

FIVE NIGHTS AT CHÂTELET



- PROMO 2030 -

**NEEV CHANDIRAMANI - MAXIMILIEN COCHET - GAUTHIER ROLAND - MAXIME
YE - MAXIME GONTRAN**

[HTTPS://FIVENIGHTSATCHATELET.NEEVCHANDIRAMANI.COM/](https://fivenightsatchatelet.neevchandiramani.com/)



TABLE DES MATIÈRES :

- I. Introduction
- II. Concept et déroulement du jeu
- III. Organisation du groupe
- IV. Répartition des tâches
- V. Site web
- VI. Modèles 3D
- VII. Conception et implémentation des mécaniques
- VIII. Multijoueur
- IX. L'interface utilisateur & Ambiance sonore
- X. Gestion de la caméra
- XI. Conclusion



I. Introduction

Five Nights At Châtelet est un jeu d'horreur et de survie inspiré de la célèbre franchise Five Nights at Freddy's.

Ce projet s'inscrit dans un cadre strictement scolaire et académique. Il a pour objectif la conception et la réalisation d'un jeu vidéo en équipe, permettant de mobiliser des compétences en conception fonctionnelle, organisation de projet et développement logiciel.

Le jeu Five Nights at Châtelet est un jeu d'horreur et de survie multijoueur en vue à la première personne, destiné à des joueurs occasionnels appréciant les ambiances angoissantes et les mécaniques simples mais immersives.



II. Concept et déroulement du jeu

Five Nights at Châtelet repose sur un gameplay **multijoueur asymétrique** en vue à la première personne. Les joueurs sont automatiquement répartis en deux camps aux objectifs opposés : les **Explorateurs**, qui doivent coopérer pour résoudre une énigme et s'échapper, et les **Infectés**, dont le but est de traquer les survivants.

Une partie s'articule autour des étapes suivantes :

- **Initialisation** : Création d'une session en ligne et répartition automatique des rôles.
- **Phase active** : Exploration et résolution d'énigmes dans une atmosphère oppressante sous contrainte de temps.
- **Dénouement** : La partie prend fin par l'évasion des explorateurs, l'infection totale du groupe, ou l'écoulement du temps imparti.



III. Organisation du groupe

Afin d'assurer une collaboration fluide et une progression constante du jeu **Five Nights At Châtelet**, notre studio **BambouX** a mis en place une organisation basée sur la communication constante et la répartition des tâches.

Gestion de la communication : La quasi intégralité des messages échangés sont échangés sur un canal de discussion sur la plateforme **Discord**.

Cette méthode nous permet de centraliser les informations afin que tous les membres du groupe soient à jour. Ce choix nous permet également de réaliser des sessions de travail régulières.

Cet outil est indispensable pour notre studio, il nous permet de résoudre rapidement les blocages techniques et de maintenir une cohésion d'équipe indispensable pour un projet de cette envergure.

Gestion du code source et versionnage : Pour le développement logiciel, nous utilisons **GitHub** comme plateforme de forge logicielle. L'usage de **Git** est essentiel pour centraliser notre code source Python, gérer les versions et permettre un travail collaboratif en parallèle sans risque de perte de données. Cette méthodologie nous offre une traçabilité complète des apports de chaque membre du groupe, conformément aux exigences de rigueur du projet SAE J3D.



IV. Répartition des tâches

La conception de *Five Nights At Châtelet* a été découpée en plusieurs pôles techniques afin de paralléliser le travail et de garantir que chaque membre apporte sa pierre à l'édifice. Nous avons défini pour chaque tâche un responsable principal (R) et un secondant (S) pour assurer la continuité du développement.

Pôle Graphisme et Environnement :

La cohérence visuelle est portée par **Gauthier Roland**, responsable du *Map design* et du *Character design*, soutenu par **Neev Chandiramani**. Ce pôle s'occupe de créer l'atmosphère oppressive du jeu à travers l'architecture des niveaux et l'apparence des entités.

Pôle Technique et Gameplay :

Ce pôle gère le "moteur" interne du jeu sous PyGame:

- **Déplacements et UI** : **Maxime Contran** dirige la gestion des mouvements et l'interface utilisateur (UI), avec l'appui de **Maximilien Cochet** et **Maxime Ye**.
- **Événements et interactions** : **Maximilien Cochet** est responsable de la logique des événements et du système de combat/interactions, secondé par **Maxime Ye** et **Maxime Contran**.



