



Cahier des charges

Thibault Gdalia
Florent Youinou
Mathilde Laplaze
Vincent Baille

17 janvier 2014



Table des matières

Introduction	2
1 Avancement du Projet	3
1.1 Moteur Physique	3
1.2 Réseau	4
1.3 Son	4
1.4 Éditeur de Map	5
1.5 Site Web	5
1.6 Graphique	5
2 Et Après ?	7
2.1 Map Editor 2.0	7
2.2 Réseaux Multijoueur	7
2.3 Nouveaux Personnage	7
2.4 Graphismes	7



Introduction

Nous sommes la Team Girafe. Un groupe de 4 étudiants composé de Mathilde "Mattou" Laplaze, Vincent "Vincæ" Baille, Florent "T4ze" Youinou, Thibault "Skeat" Gdalia.

Nous produisons actuellement un runner 2D, où le joueur parcourt nos maps à l'aide d'un Houla-Houla. Cet animal est un oiseau très gourmand, qui à chaque effort, consomme de l'énergie, sous forme de sucre. Mais étant donné que cette énergie lui est nécessaire pour voler, des bonbons sont parsemés tout au long de son parcours et son taux de sucre augmente avec le temps.

Le jeu est composé de différents modes. Le mode solo où vous devez parcourir les différentes maps attention que votre oiseau reste à l'écran car si il disparaît par la gauche : c'est perdu. Le mode multijoueur où la map est infinie et le but est donc d'aller le plus loin possible, le score étant comptabilisé selon le nombre de mètres parcourus, une fois que vous êtes mort (ça arrive à tous un jour malheureusement). Ce score est ensuite envoyé à la base de données de notre site web afin de pouvoir consulter le classement des joueurs.



Chapitre 1

Avancement du Projet

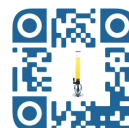
1.1 Moteur Physique

Dans un runner 2D, le moteur physique est en quelque sorte la base. L'oiseau doit être attiré par le sol, et doit être bloqué par les obstacles. Il était donc naturel que en fassions notre plus grand centre d'attention durant cette première partie de projet. Nous avons fait en sorte qu'il soit efficace et que l'on puisse le moduler aisément afin que l'on puisse s'amuser à changer les propriétés physiques d'une map à l'autre. Par exemple, la gravité est appliquée au personnage selon une variable coefficient qui peut être modifiée en fonction du niveau de difficulté. Le vol de l'oiseau peut donc être contrôlé, tout comme ses battements d'ailes qui le font avancer en même temps qu'il s'élève dans les airs.

Étant donné que la carte de jeux à les mêmes dimensions que la fenêtre, la gestion de dépassement de map a été très simple à réaliser. Un simple test sur la position du personnage par rapport à la fenêtre nous permet donc de l'empêcher de voler plus haut que le bord supérieur, ou de tomber plus bas que le bord inférieur.

Exemple :

```
if(SpritePosition.Y <= ScreenHeight)
{
    // On peut bouger horizontalement l'oiseau
}
else
{
    // On laisse l'oiseau à sa position
}
```



Du côté de la gestion des obstacles, nous les avons faits sous forme de rectangle qui contiennent les images des blocs. Nous avons dans un premier temps implémenté une collision basique, rectangulaire, qui fonctionnait avec ces rectangles. Cette collision marchait assez bien mais notre oiseau est rond, et les images des obstacles ne sont pas forcément rectangulaire non plus, ce n'était donc pas la solution la plus optimale... Pour remédier à ce problème, l'idée nous est donc venue de passer par une collision par pixel, bien plus adaptée à l'utilisation que l'on en fait dans notre jeu. Le problème de cette collision, c'est qu'elle est très gourmande puisque même en étant optimisée au maximum, elle parcourt beaucoup de pixel. Il a donc fallu assembler les deux tests, collision rectangulaire puis par pixel afin de ne pas tester une collision par pixel avec le rectangle d'un bloc si il ne touche pas le rectangle du personnage.

