|  |
| --- |
| Projet XYZ |

Table des matières

[1 Analyse préliminaire 3](#_Toc506556066)

[1.1 Introduction 3](#_Toc506556067)

[1.2 Objectifs 3](#_Toc506556068)

[1.3 Planification initiale 4](#_Toc506556069)

[2 Analyse / Conception 4](#_Toc506556070)

[2.1 Concept 4](#_Toc506556071)

[2.2 Stratégie de test 4](#_Toc506556072)

[2.3 Risques techniques 4](#_Toc506556073)

[2.4 Planification 4](#_Toc506556074)

[2.5 Dossier de conception 5](#_Toc506556075)

[2.5.1 Maquettes 6](#_Toc506556076)

[2.5.2 MLD Utilisateur 7](#_Toc506556077)

[2.5.3 MLD Elément de la biosphère 7](#_Toc506556078)

[2.5.4 Structure du fichier de projet 7](#_Toc506556079)

[2.5.5 Programmation 8](#_Toc506556080)

[3 Réalisation 8](#_Toc506556081)

[3.1 Dossier de réalisation 8](#_Toc506556082)

[3.2 Description des tests effectués 9](#_Toc506556083)

[3.3 Erreurs restantes 9](#_Toc506556084)

[3.4 Liste des documents fournis 9](#_Toc506556085)

[4 Conclusions 9](#_Toc506556086)

[5 Annexes 11](#_Toc506556087)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 11](#_Toc506556088)

[5.2 Sources – Bibliographie 11](#_Toc506556089)

[5.3 Journal de travail 11](#_Toc506556090)

[5.4 Manuel d'Installation 11](#_Toc506556091)

[5.5 Manuel d'Utilisation 11](#_Toc506556092)

[5.6 Archives du projet 11](#_Toc506556093)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS:  
Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris des spécifications de départ.*

Dans le but de préparer notre TPI je réalise un projet de programmation portant sur la génération procédurale. Pour réaliser ce projet pour approfondir mes connaissances dans le moteur 3D qu’est Unity.

Unity met à disposition de nombreux élément pour réaliser mon projet, étant donné que j’ai déjà des connaissances intermédiaires du logiciel et de son API, le choix se justifie largement.

Le but de l’application sera de pouvoir générer un environnement selon les paramètres choisis par son utilisateur.

## Objectifs

*Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.*

*Ces éléments peuvent être repris des spécifications de départ.*

L’utilisateur devra être capable :

* De générer un terrain en fonction des paramètres qu’il a choisi
* D’explorer l’environnement généré grâce à son clavier et sa souris et interagir avec les élément présent (arbres, plantes, rochers).
* D’enregistrer son projet pour le rouvrir plus tard
* De se connecter à un compte qu’il peut créer dans l’application

## 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

## Concept

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*
* *…*

## 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

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## 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:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *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…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

L’utilisateur pourra parmi plusieurs paramètres à sa disposition générer un environnement naturel conforme à ses besoins.

