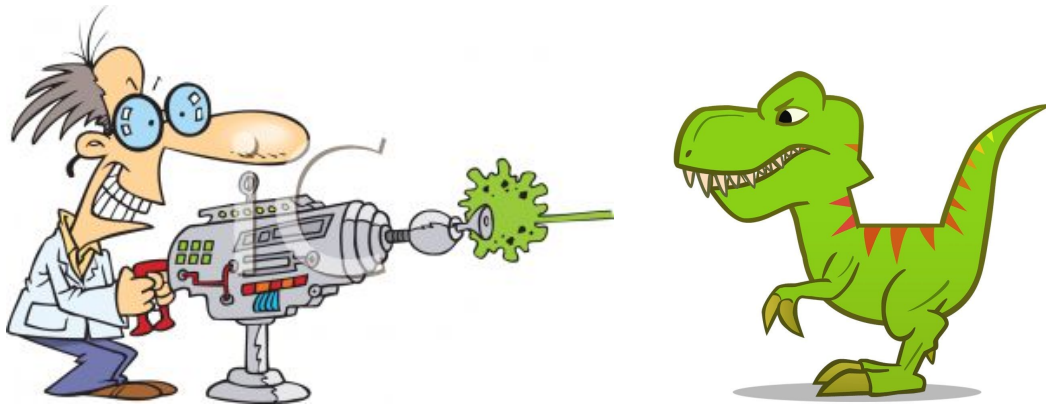


Rapport de Projet

"LOST IN DINOWORLD"



Document réalisé dans le cadre du cours de « Principes conception et développement jeux vidéo »

Esmé James JAME15539504
Ayoub Al Haddan ALHA09119400
Rénald Morice MORR14029503
Wilfried Pouchous POUW04069501

UQAC 2017/2018

Sommaire

1. Rappel du concept et des mécaniques de jeu	3
1.1. Concept	3
1.2. Histoire	3
1.3. Design	4
1.4. Gameplay	4
2. Présentations des interfaces / fenêtres du jeu	5
2.1. Interactions possibles entre les interfaces et menus	5
2.2. Menus du jeu	5
2.2.1. Menu principal	5
2.2.2. La boutique (Store)	6
2.2.3. Les crédits	6
2.2.4. Les options	7
2.3. Interfaces de jeu	7
2.3.1. Interface joueur en partie	7
2.3.2. Interface « Pause »	8
2.3.3. Interface des ingrédients	8
2.3.4. Interface de fin de partie	9
3. Architecture du projet	10
3.1. Diagramme de classes	10
3.2. Hiérarchie de la scène de jeu (PlayScene)	11
3.3. Exemple de code	12
4. Répartition des tâches	14
4.1. Division en sprints	14
4.2. Répartition des rôles	14
5. Choix de conception et technologies utilisées	15
5.1. Choix de conception	15
5.1.1. Plateforme Android	15
5.1.2. Choix des assets	15
5.2. Technologies utilisées	16
5.2.1. Gestion des sources	16
5.2.2. Communication	16
6. Description des tests effectués	16
Conclusion	17

1. Rappel du concept et des mécaniques de jeu

1.1. Concept

« Lost in Dinoworld » est un jeu **tout public** jouable sur **mobile et tablette Android**. Il est caractérisé par les genres suivants :

- **Plateforme 2D**
- **Side scrolling** : la progression du joueur s'effectue sur une ligne horizontale de gauche à droite
- **Aventure** : histoire passionnante d'un brave et talentueux scientifique

L'univers de « Lost in Dinoworld » est composé de 6 décors (forêt, savane, désert, montagnes rocheuses, banquise et ville des dinosaures) qui s'enchaînent sans transition. La caméra reste fixée sur le joueur. Seulement 2 régions ont été implémentées pour le prototype. Voici un schéma illustrant la construction de ce monde :



1.2. Histoire

L'histoire se déroule pendant l'année 2150. A ce moment là, une pandémie touche la terre entière et menace d'anéantir toute l'espèce humaine. La recette pour un antidote a été développée, cependant les ressources manquent à sa fabrication.

L'histoire commence quand notre héros du jeu, un brillant scientifique, fabrique une machine à remonter le temps afin de récolter, dans une autre époque, les ingrédients nécessaires à l'antidote. Il se téléporte donc au temps des dinosaures afin de trouver les éléments de sa recette.

