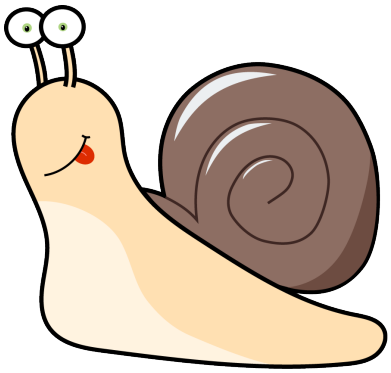
Rapport de gestion de projet

Jeu 2D pour mobile
Heliko



Auteurs
Thibaut Castanié
Jolan Konig
Noé Le Philippe
Stéphane Wouters

Encadrant
Mathieu Lafourcade
Master
IMAGINA

Année universitaire 2014-2015

Table des matières

Table des figures

1 Présentation du projet

Nous avons proposé notre propre projet dans l'optique de créer un jeu pour téléphone mobile, de sa conception jusqu'à sa mise en ligne. Le concept de base est de créer un "jeu de rythme", à l'aide du moteur de jeu Unity. Nous nommerons notre jeu **Heliko** (Escargot en Espéranto).

La difficulté relève dans un grand travail de recherche : le jeu doit être intéressant et être réalisable afin de pouvoir le mettre en ligne dans le temps donné. La méthode de gestion de projet avait une grande importance.

FIGURE 1 – Extrait du jeu final

2 Analyse de départ

Nous avons dès le départ conçu les parties du jeu pour qu'il soit jouable et en ligne au bout de 4 mois de travail. Étant donné notre légère expérience dans la gestion du temps, nous avons conçu des "couches" à développer pour former le jeu de façon incrémentale et ainsi être certain d'aboutir à un résultat (à condition de prendre en compte les tests, les finitions et la mise en ligne).

Couches incrémentales

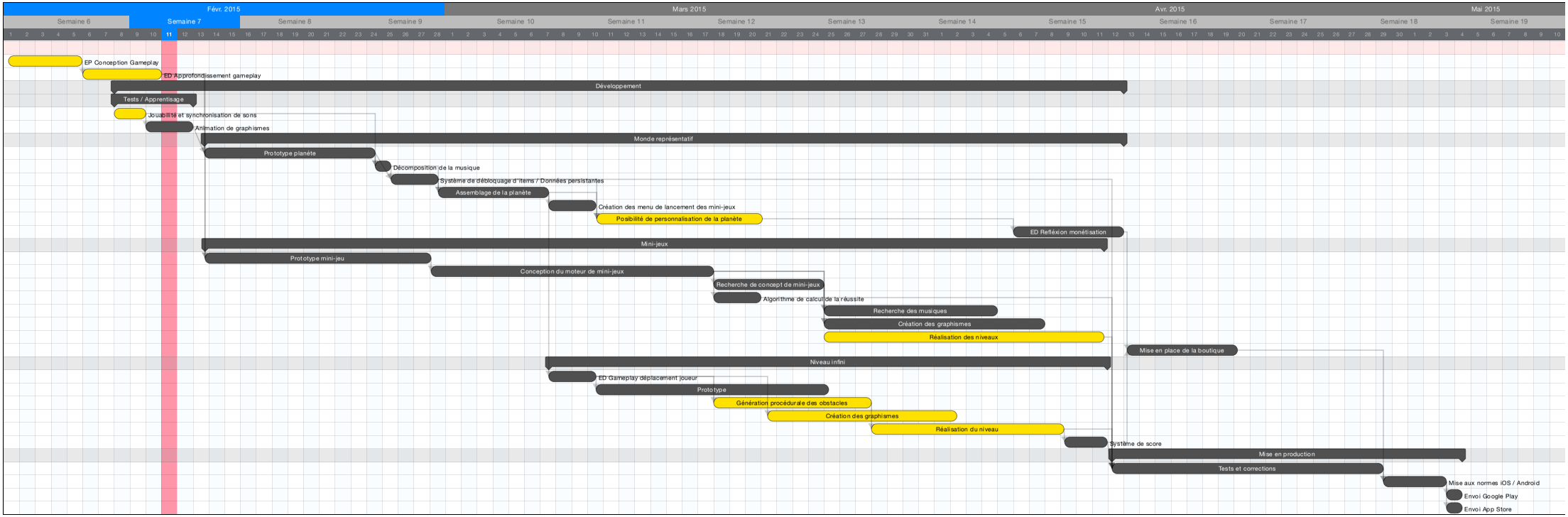
1. Créer un moteur de jeu de rythme Unity générique et créer un mini jeu jouable sur Android ;
2. Rendre l'application jouable sur tous les mobiles et supports ;
3. Créer d'autres mini jeux ;
4. Créer un moteur de tutoriel et les tutoriels associés ;