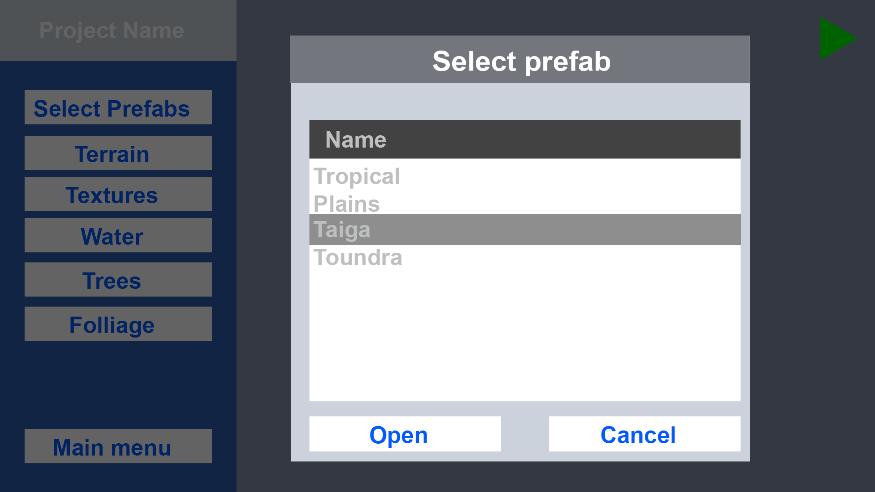
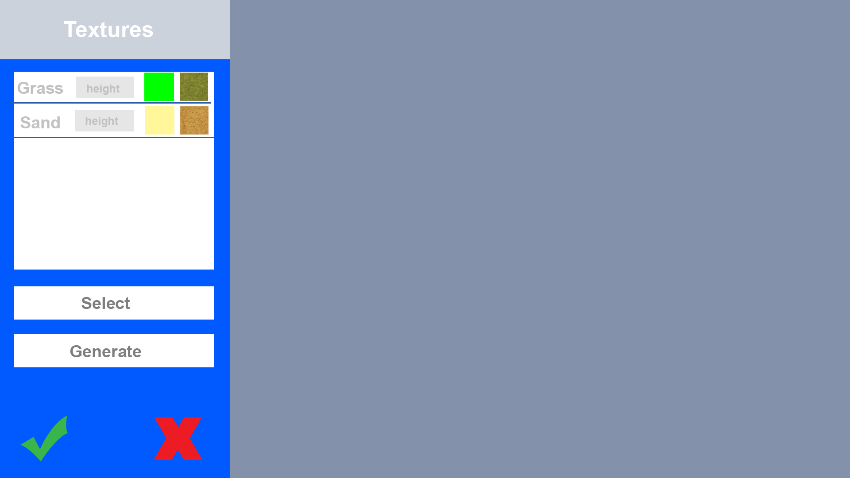
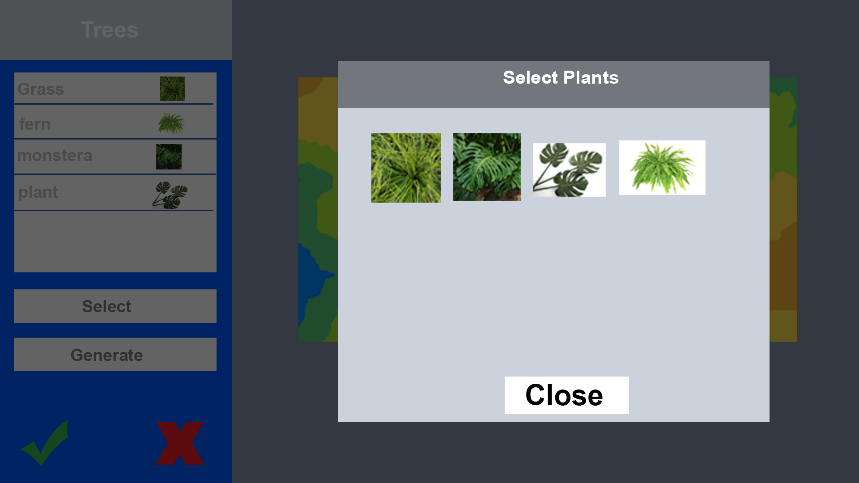
Pour générer un terrain il faudra suivre une certaine marche à suivre sinon l’application ne fonctionnera pas.

Voici le concept des marches à suivre :

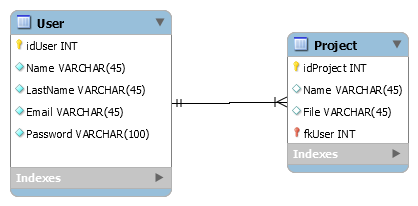
* Générer un « HeightMap[[1]](#footnote-1) »
* Générer un relief
* Générer les textures
* Générer l’environnement aquatique
* Générer les éléments naturels (rochers, arbres, plantes)

### Maquettes

Parmi ces étapes l’utilisateur pourra choisir des paramètres qui permettrons de générer le terrain désiré.