Pôle Multijoueur et Communication :

- **Réseau** : Élément obligatoire du projet, le multijoueur est piloté par **Neev Chandiramani** avec l'aide de **Maximilien Cochet**.
- **Site Web** : Le site internet, vitrine de notre projet, est géré par **Neev Chandiramani** et **Gauthier Roland**.

Pôle Audio :

L'immersion sonore, cruciale pour l'horreur, est confiée à **Maxime Ye**, responsable des sons, avec le soutien de **Maxime Gontran**.

Tâches	Login1	Login2
Map design	(R) gauthier.roland@epita.fr	(S) neev.chandiramani@epita.fr
Character design	(R) gauthier.roland@epita.fr	(S) neev.chandiramani@epita.fr
Multijoueur	(R) neev.chandiramani@epita.fr	(S) maximilien.cochet@epita.fr
Sons	(R) maxime.ye@epita.fr	(S) maxime.gontran@epita.fr
Déplacement / Mouvement	(R) maxime.gontran@epita.fr	(S) maximilien.cochet@epita.fr
Gestion des événements	(R) maximilien.cochet@epita.fr	(S) maxime.ye@epita.fr
Site Web	(R) neev.chandiramani@epita.fr	(S) gauthier.roland@epita.fr
UI	(R) maxime.gontran@epita.fr	(S) maxime.ye@epita.fr
Combats / Interactions	(R) maximilien.cochet@epita.fr	(S) maxime.gontran@epita.fr