5. Concevoir et ajouter une monétisation (gain de pièces dans les niveaux, et personnalisation du jeu) ;
6. Concevoir et ajouter un esprit communautaire (partage de l'avancement) ;
7. Créer un éditeur public de niveaux contributif.

Et le projet est sous-divisé trois grandes étapes :
— Conception et prototypes ;
— Développement du jeu ;
— Tests, finitions et mise en ligne (Play Store et Apple Store).

Ces 3 étapes demandant une gestion de projet différente, la méthode de travail a été modifiée au fil du développement.

Le **diagramme de Gantt** initial a été conçu dans l'optique de faire la totalité des couches énoncées plus haut.


Févr. 2015
Mars 2015
Avr. 2015
Mai 2015
Semaine 6
Semaine 7
Semaine 8
Semaine 9
Semaine 10
Semaine 11
Semaine 12
Semaine 13
Semaine 14
Semaine 15
Semaine 16
Semaine 17
Semaine 18
Semaine 19
EP Conception Gameplay
ED Approfondissement gameplay
Développement
Tests / Apprentisage
Jouabilité et synchronisation de sons
Animation de graphismes
Monde représentatif
Prototype planète
Décomposition de la musique
Système de débloquage d'items / Données persistantes
Assemblage de la planète
Création des menu de lancement des mini-jeux
Posibilité de personnalisation de la planète
ED Réflexion monétisation
Mini-jeux
Prototype mini-jeu
Conception du moteur de mini-jeux
Recherche de concept de mini-jeux
Algorithme de calcul de la réussite
Recherche des musiques
Création des graphismes
Réalisation des niveaux
Mise en place de la boutique
Niveau infini
ED Gameplay déplacement joueur
Prototype
Génération procédurale des obstacles
Création des graphismes
Réalisation du niveau
Système de score
Mise en production
Tests et corrections
Mise aux normes iOS / Android
Envoi Google Play
Envoi App Store

Rôles dans l'équipe Vu la charge importante de travail et la diversité des tâches, nous avons préféré nous attribuer des responsabilités fixes :

- **Stéphane Wouters**, chef de projet ;
- **Noé Le Philippe**, responsable développement technique ;
- **Thibaut Castanié**, responsable son et graphismes ;
- **Jolan Konig**, responsable intégration et publication.

Et ce, tout en faisant le choix de travailler ensemble sur toutes les parties et en s'affectant au fil du projet des micro tâches.

3 Outils et méthode de travail

3.1 Gestionnaire de tâches

Pour faire avancer le projet, nous avons utilisé toutes les capacités d'un gestionnaire de tâches en ligne, **Trello**, qui est un outil inspiré par la *méthode Kanban*. Nous nous sommes imposé de visiter ce tableau tous les jours et ses outils de notifications nous ont permis d'être continuellement connectés.

Trello permet une gestion des tâches dans le Cloud avec de nombreuses fonctionnalités :

- Création de tâches, avec titre et description ;
- Choix d'une date de fin sur une tâche ;
- Affectation de membres à une tâche ;
- Labels (tags) personnalisés ;
- Commentaires sur chaque tâche pour discussion asynchrone entre les membres du groupe sur une tâche ;
- Application Android et iOS avec notifications par *push*.

