



## 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

# 1 Présentation du projet

Nous avons proposé ce projet dans l’optique de créer un jeu pour mobile (plateforme iOS et Android), de sa conception jusqu’à sa mise en ligne. Le concept de base est de créer un “jeu de rythme”.

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

# 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. Etant 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 (A 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 tout les mobiles et supports
3. Créer d’autres mini jeux
4. Créer un moteur de tutoriel et les tutoriels associés Concevoir et ajouter une monétisation (gain de pièces dans les niveaux et personnalisation du jeu)
5. du jeu)
6. Concevoir et ajouter un esprit communautaire (partage de l’avancement)

Et le projet est sous-divisé 3 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, nous avons modifié notre organisation avec le temps.

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

Pour ce qui est des rôles dans l'équipes, vu charge importante de travail et de la diversité des tâches, nous nous avons préférés nous accorder à chacun 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

En faisant le choix de travailler ensemble sur toutes les tâches 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 utilisés toutes les capacités d'un gestionnaire de tâches en ligne, **Trello**. Nous nous sommes forcés de visiter ce tableau tout les jours et ses outils de notifications nous ont permit d'être continuellement connectés et au courant de l'avancement du projet.

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

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

Nous avons adoptés un système de rangement vertical pour les types de tâches, et des labels de couleurs en fonction de l'avancement des tâches :

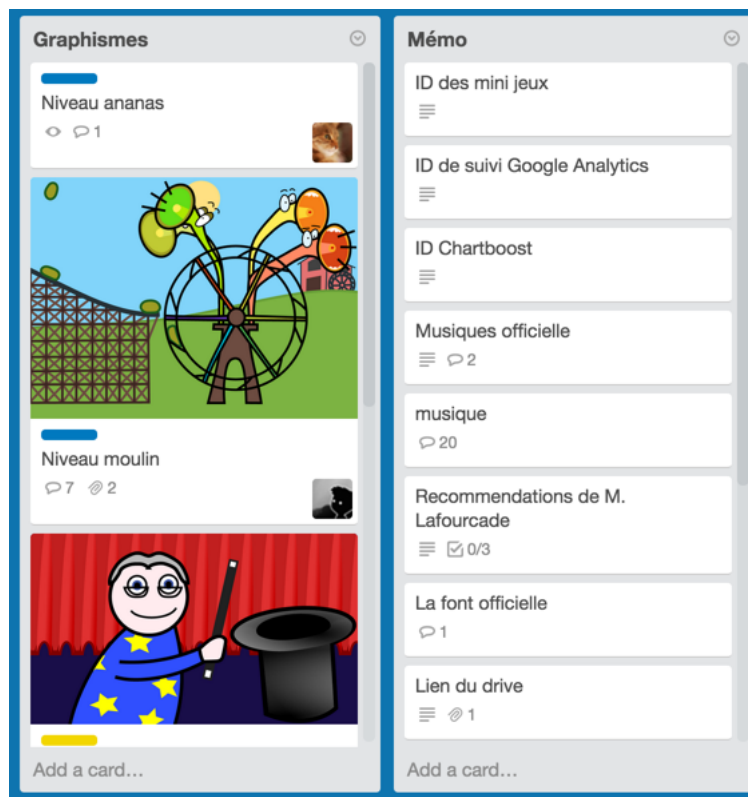
- **Bloquante** (tâche à réaliser rapidement, bloquant l'avancement du projet)
- **A discuter** / **En recherche** (tâche en cours de discussion, pour prise de décision)
- **A attribuer** (Tâche correctement spécifiée, en attente d'affectation à un membre de l'équipe)
- **En réalisation** (tâche en cours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

On crée des colonnes “TODO” par thème (développement, graphismes...), ou par cycle itératif (d’un jour à une semaine).



*Exemple de colonnes*

Trello a aussi été utilisé comme mémo et pour archiver les ressources graphiques et sonores.



*Utilisation de Trello comme mémo et archives*

Cette méthode de travail avec Trello nous a permis d'être efficace 100 : du temps au travers de l'Internet. Il n'y avait jamais de temps mort et nous savions toujours dans quelle directions nous allions tant que quelqu'un (le chef de projet) s'occupait de créer des tâches et de les organiser.

