1 Design du jeu**

#### **3.1.1 Environnement**

Le jeu se déroule dans une prison, nous devons donc concevoir une représentation en 3D de cette dernière, avec deux niveaux : les cellules en haut et la salle principale en bas, afin de coller au mieux au contexte du jeu vidéo.

Nous avons opté pour Blender, un logiciel de modélisation et d'importation d'éléments 3D, pour cette tâche. Pour obtenir les éléments nécessaires, nous avons utilisé différentes ressources, principalement le site Sketchfab, qui propose une variété d'éléments 3D. Nous avons trouvé une représentation de prison qui correspondait approximativement à nos besoins, que nous avons ensuite modifiée pour l'adapter parfaitement à notre vision du jeu.

Les modifications ont porté sur l'intérieur et l'extérieur de la prison. À l'intérieur, nous avons supprimé des cellules du niveau inférieur pour libérer de l'espace pour la salle principale et nous avons ajouté des tables. Nous avons également réduit le nombre de cellules au niveau supérieur pour qu'il y en ait quatre, correspondant aux quatre joueurs dans le cas d'une partie multijoueur. De plus, nous avons entièrement modifié le décor des cellules, y compris les lits, les toilettes et les lavabos.



FIGURE 7 – Avant-Après de l'intérieur de la prison

Concernant l'extérieur de la prison, nous avons corrigé certains problèmes esthétiques du modèle récupéré sur Sketchfab, tels que des barreaux et le sol dépassant des murs.



FIGURE 8 – Avant-Après de l'extérieur de la prison

Pour les textures, nous avons utilisé celles des modèles 3D récupérés sur Sketchfab.

---

### 3.1.2 Personnage

Pour le personnage principal nous n'avons pas encore de design définitif. En revanche, nous utilisons pour le moment un personnage 3D nommé "Y Bot" accessible sur la plateforme Mixamo représentant un robot pour pouvoir implémenter les mouvements.

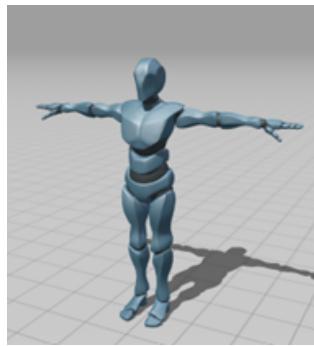


FIGURE 9 – Image de Ybot

### 3.1.3 Garde

Pour le design du garde de la prison nous avons un design qui risque d'être modifié par la suite. Nous utilisons pour le moment un personnage 3D accessible sur l'asset store de Unity représentant un garde de prison.



FIGURE 10 – Image du garde

---

## 3.2 Mécanique de jeu

### 3.2.1 Mouvements du joueur

Pour les mouvements du joueur, nous avons créé un script permettant au personnage de se déplacer verticalement et horizontalement :

```
float verticalAxis = Input.GetAxis("Vertical");
float horizontalAxis = Input.GetAxis("Horizontal");
```

FIGURE 11 – Variables des axes

Dans un premier temps nous avons récupéré l'entrée de l'axe vertical et de l'axe horizontal et les avons assignés respectivement aux variables verticalAxis et horizontalAxis.

```
else if(verticalAxis>0)
{
    PlayerAnimator.SetFloat("walk vertical", 1);
    Vector3 movement = this.transform.forward;
    movement.Normalize();
    rb.MovePosition(transform.position + movement*speed*time.fixedDeltaTime);
}
```

FIGURE 12 – Déplacements vers l'avant

Ensuite, si l'axe vertical est supérieur à 1, autrement dit si la touche “Flèche du haut” est enfoncé, un vecteur “movement” est créé représentant la direction vers l' avant du joueur , puis nous généralisons ce vecteur afin de garantir une vitesse constante enfin nous déplaçons le Rigidbody du joueur vers l'avant en ajoutant movement au produit de la vitesse de déplacement (speed), et du temps écoulé depuis la dernière frame (Time.fixedDeltaTime) à la position actuelle du joueur (transform.position).

