Page de garde

Remerciement

Contents

[Description du projet : Isidor’s Quest : Chasing the glow 5](#_Toc154782967)

[Pourquoi ce projet et pas un autre ? 5](#_Toc154782968)

[Phases d’analyses 6](#_Toc154782969)

[Le marché actuel 6](#_Toc154782970)

[Architecture fonctionnelle du produit 6](#_Toc154782971)

[Choix techniques 6](#_Toc154782972)

[Le rôle de chaque composant 6](#_Toc154782973)

[Pourquoi ces outils plutôt que d’autres ? 6](#_Toc154782974)

[Charte graphique 6](#_Toc154782975)

[Notre organisation 6](#_Toc154782976)

[Planification 6](#_Toc154782977)

[Rôles de chacun(e) 6](#_Toc154782978)

[Répartition des tâches 6](#_Toc154782979)

[Nos deux applications 6](#_Toc154782980)

[Les services proposés 6](#_Toc154782981)

[Jeu 6](#_Toc154782982)

[Site 6](#_Toc154782983)

[Nos phases de développement 6](#_Toc154782984)

[Architecture technique du produit 6](#_Toc154782985)

[Conception des applications 6](#_Toc154782986)

[Jeu 6](#_Toc154782987)

[Site 6](#_Toc154782988)

[Incorporation du jeu sur le site 6](#_Toc154782989)

[Les tests utilisateurs 6](#_Toc154782990)

[Bilan de montée en compétences 7](#_Toc154782991)

[Les difficultés rencontrées 7](#_Toc154782992)

[Ce qui reste à accomplir 7](#_Toc154782993)

[Perspectives d’amélioration 7](#_Toc154782994)

[List des annexes 8](#_Toc154782995)

Introduction

# Description du projet : Isidor’s Quest : Chasing the glow

Dans le cadre de notre troisième année de BUT informatique, il nous est demandé de concevoir une application qui est alignée avec nos projets professionnels et nos passions personnelles.

Notre jeu (qui sera accessible au travers d’un navigateur) prendra la forme d’une expérience en 2D avec un style pixel art. Au début d’une nouvelle partie, l’utilisateur sera invité à choisir parmi deux classes de héros : le guerrier et l’archer. Le joueur sera ensuite confronté à des monstres et à des énigmes de difficulté croissante, nécessitant des compétences et de la stratégie pour accéder aux niveaux suivants.

# Pourquoi ce projet et pas un autre ?

Après une longue phase de réflexion, notre choix s’est porté sur le domaine du jeu vidéo, un domaine qui suscite un intérêt majeur au sein de notre équipe. Malgré l’immensité des possibilités de création de jeux, nous avons finalement choisi de nous orienter vers un style emblématique, à l’instar de Mario, Kirby ou encore Sonic. Nous avons tous déjà joué à ces jeux plus petits et ce serait un rêve de pouvoir réaliser un tel type de jeu. De plus, la plupart des langages abordé par le projet ne sont pas tous maîtrisés par les membres de l’équipe. C’est donc aussi une occasion idéale d’enrichir notre CV.

Développement

# Phases d’analyses

## Le marché actuel

En 2023, les recettes du secteur jeux vidéo sont estimées à 245.10 milliards de dollars américains et devraient même atteindre 378.08 milliards de dollars américains en 2028. Ce chiffre s’explique par l’augment des plateformes disponibles de nos jours, on peut désormais jouer à des jeux plutôt puissants sur nos smartphones, ce qui n’était pas possible dans les années 80 et 90. De plus, la diversité de type de jeux attire de nombreux consommateurs, en 2022, il existe un peu moins d'1,8 milliard de joueurs dans le monde.

## Choix techniques

### Le rôle de chaque composant

#### Jeu

Unity Engine : pour développement du jeu

Visual Studio : Environnement de développement conseillé par Unity

Piskelapp : Permet de créer des designs en 2D pixelisé

Pinetools : Utiliser pour pixeliser une image

OpenGameArt & CraftPix : Recherches de Sprite (design utilisé pour le jeu)

#### Web

Figma : Maquettage du site

React Native : Framework utilisé pour développer la partie frontend

Node.JS : Plateforme de développement utilisé pour la partie backend

Visual Studio code : Environnement de développement utilisé pour le site

MongoDB : Base de données non relationnelle, utilisée pour stockage des données des utilisateurs

#### Gestion et organisation

Trello : Organisation des taches

Google Drive : Echange de document

GitHub : Gestionnaire de version

Discord : Communication

Notion : Création du diagramme de Gantt

Visual Paradigm : Mise en place des diagrammes UML

### Pourquoi ces outils plutôt que d’autres ?

#### Jeu

Nous avons décidé d’utiliser Unity, car il s’agit d’une plateforme de jeu gratuite qui est multi-plateforme cela permet de facilement générer une version web de notre jeu (WebGL), de plus, la documentation est facile d’utilisation et la communauté Unity est très active, il existe de nombreux tutoriels (vidéo) ainsi que de résolution de problème proposé par la communauté sur Unity Community par exemple. On peut également noter que les designs pour le jeu comme les sorites sont facile d’accès sur des sites tels qu'OpenGameArt et nous pouvons réaliser nos propres design 2D grâce à différents outils telles que piskelapp. De plus, certains membres de l’équipe n’ont jamais travaillé avec Unity, c’était l’occasion idéal pour pouvoir apprendre l’utilisation d’une nouvelle technologie.

