

Duck Adventures

Par
Les Canards
Sauvages

Équipe projet :

Sarah François, Elouan Coleno
Hugo Lefay, Arthur Raillon

Contents

1 Introduction	2
2 Présentation des membres du groupe	2
2.1 Arthur Raillon	2
2.2 Sarah François	3
2.3 Hugo Lefay	3
2.4 Elouan Coleno	3
3 État de l'art	4
4 Choix techniques	4
4.1 Moteur graphique	4
4.2 Outil de dessin	5
4.3 EDI C#	5
4.4 EDI Web	5
4.5 Outil de création de l'installateur	6
4.6 Serveurs multijoueur	6
4.7 Coordination, hébergement web	7
5 Avancement, obstacles rencontrés et résolution des problèmes	8
5.1 Avant la première soutenance	8
5.1.1 Arthur Raillon	8
5.1.2 Sarah François	11
5.1.3 Hugo Lefay	12
5.1.4 Elouan Coleno	13
5.2 Avant la deuxième soutenance	15
6 Mise en relation avec le diagramme de Gantt	18
6.1 Avancements selon le diagramme de Gantt	18
7 Conclusion	20

Introduction

Les Canards Sauvages est une entreprise fondée en 2024 par Arthur Raillon, Sarah François, Hugo Lefay et Elouan Colleno, 4 étudiants de l'EPITA. Peu après la création, l'entreprise annonce la sortie de son produit phare en mai 2025 : le jeu vidéo *Duck Adventures*, un jeu d'aventure en vue 2D isométrique, mettant en scène Hogu, un jeune canard rejeté par sa famille, dans un monde qui semble paisible. Dans ce rapport, après une présentation de chaque membre ainsi que de son rôle dans l'équipe, nous présenterons les choix techniques que nous avons faits afin de mener à bien notre projet. Ensuite, nous aborderons les avancements de chaque membre, les obstacles rencontrés par chacun, et enfin, les moyens mis en place pour surmonter ces difficultés.

Présentation des membres du groupe



Figure 1: L'équipe de Duck Adventures

Arthur Raillon

Chef de projet

Développement IA

Jeu

Arthur, passionné de cybersécurité et de red teaming, a découvert ces domaines pendant la période du COVID, ce qui l'a conduit à maîtriser les aspects techniques du développement du jeu. Il fait en sorte que

le jeu fonctionne. Son intérêt pour l'intelligence artificielle l'a poussé à intégrer cette technologie dans le jeu, en optimisant les comportements des personnages. Grâce à ses qualités de leader et à ses soft-skills, il a su fédérer une équipe autour d'une vision claire, tout en favorisant la collaboration et l'innovation.

Sarah François

Leader développement graphique

Venant d'une famille d'informaticiens et de nature très curieuse, Sarah a été introduite tôt à la programmation. Passionnée de jeux vidéo et d'art, elle réunit ses passions dans cette entreprise, où elle met en pratique ses techniques et son imagination sans limite dans la création du jeu. Elle est responsable des dessins des personnages, des paysages et des animations.

Hugo Lefay

Développement multijoueur

Passionné par l'informatique, domaine qui stimule sa curiosité et sa soif d'apprendre, Hugo est toujours prêt à s'investir pleinement. Il met toute son énergie et son engagement dans chaque projet qu'il entreprend. Curieux de découvrir de nouvelles technologies, il explore sans cesse de nouvelles idées pour élargir ses compétences, notamment dans le développement multijoueur et l'intégration des serveurs.

Elouan Coleno

Responsable de l'histoire

Développement du jeu

Depuis son plus jeune âge, Elouan est passionné de jeux vidéo et d'histoires en tous genres. Il a découvert l'informatique pendant la période du COVID, domaine qui l'a tout de suite intéressé. Ce projet lui permet donc d'allier ses passions, afin de créer le monde d'Hogu, doté de canards dont les ailes se transforment en lance-missiles. Il est également responsable de l'écriture du lore et de la géopolitique du monde du jeu.

État de l'art

Avant de débuter la création de notre projet, nous avons écumé Internet afin de trouver des éléments existants sur lesquels nous appuyer, et d'identifier les éléments à créer à partir de zéro.

Nous avons donc trouvé de nombreuses sources :

- **Dessin original** : Mimi Panda
- **HTML/CSS/JS** : Mozilla Developer
- **Installeur** : TUTO UNITY FR
- **Unity** :
 - Lawless Games
 - Tamara Makes Games
 - TUTO UNITY FR

Choix techniques

Afin de mener à bien notre projet, nous avons dû décomposer notre projet en différentes tâches, et choisir des outils spécifiques pour mener à bien chacune de ces tâches.