---

Cette condition est répétée pour le cas où l'axe vertical est inférieur à 1 à la seule différence que le vecteur “movement” est opposé, de plus la variable de vitesse de déplacement “speeder” est inférieure a “speed” :

```
else if(verticalAxis<0)
{
    PlayerAnimator.SetFloat("walk vertical", -1);
    Vector3 movement = this.transform.forward * verticalAxis;
    movement.Normalize();
    rb.MovePosition(transform.position + movement*speedar*Time.fixedDeltaTime);
}
```

FIGURE 13 – Déplacements vers l’arrière

Nous avons procédé de la même manière pour les mouvements horizontaux, sauf que le vecteur assigné à la variable “movement” est transform.right (représentant le vecteur pointant vers la droite du joueur) et ici “speedcote” est égal a “speed” :

```
else if(horizontalAxis>0)
{
    PlayerAnimator.SetFloat("walk horizontal", 1);
    Vector3 movement = this.transform.right;
    movement.Normalize();
    rb.MovePosition(transform.position + movement*speedcote*Time.fixedDeltaTime);
}
else if(horizontalAxis<0 )
{
    PlayerAnimator.SetFloat("walk horizontal", -1);
    Vector3 movement = -this.transform.right;
    movement.Normalize();
    rb.MovePosition(transform.position + movement*speedcote*Time.fixedDeltaTime);
}
```

FIGURE 14 – Déplacements vers la droite et vers la gauche

---

De plus nous avons créé une condition permettant au joueur d'accélérer (sprint) :

```
if (Input.GetKey(KeyCode.LeftShift) && verticalAxis>0 )
{
    PlayerAnimator.SetFloat("run", 1);
    PlayerAnimator.SetFloat("walk vertical", 1);
    Vector3 movement = this.transform.forward * verticalAxis;
    movement.Normalize();
    rb.MovePosition(transform.position + movement*speedf*Time.deltaTime);
}
```

FIGURE 15 – Sprint

Si les touches “flèche du haut” et “maj gauche” sont enfoncées, le même code est appliqué que celui pour se déplacer vers l'avant sauf que la vitesse de déplacement est plus importante (speedf > speed).

Pour finir nous avons ajouté un script de saut :

```
bool canJump = Time.time - lastJumpTime >= 1.5f;
```

FIGURE 16 – Condition pour sauter

```
if (Input.GetButton("Jump") && canJump )
{
    rb.AddForce(Vector3.up*4,ForceMode.Impulse);
    PlayerAnimator.SetFloat("jump", 1);
    lastJumpTime = Time.time;
```

FIGURE 17 – Saut

Ce script crée une variable booléen qui renvoie si le joueur peut sauter, (le joueur peut sauter chaque 1.5 secondes), si le bouton espace est enfoncé et que le joueur peut sauter, nous ajoutons une force au rigidbody du joueur, la force est dans la direction “Vector3.up” (vers le haut) multipliée par 4 unités, et appliquée avec “ForceMode.Impulse”, signifiant qu'elle est appliquée une seule fois. Enfin, la dernière ligne enregistre le moment où le

---

saut a eu lieu en stockant la valeur actuelle du temps dans la variable lastJumpTime. Cela sera utile pour limiter le nombre de saut.

### 3.2.2 Animations du joueur

Nous avons amélioré l'expérience de jeu en introduisant des animations pour le joueur, activées dans des circonstances spécifiques. Une animation de respiration se déclenche lorsque le joueur est immobile :

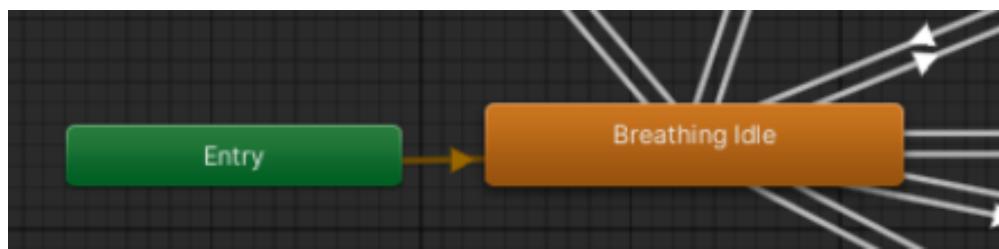


FIGURE 18 – Animation par défaut

Pour faire ceci, après avoir importé l'animation depuis le site Mixamo, nous avons ajouté une transition du bloc "Entry" à notre animation "Breathing Idle", permettant donc à cette animation d'être activée par défaut sur notre personnage.

Pour les autres animations nous avons dû créer des variables et des conditions sur celles-ci :

Ces quatre variables sont des nombres réels (float), et les conditions associées à ces variables sont celles-ci :

1. **Walking** : si la variable "walk vertical" est supérieure à 0.1 et que "run" est inférieur à 0.1, alors l'animation de marche est déclenchée. et si "walk vertical" est inférieur à 0.1 l'animation repasse en "Breathing Idle" (animation par défaut).