### 3.2 Réunions

Même si Trello nous permettait de travailler de façon indépendante, nous nous réunissions très régulièrement pour travailler ensemble et fixer les nouveaux objectifs, de une à deux fois par semaines.

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

- 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 dédiés 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

Nous avons adopté dès le départ le développement par cycle itératif, bien adapté pour le développement d'un jeu vidéo et surtout à cause de la contrainte principale : notre liberté sur les choix.

Nous avons dédiés les premières semaines à la réalisation de prototypes et de tests en augmentant toujours la difficultés, 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'une 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 3 - 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 4 - Deuxième version du moteur de rythme (3 jours)
- Prototype 5 - 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
- Etc.



*Nos différents projets Unity de tests*



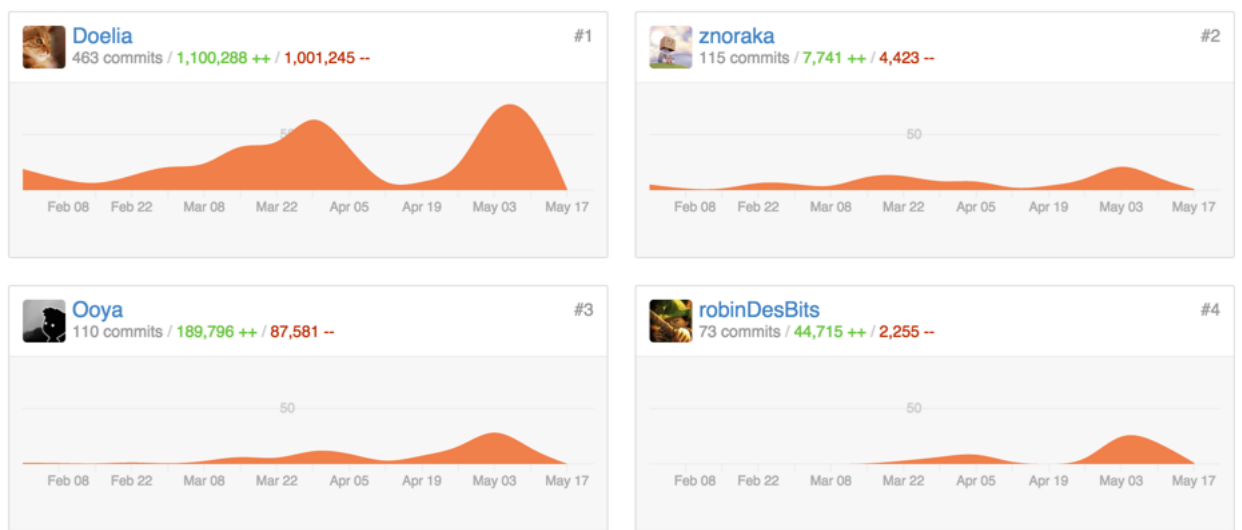
Avec à chaque nouveau test, un nouveau projet Unity. Au test 14, nous nous sommes rendu compte que le moteur correspondait à nos attentes, et nous avons ensuite itéré directement sur ce projet. C'est à ce moment (environ 1 mois après le début du projet) que le système de fonctionnement a changé et que 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

Nous avons bien sûr utilisé un gestionnaire de version pour notre projet et nous avons choisi le gestionnaire GitHub qui offre de nombreux outils de statistiques très intéressants.