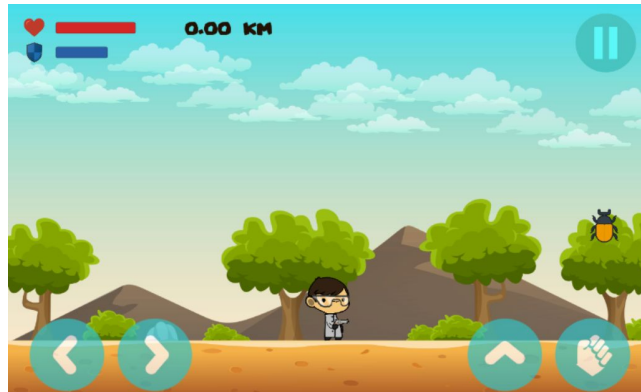
Une fois arrivé à cette époque, le scientifique aura pour but de parcourir le monde des dinosaures en récupérant les ingrédients de l'antidote tout en affrontant les ennemis qu'il croisera sur son chemin. Au fur et mesure, notre héros gagnera en connaissances et en compétences ce qui lui permettra de combattre plus efficacement ses ennemis, tout cela dans le but de regagner son téléporteur avec tous les ingrédients de l'antidote.

1.3. Design

Il a été choisi de créer une ambiance « cartoon » dans l'objectif de :

- toucher un public le plus large possible
- faciliter le développement du jeu

Ainsi les formes ne sont pas agressives et les couleurs sont plutôt pétillantes. Voici une capture d'écran de la région 2 pour illustrer ce propos :



1.4. Gameplay

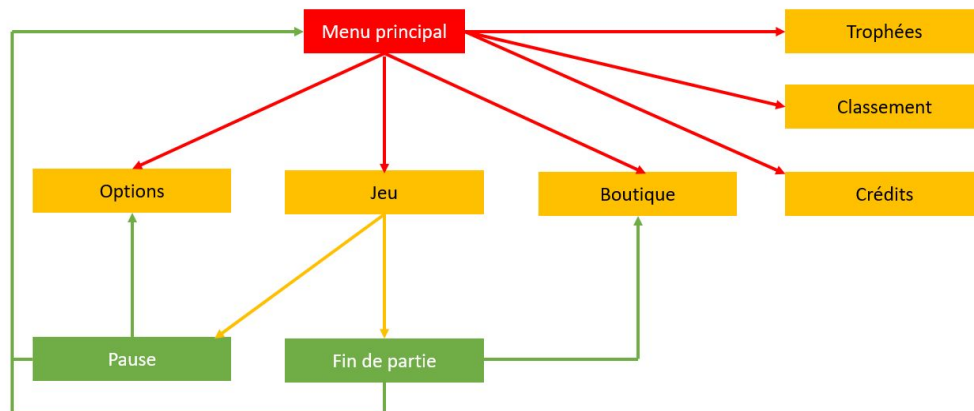
Durant la progression du joueur parmi les 6 régions du monde des dinosaures, la difficulté augmentera au fur et à mesure du défilement de ces régions. Ainsi, le joueur sera d'autant plus challengé que sa progression sera importante. Pour pouvoir gagner, le joueur devra équiper de mieux en mieux son scientifique pour rendre plus accessible les combats des dernières régions. Une boutique est à sa disposition (uniquement accessible dans le menu principal ou en fin de partie) pour acheter et s'équiper d'armes / armures / habiletés spéciales.

Une partie importante du gameplay repose sur l'équilibre entre la difficulté des régions et la diversité des équipements du personnage principal.

2. Présentations des interfaces / fenêtres du jeu

2.1. Interactions possibles entre les interfaces et menus

Le graphe ci-dessous expose la navigation possible entre les différents écrans du jeu :



Remarque : Les fonctionnalités « Trophées » ainsi que « Classement » ne sont pas présentes dans le prototype. Leur menu respectif est vide !

