|  |
| --- |
| Projet Pré-TPI  Générateur de labyrinthe |

Par David Dubey, SI-CA2a

Table des matières

[1 Analyse préliminaire 3](#_Toc67392908)

[*1.1* Introduction 3](#_Toc67392909)

[*1.2* Objectifs 3](#_Toc67392910)

[*1.3* Gestion de projet et planification 3](#_Toc67392911)

[1.3.1 Sprint 1 : Analyse préliminaire (Échéance le 12.02.2021) 3](#_Toc67392912)

[1.3.2 Sprint 2 : Conception de base (Échéance le 26.02.2021) 3](#_Toc67392913)

[1.3.3 Sprint 3 : Réalisation du projet (Échéance le 26.03.2021) 4](#_Toc67392914)

[Ce sprint est consacré à la réalisation complète du projet. À ce stade-là, il n’y a plus de doute sur le choix de la technologie. Il y a, cependant, une phase de d’apprentissage du moteur Unity. Le langage C# est déjà connu. 4](#_Toc67392915)

[1.3.4 Sprint 4 : Présentation (Échéance actuellement non définie) 4](#_Toc67392916)

[*1.4* Planification initiale 5](#_Toc67392917)

[*1.5* Planification Sprint 3 5](#_Toc67392918)

[*1.6* Sources, acquisition du savoir 5](#_Toc67392919)

[2 Analyse / Conception 5](#_Toc67392920)

[*2.1* Concept 5](#_Toc67392921)

[*2.2* Environnement et limite du système 6](#_Toc67392922)

[*2.3* Stratégie de test 6](#_Toc67392923)

[*2.4* Risques techniques 6](#_Toc67392924)

[*2.5* Planification 6](#_Toc67392925)

[*2.6* Dossier de conception 6](#_Toc67392926)

[3 Réalisation 7](#_Toc67392927)

[*3.1* Dossier de réalisation 7](#_Toc67392928)

[*3.2* Description des tests effectués 7](#_Toc67392929)

[*3.3* Erreurs restantes 7](#_Toc67392930)

[*3.4* Liste des documents fournis 7](#_Toc67392931)

[4 Conclusions 8](#_Toc67392932)

[5 Annexes 9](#_Toc67392933)

[*5.1* Résumé du rapport du TPI / version succincte de la documentation 9](#_Toc67392934)

[*5.2* Réflexions 9](#_Toc67392935)

[*5.3* Sources – Bibliographie 9](#_Toc67392936)

[*5.4* Journal de travail 9](#_Toc67392937)

[*5.5* Manuel d'Installation 9](#_Toc67392938)

[*5.6* Manuel d'Utilisation 9](#_Toc67392939)

[*5.7* Archives du projet 9](#_Toc67392940)

# Analyse préliminaire

## Introduction

Le projet est un jeu vidéo codé en C++ avec le moteur Enreal. Il s’agit d’un jeu de création de labyrinthe avec un bot qui devrait trouver le bon chemin pour le traverser. Le but est d’apprendre à gérer le pathfinding car ce projet sert avant tout de base pour le future projet type Tower defense pour le vrai TPI. Le jeu en question devra avoir une interface ergonomique pour la souris et aussi les écrans tactiles.

## Objectifs

Les principaux objectifs sont :

* Dans un premier temps, apprendre le C++ et Unreal Engine
* Dans un second temps, créer un jeu basé sur les technologies citées précédemment.
* Apprendre à gérer le pathfinding.
* Créer une interface utilisateur ergonomique pour la souris et le tactile.
* Dans le jeu : Pouvoir créer un labyrinthe et des morceaux de mur.
* Lorsque l’utilisateur appelle le bot, celui-ci doit résoudre le labyrinthe.

## Gestion de projet et planification

Le projet sera entièrement réalisé en suivant la méthode agile. Les moment clés du projet est séparé en quatre sprints.