#### Web

Nous avons décidé d’utiliser Figma, car il s’agit d’un outil gratuit dont nous avons déjà utilisé auparavant pour réaliser les maquettes de nos sites web, il s’agit d’un outil simple d’utilisation qui permet de travailler en collaboration en simultané, plusieurs personnes en même temps peuvent travailler sur cet outil.

Concernant le site web, nous avons utilisé différentes technologie, le Framework React-Native pour le front-end, car le Framework React est de plus en plus utilisé dans le monde professionnelle et à travers la réalisation de notre site web, nous souhaitons pouvoir maîtriser ce Framework, nous avons pu apprendre de nouvelles thématiques, dont nous jamais abordé telles que les notions de JSX, Hooks par exemple.

Nous avons choisi Node.js pour le back-end et le Framework Express.js permettant de réaliser facilement et rapidement des fonctions de traitement pour différentes requêtes http, de gérer les sessions de l’utilisateur ou de se servir des cookies par exemple. Il s’agit d’un Framework qui offre un ensemble complet d’attribut pour les sites web et Node.js est l’un des back-end les plus utilisés pour le Framework React.

MongoDB est la base de données que nous avons utilisées pour le site web, nous avons décidé de travailler avec cette base, car pour la plupart des membres de l’équipe, nous avons jamais utilisé de base de données non relationnel (no SQL), MongoDB est la base de donnée non relationnel la plus connue et il existe en node.js, une bibliothèque permettant de facilement réaliser des opérations sur MongoDB nommé mongoose, l’alchimie entre node.js et MongoDB était simple et c’était également pour l’équipe de découvrir une nouvelle technologie via ce projet.

Nous avons décidé avec l’éditeur de code Visual studio code, car il propose de nombreuses extensions permettant de rendre le travail plus simple comme le live Share permettant simultanément sur le même code, de plus il s’agit d’un éditeur adapté aux langages JavaScript donc aux Framework React et au langage Node.js. On peut facilement utiliser son terminal et GitHub avec Visual Studio Code.

#### Gestion et organisation

Créer des maquettes d’application web en collaborant avec les autres membres, tout en étant gratuit et accessible à tous ? Figma est alors l’outil parfait ! Contrairement à ses concurrents, qui sont payants ou qui deviennent payantes après une phase de test. Figma est un outil qui est facile à prendre en main, elle permet aussi de réaliser des maquettes pour applications mobiles.

Qui ne connaît pas Google ? Un des GAFAM ! L’entreprise a développé une application, Google Drive, qui permet également de collaborer avec les membres de l’équipe en plus de pouvoir créer, éditer et supprimer différents types de fichiers (documents, tableurs, présentations). Nous utilisons cet outil car c’est celui qui offre le plus d’espaces de stockage sans frais, avec 15 GO.

Les outils de versionning sont nombreux, pourtant il y en a certains qui sont plus pertinents que d’autres, comme la gestion de version décentralisée (chaque développeur possède ses propres dépôts et copie locale), ce qui permet à chacun de travailler à son rythme. Nous nous sommes orientés vers Git, et plus précisément GitHub. Offrant la possibilité de créer plusieurs branches locales indépendantes, un tableau de bord personnel pour suivre les issues et les pull requests. De plus, si un problème survient, aucun souci, grâce à la grande communauté qui s’y trouve.

Discord, un outil de communication, de visioconférence et d’envoi de fichiers, se distingue par sa capacité à évoluer constamment grâce à la possibilité d’ajouter des bots sur nos serveurs. Étant tous sur la plateforme depuis plus de 2 ans, nous l’avons sélectionnée sans hésitation.

Lorsqu’on évoque Notion, l’outil no-code de productivité vient immédiatement à l’esprit. Néanmoins, elle permet aussi de faire des diagrammes de Gantt. Cette fonctionnalité proposée par l’application a été notre dernier recours. En effet, nous avons dans un premier temps porté nos recherches sur d’autres applications, mais elles étaient soit payantes, soit mal conçues (constitution du diagramme avec un assemblage de rectangles).