1.2 Réseau

1.3 Son

Nous savons que tout bon jeu est accompagné d'une vraie bibliothèque de sons, c'est pourquoi Mattou s'est vraiment concentrée sur cet aspect du projet. Il a fallu tout d'abord qu'elle recherche les musiques et les effets sonores adéquats pour que l'ambiance lorsque vous êtes dans le jeu soit optimale. Puis nous voulions que ce soit très structuré dans le projet pour ne pas avoir des sons en vrac dans nos Contents, notre solution : utiliser Xact.



1.4 Éditeur de Map

Nos Maps sont basées sur des fichiers textes (en .txt) que nous renommons en .lvl pour level (sans blagues). Le fichier est composé de : sur la première ligne le nombre d'élément pouvant se trouvé sur une même colonne, sur la deuxième ligne le nombre d'élément sur une ligne, puis sur le reste c'est la définition de la map les "0" correspondent à un espace vide, les 4 sont eux des obstacles et les 3 sont les fameux bonbons permettant de regagner de l'énergie.

Il est bien entendu que nous n'allons pas écrire toutes nos maps à la main, cela serait trop long et très très lassant. Donc nous avons créé un éditeur de maps qui écrira dans le fichier .lvl pour que nous ayons juste à cliquer un peu partout pour avoir une map (SUPER)

1.5 Site Web

Notre site, réalisé entièrement sans CMS, regroupe une présentation rapide de notre projet accompagnée d'images illustrant les différentes parties de notre jeu. Chaque membres du groupe a sa page personnelle, avec une petite présentation et une photo (la classe). Pour la partie pratique, vous pouvez télécharger nos différents rapports, en LaTeX ou en PDF, ainsi que la version du code source présentée lors des différentes soutenances directement depuis le site.

De plus, une page est réservée à l'inscription (avec enregistrement de carte bleu tout ça tout ça... histoire de rembourser tout nos frais de l'année). Cette partie n'est pas obligatoire mais recommandée puisque qu'il faut se connecter au jeux pour accéder au mode multijoueur et avoir l'envoi de score en ligne en fin de partie. Pour rester à l'écoute de nos admirateurs, une page de contact est disponible. Vous avez donc la possibilité de nous laisser des commentaires ou de nous suggérer des améliorations (pas de signalement de bugs puisque nous codons assez bien pour qu'il n'y en ai pas).

1.6 Graphique

La partie a était compliquée suite au départ d'Adrien qui n'était pas prévue, il était le plus expérimenté d'entre nous avec les logiciels de graphismes (Photoshop en l'occurrence). Il fallu très vite apprendre les bases de ce logiciel afin de pouvoir réutiliser les travaux déjà produit par Adrien. Nous avons donc du faire face a ce petit problème car un jeu avec des graphismes mal fini n'ai pas très attirant et le savons bien.



Nous n'avons pas de personne assigner au graphismes, chacun crée les graphismes ont il a besoin au fur et à mesure que nous avançons dans nos parties. Ce n'est pas forcément très efficace mais cela évite qu'un des membres se consacre entièrement au graphismes au détriment du code ce qui n'est pas le but de ce projet. Nous souhaitons que chaque membres puisse apprendre ce qu'il souhaite, c'est pour cela que nous sommes parfois peut-être trop nombreux sur une tâche, mais cette méthode a l'avantage d'impliquer l'ensemble de l'équipe à 100% dans le projet.

Nous avons tout de même défini une charte dans nos graphismes pour ne pas tomber dans un trop gros décalage entre deux parties du jeu. Nous avons donc des modèles de départ pour partir sur les mêmes bases, puis chacun les modifie, tout en montrant aux membres de l'équipe pour qu'ils puissent donner leur avis, et voir si cela reste en accord avec le reste du projet



Chapitre 2

Et Après ?

2.1 Map Editor 2.0

L'éditeur de map serat intégrer au jeu afin que vous puissiez créer vos propres map et ainsi étendre la durée de vie de notre jeu (qui est déjà infini), vous aurez aussi mettre au défi vos amis en partageant vos créations, ce qui rendra le jeu quelque peu plus addictif.

2.2 Réseaux Multijoueur

2.3 Nouveaux Personnage

Nous savons que nos bel oiseau est très attachant, nous avons décider de rajouter de nouveau personnages, qui auront des caractéristiques physiques différentes, pour que vous puissiez choisir choisir un personnage qui correspondent le mieux a votre style de jeu.

2.4 Graphismes



Conclusion