2.2. Menus du jeu

2.2.1. Menu principal

Le menu illustré par l'image ci-dessous correspond au menu du jeu :

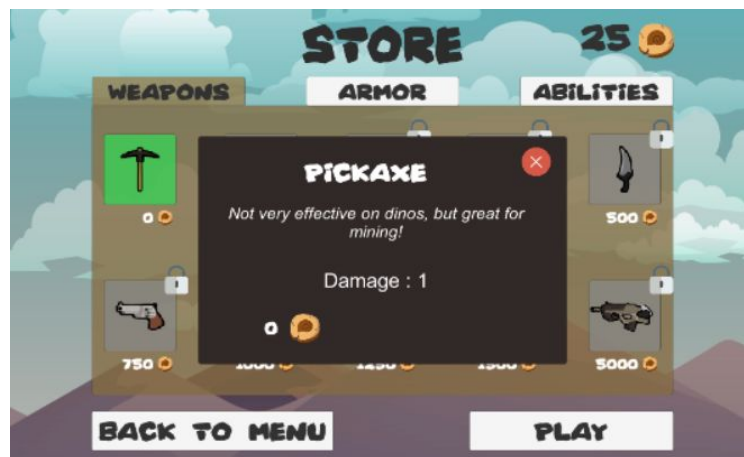


Le joueur peut :

- lancer une nouvelle partie
- accéder à la boutique
- modifier les options
- visualiser les crédits

2.2.2. La boutique (Store)

Le menu illustré par l'image ci-dessous correspond à la boutique du jeu :

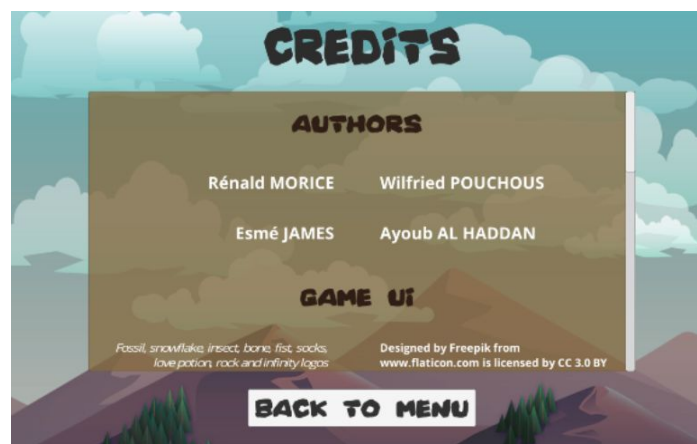


Le joueur peut:

- visualiser son nombre de fossiles
- acheter des armes et s'en équiper
- acheter des armures et s'en équiper
- acheter des habiletés et s'en équiper

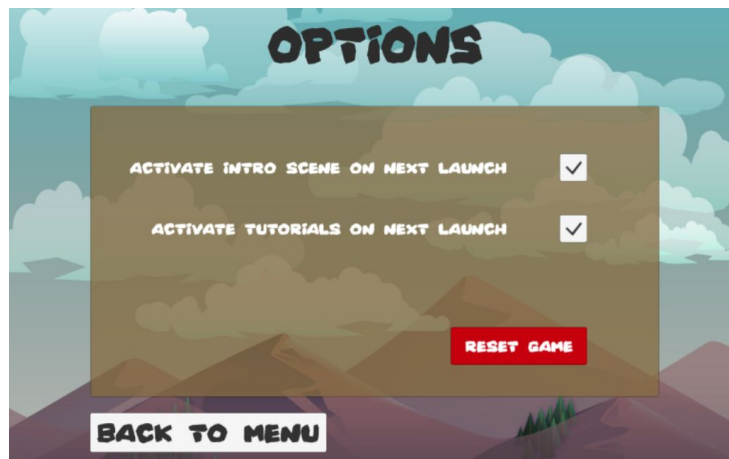
2.2.3. Les crédits

Le menu illustré par l'image ci-dessous correspond aux crédits du jeu :



2.2.4. Les options

Le menu illustré par l'image ci-dessous correspond aux options du jeu :



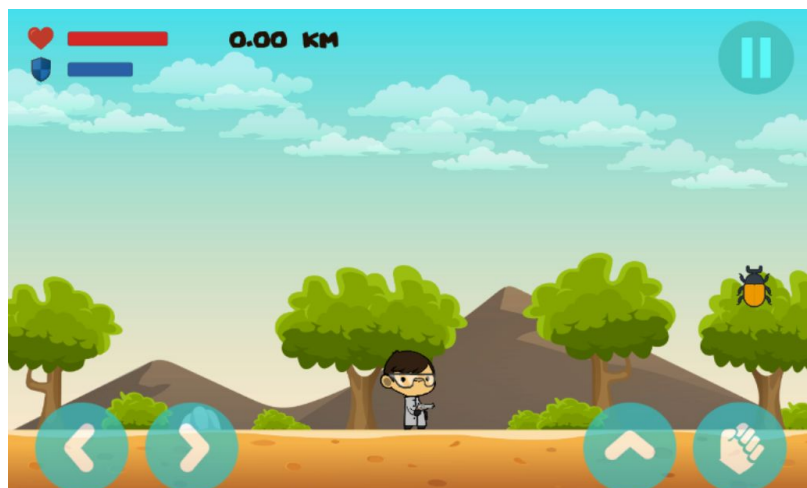
Le joueur peut choisir de :