cf cahier des spécifications techniques

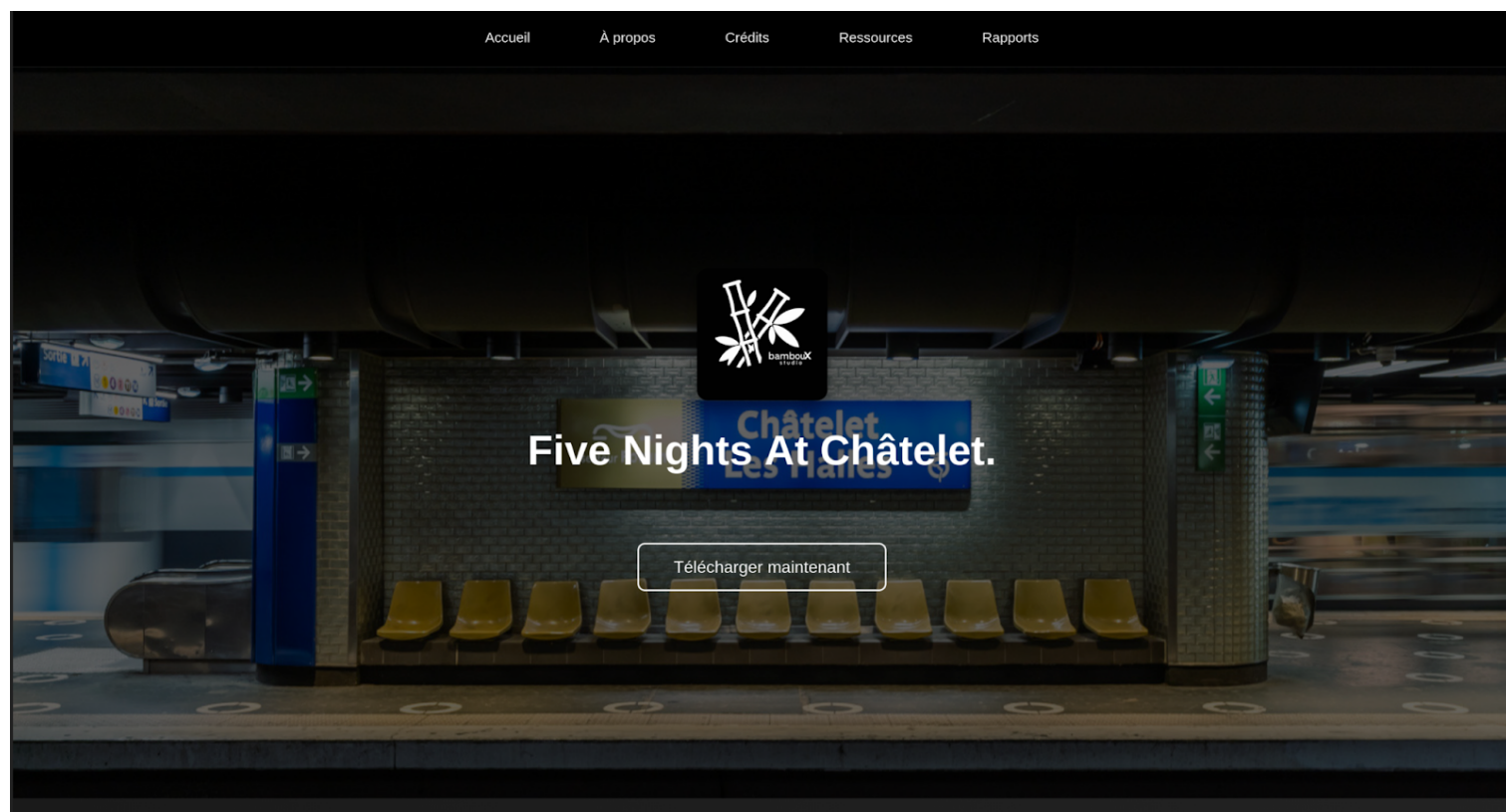


V. Site web

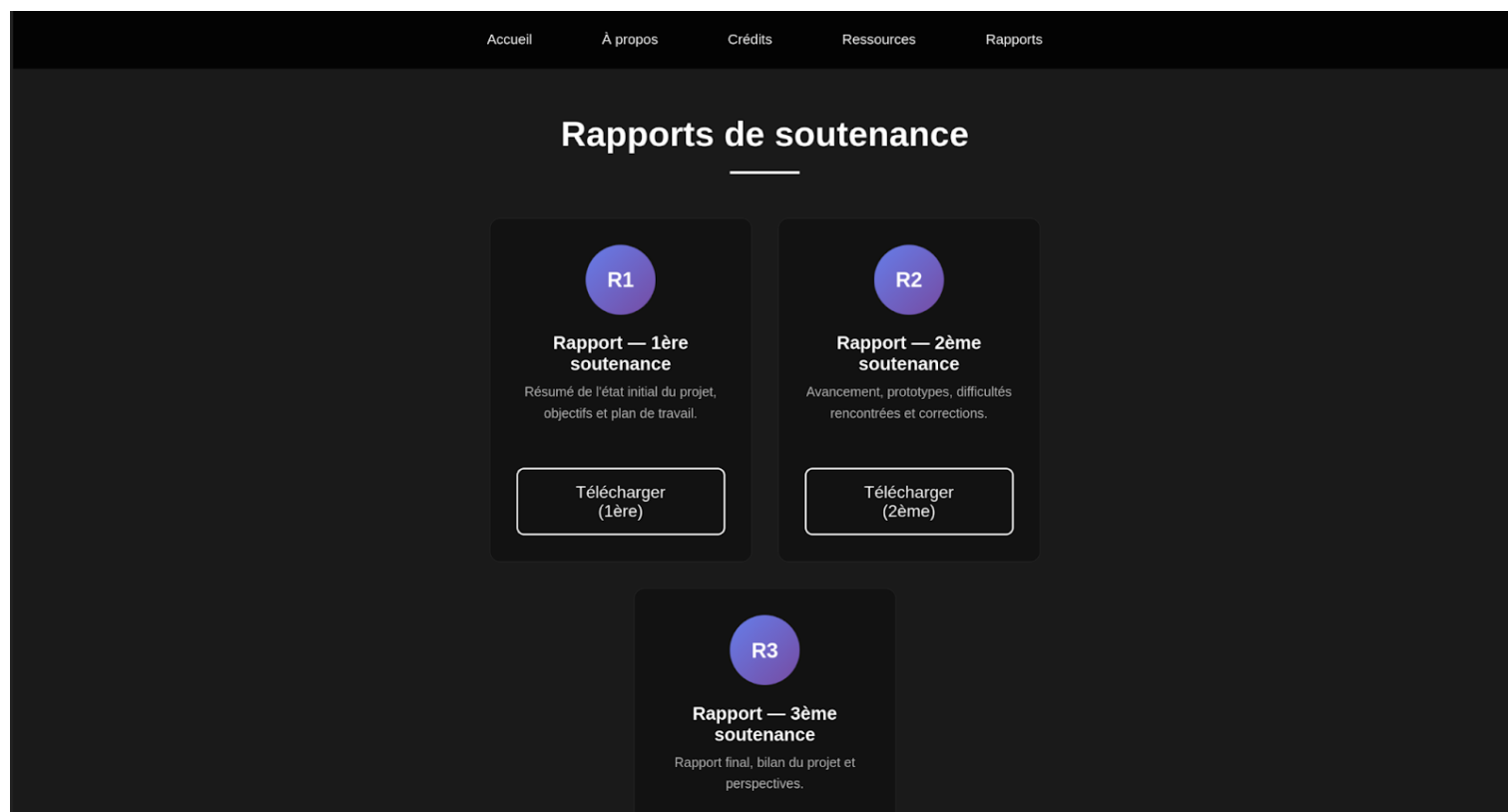
Conformément aux exigences du cahier des charges, nous avons développé un site internet qui fait office de vitrine et de centre de ressources pour *Five Nights At Châtelet*. Ce support de communication, mis à jour à chaque étape du projet, remplit plusieurs fonctions essentielles :

