

Rapport préliminaire du projet

Stéphane MANIACI
Baptiste SALEIL
tutoré par : Abdelkader GOUAICH

19 octobre 2009

Table des matières

I	Présentation du projet	2
II	Inspirations	3
III	Spécifications techniques	4
1	Langage de programmation	4
2	Infrastructure de développement	4
3	Éléments du jeu	4
IV	Feuille de route	5

Première partie

Présentation du projet

L'objectif du projet est de créer une série de mini-jeux basés sur le mouvements pour permettre à des personnes victimes d'un AVC de pouvoir travailler leur motricité d'une façon ludique. Ils auront pour cela une clé USB à leur disposition sur laquelle ils pourront directement démarrer les jeux et s'entraîner avec un contrôleur sans-fil (*i.e* une Wiimote) .

Le programme doit être capable de tracer leur mouvements et leur progression (système de score) afin de pouvoir exploiter les résultats de cette évolution si ils s'avèrent utiles aux médecins qui les prennent en charge. On envisage notamment de remonter les données sur un serveur principal dans le centre du patient.

Ce projet est proposé et tutoré par Mr. Abdelkader GOUAICH.

Deuxième partie

Inspirations

Les jeux seront divisés en catégories pour permettre une diversité ludique avec quand même une progression globale pour conserver une certaine linéarité malgré les différents types de jeu.

Jeu de tir à la première personne

Le but du jeu est de tirer sur des objets volants, d'un point de tir statique. Inspiré d'un jeu de *Wii Play*, nous développerons le *game design* une fois que l'on aura obtenu un prototype fonctionnel.



FIG. 1 – Capture du jeu de tir sur Wii Play

Jeu de dessin

Inspirée par le jeu *Rayman contre les lapins crétins* sur Wii, le but est de surligner à l'écran des motifs indiqués par le jeu. Là encore, on attendra d'avoir un prototype fonctionnel pour décider les détails artistiques.

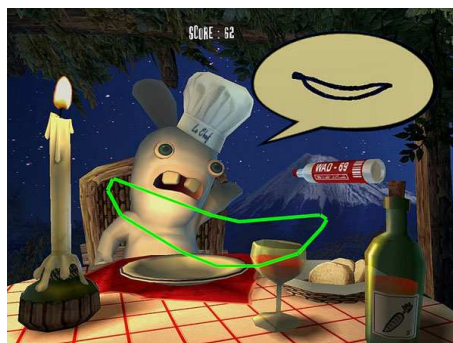


FIG. 2 – Capture du jeu de tir sur Wii Play

Troisième partie

Spécifications techniques

1 Langage de programmation

Comme proposé dans l'énoncé du projet, nous utiliserons C++ et la librairie graphique QT dans sa version embarquée, cette dernière permettant des économies de ressources et une portabilité maximale. Google Code offre une gestion du code avec Subversion, qui est un système de révision du code, mais nous utiliserons Git à la place, un système plus récent (développé pour le noyau Linux) qui permet de travailler sur le projet en hors-ligne.

2 Infrastructure de développement

Nous allons utiliser au maximum la plateforme Google pour mener à bien notre projet :

- **Google Code** pour l'hébergement du code, bugtracking, wiki ;
- **Google Documents** pour le partage de notes et documents ;
- **Google Mail** pour la communication et le chat entre développeurs ;
- **Google Groups** si nous en venons à utiliser une liste de discussions.

3 Éléments du jeu

Le patient lancera le jeu depuis sa clé USB, et arrivera sur un menu principal simple et concis, pour lui permettre d'accéder aux jeux triés par catégorie. Il nous faudra donc coder dans un premier temps un **moteur central** capable de gérer **l'affichage 2D, la journalisation**, et qui fournisse une **architecture de base** avec les classes de base sur lesquelles seront construites les jeux. L'affichage doit **se passer d'un serveur X** (par économie de ressources), nous écrirons donc directement sur le framebuffer grâce à Qt Embedded. La journalisation se fera dans des fichiers clairs (plain text), nous n'avons pas encore défini sous quelle forme, ce sera décrit dans le premier compte-rendu. Les diagrammes fournis par l'analyse UML apporteront plus de précision dans quelques semaines.

Quatrième partie

Feuille de route

Nous avons défini dans un premier temps les étapes nécessaires à la réalisation du projet mais les dates arrêtées seront décidées dans le prochain rapport. Voici donc la liste des objectifs principaux à atteindre.

1. Définition du projet et des moyens à mettre en œuvre ;
2. Analyse UML du projet ;
3. Prise en main de Qt et C++
4. Mise en place d'un environnement de développement ;
5. Réalisation d'un premier jeu fonctionnel dont sera dégagé le moteur ;
6. Optimisation et finalisation du moteur ;
7. Réalisation des jeux du projet ;
8. Optimisation, finalisation et documentation du projet ;
9. Champagne !