Ainsi, nous avons fait différents choix techniques :

Moteur graphique

L'outil principal que nous utilisons pour développer notre jeu est le moteur de jeu Unity. Cet outil est l'un des deux moteurs les plus utilisés dans l'industrie du jeu vidéo, aussi bien par les studios spécialisés que par les amateurs et les créateurs de jeux dits "indépendants".

Nous avons donc choisi Unity car c'est un moteur universellement reconnu et largement utilisé. De plus, la prise en main de cet outil est facile, c'est d'ailleurs un des aspects qui a mené à sa popularité. Enfin, Unity étant très populaire, de nombreux articles, vidéos et tutoriels de documentation sur l'outil sont disponibles.

Outil de dessin

L'intégralité des personnages, des pièces d'équipements et, de manière générale, la majorité des “assets” que nous utilisons, sont dessinés à la main par Sarah, notre graphiste. En conséquence, un logiciel de dessin a été nécessaire pour la réalisation de ces “assets”. Sarah a donc opté pour le logiciel Ibis Paint, un outil de dessin sur écran auquel elle était déjà habituée, et qui lui a donc permis une prise en main rapide des fonctions avancées.

EDI C#

Le moteur Unity est complet et permet, entre autres, d'importer des “assets” dans le jeu afin de créer les zones de jeu, par exemple. Cependant, un moteur seul ne permet pas de faire un jeu aussi complet et personnel que ce que souhaite l'équipe en charge de *Duck Adventures*. En effet, les fonctionnalités propres à notre jeu nécessitent d'être développées en langage C#.

Pour mener à bien cette tâche, les développeurs utilisent des EDI (Environnements de Développement Intégrés), qui sont des logiciels permettant d'écrire et d'exécuter du code de manière pratique.

Pour le C#, nous avons choisi deux EDI différents : d'une part, Visual Studio Code (abrégé VsCode), un EDI versatile permettant de coder en tous langages, et d'autre part, JetBrains Rider, un EDI spécialisé en C# nécessitant un travail de prise en main plus important.

VsCode a été choisi pour sa versatilité et sa facilité de prise en main. De plus, certains membres utilisaient déjà cet outil, ils sont donc restés dessus.

Les autres membres ont plutôt opté pour JetBrains Rider, un outil spécialisé pour le développement en C#. Cet outil est payant mais l'EPITA fournit à ses élèves une licence JetBrains, ce qui leur permet d'utiliser ce produit gratuitement.

EDI Web

Pour promouvoir son jeu, et permettre le téléchargement de ce dernier aux joueurs, Les Canards Sauvages a décidé de créer un site web. Pour ce

faire, l'équipe a utilisé le langage de balisage HTML, couplé avec le CSS et le JavaScript, trois langages informatiques utilisés pour la création de pages web.

Afin de créer ce site, nous avons utilisé l'EDI VsCode, qui permet une prise en main très simple et qui, par ailleurs, était déjà utilisé par l'intégralité des membres.

Outil de création de l'installateur

Un installateur est un programme servant à installer facilement un logiciel sur un ordinateur. L'équipe a donc créé un installateur pour le jeu *Duck Adventures*.

Pour mener à bien cette tâche, l'équipe a utilisé un logiciel tiers, Inno Setup. Ce logiciel a été choisi pour trois raisons majeures : premièrement, ce logiciel est largement reconnu comme une référence dans la création d'installateurs. Deuxièmement, conséquence du premier point, la documentation sur le logiciel est nombreuse et très fournie. Enfin, la prise en main de ce logiciel est relativement simple.

```

[Name: "desktopicon"; Description: "{cm>CreateDesktopIcon}"; GroupDescription: "{cm\AdditionalIcons}"; Flags: unchecked]

[Files]
; Fichier principal (l'exécutable)
Source: "C:\Users\hugo\Desktop\unity projet\build\{#MyAppExeName}"; DestDir: "{app}"; Flags: ignoreversion
; UnityPlayer.dll (essentiel pour Unity)
Source: "C:\Users\hugo\Desktop\unity projet\build\UnityPlayer.dll"; DestDir: "{app}"; Flags: ignoreversion
; UnityCrashHandler64.exe (généré par Unity)
Source: "C:\Users\hugo\Desktop\unity projet\build\UnityCrashHandler64.exe"; DestDir: "{app}"; Flags: ignoreversion
; Dossier _Data (données du jeu)
Source: "C:\Users\hugo\Desktop\unity projet\build\My project (2)_Data\*"; DestDir: "{app}\My project (2)_Data"; Flags: ignoreversion recurseallsubdirs createallsubdirs
; Dossier MonoBleedingEdge (éci généré par Unity)
Source: "C:\Users\hugo\Desktop\unity projet\build\MonoBleedingEdge\*"; DestDir: "{app}\MonoBleedingEdge"; Flags: ignoreversion recurseallsubdirs createallsubdirs

[Registry]
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocExt}\OpenWithProgids"; ValueType: string; ValueName: "{#MyAppAssocKey}"; ValueData: ""; Flags: uninsdeletevalue
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocExt}\ValueTypes"; ValueType: string; ValueName: "{#MyAppAssocKey}"; ValueData: "ValueTypes"; Flags: uninsdeletekey
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocExt}\DevsupportedTypes"; ValueType: string; ValueName: "DevsupportedTypes"; ValueData: "DevsupportedTypes"; Flags: uninsdeletevalue
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocKey}\Label1\open\command"; ValueType: string; ValueName: ""; ValueData: "{app}\{#MyAppExeName}"; Flags: uninsdeletevalue
Root: HKA; Subkey: "Software\Classes\{#MyAppAssocExt}\SupportedFileTypes"; ValueType: string; ValueName: ".mvp"; ValueData: ""

[Icons]
Name: "{autoprograms}\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"
Name: "(autodesktop)\{#MyAppName}"; Filename: "{app}\{#MyAppExeName}"; Tasks: desktopicon

[Run]
Filename: "{app}\{#MyAppExeName}"; Description: "{cm\LaunchProgram,{#StringChange(MyAppName, '\ ', '\4')}}"; Flags: nowait postinstall skipifsilent