- **Présentation** : Le site propose une description détaillée du concept de jeu ainsi qu'une présentation de notre équipe et de ses membres.
- **Centre de documentation** : Une section dédiée regroupe l'ensemble des ressources externes mobilisées, incluant les références bibliographiques, les bibliothèques logicielles utilisées ainsi que les banques d'assets sonores et graphiques.
- **Plateforme de téléchargement** : Le site offre un accès direct aux livrables numériques, permettant notamment de récupérer le rapport de soutenance et les dernières versions du projet.

Le site web est accessible à l'adresse : <https://fivenightsatchatelet.neevchandiramani.com/>



Page d'accueil du site web



Page de téléchargement des rapports du site web



À propos

Le Jeu

Five Nights At Châtelet est un jeu d'horreur et de survie inspiré de la célèbre franchise Five Nights at Freddy's. Ce projet s'inscrit dans un cadre strictement scolaire et académique. Il a pour objectif la conception et la réalisation d'un jeu vidéo en équipe, permettant de mobiliser des compétences en conception fonctionnelle, organisation de projet et développement logiciel. Le jeu Five Nights at Châtelet est un jeu d'horreur et de survie multijoueur en vue à la première personne, destiné à des joueurs occasionnels appréciant les ambiances angoissantes et les mécaniques simples mais immersives.

L'Équipe

Une équipe d'étudiants passionnés par le jeu vidéo, le design et l'audio, réunie pour concevoir ce prototype académique et expérimenter des mécaniques immersives.

NC

Neev Chandiramani
Développeur

Créateur et développeur principal
de Five Nights At Châtelet.

MC

Maximilien Cochet
Développeur

Participe au développement et à
l'implémentation des mécaniques

début de la page "à propos" du site web

VI. Modèles 3D

Dans le projet, nous nous occupons de tout l'**aspect graphique** du jeu : la création de la map, des personnages, des animations, mais aussi de leur **implémentation directement en Python**.

Pour l'instant, nous avons principalement travaillé sur les **personnages**. Nous en avons créé deux, qui représentent les deux équipes du jeu : un **explorateur** et un **infecté**. Le jeu a une ambiance volontairement horripilante, qui se déroule dans un centre commercial. Pourtant, nous avons fait le choix d'un **style graphique cartoon**, avec des grosses têtes et de grands yeux un peu amusants. L'objectif était de créer un **contraste** entre le visuel des personnages et l'ambiance oppressante du jeu, pour rendre l'ensemble plus marquant et original.



modèle 3D d'un "explorateur"



modèle 3D d'un "infecté"