= walk vertical	0
= walk horizontal	0
= jump	0
= run	0

FIGURE 19 – Variables de l'animator

Conditions

= walk vertical	Greater	0.1
= run	Less	0.1

(a) Condition de démarrage "Walking"

Conditions

= walk vertical	Less	0.1
-----------------	------	-----

(b) Condition d'arrêt "Walking"

FIGURE 20 – Illustration des conditions de démarrage et d'arrêt "Walking"

2. **Left/Right Strafe Walking** : même principe, mais avec la variable "walk horizontal" pour se diriger vers la droite et la gauche.

3. **Walking Backwards** : similaire au fonctionnement des autres animations.

4. **Jumping** : déclenché de manière similaire en fonction de la condition de saut.

5. **Run** : cette animation d'accélération peut être activée à deux moments, donc elle possède deux conditions de démarrage et deux conditions d'arrêt, contrairement aux autres animations. Ces conditions sont également basées sur les mouvements du personnage.

Conditions

= run	Greater	0.1
= walk vertical	Greater	0.1

(a) 1ère condition de démarrage "Run"

Conditions

= run	Less	0.1
-------	------	-----

(b) 1ère condition d'arrêt "Run"

FIGURE 21 – Illustration des conditions de démarrage et d'arrêt "Run"

Toutes ces conditions sont implémentées dans des transitions partant de l'état "Breathing Idle" dans l'Animator.

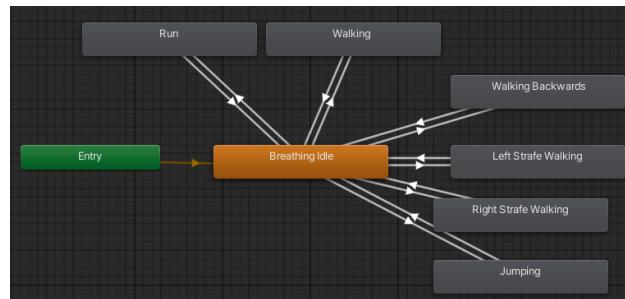


FIGURE 22 – Animator de “Y Bot”

Sauf ces deux dernières permettant de passer de l'état "Walking" à "Run" et vice-versa :

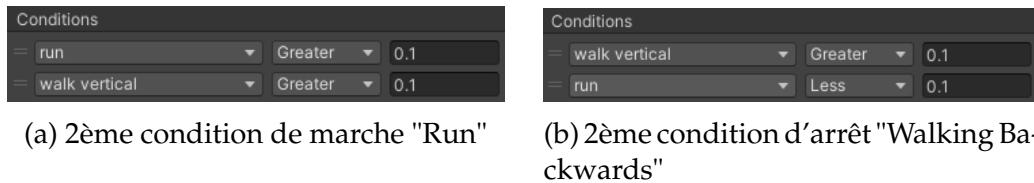


FIGURE 23 – Illustration des deux conditions

Pour illustrer, l'Animator final prend la forme suivante :

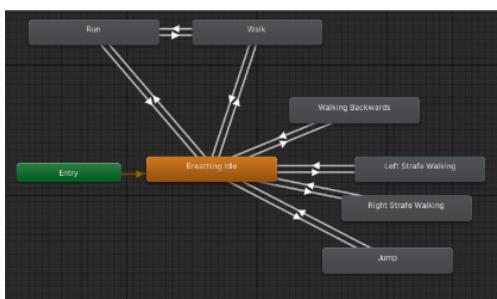


FIGURE 24 – Animator Final de “Y Bot”

---

Il est également important de noter que les variables mentionnées précédemment sont modifiées dans le script move.cs (figure 11-17). Par exemple, la variable "Jump" est initialisée à 0 au début du script et est ensuite réglée à 1 lorsque la condition de saut est vérifiée.

```
PlayerAnimator.SetFloat("jump", 0);
```

(a) Initialisation de "jump"

```
PlayerAnimator.SetFloat("jump", 1);
```

(b) Incrémentation de "jump"

FIGURE 25 – Illustrations de l'initialisation et de l'incrémentation de "jump"

---

### 3.3 Intelligence Artificielle (IA)

#### 3.3.1 Navmesh

Nous avons réalisé l'application du Nav Mesh sur le sol de la prison, suivie de l'application de modificateurs Nav Mesh aux piliers de la salle dans le but d'augmenter la surface du Nav Mesh.