- relancer la cinématique d'introduction au lancement de la prochaine partie
- activer les tutoriels pour la prochaine partie
- réinitialiser l'ensemble de son jeu (données joueur supprimées)

2.3. Interfaces de jeu

2.3.1. Interface joueur en partie

L'interface illustrée par l'image ci-dessous correspond à l'interface utilisée par le joueur pour interagir avec le monde du jeu :



Le joueur peut :

- visualiser son niveau de vie
- visualiser son niveau d'armure
- visualiser le nombre de kilomètres parcourus
- appuyer sur le bouton « Flèche vers la gauche » pour faire avancer le personnage vers la gauche
- appuyer sur le bouton « Flèche vers la droite » pour faire avancer le personnage vers la droite
- appuyer sur le bouton « Flèche vers le haut » pour faire sauter le personnage
- appuyer sur le bouton « Coup de poing » pour frapper avec le personnage
- appuyer sur le bouton en haut à droite pour accéder à l'interface de « Pause »

2.3.2. Interface « Pause »

L'interface illustrée par l'image ci-dessous correspond à l'interface de pause :



Le joueur peut :

- accéder aux options
- lancer une nouvelle partie
- revenir au menu principal
- re-appuyer sur le bouton en haut à droite pour revenir en jeu

2.3.3. Interface des ingrédients

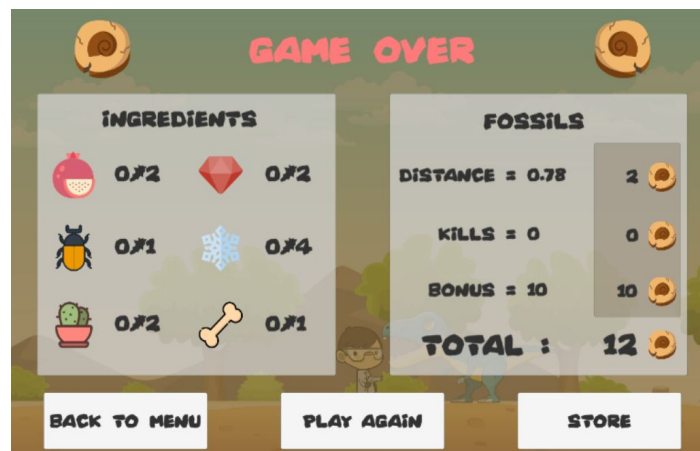
L'interface illustrée par l'image ci-dessous correspond à l'interface permettant au joueur de visualiser l'état de sa récolte des ingrédients :



Cette interface apparaît uniquement à titre d'information, le joueur ne peut pas interagir avec celui-ci.

2.3.4. Interface de fin de partie

L'interface illustrée par l'image ci-dessous correspond à l'interface qui s'affiche lorsque le joueur meurt en partie ou lorsqu'il parvient à atteindre le téléporteur accessible à la fin du jeu :