Processus de création 3D sur Blender

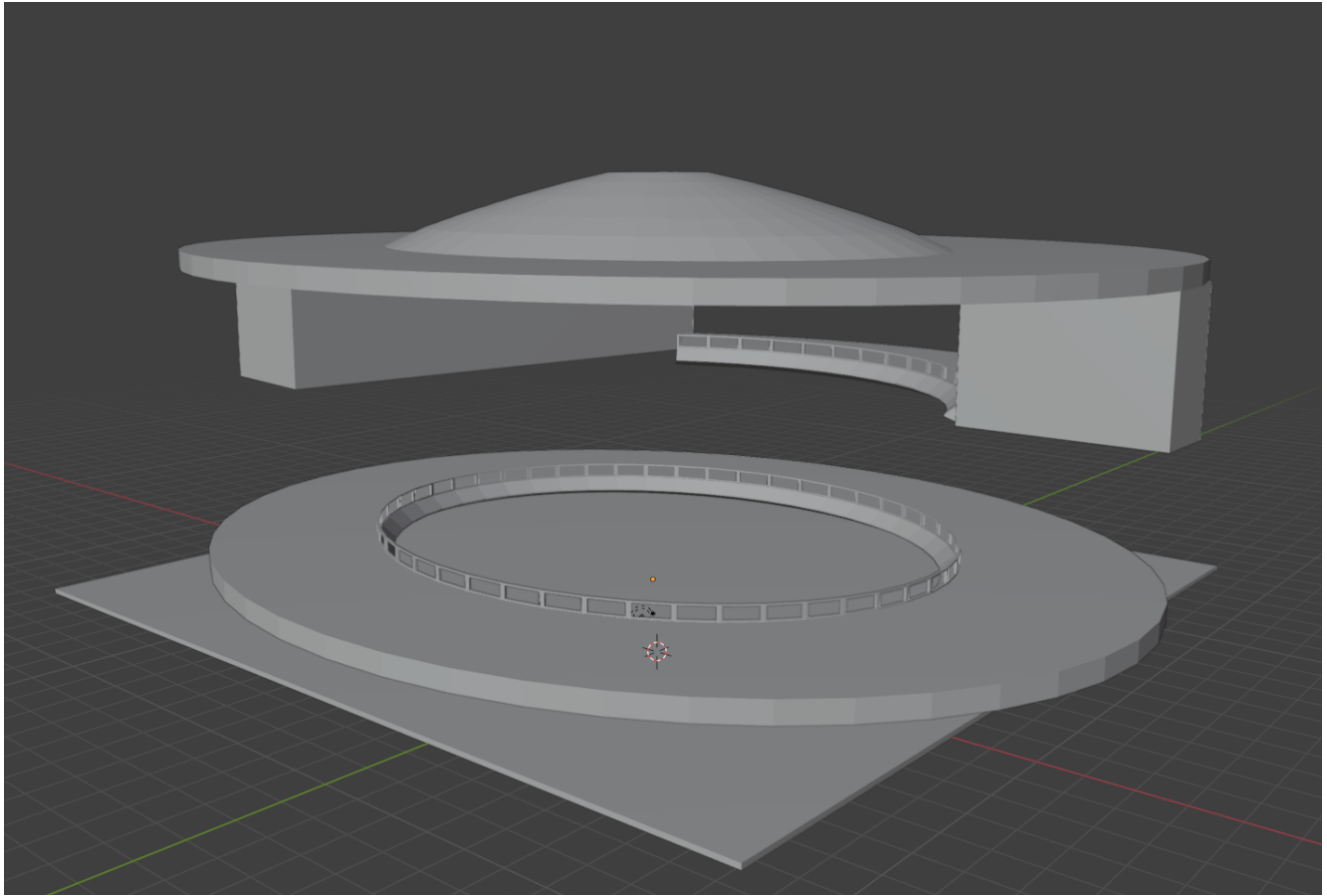
Avant de commencer la modélisation, nous avons passé beaucoup de temps sur la phase de design. Nous avons effectué des recherches et testé plusieurs idées jusqu'à arriver à une direction visuelle cohérente.

Une fois les idées définies, nous sommes passés à la modélisation 3D sur Blender, en veillant à ce que les personnages soient expressifs et adaptés au style du jeu.

Développement de l'environnement (Map)

En parallèle, nous avons aussi commencé à travailler sur la **map**. Nous avons réalisé un **premier squelette du centre commercial**, en délimitant les différents étages et leur organisation.

Le but était d'avoir une **vision globale de la structure** avant d'entrer dans les détails. Nous avons ensuite commencé à écrire le code permettant d'**implémenter cette map en Python**, pour qu'elle soit fonctionnelle et exploitable pour le gameplay.



modèle 3D de la map

Perspectives et objectifs

Pour la suite, nous prévoyons d'affiner la map en ajoutant des **détails visuels** et des **éléments interactifs**, de continuer les **animations des personnages**, et de créer d'autres modèles si besoin. L'objectif est d'avoir un environnement et des personnages **cohérents visuellement et techniquement**.



VII. Conception et implémentation des mécaniques

Gestion du sprint et de l'endurance

Nous avons mis en place une **mécanique de sprint** accompagnée d'un système d'**endurance (stamina)**. Lorsque le joueur maintient la touche Shift, la vitesse de déplacement augmente, mais l'endurance diminue progressivement. Une fois l'endurance épuisée, le sprint est automatiquement désactivé. Lorsque le joueur ne sprinte plus, l'endurance se régénère progressivement. Un **affichage à l'écran (HUD)** permet au joueur de visualiser en temps réel son niveau d'endurance. Cette mécanique ajoute une **dimension stratégique** au gameplay en empêchant le sprint illimité.