```

Figure 2: Exemple de code Inno Setup

Serveurs multijoueur

Notre jeu ayant un mode multijoueur, nous avons eu besoin d'un logiciel pour permettre l'hébergement des parties multijoueur. Pour ce faire, nous avons choisi le logiciel Photon Engine, pour sa simplicité d'utilisation et la facilité de mise en relation entre Photon Engine et Unity.

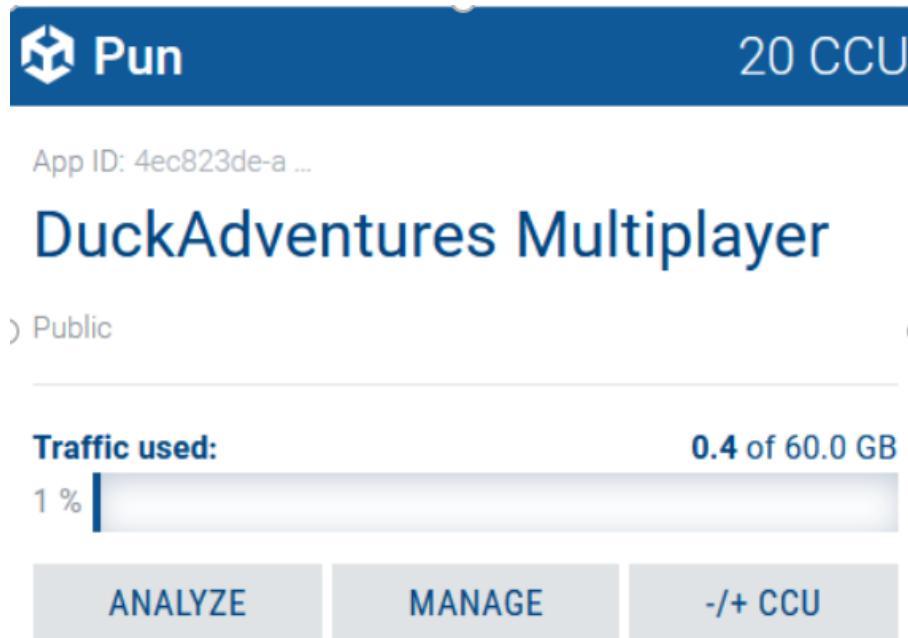


Figure 3: Capture d'écran du lobby multijoueur

Coordination, hébergement web

Afin de pouvoir avoir un travail coordonné et de synchroniser facilement toutes nos avancées, nous avons utilisé le service GitHub et l'outil Git Bash, qui nous permettent d'héberger notre site Web et notre projet, et permettent à l'équipe de travailler sur la même version sans difficulté de synchronisation.