Un système de rangement vertical a été adopté pour les types de tâches, et des labels de couleurs en fonction de l'avancement des tâches :

- **Bloquante** (tâche à réaliser rapidement, bloquant l'avancement du projet) ;
- **À discuter / en recherche** (tâche en cours de discussion, pour prise de décision) ;
- **À attribuer** (Tâche correctement spécifiée, en attente d'affectation à un membre de l'équipe ;
- **En réalisation** (tâche en cours de réalisation par un ou plusieurs membres de l'équipe) ;
- **À tester / à contrôler** (tâche réalisée, à tester pour confirmation) ;
- **Fait** (tâche réalisée et fonctionnelle, prête à être archivée).

Les tâches sont classées dans des colonnes “TODO” triées par thèmes (développement, graphismes. . .), ou par cycle itératif (d’un jour à une semaine).

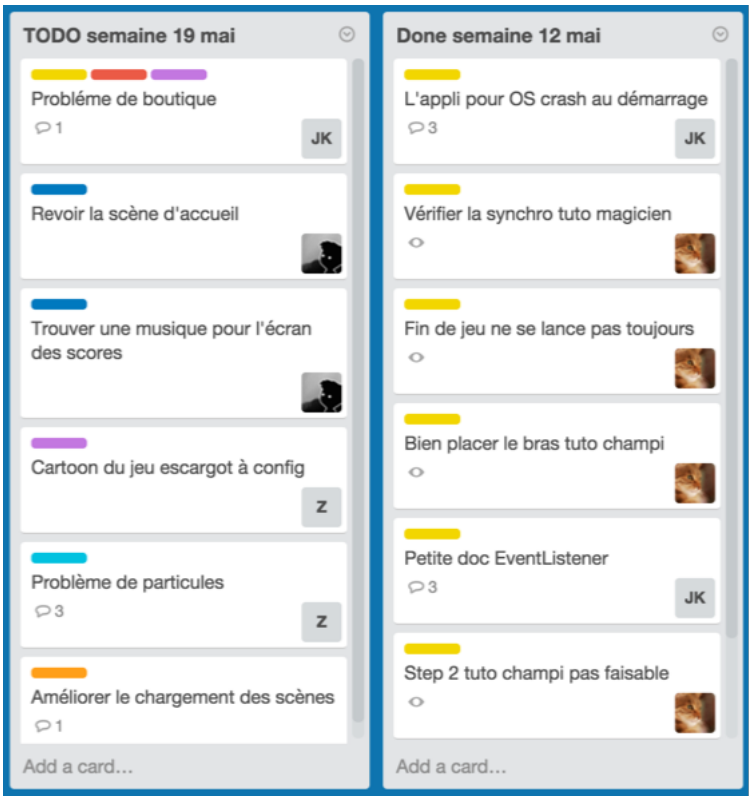


FIGURE 2 – Exemple de colonnes en fin de projet

Trello a aussi été utilisé comme mémo et pour archiver les ressources graphiques et sonores (figure 3).