Système de saut et gestion de la gravité

Nous avons implémenté un **système de saut manuel** sans utiliser de contrôleur prédéfini. Lorsque le joueur appuie sur la touche Espace, une **force verticale** est appliquée au personnage. La **gravité** est ensuite simulée par une accélération constante vers le bas. Une **détection du sol** empêche le joueur de sauter plusieurs fois sans avoir atterri. Ce système permet d'obtenir un comportement de saut réaliste et contrôlé.



Interaction avec l'environnement

Nous avons développé une **mécanique d'interaction** avec les objets proches du joueur. Une distance d'interaction est définie : lorsque le joueur se trouve suffisamment près d'un objet interactif et appuie sur la touche E, une **interface visuelle** apparaît à l'écran. Si le joueur s'éloigne de l'objet, l'interface disparaît automatiquement. Ce système peut être réutilisé pour différents types d'interactions comme des **portes, des boutons ou des objets à activer**.

Caméra et orientation du personnage

Nous avons travaillé sur la **gestion de la caméra à la souris** et sur la **synchronisation** entre la direction de la caméra et l'orientation du personnage. La caméra est attachée à un pivot lié au joueur, et les mouvements de la souris modifient la rotation de ce pivot. La **rotation horizontale** est ensuite appliquée au personnage afin qu'il regarde dans la même direction que la caméra. Cette fonctionnalité est fonctionnelle mais encore perfectible : nous prévoyons d'améliorer la fluidité via un système de **lissage ou d'interpolation**.



VIII. Multijoueur

Le projet repose sur une architecture **client-serveur** destinée à implémenter un mode multijoueur en **réseau**.

Un serveur dédié, hébergé sur un VPS (de l'association La Banquise), est chargé de la gestion des connexions et de la synchronisation de l'état du jeu entre les différents clients.

Il fonctionne de manière indépendante du moteur graphique et utilise des sockets **TCP** pour recevoir les données envoyées par les clients et leur renvoyer l'état global du jeu.

Chaque client, exécuté localement sur l'ordinateur du joueur, s'appuie sur le moteur Ursina pour le rendu graphique et la gestion des interactions.

À intervalles réguliers, le client transmet au serveur les informations relatives à son joueur, telles que sa position dans l'espace de jeu.

Le serveur centralise ces données, les maintient à jour pour l'ensemble des joueurs connectés, puis les redistribue afin d'assurer la synchronisation des entités visibles dans chaque instance du jeu.

Cette séparation stricte entre logique réseau et rendu graphique permet d'améliorer la modularité du projet, de limiter les dépendances et de faciliter les évolutions futures, notamment l'optimisation des échanges réseau ou l'ajout de nouvelles mécaniques multijoueur.



Pour le moment, l'aspect réseau et multijoueur n'est pas totalement fonctionnel.

A ce stade du projet, nous testons l'aspect réseau sur des "jeux" beaucoup plus simples, par exemple uniquement des cubes qui bougent en 3D. Nous nous concentrons sur l'aspect technique du réseau afin de le rendre fonctionnel sur un jeu simple pour ensuite déployer le multijoueur sur *Five Nights at Châtelet*.



IX. L'interface utilisateur & Ambiance sonore

1. Identité visuelle et sonore du menu

Nous nous sommes concentrés sur la création d'une **identité forte** pour notre jeu *Five Nights at Châtelet*. Notre travail a débuté par la **conception intégrale du menu principal** sous PyGame. Ce travail inclut :

- **Recherche et intégration visuelle** : Nous avons personnellement recherché et inclus une image de fond cohérente avec l'ambiance du métro parisien, sur laquelle nous avons appliqué un **voile sombre** pour garantir la lisibilité des options tout en installant immédiatement l'ambiance obscure et pesante recherchée pour un jeu d'horreur.
- **Identité visuelle** : Nous avons développé un rendu visuel de type **caméra de surveillance** en superposant un effet de « **bruit visuel** » (grain/neige) et un **balayage CRT** (*Cathode Ray Tube*) composé de lignes horizontales pour coller à l'univers horrifique.
- **Effets dynamiques** : Nous avons également intégré des effets de **glitches aléatoires** sur les textes pour donner une impression d'instabilité technique. Cette instabilité suggère un matériel défaillant ou une présence paranormale, renforçant l'immersion dès le menu principal.