Concernant l’outil pour les diagrammes UML, il y a mieux que Visual Paradigm, comme Lucidchart. Cependant, son utilisation est restreinte. Nous pouvons uniquement créer 3 documents et pas plus. Alors que Visual Paradigm permet d’ajouter et de confectionner une infinité de fichiers tant que la limite de stockage n’est pas dépassée (1 GO).

## Charte graphique

A screenshot of a video game

Description automatically generated

Figure : charte graphique utilisée pour l'application

## Notre organisation

### Planification

Avant de commencer le projet, nous avons mis en place un diagramme de Gantt que nous avons intégré, sous forme d’un tableau dans le cahier des charges (voir en annexe 6). Il y a un lien qui l’accompagne, étant donné que celui-ci est devenu un lien mort, mais aussi que la lecture du diagramme sur un tableau n’est pas la plus idéale, nous avons mis en annexe 4 des images l’illustrant. Le diagramme de Gantt est présent pour titre indicatif, durant le projet, nous avons rajouté des tâches ou encore réalité des fonctionnalités plus tôt que la date indiquée sur le diagramme.

### Rôles de chacun(e)

Outre le fait que le projet doit correspondre à nos envies personnel et professionnel, c’est avant tout une manière d’apprentissage. Nous avons donc chacun choisis plusieurs rôles, qui nous semble pertinent.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | COLLOMBET Nathan | LIN  Oscar | SELVARATNAM Akash | ZHANG Anxian | LIN  Xingtong |
| Web designer | ✓ |  | ✓ | ✓ |  |
| Game designer |  |  |  | ✓ | ✓ |
| Développeur Web | ✓ | ✓ | ✓ | ✓ | ✓ |
| Développeur Logiciel |  | ✓ | ✓ | ✓ | ✓ |
| Testeur | ✓ |  |  | ✓ |  |

### 

### 

### Répartition des tâches

Dans notre organisation il n’y pas à proprement parler de chef projet, nous utilisons une application de gestion des tâches, Trello, qui nous permet d’ajouter des missions et de les labelliser. Cela veut dire qu’une fois une mission débutée par une personne, elle se doit de mettre en tant que membre sur la carte afin de ne pas refaire des choses qui ont déjà été réalisées.

A black background with white text

Description automatically generated

Figure : exemple de tâche effectuée et labellisée « Terminer »

# Nos deux applications

## Les services proposés (architecture fonctionnel)

### Jeu

### Site

## Nos phases de développement

## Architecture technique du produit

## Conception des applications

### Jeu

### Site

#### Frontend

#### Backend

## Incorporation du jeu sur le site

## Les tests utilisateurs

Conclusion

# Bilan de montée en compétences

# Les difficultés rencontrées

# Ce qui reste à accomplir

# Perspectives d’amélioration

# List des annexes

[Annexe 1 : Résumé 9](#_Toc154782996)

[Annexe 2 : Abstract 9](#_Toc154782997)

[Annexe 3 : Les sources 9](#_Toc154782998)

[Annexe 4 : Images du digramme de Gantt 9](#_Toc154782999)

[Annexe 5 : Table des illustrations 9](#_Toc154783000)

[Annexe 6 : Cahier des charges 9](#_Toc154783001)

[Annexe 7 : Poster 9](#_Toc154783002)

# Annexe 1 : Résumé

# Annexe 2 : Abstract

# Annexe 3 : Les sources

Collectif. « Le marché du jeu vidéo – Faits et chiffres ». 13 décembre 2023. Statista.

<https://fr.statista.com/themes/9063/le-marche-du-jeu-video/#topicOverview>

Collectif. « Industrie du jeu - Analyse de la taille et des parts – Tendances et prévisions de croissance (2023 - 2028) ». 2018. Mordor Intelligence.

<https://www.mordorintelligence.com/fr/industry-reports/global-gaming-market>

Pauline Callies. « Les meilleurs outils pour créer votre maquette d’application mobile ». 19 mai 2022. Aventique.

<https://aventique.paris/maquette-application-mobile/#:~:text=Figma%20est%20un%20outil%20de,de%20ses%20fonctionnalit%C3%A9s%20est%20payant>.

Collectif. « 6 alternatives à Google Drive ». 2021. Blogdumoderateur.

<https://www.blogdumoderateur.com/tools/alternatives/google-drive/>

Eloïse Salson. « Suivez vos modifications à la trace avec les 8 meilleurs logiciels de versionning ». 19 janvier 2022. Appvizer.

<https://www.appvizer.fr/magazine/services-informatiques/gestion-versions/outils-versionning>

# Annexe 4 : Images du digramme de Gantt

**A screenshot of a computer

Description automatically generated**

Figure : Digramme de Gantt, première partie

A screenshot of a computer

Description automatically generated

Figure : Diagramme de Gantt deuxième partie

# Annexe 5 : Table des illustrations

# Annexe 6 : Cahier des charges

# Annexe 7 : Poster