### Sprint 1 : Analyse préliminaire (Échéance le 12.02.2021)

Ce premier sprint est consacré à la prise de connaissance du cahier des charges, la mise en place de l’infrastructure et l’apprentissage des technologies utilisées pour ce projet. Ici, il est consacré à l’apprentissage du C++ et Unreal Engine. Cependant pour des raisons détaillées plus loin, le choix de la technologie a changé.

### Sprint 2 : Conception de base (Échéance le 26.02.2021)

Ce deuxième sprint et consacré à la réalisation d’une partie du projet en C++ avec le moteur Unreal et de juger si les technologies testées sont les plus adaptées. Pour le projet. À la fin de ce sprint, il fallait prendre la décision de continué avec Unreal et C++ ou de passé sur le monteur Unity en C#.

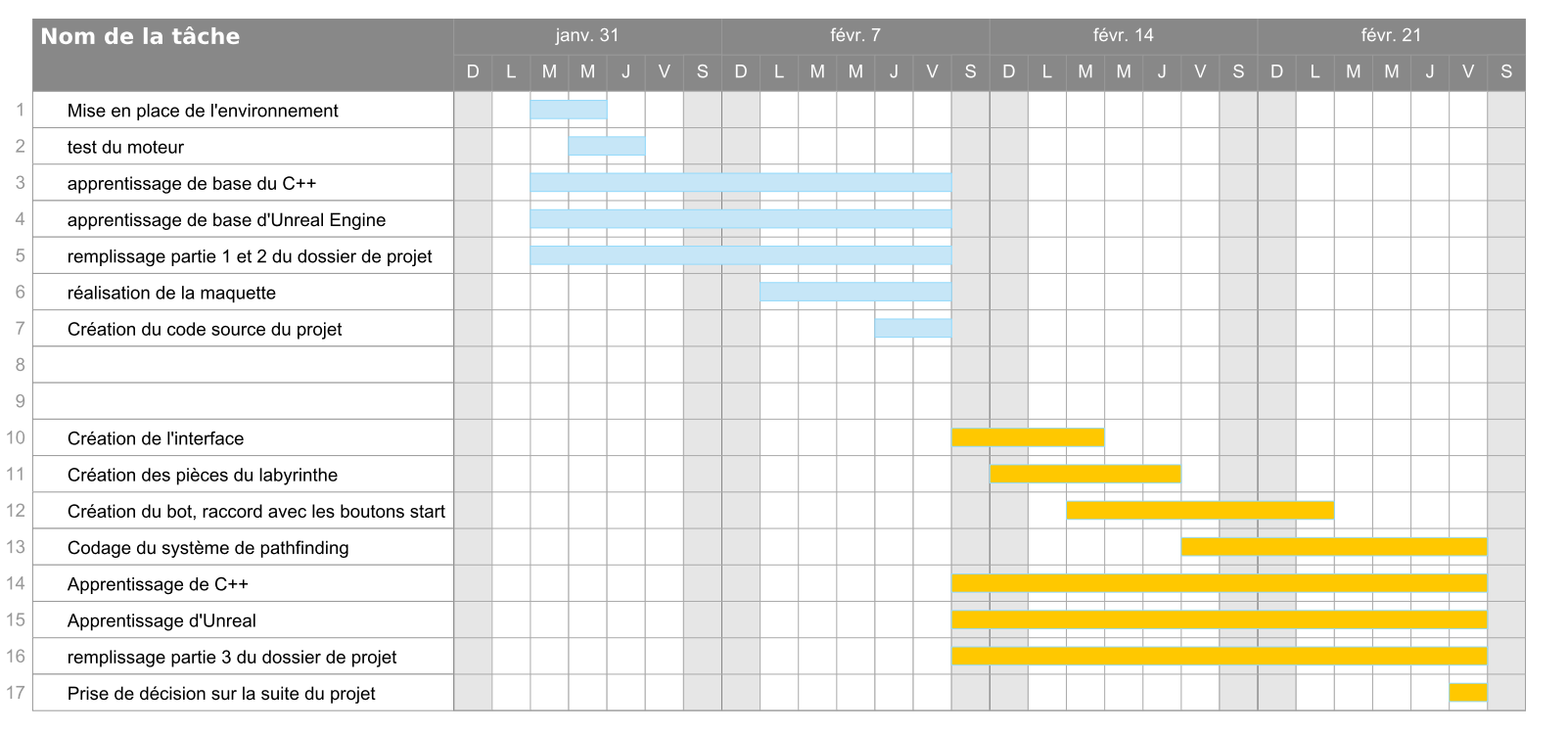
### Sprint 3 : Réalisation du projet (Échéance le 26.03.2021)