- **Ambiance sonore** : Nous avons recherché et intégré une musique de fond spécifique. Cette piste audio a été choisie pour sa capacité à instaurer une **tension immédiate**.



image de fond du menu : <https://bit.ly/45vpptd>

2. Ergonomie et navigation hybride

Au niveau de l'ergonomie, nous avons mis en place une **navigation fluide** pour accompagner l'utilisateur dans la prise en main :

- **Navigation hybride** : Nous avons programmé l'interface pour qu'elle réponde de manière fluide aux



entrées du clavier via les **flèches directionnelles** et à la **souris**.

- **Options du menu** : Cela permet de sélectionner intuitivement les trois options principales : **Start**, **Options** et **Exit**.



image du menu

3. Recherche et intégration sonore

Le son étant un pilier du genre survival-horror, nous avons entamé un travail de fond sur l'environnement sonore du jeu :

- **Sélection audio** : Nous avons effectué des recherches approfondies pour identifier, sélectionner et inclure une **musique de fond oppressante** ainsi que des **sons**



d'ambiance pesants qui correspondent à l'atmosphère de Châtelet version horreur.

- **Contrôle interactif** : Pour offrir un confort optimal au joueur, nous avons conçu et relié une **barre de volume interactive** directement au mixeur audio de PyGame, permettant de configurer le son en temps réel.
- **Planification** : La liste des effets sonores liés aux futures actions de jeu (déplacements des infectés, ouverture des portes) est en cours de développement.

cf musique de fond : <https://freesound.org/people/Xanco123/sounds/233311/>



X. Gestion de la caméra

Point de vue technique :

Dans un premier temps, on essaie d'importer la fonction `run_menu` du fichier `menu` avec le `try`.

Si tout se passe bien, la fonction `run_menu` est détectée et est utilisable.

Cependant si une erreur survient lors de la connexion avec le fichier `menu` alors on stock l'erreur dans la variable `'e'` de la ligne `except`. Elle est ensuite affichée à la fin du message d'erreur dans la console.

Avec le `if`, on teste ce qu'il s'est passé au-dessus. Si il y'a eu une erreur alors le `if` sera ignoré, par conséquent, le menu ne se lancera pas. Si la fonction `run_menu` a bien été trouvée alors la variable `run_menu` est égale à `True`.

On entre donc dans le `if`. Ensuite on teste le cas où le joueur quitte le menu en appuyant sur le bouton `'exit'`, si le joueur active l'`exit` alors la fonction `run_menu` retourne `'exit'` ce qui est bien différent de `start` ainsi le jeu se ferme.

Si le joueur appuie sur `'start'` alors la valeur de `run_menu` est égale à `'start'` on ignore donc le deuxième `if` et l'application se lance. Un écran de chargement s'affiche pour que la transition de la fenêtre menu à la fenêtre du jeu ne soit pas visible.



```
try:
    from Menu import run_menu
except Exception as e:
    run_menu = None
    print("Warning: impossible d'importer `Menu.run_menu()` : ", e)

if run_menu:
    result = run_menu()
    if result != 'start':
        sys.exit()
```

exemple de code du jeu

Ensuite, on initialise les premiers éléments du jeu. Nous avons pour l'instant un cube test de couleur rouge et de côté 3 représentant le joueur.

On initialise un autre cube test décoratif de couleur bleu pour tester la fonction de déplacement et de caméra.

```
joueur = Entity(
    position= (0,5,0),
    model = 'cube',
    color = color.red,
    scale_y = 3,
    collider = 'box'
)

test_cube = Entity(
    model='cube',
    color=color.azure,
    position=(2,1,2),
    shader=lit_with_shadows_shader
)
```

exemple de code du jeu



Lorsque le joueur est dans la partie, le mouvement de la souris est bloqué dans la fenêtre.

On rend aussi le curseur invisible. Il est important de noter que cela marche uniquement avec la souris, le pavé tactile fonctionne différemment et est buggé avec ce système.

Ensuite on crée le point de rotation de la caméra. Ici, il est accroché au joueur et est à hauteur 2 au-dessus du sol.

L'attribut Camera.parent sert à faire en sorte que la caméra suive le joueur tout le temps. On oriente ensuite la caméra vers le bas de 15° pour qu'elle soit adaptée au cube en 3ème personne.

```
mouse.locked = True
mouse.visible = False

camera_pivot = Entity(parent=joueur, y=2)
camera.parent = camera_pivot
camera.fov = 90
camera.rotation = (15, 0, 0)
```

exemple de code du jeu

On crée une fonction pour gérer le mouvement de la caméra entre chaque frame.