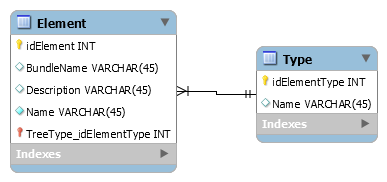
### MLD Utilisateur

L’utilisateur pourra aussi enregistrer ses projets dans une base de données

### MLD Elément de la biosphère

En ce qui concerne les modèles utilisés ils seront stockés aussi sur un Site Hébergeur, ou l’application auras l’autorisation de télécharger les « AssetBundles [[2]](#footnote-2)» nécessaire.

Pour les éléments avec lequel l’utilisateur pourra interagir une autre base de donnée est créé pour stocker les détails des biomes. La composition de cette base de données dépend grandement de la granularité définie, car il est possible d’aller très loin dans la hiérarchie de la biosphère.



### Structure du fichier de projet

Pour réduire la complexité de la base de données le projet sera exporté dans un fichier qui contient tous les détails de la scène. La syntaxe du fichier pourra encore évoluer durant le projet mais le principe est de pouvoir enregistrer chaque paramètre rentré par l’utilisateur, les lister par catégories

Exemple[[3]](#footnote-3) :

* @Terrain
  + <Dimensions – 10 x 10 >
  + <NoiseScale – 1.4 >
  + <RandomSeed – 2 >
* @Textures
  + <#Item 1>
  + <Name – Grass >
  + <Color – 255,0,0 >
  + <Texture – terrainTexture.grass >
  + <#Item 2>
  + <Name – Sand >
  + <Color – 0,255,0 >
  + <Texture – terrainTexture.sand >
* …

Pour résumé Une Section est définie par @NomSection, les valeurs son entre « <> »

Et Si la valeur contient un « # » Au début du nom cela signifie que c’est une liste.

### Programmation

Comme cité dans l’introduction le langage de programmation utilisé est le C#, Car le logiciel utilisé (Unity) compile ses scripts dans ce langage – JavaScript est aussi disponible mais mes connaissances et mon intérêt pour le C# m’amène à coder dans ce langage --.

Etant donné qu’il y a une interface graphique une structure MVC est plus que bienvenue, les scripts gérant la génération, la navigation, et la liaison des données entre deux sont séparé dans des dossiers et des sous dossier respectant leurs rôles et l’appartenance à leurs catégories.

### Exemple de code Modèle ###

### Exemple de code Vue ###

### Exemple de code Contrôleur###

#### Le principe de la génération procédurale

Unity met à disposition divers éléments pour faciliter le développement d’une telle application. Il n’est même pas obligé d’utiliser des scripts pour générer un environnement personnalisé ce qui est pas le but dans ce projet.

Pour commencer la génération de terrain il faut avoir un « HeightMap » à sa disposition.

###Image de HeightMap###

Pour obtenir un tel résultat Unity met à disposition dans sa classe Mathf une fonction qui permet de générer du bruit « aléatoirement » (Mathf.PerlinNoise), en effet ce bruit n’est pas totalement aléatoire sinon on obtiendra un résultat plus comme cela.

### Image de Bruit aléatoire####

Après avoir générer le HeightMap Il est possible de générer un Terrain Avec la Classe terrain de Unity.

PseudoCode pour la génération de terrain

* Je veux générer un HeightMap
  + Je Change les valeurs des propriétés du HeightMap
  + Si les valeurs sont plus grandes que 0
  + Générer un HeightMap(Texture2D)
    - Créer un Plane (Mesh) Sur Unity
    - Affecter le HeightMap (Texture2D) au plane (Mesh)
* Je veux générer un Terrain
  + Je Récupère les valeurs du HeightMap Générer
  + J’affecte les valeurs dans la section TerrainData
  + Je génère le terrain

# 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*

1. HeightMap : Texture en 2D avec des pixels variant du blanc au noir, blanc étant le point le plus bas et noir étant le point le plus haut [↑](#footnote-ref-1)
2. AssetBundle : Terme utilisé par Unity pour définir un fichier qui contient tous les assets disponibles pour l’application [↑](#footnote-ref-2)
3. Valeurs récupérée des maquettes [↑](#footnote-ref-3)