**Projet de Fin d’Année**



Groupe JeanKévin

Félix Pognard

Quentin Sanchez

Valentin Dol

Mattis Faucheux

Table des matières

[Membres de l’équipe 3](#_Toc40364249)

[Analyse du moteur : 4](#_Toc40364250)

[Analyse du projet 7](#_Toc40364251)

[UML 8](#_Toc40364252)

# Membres de l’équipe

* Félix POGNARD (Lead Tech) :
  + Responsable Core
  + Responsable Save
* Quentin SANCHEZ :
  + Responsable Rendering
  + Responsable Génération du Monde
* Valentin DOL :
  + Game State Machine
  + Responsable UI
  + Responsable son
* Mattis FAUCHEUX :
  + Responsable Physique
  + Responsable Gameplay

# Analyse du moteur :

  Core :

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| Maths | Non | Intégrer GLM |
| State Machine | Oui | Créer les States nécessaire : Menu, Game, Pause |
| Input Manager | Oui | Amélioration : utiliser le design pattern : Command  <http://gameprogrammingpatterns.com/command.html> |
| Sound Manager | Non | Lier avec l’implémentation de irrKlang |

-          Save / Load de Game:

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| Sauvegarder au moment de quitter le jeu, pas d’update de la sauvegarde en live | Non | Sauvegarder le monde chunk par chunk (256\*16\*16)  Utiliser TinyXML ou RapidJson pour charger les sauvegardes |
| Possibilité de charger plusieurs parties différentes | Non | Charger les différentes sauvegardes dans un écran de sélection de parties |

-          Rendering/LIBGFX :

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| Light : Directional, Point | Oui | Optionnel: retirer Spot Light |
| Texture : Charger une texture pour chaque bloc | Texture manager,  Parser de texture | Rien |
| Model : Charger des cubes, des rectangles et des models plus complexes | Parser .obj | Implémenter Assimp pour gérer les fichiers chargement de models |
| Frustum culling et occlusion culling | Non | Etudier le fonctionnement et essayer de l’intégrer |

-          UI :

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| Main Menu :   * New Game * Load Game * Options * Exit | Oui | Améliorer la partie visuel |
| Load  Game :   * Afficher les différentes sauvegardes accessibles | Non | Récupérer les noms et les références vers le fichier à load |
| Options :   * Gestions des inputs * Gestions des volumes sonores | Gestion des inputs | Créer une gestion des volumes sonores |
| In Game :   * Vie * Bar d’action | Non | * Implémenter des images pour représenter la vie |
| Inventaire (personnel et coffre) | Non | * Craft grid dans l’inventaire * Inventaire personnel * Inventaire coffre. |
| Menu pause :   * Reprendre * Options * Save * Save to menu |  |  |

-          Son

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| Librairie Sonore | Non | Implémenter irrKlang dans le code |
| Background musique calme en permanence | Non | Récupérer OST Minecraft  et la jouer |
| Sons spatialisés : marche, attaque, rivière, casser blocs, etc…) | Non | Récupérer les sons et les implémenter /-> faire les sons à implémenter |

-          Physique :

|  |  |  |
| --- | --- | --- |
| Besoin | Existant | À  faire |
| World Physics | Non | Créer un world physics et une gravité |
| RigidBody (rectangulaire) | Non | Créer RigidBody |
| Raycast | Non | Créer des raycasts |

# Analyse du projet

|  |
| --- |
| **Besoin** |
| Intégrer 5 types de blocs :  Terre, Gravier, Eau, Lave, Bois |
| Mouvement personnage (ZQSD ou  WASD) |
| Ne traverse pas les murs  Subit la gravité |
| Fall damage |
| Saut avec un peu d'air-contrôle |
| Casser et poser des blocs  (de manière logique pas de  bloc sur de l’eau) |
| Inventaire :   * Limiter chaque stack à 64 |
| Crafting :   * Craft dans l’inventaire * Coffre * Arme * *Bonus (arc, flèches et  torches)* |
| Ennemi :   * Se dirige vers le joueur * Inflige X dégâts par secondes lorsque proche du joueur * Collision pour ne pas rentrer dans les murs * Un seul type d’ennemi, mesh fixe non animé |
| Équiper des armes :   * Fait des dégâts aux ennemis (les tuer s’ils n’ont plus d’HP) * Armes durabilités et se détruisent |
| Mort du joueur :   * Message “You Died” pendant quelques secondes * Respawn endroit prédéfini |
| Charger un monde aléatoire de taille fixe |

# UML