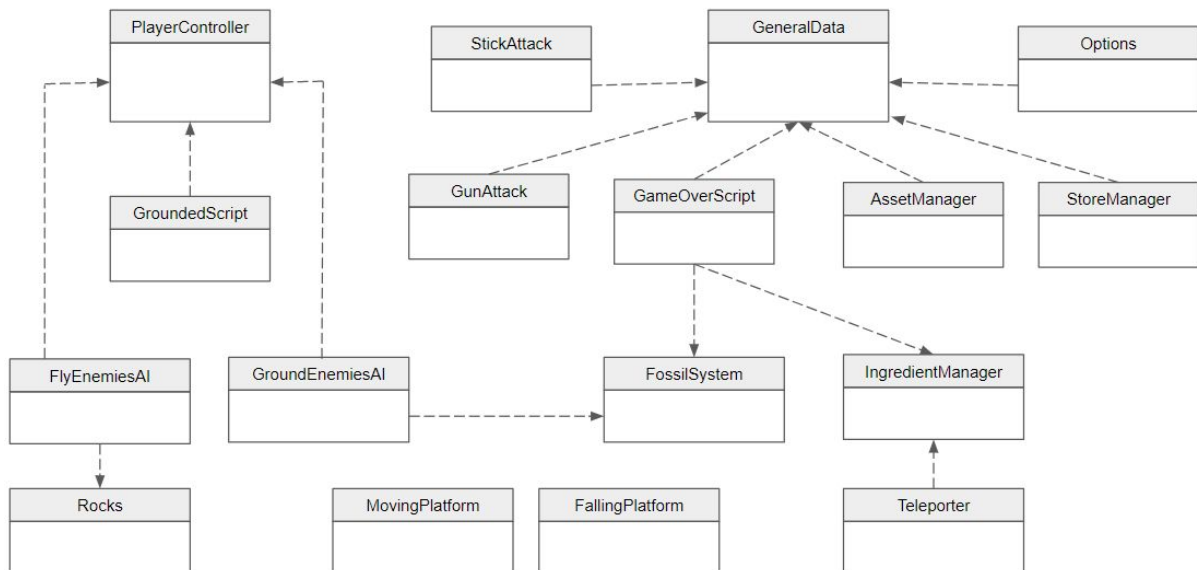
Le joueur peut:

- retourner au menu principal
- lancer une nouvelle partie
- accéder à la boutique
- visualiser le résumé de sa partie

3. Architecture du projet

3.1. Diagramme de classes

Le diagramme ci-dessous représente les relations entre les différentes classes principales de « Lost In DinoWorld » :



Remarque : Seules les classes importantes sont présentes dans ce diagramme afin de faciliter la compréhension de l'architecture du jeu.

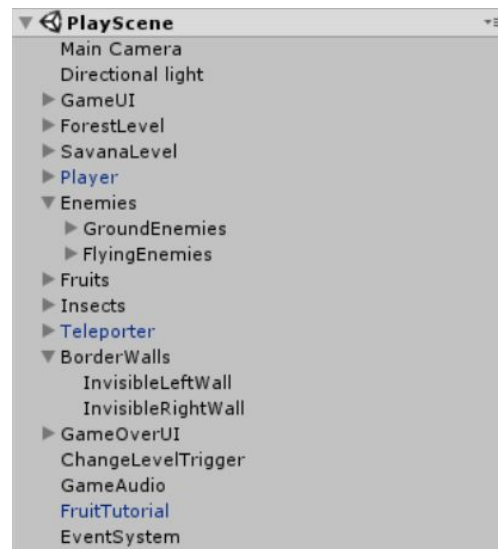
Dans ce diagramme, on peut retrouver les classes suivantes :

- *PlayerController* : permet de gérer le personnage principal contrôlé par le joueur (déplacement, santé, armure, animations, etc...)
- *GroundedScript* : permet de gérer l'entrée dans les airs du personnage ainsi que son atterrissage afin de savoir qu'il est sur un sol
- *FlyEnemiesAI* : permet de gérer le comportement des dinosaures volants (déplacement aérien, estimation de la position future du joueur, jeté de pierres, etc...)
- *Rocks* : permet de créer une pierre qui sera jetée par un dinosaure volant
- *GroundEnemiesAI* : permet de gérer le comportement des dinosaures terrestres (déplacement en solo, déplacement en groupe, attaque, santé, etc...)
- *MovingPlatform* : permet de gérer le déplacement de la plateforme mobile

- *FallingPlatform* : permet de gérer le déplacement de la plateforme évanescence
- *GeneralData* : contient toutes les données du joueur qui peuvent être chargées, sauvegardées voire même réinitialisées
- *StickAttack* : permet de gérer toutes les attaques du personnage principal de type corps-à-corps à une main
- *GunAttack* : permet de gérer toutes les attaques du personnage principal de type arme à distance
- *Options* : permet les options du jeu (accessibles via le menu Options)
- *StoreManager* : permet de gérer la boutique du jeu (achat et / ou équipement d'une arme / armure / habileté)
- *AssetManager* : permet de gérer le chargement de l'arme / armure / habileté équipées par le joueur lors de lancement d'une nouvelle partie
- *GameOverScript* : permet de gérer l'écran de fin de partie (ingrédients récoltés, fossiles emmagasinés, etc...)
- *FossilSystem* : permet de comptabiliser le nombre de fossiles gagnés en partie (ennemis tués, distance parcourue, régions franchies, etc...)
- *IngredientManager* : permet de gérer la récolte des ingrédients en partie
- *Teleporter* : permet de gérer le téléporteur présent en fin de jeu (tout à droite de la dernière région)