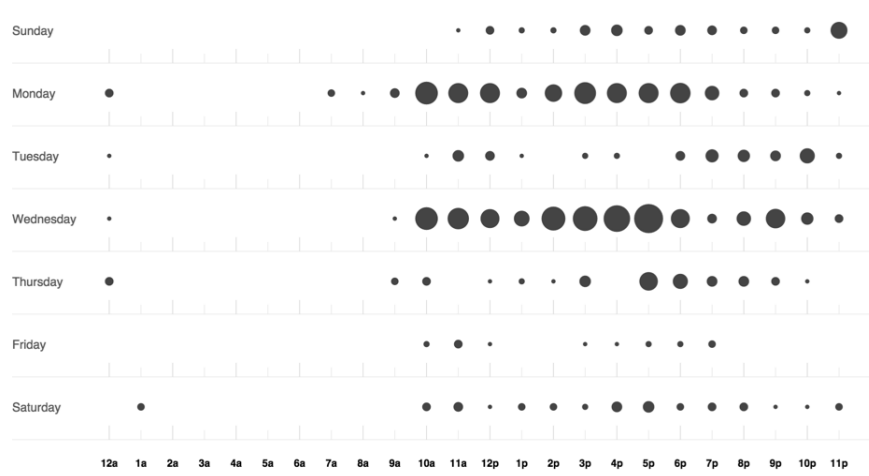
Déjà habitués à Git, nous avons su l'utiliser pleinement et nous avons envoyés des commits pour chaque modification fonctionnelle. Nous sommes ainsi arrivés en fin de projet avec un cumul de **880 commits**, globalement bien répartis entre nous quatre. (Avec plus de 2 millions de modifications dans les fichiers...).



*Répartition des commits dans le temps*

On remarque aussi de nombreuses suppressions (presque autant que d'addition), prouvant l'évolution du projet et l'application des cycles itératifs. On note également que le développement s'est fait continuellement dans le temps, sans aucune coupure.

### 3.4.1 Ponctualité hebdomadaire

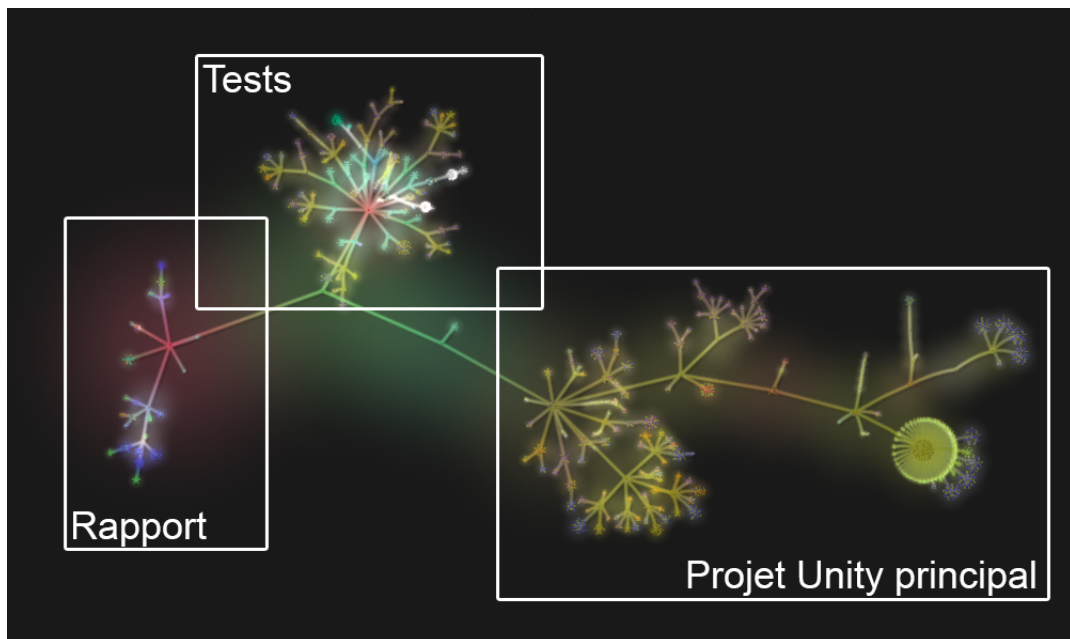


*Punchcard de Github*

D'après les statistiques Github, nous avons travaillé tout les jours de la semaine, mais avec un avantage pour le lundi et le mercredi en après midi et jusqu'en fin de soirée.

### 3.4.2 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.



### *Modélisation Gource en fin de projet*

Cette capture représente l'état final de l'architecture des fichiers du projet. On remarque la taille de la branche des tests qui a effectivement constituée une grande partie du projet (Environs 30