Le Nav Mesh, ou maillage de navigation, est une représentation tridimensionnelle de l'environnement de jeu. Il est utilisé par les agents de navigation pour se déplacer efficacement, en divisant l'espace en régions accessibles et inaccessibles.

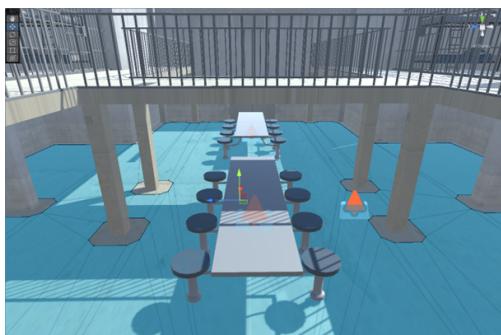


FIGURE 26 – Nav Mesh avant le Nav Mesh Modifier

Quant aux Nav Mesh Modifier, ils sont des composants permettant d'ajuster la surface du Nav Mesh. Dans ce contexte, leur application aux piliers de la salle vise à étendre la zone navigable du Nav Mesh autour des obstacles, comme les piliers. Cela permet aux agents de navigation de contourner ces obstacles de manière fluide tout en explorant l'environnement de jeu.



FIGURE 27 – Nav Mesh après le Nav Mesh Modifier

---

Étant donné que les marches étaient trop courtes pour être correctement franchies par les agents de navigation, malgré nos tentatives de réglage des paramètres tels que la hauteur de marche (step height) ou la pente maximale (max slope), nous avons été contraints d'utiliser un NavMesh Link. Ce dernier permet de créer une connexion entre deux surfaces du NavMesh qui ne sont pas directement accessibles par les algorithmes de navigation standard. En substance, cela permet aux agents de navigation de passer d'une zone à une autre de manière fluide, même si elles ne sont pas directement reliées sur le NavMesh initial.

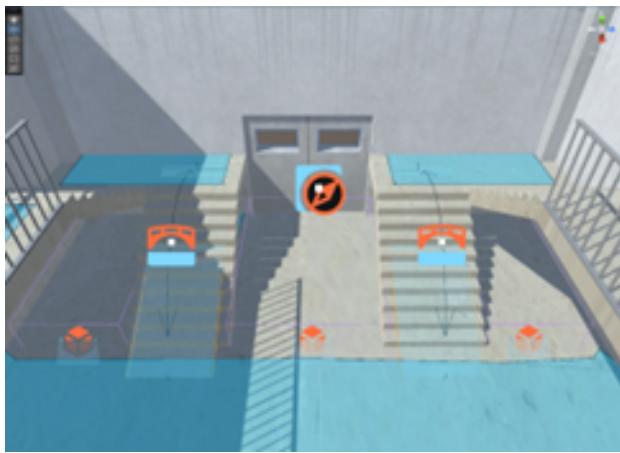


FIGURE 28 – Nav Mesh Link reliant les deux Nav Mesh Surface

---

### 3.3.2 Mouvement IA

Le script déplacement.cs est responsable de la navigation du garde dans l'environnement du jeu.

Au démarrage du jeu, la méthode Start() est appelée. Dans cette méthode, le script récupère le composant NavMeshAgent attaché à l'objet (le garde) et l'assigne à la variable agent. Le composant NavMeshAgent est responsable de la navigation du garde dans l'environnement du jeu.

```
void Start()
{
    agent = GetComponent<NavMeshAgent>();
}
```

FIGURE 29 – méthode Start()

À chaque frame, la méthode Update() est appelée. Dans cette méthode, la destination de l'agent de navigation (agent.destination) est mise à jour pour être égale à la position du joueur (YBot.position). Cela permet au garde de se déplacer automatiquement vers la position du joueur à chaque frame.

```
void Update()
{
    agent.destination = YBot.position;
}
```

FIGURE 30 – méthode Update()

Le code utilise donc le composant NavMeshAgent de Unity pour permettre au garde de suivre le joueur de manière dynamique. Cette approche simplifie la gestion du mouvement du garde, car le système de navigation de Unity se charge de la détection d'obstacles et du calcul des chemins optimaux.

---

## 3.4 Site Web

### 3.4.1 Architecture du site Web

Les carrés symbolisent les pages du site, chacune étant identifiée par son nom situé juste au-dessus. Les flèches placées au-dessus représentent les liens permettant d'accéder à ces pages, tandis que celles situées en-dessous décrivent les diverses rubriques qu'elles contiennent.