3.2. Hiérarchie de la scène de jeu (PlayScene)

La hiérarchie de la scène principale de jeu (PlayScene) a été pensée et organisée au mieux afin d'assurer une facilité dans la création et la modification de celle-ci. Ci-dessous une capture d'écran de cette hiérarchie :



3.3. Exemple de code

Pour comprendre l'architecture globale du code ainsi que les liens entre les classes, voir le point 3.1 de ce document.

Voici un exemple de code avec la fonction de "Move()" qui permet de déplacer horizontalement un dinosaure terrestre vers sa cible qui est le personnage principale :

```

145 //Move Dino if needed
146 //-----
147 //référence
148 private void Move() {
149     if (IsInCameraFOV()
150         && !HasDelimiterConstraint()
151         && !IsAtEndOfGround()
152         && !HasAllyOnTheWay()
153         && !IsNearTarget()
154     ) {
155
156         //Flip sprite if necessary
157         Flip();
158
159         //Be sure that Dino is following only horizontally
160         Vector2 targetPos = target.position;
161         targetPos.y = transform.position.y;
162
163         //Move
164         transform.position = Vector2.MoveTowards(transform.position, targetPos, speed * Time.deltaTime);
165
166         //Run animation
167         Run();
168
169         // Play the sound of footsteps
170         if (!Audio.isPlaying) Audio.PlayOneShot(StepsSound);
171     }
172     //If it can move
173     else if(!animator.GetCurrentAnimatorStateInfo(0).IsName("Raptor_Hurt")) StopRun();
174 }
175 }
176

```

Explications :

- *Lignes 149 à 153* : le dinosaure terrestre peut bouger :
 - s'il est dans le champ de vision du joueur
 - s'il n'est pas en dehors des limites spatiales qu'il lui sont imposées
 - s'il n'est pas proche d'une extrémité du sol (risque de tomber dans le vide)
 - s'il n'est pas collé (trop proche) à un allié qu'il suit dans la même direction
 - s'il n'est pas collé (trop proche) à sa cible
- *Ligne 157* : le dinosaure terrestre change d'orientation horizontale si le joueur passe dans son dos.
- *Ligne 164* : le dinosaure terrestre est translaté afin de le mouvoir.
- *Ligne 167* : la fonction "Run()" est appelée pour lancer l'animation "Courir" du dinosaure.
- *Ligne 170* : le bruit sonore de déplacement du dinosaure terrestre est lancé s'il n'est pas déjà actif.
- *Ligne 173* : Si une ou plusieurs conditions ne sont pas remplies aux lignes 149 à 153, c'est que le dinosaure terrestre ne doit pas bouger. Dans ce cas la méthode "StopRun()" est appelée à condition que le dinosaure ne soit pas dans l'état "Hurt".

4. Répartition des tâches

4.1. Division en sprints

Pour la répartition des tâches, nous avons réalisé un Product Backlog recensant l'ensemble des tâches à effectuer en début de projet. Ensuite, nous avons divisé celui-ci en Sprints Backlog répartis sur les quatre semaines de développement par priorité : les tâches essentielles en premier et les tâches secondaires dans les dernières semaines. Ci-dessous un tableau résumant les tâches réalisées par sprint :

Sprint 1	Sprint 2	Sprint 3	Sprint 4
Contrôles joueur Animations héro Décors régions Plateformes	Animations IA ennemis Android Vie et Ingrédients	Système de fossiles Store Level design Sons et musiques	Armes Sauvegarde Cinématiques Tutoriels

Les semaines après le sprint 4 ont été allouées à la correction des bugs et aux diverses tâches de « nettoyage » de fin de projet.