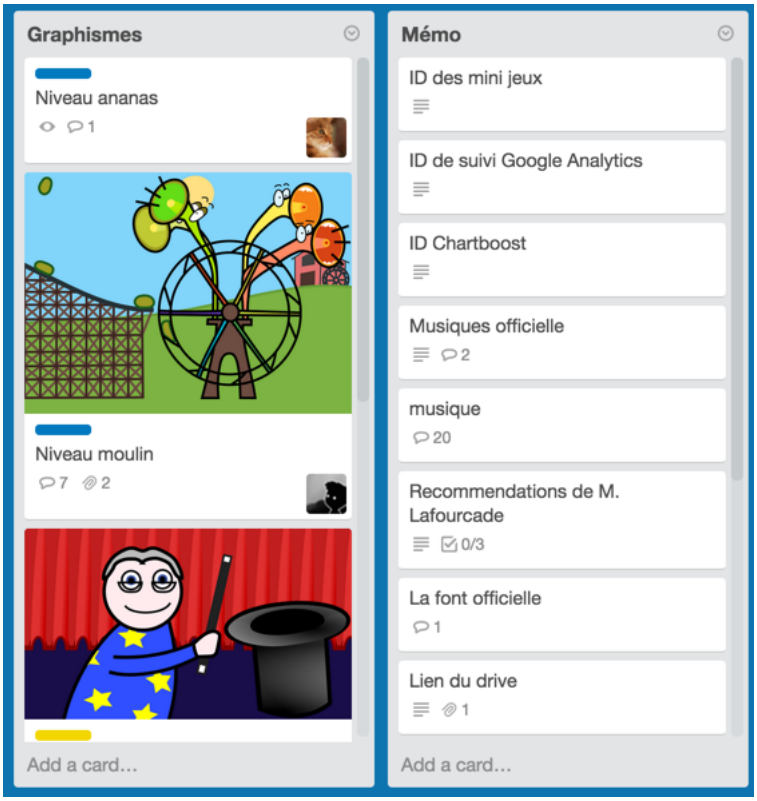


FIGURE 3 – Utilisation de Trello comme mémo et archives

Cette méthode de travail avec Trello nous a permis d'être efficace 100% du temps au travers d'Internet. Il n'y avait jamais de temps mort et nous étions toujours clairs dans notre direction tant que quelqu'un (chef de projet) s'occupait de créer des tâches et de les organiser.

3.2 Réunions

Même si *Trello* nous permet de travailler de façon indépendante, nous nous réunissions très régulièrement pour travailler ensemble et fixer les nouveaux objectifs, d'une à quatre fois par semaine.

Pendant les longues séances de travail en collaboration (de 9h à 18h par exemple), nous utilisions un véritable tableau blanc pour noter les tâches en cours et les affectations. Exemples de tâches :

— Réadapter la taille du logo *pause* sur Android v4.1 en écran 16 :10 ;
— Couper les 150ms du début du son `tic.wav` ;
— Revoir l'animation du bras gauche du champignon.

Les tâches étant à 80% de ce type (courtes et rapides), l'organisation est très importante pour être efficace.

De nombreuses heures de réunion étaient dédiées à la recherche de concept ou de remise en question des objectifs. Par exemple abandonner le développement d'un mini jeu trop complexe, ou en inventer de plus simples.

3.3 Cycles itératifs

Dès le départ, un développement par cycles itératifs a été choisi. Bien adapté pour le développement d'un jeu vidéo et surtout à cause de notre contrainte principale : notre liberté sur les choix.

Les premières semaines ont été dédiées à la réalisation de prototypes et de tests en augmentant toujours la difficulté, dans l'objectif de produire un moteur de jeu de rythme fonctionnel qui correspond aux besoins définis dans le cahier des charges.

- **Prototype 1** - Réalisation d'un cube qui bat à un rythme constant (3 jours)
- **Prototype 2** - Réalisation d'un prototype de test de réussite (2 jours)
- **Prototype 3** - Création de la première version du moteur de rythme (6 jours)
- **Prototype 4** - Réalisation d'un prototype d'animations, connectés au moteur de rythme (4 jours) -> *Le moteur n'est pas assez précis et doit être revu*
- **Prototype 5** - Deuxième version du moteur de rythme (3 jours)
- **Prototype 6** - Nouvelle tentative de connexion à des animations (1 jour) -> *Test OK. On s'aperçoit que nous avons besoin de quart de temps dans le modèle et qu'il faut recommencer une nouvelle fois sa conception*
- **Prototype 7** - Troisième version du moteur de rythme (4 jours)
- Etc.

FIGURE 4 – Les différents projets Unity de tests

On voit sur la figure 4 l'ensemble de nos tests. A chaque nouveau test, un nouveau projet Unity. Au **test 14**, nous avons jugé que le moteur correspondait à nos attentes et nous avons ensuite itéré directement sur ce projet. C'est à ce moment (environ 1 mois après le début du projet) que le système de fonctionnement à changé et que nous avons fonctionné en micro tâches.

Les dernières semaines ont été dédiées aux finitions sur le projet afin de rendre le jeu agréable avec un aspect "fini".

3.4 Github

Un gestionnaire de version pour notre projet a bien sûr été utilisé et nous avons choisi le gestionnaire *GitHub* qui offre de nombreux outils de statistiques très intéressants.

Déjà habitués à Git, nous l'avons utilisé pleinement et nous avons envoyé des *commits* pour chaque modification fonctionnelle. Nous sommes ainsi arrivés en fin de projet avec un cumul de **900 commits**, globalement bien répartis entre nous quatre. (Avec plus de 2 millions de modifications dans les fichiers...).