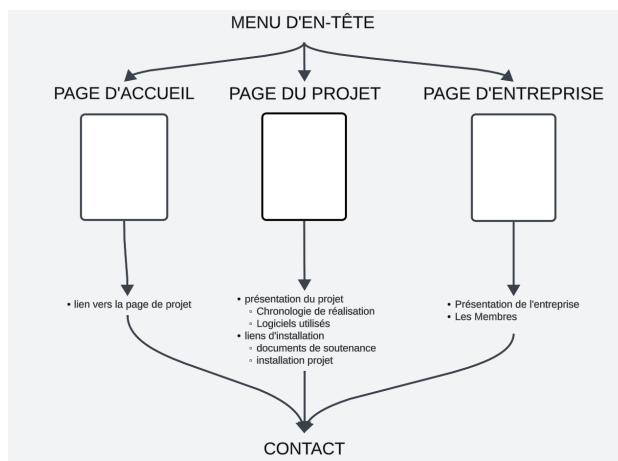


FIGURE 31 – architecture du site Web

### 3.4.2 Sections du site Web

Afin d'avoir un rendu visuel plus agréable du site Web, nous l'avons mis en forme avec du CSS, un langage de programmation destiné au formatage des pages HTML. La première page fait office de page d'accueil avec notamment la couverture du jeu à la une de notre site, ici La Grande Évasion.

---

Ensuite vient la page du projet, cette page indique toute information utile sur l'avancement des tâches ou les logiciels utilisés. Il y a aussi des liens de téléchargement du jeu et des documents de soutenance tels que les rapports de soutenances, les plans de soutenances ou encore le rapport de projet. Cependant, le bouton pour le téléchargement du jeu ou pour le rapport de projet ne sont pas encore fonctionnels car le projet n'est pas encore terminé. Sur la dernière page, nous y avons mis les informations plus centrées sur l'entreprise telles que les membres de celle-ci.

Enfin, nous avons un menu d'en-tête et un pied-de-page présents sur toutes les pages. Le menu d'en-tête sert à sélectionner la page souhaitée tandis que le pied-de-page donne des informations utiles pour nous contacter.

## 4 Prévisions de l'avancement

### 4.1 Design du jeu

Des ajustements doivent être apportés au design du jeu afin d'optimiser son rendu final. Ces modifications visent à améliorer différents aspects pour garantir une expérience de jeu optimale et immersive.

#### 4.1.1 Prison

Pour la suite du projet, nous devons concentrer nos efforts sur la modification de la prison à l'intérieur pour améliorer le décor, afin qu'il corresponde parfaitement à l'environnement souhaité pour le scénario du jeu. Cette adaptation est essentielle pour créer une atmosphère immersive et cohérente avec l'intrigue. Nous devons concevoir un environnement qui non seulement reflète l'aspect carcéral, mais qui offre également des éléments propices à l'intégration d'éénigmes et de défis pour les joueurs.

---

En nous basant sur le scénario, nous devons envisager des décors intérieurs qui évoquent à la fois le sentiment d'oppression d'une prison et le contexte de l'expérimentation scientifique menée par le personnage du docteur. Cela pourrait inclure des détails tels que des équipements de laboratoire abandonnés, des graffitis laissés par d'autres prisonniers, des indices sur les expériences menées par le scientifique, etc. Nous devons également veiller à ce que ces éléments décoratifs servent de pistes pour résoudre les énigmes et avancer dans le jeu.

En ce qui concerne l'extérieur de la prison, bien que nous n'ayons pas prévu de modifications spécifiques, nous devons rester ouverts aux ajustements nécessaires. Cela pourrait inclure des améliorations des textures des murs pour renforcer l'aspect visuel, ou même des révisions des portes de sortie pour les rendre plus cohérentes avec le reste du design de la prison.

En résumé, notre prochaine étape consiste à peaufiner le décor intérieur de la prison pour qu'il corresponde parfaitement au scénario et à l'ambiance recherchée, tout en restant attentifs aux éventuelles améliorations à apporter à l'extérieur.

#### **4.1.2 Personnage**

Pour adapter le personnage principal au personnage souhaité lié au scénario, nous devons trouver et mettre en place un modèle 3D d'un prisonnier qui puisse être intégré à Unity et adapté à toutes les mécaniques du jeu. Nous veillerons également à optimiser le modèle pour garantir des performances fluides du jeu.