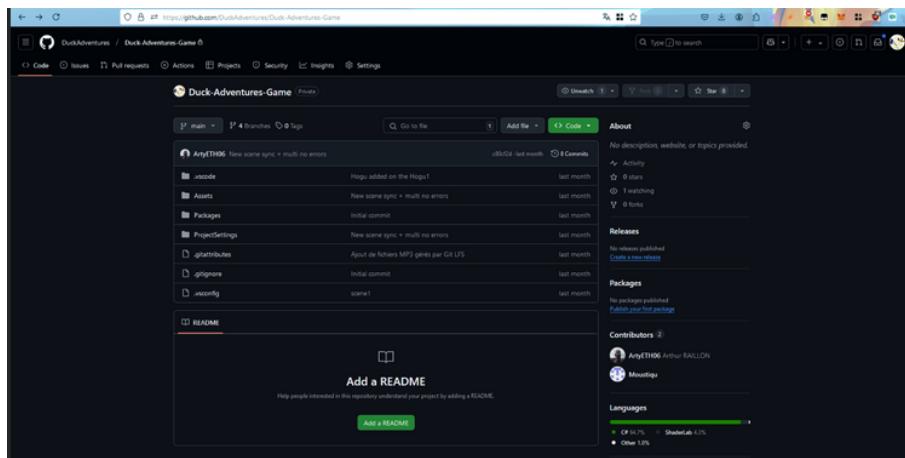


Figure 4: Interface GitHub pour la gestion de projet

Avancement, obstacles rencontrés et résolution des problèmes

Avant la première soutenance

Arthur Raillon

Arthur a principalement travaillé sur le développement du jeu, solo comme multijoueur avec Unity. Initialement, le partage du code était compliqué, mais la mise en place de GitHub a résolu ce problème. Il a également aidé à l'hébergement avec GitHub grâce à son expérience précédente ainsi que pour les différents montages vidéos.

Outils utilisés :

- Unity
- VsCode (Éditeur de code)
- GitHub / GitHub Desktop

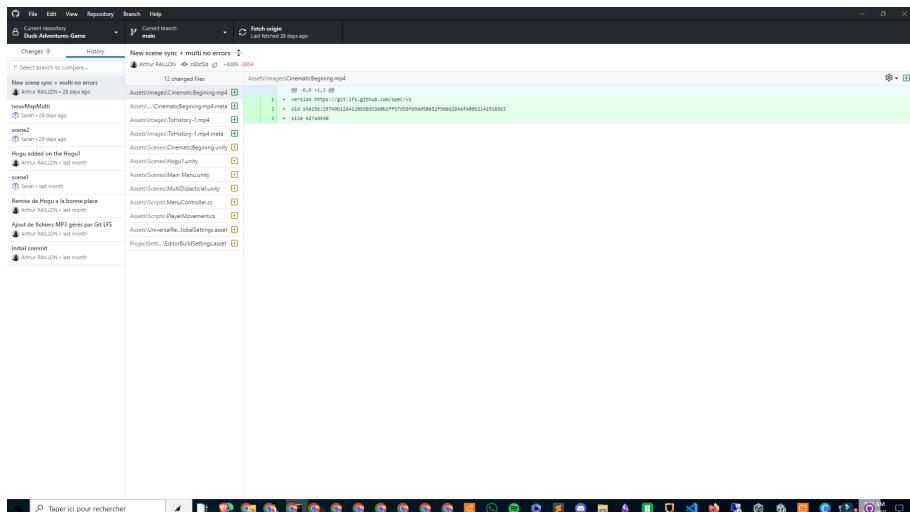


Figure 5: GitHub Desktop

- Filmora pour montage vidéo (Expérience précédente)



Figure 6: Montage Vidéo avec Filmora

- HTML, CSS, JS, C# (Expérience précédente et scolaire)
- Photon Engine / Fusion (Hébergement multijoueur gratuit, latence)

Prises en main et créations :

- Prise en main de Unity
- Création d'animations

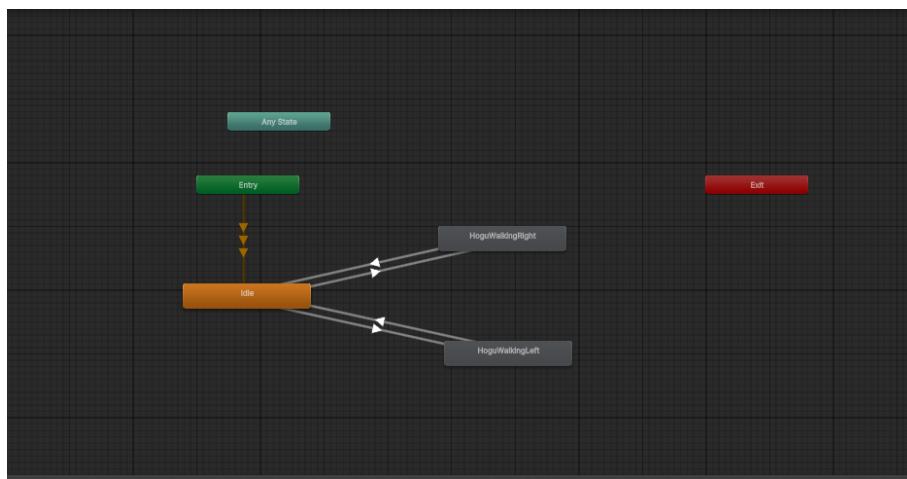


Figure 7: Animations créées

- Montage vidéo avec Filmora
- Création des boîtes de dialogues, recréation de systèmes connus

- Gestion des déplacements des joueurs en mode solo et multijoueur avec synchronisation
- Gestion de la caméra pour suivre le joueur

