|  |
| --- |
| Implémentation d’un jeu de révision des livrets |

Table des matières

[1 Analyse préliminaire 3](#_Toc499021832)

[1.1 Introduction 3](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 4](#_Toc499021836)

[2.1 Concept 4](#_Toc499021837)

[2.2 Stratégie de test 4](#_Toc499021838)

[2.3 Risques techniques 4](#_Toc499021839)

[2.4 Planification 4](#_Toc499021840)

[2.5 Dossier de conception 5](#_Toc499021841)

[3 Réalisation 5](#_Toc499021842)

[3.1 Dossier de réalisation 5](#_Toc499021843)

[3.2 Description des tests effectués 6](#_Toc499021844)

[3.3 Erreurs restantes 6](#_Toc499021845)

[3.4 Liste des documents fournis 6](#_Toc499021846)

[4 Conclusions 6](#_Toc499021847)

[5 Annexes 7](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 7](#_Toc499021849)

[5.2 Sources – Bibliographie 7](#_Toc499021850)

[5.3 Journal de travail 7](#_Toc499021851)

[5.4 Manuel d'Installation 7](#_Toc499021852)

[5.5 Manuel d'Utilisation 7](#_Toc499021853)

[5.6 Archives du projet 7](#_Toc499021854)

# Analyse préliminaire

## Introduction

Ce projet est réalisé dans le cadre du Pré-TPI. Suite au module Projet (prog/web), je n’étais pas satisfait du résultat de mon projet alors j’ai décidé de repartir sur du C# pour m’améliorer. En plus de ça, c’est l’occasion pour moi de découvrir le fonctionnement du logiciel « Unity ».

## Objectifs

* Le programme devra proposer des multiplications (livrets de 1 à 12) dans un ordre aléatoire. Une liste de résultats possibles sera proposée et le joueur devra choisir le résultat parmi cette liste.
* Dès le début de la partie, une figurine représentant le héros du jeu (le joueur) sera affichée. Pendant tout la durée du jeu, des guerriers apparaîtront et s’approcheront du héros au fur et à mesure, régulièrement, pour le tuer. Au fur et à mesure des réponses justes, les guerriers perdront leurs points de vie et mourront. S’ils parviennent jusqu’au héros du jeu, la partie en cours s’arrête.
* Chaque 10 opérations (paramétrable)proposées, le résultat de la partie qui vient de se dérouler est annoncé au joueur. Il peut ensuite passer à la partie suivante ou arrêter de joueur. Après 5 niveaux (paramétrable), le résultat final est annoncé au joueur et la partie s’arrête.
* Deux types de guerriers pourront apparaître : de simples guerriers et leur chef. Les points de vie de ces personnages dépendront de leurs types.
* Le score du joueur sera sauvegardé (pourcentage de résultats corrects). Ses 10 meilleurs scores pourront s’afficher à sa demande.

## Planification initiale

Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.

Ces éléments peuvent être repris des spécifications de départ.

# Analyse / Conception

## Choix des technologies

Pour la réalisation du jeu, j’ai décidé d’utiliser « Unity » malgré la liberté qu’offre la librairie « MonoGame ». MonoGame est une librairie basée sur « XNA ». Celui-ce me permettrait de construire mon jeu de A à Z et arriver à un résultat similaire à un projet Unity. Le problème est que je passerais beaucoup plus de temps sur la partie affichage des textures et du texte en utilisant MonoGame. Il me semble plus adapté à ce projet d’utiliser un moteur de jeu car cela me facilitera beaucoup la gestion de l’affichage et des ressources pour me concentrer sur les fonctionnalités principales du jeu.

## Conception

Maquette du jeu fournie : « MaquetteMath-Battle.pdf »

MCD fourni : « MCD-MathBattle.pdf »

MLD fourni : « MLD-MathBattle.pdf »

Inspiré du jeu : <https://www.logicieleducatif.fr/math/logique/math-kingdom.php>

## Stratégie de test

Décrire la stratégie globale de test:

* types de des tests et ordre dans lequel ils seront effectués.
* les moyens à mettre en œuvre.
* couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).
* données de test à prévoir (données réelles ?).
* les testeurs extérieurs éventuels.

## Risques techniques

* Manque de compétences en l’utilisation du logiciel Unity.

## Planification

Révision de la planification initiale du projet :

* planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.
* partage des tâches en cas de travail à plusieurs.

Il s’agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.

## Dossier de conception

Fournir tous les document de conception:

* Matériel Hardware disponible au CPNV. (Optiplex 7070)
* OS de réalisation et utilisation: Windows.
* Logiciels pour la réalisation et l'utilisation : Unity, Visual Studio, Balsamiq, MySQL, Visual Studio Code, Draw.io.
* bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.
* programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…

# Réalisation

## Dossier de réalisation

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

## Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)

## Journal de travail

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

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

Media, … dans une fourre en plastique