---

#### **4.1.3 Garde**

Nous prévoyons de modifier le design du garde de la prison pour qu'il corresponde à l'apparence du docteur présent dans le scénario. Notre prochaine étape consistera donc à trouver ou créer un modèle 3D du docteur et à l'intégrer dans Unity pour qu'il puisse parcourir la prison et interagir avec les joueurs selon les mécaniques de jeu établies.

#### **4.1.4 Objets**

L'importation d'objets 3D essentiels, tels que des clés ou des boîtes, sera nécessaire pour résoudre les énigmes du jeu. Ces objets joueront un rôle crucial dans l'avancement du joueur en lui permettant de surmonter les obstacles et de progresser dans l'histoire.

#### **4.1.5 Menu du jeu**

Nous projetons de concevoir le design du menu du jeu, une composante essentielle pour l'expérience vidéoludique. Nous avons déjà élaboré une page de couverture pour le jeu et nous avons l'intention d'adapter ce design pour le menu. Cette démarche a été facilitée par l'utilisation d'une intelligence artificielle.

### **4.2 Mécanique de jeu**

Pour ce qui est des mécaniques du jeu nous prévoyons d'ajouter des mouvements et des animations pour pouvoir se déplacer tout en étant accroupis, pouvoir accélérer lors du déplacement vers l'arrière et pouvoir sauter vers l'avant lorsque l'on sprint et saute simultanément.

---

#### **4.2.1 Inventaire**

Nous incluons également la création d'un système d'inventaire dans nos objectifs pour finaliser le projet. Cet inventaire permettra aux joueurs de stocker les différents objets nécessaires pour résoudre les énigmes et s'évader de la prison. Nous avons l'intention de concevoir un inventaire capable de stocker jusqu'à cinq objets visibles en plein jeu, ainsi qu'un inventaire étendu pouvant être ouvert pour présenter plusieurs espaces disponibles pour différents objets. Les objets pourront être récupérés et placés directement dans l'inventaire, et les joueurs auront également la possibilité de jeter des éléments de leur inventaire selon les besoins de la situation.

### **4.3 Intelligence Artificielle (IA)**

#### **4.3.1 Mouvement IA**

Il reste à ajouter une partie de code permettant au garde de patrouiller dans une partie de la salle, et de poursuivre le prisonnier uniquement s'il entre dans son champ de vision et/ou d'audition. De plus, un système de dégâts doit être intégré au code afin que le garde puisse renvoyer le prisonnier dans sa cellule.

#### **4.3.2 Animation IA**

On ajoutera une animation qui fera en sorte que le garde ne soit pas en position bras tendu : nous ajouterons une animation lorsqu'il est immobile et lorsqu'il est en train de se déplacer.

---

## **4.4 Mode multijoueur**

Le mode multijoueur est un élément essentiel de notre projet, et nous avons déjà entamé des recherches à ce sujet. Nous avons identifié Photon Engine comme un moteur de réseau en temps réel spécialement conçu pour les jeux multijoueurs. Ce moteur propose des fonctionnalités avancées telles que le matchmaking et la communication entre les joueurs. Ainsi, nous avons initié le processus d'intégration des fondations de ce moteur dans Unity pour développer notre mode multijoueur.

## **4.5 Site Web**

Pour le développement du site Web, nous prévoyons d'intégrer des étapes supplémentaires au fil de l'avancement du jeu. Nous désactiverons temporairement les boutons de téléchargement qui ne sont pas encore opérationnels, mais qui seront rendus disponibles dès que les documents correspondants pourront être mis en ligne.

# **5 Conclusion**

Au cours de cette période, nous avons avancé de manière significative dans le développement de notre projet "LA GRANDE ÉVASION". Nous avons réussi à pallier au retard pris sur le mode multijoueur en commençant à développer notre Intelligence Artificielle. Quant à la programmation, à la modélisation 3D et au site Web, ils ont tous été développés dans le temps prévu. Nous avons d'autant plus élaboré un scénario captivant mettant en scène un groupe de prodiges cherchant à regagner leur liberté. Cependant, il reste encore quelques efforts à fournir pour finaliser notre jeu. Nous devons nous concentrer sur le mode multijoueur, l'intégration des mécaniques de jeu restantes, et l'amélioration de l'intelligence artificielle. En conclusion, bien que nous soyons satisfaits des progrès réalisés, nous restons concentrés et engagés pour mener à bien notre projet et livrer une expérience de jeu captivante.