Problèmes rencontrés :

- Gestion de gros fichiers avec Git LFS (Large File Storage)
- Intégration du personnage en 2D pour un rendu isométrique



Figure 8: Intégration de Hogo sur la tilemap isométrique

- Utilisation de nombreux scripts pour des tâches précises

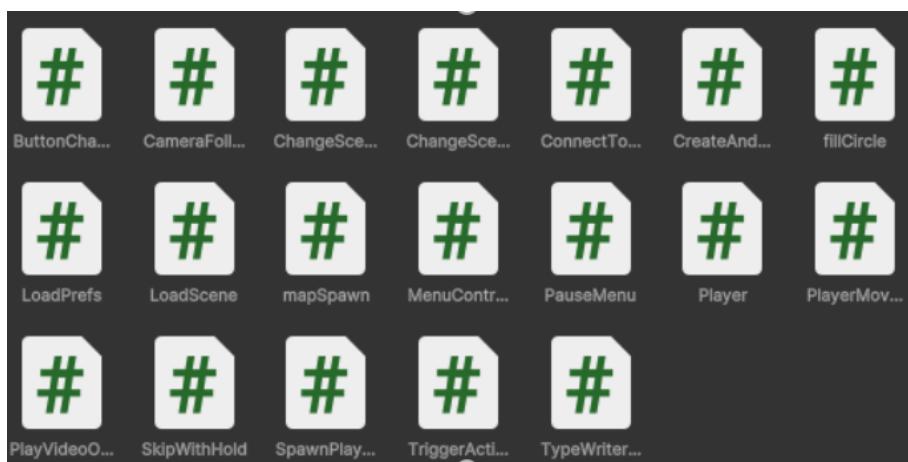


Figure 9: Différents scripts dans le projet

- Chargement de vidéos après une scène

Sarah François**Maps :**

Le but : Plonger le joueur dans le monde de L'Étang avec un univers naturel rappelant un Moyen-Âge fantastique.

Étapes :

- Création du sol, mélange d'herbe et de chemin
- Création de la forêt environnante
- Ajout des collisions pour tous les objets présents

Résultat : (*Ajouter captures d'écran ou descriptions détaillées*)

Réseaux sociaux : Sarah a choisi de poster principalement sur Instagram, le réseau social le plus populaire parmi notre cible (jeunes de 12 à 25 ans). Ces posts visent à faire connaître notre jeu et à obtenir des retours.

Dessins :

Le but : Créer des personnages mignons et « enfantins » avec un décalage par rapport à l'histoire plus sombre. Au fur et à mesure du jeu, le personnage principal obtient des améliorations qui changent son apparence pour le rendre plus impressionnant et mécanique.

Étapes :

- Choix d'un design cohérent avec l'univers du jeu
- Dessin initial d'un canard, jugé trop réaliste
- Recherche d'inspiration sur Internet et obtention des droits d'auteur
- Création des designs des autres personnages en cohérence avec le premier



Figure 10: À gauche notre résultat, à droite notre inspiration

Résultat : (*Ajouter illustrations ou descriptions détaillées*)

Hugo Lefay

Hugo a créé l'installateur grâce à Inno Setup, un logiciel libre permettant de créer des installateurs pour Windows. Il a choisi Inno Setup car il supporte toutes les versions de Windows, a une interface facilitant l'utilisation, permet de personnaliser le processus d'installation et est très populaire avec une abondante documentation.

Étapes :

- Suivi d'une vidéo explicative : Tutoriel Inno Setup
- Crédit de l'installateur en respectant l'organisation des fichiers

Problèmes rencontrés :

- Erreur « Failed to load mono » lors de la création de l'installateur
- Solution trouvée en ajoutant une ligne de code dans le script d'installation

Code utilisé :

```
Source: "C:\Users\hugo\Desktop\unity projet\build\My project (2)
```

```
Destdir: "{app}\My project (2)_Data";
```

```
Flags: ignoreversion recursesubdirs createallsubdirs
```

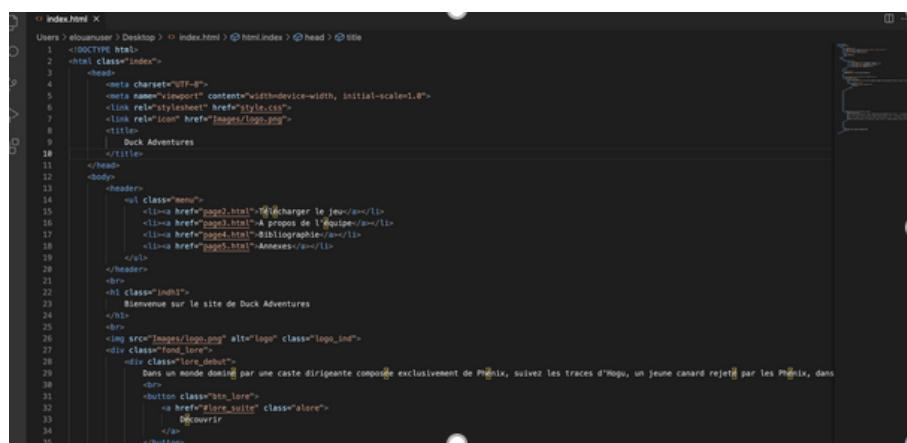
Elouan Coleno

Site Web :