Velocity[0] représente l'axe y (haut/bas) et velocity[1] représente l'axe x (gauche/droite). On multiplie par 80 pour représenter la sensibilité de la souris.

La méthode clamp sert à forcer la caméra à rester entre l'intervalle [-30,45] sur l'axe y. Le joueur ne peut pas regarder trop vers le ciel ni trop vers le sol



Et enfin, on place la caméra derrière le personnage avec la position à -5 grâce à l'attribut `camera.position`.

Grâce à ursina, il n'est pas compliqué de passer d'une caméra fps à une caméra 3ème personne.

Il suffit de changer la position de la caméra ainsi que la rotation et la fov.

```
def mouvement_camera():  
    camera_pivot.rotation_y += mouse.velocity[0] * 80  
    camera_pivot.rotation_x -= mouse.velocity[1] * 80  
    camera_pivot.rotation_x = clamp(camera_pivot.rotation_x, -30, 45)  
    camera.position = (0, 0, -5)
```

exemple de code du jeu

On doit ensuite appeler cette fonction dans la fonction `update`. Grâce à `update`, on recalcule à chaque frame l'orientation de la caméra. Sans ça, la caméra ne bougerait pas.

```
def update():  
    mouvement_joueur()  
    mouvement_camera()
```

exemple de code du jeu

Pour finir, on initialise une fonction `input` qui gèrera toutes les actions sur le clavier effectué par le joueur.



Pour l'instant seulement la touche échap peut être utilisée (autre que les touches de déplacement) pour quitter le jeu.

```
def input(key):  
    if key == 'escape':  
        application.quit()
```

exemple de code du jeu

Améliorations / changement prévu sur le code présenté ci-dessus :

- Gravité pour que le cube test du joueur ne flotte pas mais tombe sur le sol sans le traverser.
- Création d'une fenêtre pour que lorsque la touche échap est pressée une petite fenêtre s'ouvre pour changer les paramètres, revenir au menu en quittant la partie.

Il est possible que nous modifions le fonctionnement de la caméra en fonction de l'évolution des mécaniques de déplacement des personnages.



XI. Conclusion

Le projet *Five Nights at Châtelet* représente une aventure collaborative ambitieuse menée par notre équipe BambouX, composée de Neev Chandiramani, Maximilien Cochet, Gauthier Roland, Maxime Ye et Maxime Gontran. Chacun a apporté ses compétences techniques, sa créativité et son investissement personnel afin de poser les bases d'un jeu d'horreur multijoueur immersif, encore en cours de développement mais déjà fonctionnel sur de nombreux aspects.

L'une des principales forces de notre groupe réside dans la complémentarité de nos profils. Gauthier Roland a joué un rôle central dans la direction artistique du projet, en concevant les modèles 3D, les personnages et l'environnement du jeu, contribuant fortement à l'identité visuelle particulière de *Five Nights at Châtelet*. De son côté, Neev Chandiramani s'est investi dans les aspects réseau et multijoueur, ainsi que dans la création et la gestion du site web, garantissant à la fois la dimension technique du projet et sa visibilité extérieure.

Maxime Gontran s'est concentré sur les déplacements du joueur, l'interface utilisateur et certaines mécaniques essentielles du gameplay, tandis que Maximilien Cochet a travaillé sur la logique des événements et des interactions, assurant la cohérence du fonctionnement global du jeu. Enfin, Maxime Ye a apporté une contribution importante à l'ambiance sonore, élément clé de l'expérience de jeu, tout en participant aux aspects techniques du projet.

Malgré les difficultés rencontrées, notamment liées à la complexité du multijoueur et aux contraintes de temps, nous avons su maintenir une communication efficace et une organisation rigoureuse, en nous appuyant sur des outils



collaboratifs adaptés. Ce premier rapport de soutenance reflète ainsi une phase de structuration, d'expérimentation et de mise en place des fondations techniques du projet.

Five Nights at Châtelet n'en est encore qu'à une étape intermédiaire de son développement. Les bases posées au cours de cette première phase nous permettent désormais d'envisager la suite du projet avec une vision plus claire, en poursuivant l'amélioration des mécaniques, du multijoueur et de l'immersion globale. Ce travail marque une progression importante dans notre apprentissage et constitue un socle solide pour les prochaines étapes du développement.