### Ce sprint est consacré à la réalisation complète du projet. À ce stade-là, il n’y a plus de doute sur le choix de la technologie. Il y a, cependant, une phase de d’apprentissage du moteur Unity. Le langage C# est déjà connu.

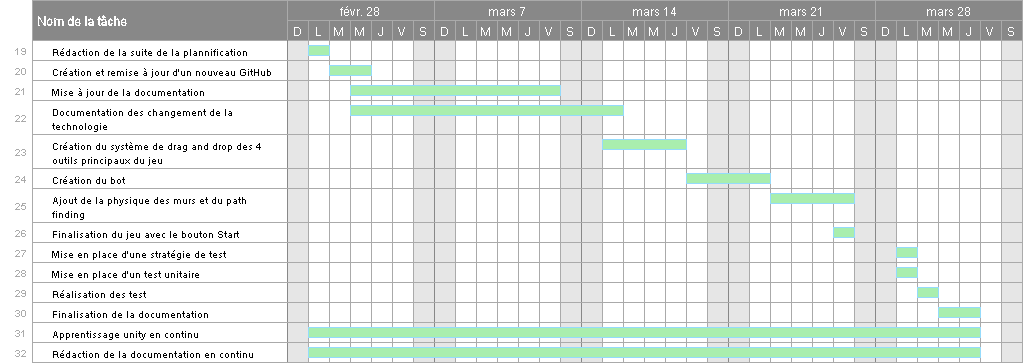
### Sprint 4 : Présentation (Échéance actuellement non définie)

Ce sprint est le dernier du projet et est consacré uniquement à la réalisation de la présentation orale.

## Planification initiale



## Planification Sprint 3



## Acquisition du savoir

# Analyse / Conception

## Concept

Le projet sera programmé en C++ avec un moteur de jeu « Unreal Engine ». Il créera un exécutable qui lancera directement le jeu.

Mise à jour du 21 février : La suite du projet se fera finalement en C# et sur le moteur de jeu Unity.

## Environnement et limite du système

Le projet sera est développé en C# sur un ordinateur Windows 10. La compilation du code source donne un exécutable (.exe) pour Windows. Il n’est pas prévu de version pour un autre système d’exploitation.

## Stratégie de test

Dans ce projet, comme demandé dans le cahier des charges, il y aura au moins un test unitaire tenu à jour

## Risques techniques

* Apprentissage d’un nouveau langage (C++, pointeurs)
* Apprentissage d’un moteur de jeu (Unreal)

Mise à jour

Pour ces deux points, il y aura une partie du temps total du projet qui sera consacré à l’apprentissage de ces technologies.

## Planification

Considérant l’éventualité d’un changement de technologie lors pendant la 4ème semaine, pour l’instant, le projet se tient sur la planification initiale.

Sprint 1 : Analyse préliminaire (01/02/2021 – 12/02/2021)

Sprint 2 : Conception de base (13/02/2021 – 26/02/2021)

Mise à jour du 21 février : Le projet continue avec Unity en C#.

Sprint 3 : Réalisation du projet (27/02/2021 – 26/03/2021)

Prolongement du sprint 3 pour le rendu final au 01/04/2021

Sprint 4 : Présentation (27/03/2021 – 22/04/2021)

Voir point 1.4 et 1.5

## Dossier de conception

Matériel Hardware :

* Ordinateur client du CPNV, dell OptiPlex 7070 (16go RAM, 500go SSD, intel core i5 9500 3GHz)

Logiciels et sites

* Git
* Github
* Git LFS 2.13.2
* Microsoft Visual Studio Enterprise 2019 Version 16.7.3
* Visual Studio Code 1.53.2
* Unreal Engine 4
* MS-Office 2016
* smartsheet.com

Programmation

Le code sera organisé de manière modulaire avec le concept de séparation des préoccupations. Il sera en tout temps disponible sur le Github.