Le but : Créer un site web complet qui présente le jeu, permet de le télécharger et donne des informations sur l'équipe et l'avancée du projet.

Les étapes :

- Le site a été créé à partir de rien. Il a donc fallu créer la structure en HTML à l'aide de connaissances acquises auparavant et de ressources présentes sur Internet (W3Schools, Mozilla Developers), en utilisant l'EDI VSCode.
- **Structure HTML de la page principale**



```

<!DOCTYPE html>
<html class="index">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <link rel="icon" href="Images/logo.png">
    <title>Duck Adventures</title>
  </head>
  <body>
    <header>
      <ul class="menu">
        <li><a href="page1.html">Télécharger le jeu</a></li>
        <li><a href="page2.html">À propos de l'équipe</a></li>
        <li><a href="page3.html">À propos de l'équipe</a></li>
        <li><a href="page4.html">Années</a></li>
      </ul>
    </header>
    <h1 class="ind1">Bienvenue sur le site de Duck Adventures</h1>
    <br>
    
    <div class="text_debut">
      <p>Dans un monde dominé par une caste dirigeante composée exclusivement de Phoenix, suivez les traces d'Hogu, un jeune canard rejeté par les Phoenix, dans</p>
      <br>
      <button class="btn_lire">
        <a href="#">Lire suite</a>
        <img alt="arrow icon" style="vertical-align: middle;">
      </button>
    </div>
  </body>
</html>

```

Figure 11: Structure HTML de la page principale

- Ensuite, le site a été mis en forme selon nos exigences, grâce à un fichier de feuilles de style en cascade (CSS).

Feuille de style du site

- Enfin, les fichiers du site ont été envoyés sur GitHub afin d'avoir un hébergement du site.

Repo GitHub du site

```

<!DOCTYPE html>
<html class="index">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <link rel="icon" href="Images/logo.png">
  </head>
  <title>Duck Adventures</title>
  <style>
    /* Smooth scroll */
    html {
      scroll-behavior: smooth;
    }
  </style>
  </head>
  <body>
    <header>
      <ul class="menu">
        <li><a href="page2.html">Télécharger le jeu</a></li>
        <li><a href="page3.html">A propos de l'équipe</a></li>
        <li><a href="page4.html">Bibliographie</a></li>
        <li><a href="page5.html">Annexes</a></li>
      </ul>
    </header>
    ...
  </body>
</html>

```

Figure 12: Repo GitHub du site

Résultat :

- Page d'accueil du site



Figure 13: Page d'accueil du site

- Page de présentation de l'équipe

Problèmes rencontrés :

- Reprise du CSS difficile, nécessite de tout réapprendre, concepts avancés à appréhender.

Avant la deuxième soutenance

Concernant le mode multijoueur : Le mode multijoueur sera du PvE à 2 joueurs. Les bots ont été créés de façon à apparaître toutes les x secondes. Cette valeur sera adaptée dans le futur afin d'ajuster au mieux l'expérience utilisateur. Une attaque a aussi été implémentée : un jet de plume par le joueur, qui est utilisable en appuyant sur la touche espace. Cet objet a été configuré de sorte à enlever de la vie aux ennemis en cas de contact. De plus, si la vie de l'ennemi tombe à 0 après le coup, l'ennemi a été configuré pour disparaître. L'implémentation de cette mécanique de jeu a posé un problème de synchronisation entre les joueurs, en effet, un joueur spectateur voyait l'attaque différemment. Ce problème était lié au gestionnaire de paquets, Photon, et a pu être corrigée rapidement. Une attaque ennemie rudimentaire a aussi été implémentée : le jeu enlève de la vie au joueur en cas de contact avec un ennemi.

Concernant le mode solo : Nous avons imaginé et dessiné la carte du jeu, et créé une nouvelle zone de jeu, un village avec des bâtiments toujours dessinés à la main, intérieur comme extérieur. Il est possible de se rendre dans ce village depuis la première zone. Dessiner les bâtiments en respectant l'angle de vue a été une tâche fastidieuse mais néanmoins réussie.

Nous avons également ajouté une interface de type affichage tête haute (ATH) sur lequel figure une jauge de vie fonctionnelle, une jauge d'énergie fonctionnelle qui permet au joueur de sprinter pendant un temps limité, et nous avons ajouté un accès à la carte du monde en appuyant sur TAB.