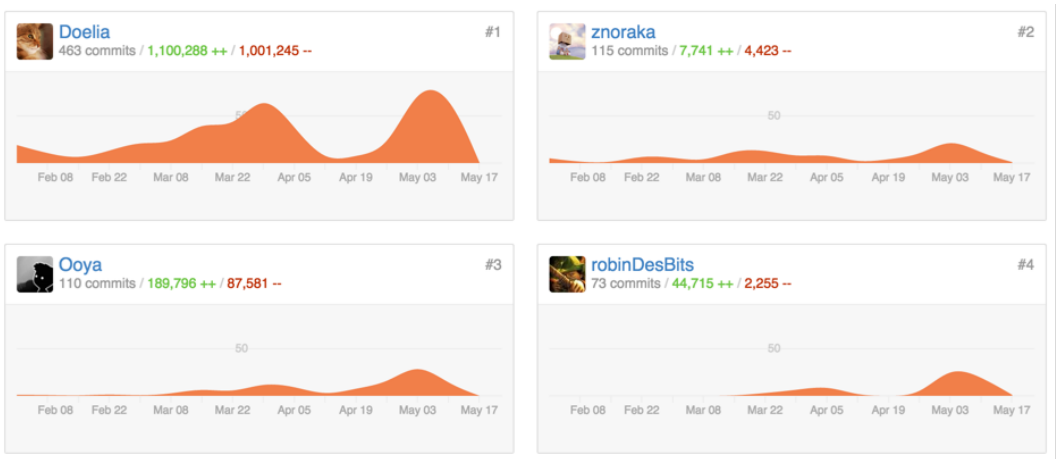


FIGURE 5 – Répartition des commits dans le temps

On remarque d'après la figure 5 de nombreuses suppressions (presque autant que d'additions), prouvant l'évolution du projet et l'application des cycles itératifs. Le développement s'est fait continuellement dans le temps.

3.4.1 Ponctualité hebdomadaire

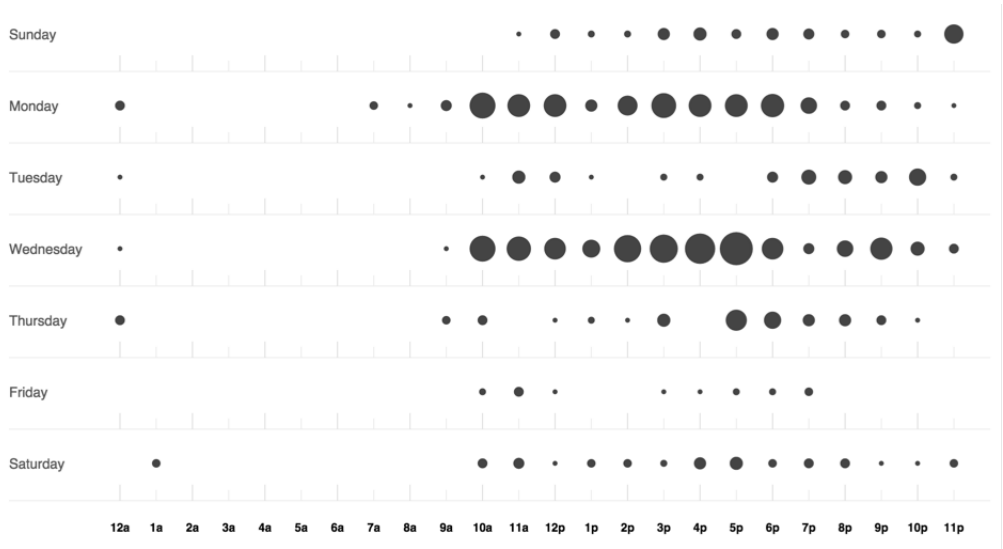


FIGURE 6 – Punch card de Github

D'après les statistiques *Github* (figure 6), nous avons travaillé tous les jours de la semaine, avec une préférence pour le lundi et le mercredi en après-midi et jusqu'en fin de soirée.