4.2. Répartition des rôles

Pour la répartition des rôles, nous décidions à chaque début de sprint qui se chargerait de réaliser une tâche donnée. Les noms des responsables étaient inscrits dans la ligne correspondant à cette tâche dans le sprint backlog associé afin de faciliter l'organisation des semaines. Dans l'esprit de garder une vision globale du projet, chaque membre s'est vu faire des tâches diverses dans le projet. Ci-dessous un tableau résumant les responsabilités majeures de chaque membre :

Wilfried	Rénald	Esmé	Ayoub
Tutoriels Store Ingrédients Animations Correctifs	Level design Système de vie / dégât Android Animations Sauvegarde Plateformes	Système de fossiles Store Android Animations Cinématiques Menus	Sons et musiques Système de vie / dégât IA ennemis Animations Armes

Le tableau ne reporte que les tâches les plus globales du projet et ne prend pas en compte le temps de travail de chacune d'entre elles. Les tâches réalisées à plusieurs sont affichées en couleurs.

5. Choix de conception et technologies utilisées

5.1. Choix de conception

5.1.1. Plateforme Android

Nous avons fait le choix d'orienter le développement de notre jeu pour une sortie sur plateforme Android. Ce choix a été retenu car il permet une bonne visibilité du jeu à travers le Play Store. De plus, la facilité de téléchargement d'un jeu mobile depuis le Play Store permet de le rendre plus accessible à un public large.

La mise en place des technologies permettant une sortie sur Android a été faite à travers l'utilisation des assets généraux de Unity prévus à cet effet.

5.1.2. Choix des assets

Nous avons utilisé en majorité des assets gratuits et libres de droits pour le personnage principal, les décors et les musiques. Nous avons choisi des assets de différentes origines mais qui ont une cohérence graphique et/ou d'ambiance dans le but de rester dans le thème « Cartoon » voulu dans le jeu.

Pour le personnages ainsi que les dinosaures, nous avons utilisé des sprites découpés par "partie du corps". Ainsi, il a fallu créer toutes les animations de chaque état des personnages du jeu grâce à l'outil Unity permettant d'effectuer des transformations et des rotations sur chaque sprite les constituant. Pour le héros par exemple, il a fallu créer des animations pour les états :

- Idle (le personnage est en attente)
- Courir
- Sauter
- Frapper
- Tirer
- Blessé
- Mort

5.2. Technologies utilisées

5.2.1. Gestion des sources

Pour la gestion des sources du projet, nous avons utilisé l'outil Collaborate intégré à Unity. Ceci a permis de garder un projet consistant selon les modifications de chacun au fil du temps et donc de perdre le moins de temps possible dans la gestion des conflits du code source.

5.2.2. Communication

Pour communiquer et réaliser des réunions virtuelles régulières, nous avons utilisé l'outil Discord. Ceci nous permettait de discuter des modifications faites ou à faire et donc de contrôler les étapes de développement du projet.

6. Description des tests effectués

En ce qui concerne les tests effectués, chaque membre de l'équipe avait pour responsabilité de tester les fonctionnalités qu'il venait de développer. Une fois vérifié et corrigé si nécessaire, le développement était intégré à la version stable du jeu.

Par la suite, au moins un autre membre du groupe se désignait pour tester les fonctionnalités qui venaient d'être ajoutées. Si des bugs étaient remarqués, ils étaient signalés à la personne responsable. En fonction de l'importance du bug et de sa rapidité à être résolu, le bug était soit :

- bloquant : dans ce cas il était corrigé immédiatement
- non bloquant : dans ce cas il était noté dans une liste de corrections / améliorations à effectuer plus tard

En fin de sprint, une discussion avait lieu à propos des tâches à effectuer pour le prochain sprint et la possibilité de corriger les bugs non bloquants s'il nous restait du temps de disponible. Par exemple, à la fin du Sprint 2, cette discussion a mené à la mise en pause du Sprint 3 tant que toutes les corrections notées n'étaient pas réglées.

De plus, Rénaud ou Esmé étaient chargés de tester l'application sur Android au fur et à mesure de la semaine.

Conclusion

Au bout d'un mois de développement, nous sommes parvenus à développer un prototype remplissant 90% des attentes prévues. Nous considérons donc ce prototype comme complet, étant donné que les tâches restantes non développées n'étaient pas primordiales à nos yeux concernant le prototype.

De plus, ce prototype a été conçu de façon à ce qu'il soit facilement évolutif par la suite. En effet, nous souhaitons poursuivre la réalisation de ce projet (développer les 4 régions manquantes) pour qu'il soit disponible très prochainement sur le Play Store (un email vous sera envoyé lors de la sortie du jeu).