Enfin, nous avons implémenté une IA sommaire qui se dirige vers le joueur afin de lui infliger des dégâts. Dans le cas où la vie du joueur tomberait à 0, un écran de mort l'attend, depuis lequel le joueur peut recommencer une partie ou retourner au menu principal.

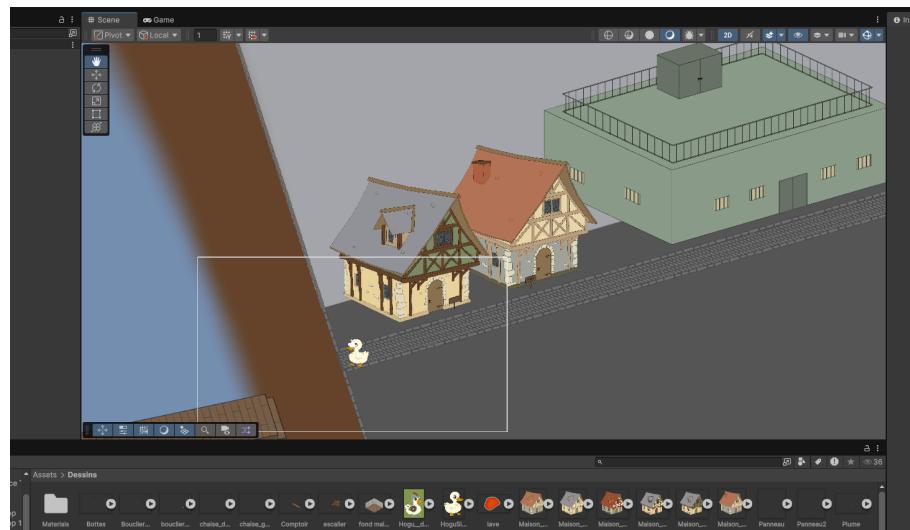


Figure 14: Deuxième zone du jeu : le village



Figure 15: Intérieur d'un bâtiment, l'armurerie



Figure 16: La carte du jeu, qui s'affiche en appuyant sur TAB



Figure 17: Le pont permettant de changer de zone



Figure 18: L'écran de mort du joueur

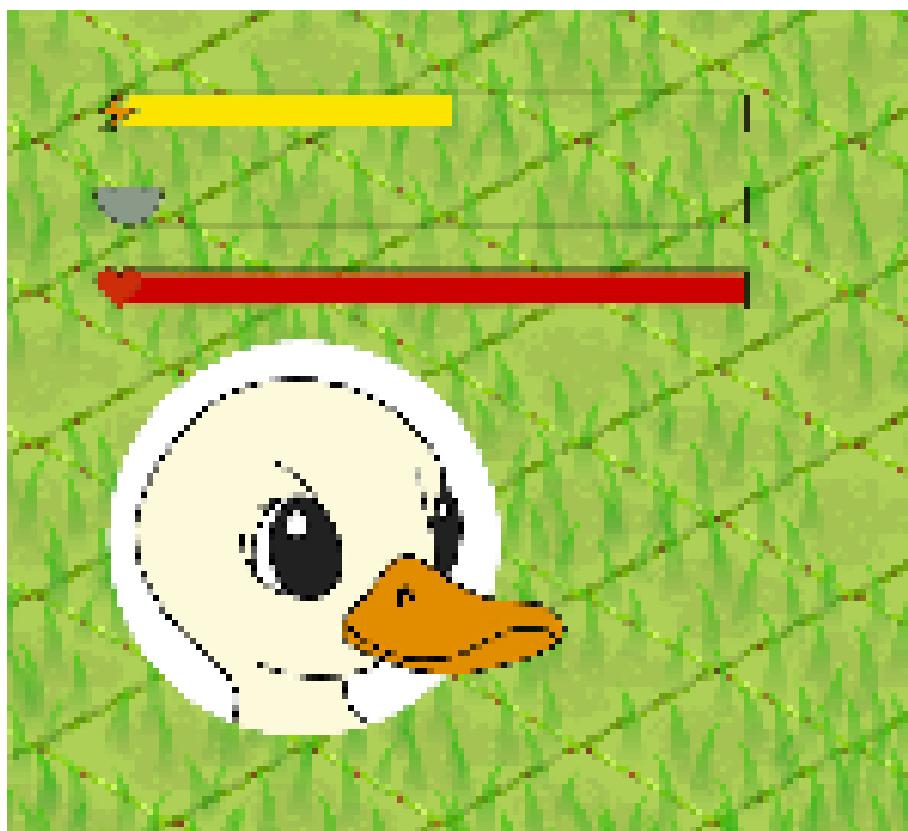


Figure 19: Les jauge de vie, de bouclier et d'énergie

Mise en relation avec le diagramme de Gantt

Tableau d'avancement des tâches						
TASK	SEPTEMBRE	NOVEMBRE	JANVIER	FÉVRIER	AVRIL	MAI
CRÉATION VISUELS						
ÉCRITURE HISTOIRE						
CRÉATION SITE WEB						
MODE SOLO						
MODE MULTI						
IA						
INSTALLEUR						
COMMUNICATION						

Figure 20: Diagramme de Gantt

Avancements selon le diagramme de Gantt

Visuels : Avancement conforme aux prévisions.