Annexes

* Maquette
* Journal de travail

# Réalisation

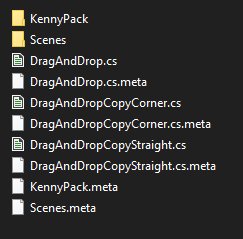
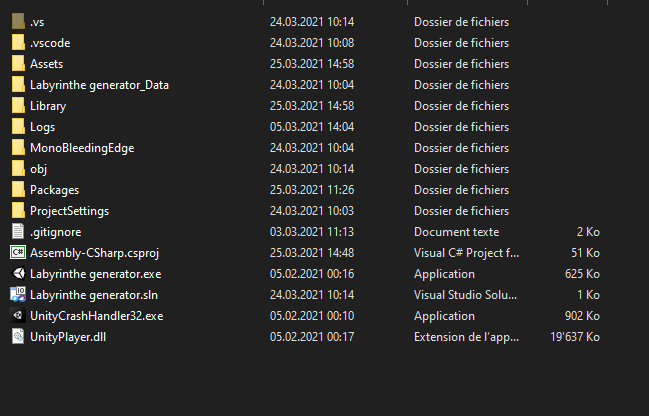
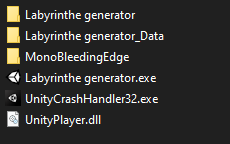
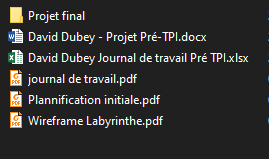
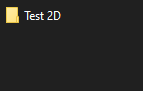
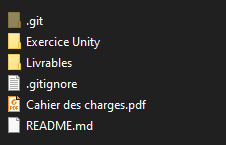
## Dossier de réalisation

L’entièreté de projet est hébergée sur Github. Lorsque la décision a été prise de changer de technologie, il y a quelques complications sur le repository. En effet, des problèmes avec les fichier « .gitignore » empêchant l’upload d’un projet de type Unity. Il a donc fallu créer un nouveau repository spécialement conçu pour la création d’un projet de type Unity. L’ancien a donc été abandonné pour continuer sur le nouveau.

Github 1: https://github.com/20David20/Pre-TPI-2021-Labyrinthe (Abandonné, plus à jour.)

Github 2 : https://github.com/20David20/Pre-TPI-Labyrinthe-Unity/projects (À jour.)

Les répertoires sont organisés comme ceci :



Il y a le fichier « .gitignore », le « README.md » et une copie du cahier des charges du projet. On remarque deux dossiers : Exercice Unity et Livrables

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Réflexions

## Sources – Bibliographie

* Anthony Cardinale. (2021, février) Créer des JEUX 2D qualité pro avec UNITY (AVEC ou SANS CODE) [Cours vidéo]. Udemy.  
  https://www.udemy.com/course/cours-creer-des-jeux-2d-qualite-pro-avec-unity-2d-avec-ou-sans-code/
* Savvu the GameDevel. (2020, 5 avril). Drag and Drop System in Unity - Puzzle Game Example (PC and Mobile) [Vidéo]. YouTube.   
  https://www.youtube.com/watch?v=uk\_E\_cGrlQc
* wRadion. (2018, 27 janvier). Faire un Tower Defense avec Unity #1 - Ennemi (spawn et « path-finding ») [Vidéo]. YouTube.  
  https://www.youtube.com/watch?v=uvP4gZ1QAGI&t=4151s
* openclassrooms. (2020, 23 mars). Apprenez à programmer en C# sur .NET https://openclassrooms.com/fr/courses/218202-apprenez-a-programmer-en-c-sur-net
* Gervais, L. (2019). Apprendre la Programmation Orientée Objet avec le langage C# (3e édition) (French Edition) (3e éd.). ENI.

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*