3.4.2 Estimation du temps passé

Estimation globale Nous nous sommes réunis au moins une fois par semaine pour travailler, sur des séances d'environ 8h sur une journée. Sur 4 mois, cela représente un total de 128 heures par personnes, et 512 heures au total. A cela nous pouvons y ajouter toutes les heures de travail indivuduels, que nous chiffrons facilement à 30% du projet. Soit **665 heures** au total.

Estimation assistée Etant donné que nous envoyons très régulièrement des commits, nous avons essayés d'utiliser un outil de calcul : **git-hours**. Il calcule les heures de travail en fonction des écarts entre les commits. (Mais ne compte évidemment que le temps de développement pure, les moments passés à disctuter et débattre autour du projet en équipe ne sont donc pas comptabilisés. L'application retourne un total de **29822** minutes, soit **497 heures**. Ce qui semble totalement cohérant avec notre première éstimation, en ajoutant les heures de débats en réunion.

Convesion en jours/hommes En prenant une base de 35h par semaines, cela représente un travail de **133 jours/homme** *(665/35 * 7 jours)*.

3.4.3 Modélisation Gource

Gource est une application qui permet de tracer en vidéo l'historique des commits qui construit l'architecture des fichiers d'un projet GIT.

Cette capture (figure 7) représente l'état final de l'architecture des fichiers du projet. La taille de la branche des tests démontre qu'ils ont effectivement constitué une grande partie du projet (Environs 30%).

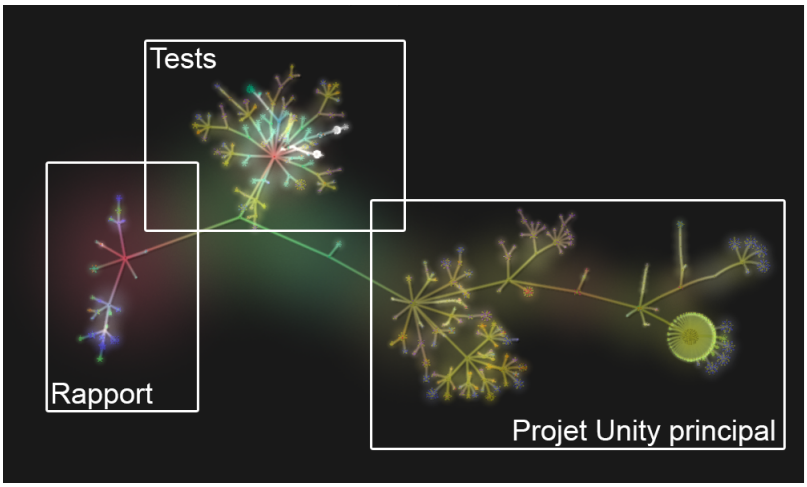


FIGURE 7 – Modélisation Gource en fin de projet

4 Résultat et comparaison avec le prévisionnel

Globalement à 2 semaines avant la date finale, on peut admettre que l'objectif a été atteint. Le jeu est jouable et tout est prêt pour qu'il soit mis en ligne sur l'Apple Store et le Play Store. Les tests auprès des utilisateurs ont été concluants.

La liste des *couches* (prévues dans l'analyse) qui ont été développées est la suivante :

- 3 mini jeux pleinement fonctionnels ;
- Des tutoriels pour chacun des mini jeux ;
- Jeu jouable sur tous les téléphones sous Android et iOS ;
- Système de déblocage des mini jeux et enregistrement du taux réussite ;
- Monétisation avec de la publicité.

Les éléments suivants n'ont pas été développés, par choix :

- Système de personnalisation ;
- Mode de jeu infini répétitif ;
- Éditeur de niveau collaboratif.

Ces choix de retrait se sont faits suffisamment tôt dans le projet (à mi-parcours pour être certain d'aboutir à un jeu fonctionnel). Nous avons préféré nous concentrer sur les tâches listées si dessus. Ce fut tout de même un travail important.

Nous sommes globalement satisfaits de notre gestion du temps, et nous pensons que notre organisation y a beaucoup contribué. L'expérience de créer un jeu de A à Z nous a beaucoup appris et nous a permis de mettre en oeuvre beaucoup de concepts sur la gestion de projets.