Concernant les personnages : Le personnage principal (Hogu), les PNJ standards (Cygnes) et les ennemis (Phénix et Oies mécaniques) sont dessinés, des pièces d'équipement ont été dessinées à la main par Sarah, certaines animations sont dessinées.

Concernant la map : Les deux premières zones du mode solo sont dessinées et implémentées, les zones suivantes sont en cours de dessin. La première map multijoueur est implémentée, mais peut être modifiée.

Concernant l'histoire de l'univers : Les fondements de l'univers du jeu sont écrits et disponibles sur le site web.

Concernant l'histoire de l'intrigue principale : L'histoire est écrite, une cinématique résumant le début est implantée dans le jeu, les quêtes principales sont écrites.

Concernant les intrigues secondaires : Plusieurs quêtes secondaires ont été écrites (ex : quête secondaire sur les couleurs).

Création du site web : Avancement conforme aux prévisions.

La structure du site : La structure du site a été intégralement créée en HTML, le design du site a été créé en CSS, le site est hébergé sur GitHub. Une biographie de chaque membre de l'équipe est en ligne, ainsi que le lore du jeu et un lien de téléchargement de la version actuelle.

Mode solo : Avancement conforme aux prévisions.

Le système de déplacements est créé et implanté, ainsi que les premières zone. Interactions avec l'environnement (boîtes de dialogue, système de dégâts par les ennemis, systèmes de changement de zone, habitations). Une interface est également implantée.

Mode multi : En avance par rapport aux prévisions.

Le système de lobbies est en place, avec une possibilité de créer une instance ou d'en rejoindre une existante à l'aide d'un mot de passe. Les déplacements sont également implémentés, et synchronisés entre les joueurs. Un système de combat rudimentaire et des bots rudimentaires sont implémentés.

IA :

En temps par rapport aux prévisions.

Une IA rudimentaire qui cherche le joueur et lui inflinge des dégâts est implémentée.

Installeur :

Largement en avance par rapport aux prévisions.
(État final)

L'installateur est fini, fonctionnel et en ligne.

Communication :

Avancement conforme.

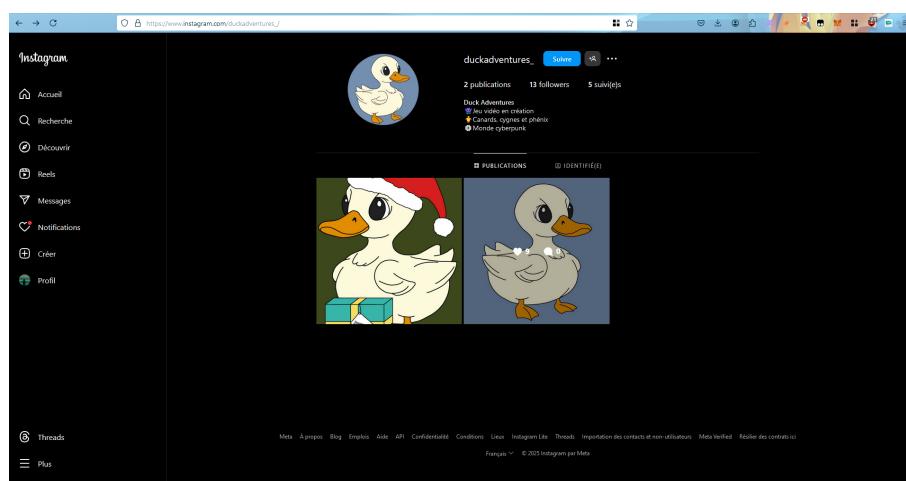


Figure 21: Page Instagram de Hogu, notre mascotte

Un compte Instagram a été créé, deux posts à date.

Conclusion

Le projet *Duck Adventures* est un défi stimulant qui combine développement, graphisme et conception de gameplay. Notre groupe, *Les Canards Sauvages*, s'efforce de créer un jeu unique et divertissant tout en

renforçant nos compétences techniques et collaboratives. Nous avons hâte de vous présenter nos avancées lors des différentes soutenances. Nous sommes également ravis de pouvoir appliquer les compétences en C# que nous avons acquises au cours de nos TPs de programmation.