



# RAPPORT – MARATHON DES MILLIONS

## 1 CONTENTS

1	Contents .....	1
2	Introduction.....	2
3	Description .....	2
4	Modalités utilisées .....	3
5	Choix de conception .....	3
6	Librairies utilisées .....	4
6.1	Gestion de la Wiimote .....	4
6.2	Reconnaissance de la voix et synthèse vocale.....	5
7	Matériel utilisé .....	5
8	Captures d'écran .....	6
8.1	Questions.....	6
8.2	Run.....	7
8.3	ClickMe .....	8
8.4	Labyrinth.....	9
8.5	Buttons .....	10
9	Procédure installation .....	11
10	Améliorations possibles .....	12
11	Annexes .....	12
12	Conclusion .....	12



## 2 INTRODUCTION

Le projet Marathon des Millions est un projet de jeu développé pendant le cours d'interfaces multimodales 2011.

Ce projet a été mené à bien par les étudiants de Bachelor suivant :

- David Dutch
- Vincent Ischi
- Baptiste Wicht

## 3 DESCRIPTION

Le marathon des millions est un jeu inspiré de deux autres jeux :

- Qui veut gagner des millions : Qui est un jeu dans lequel il faut répondre à des questions pour gagner de l'argent
- Les lapins crétins : Il s'agit d'un jeu pour la Wii dans lequel il faut jouer à une série de mini-jeux assez crétins pour gagner une compétition.

Dans ce jeu, nous avons donc toujours 2 choses à faire en parallèle. Premièrement, l'utilisateur doit répondre aux questions affichées sur le panneau de gauche du jeu. C'est la principale chose à faire dans le jeu. Si l'utilisateur n'arrive pas à répondre à une question, le jeu entier est perdu. Les questions croissent en difficulté au fur et à mesure que l'utilisateur répond à de nouvelles questions.



A côté de cela, les mini-jeux permettent à l'utilisateur d'accumuler des bonus, ces bonus vont multiplier l'argent obtenu à chaque question répondue correctement.

Voici les mini-jeux que nous avons développés pour ce projet :

- Run : L'utilisateur doit agiter la manette Wii de haut en bas le plus vite possible pour faire avancer une voiture sur la route et gagner la course. Si la voiture de l'ordinateur atteint la ligne d'arrivée avant celle de l'utilisateur, l'utilisateur a perdu.
- Buttons : L'utilisateur doit simplement appuyer sur les boutons indiqués avant qu'ils ne disparaissent.
- ClickMe : L'utilisateur doit atteindre les marmottes avec le pointeur le plus vite possible et appuyer sur B pour tirer. Il doit toucher 10 marmottes en un certain nombre de secondes. Si le temps est écoulé et qu'il n'a pas atteint assez de marmottes il a perdu.
- Labyrinth : L'utilisateur doit déplacer un personnage (Bryan) dans le labyrinthe sans toucher les murs du labyrinthe. S'il réussit à atteindre la fin du labyrinthe, il a gagné et s'il a touché les murs il a perdu.

Ces mini-jeux s'alternent de façon aléatoire l'un après l'autre.



En plus des mini-jeux, lorsque l'utilisateur a répondu à 5 questions, il doit jouer à un nouveau jeu, sans questions en parallèle cette fois. Ce mini-jeu est le même que ClickMe à la différence près qu'il ne doit pas appuyer sur B, mais crier pour tirer.

A première vue ce jeu peut sembler difficile à prendre en main du fait de devoir jouer à deux jeux en parallèle. En plus de cela, certains mini-jeux sont plutôt prenant et parfois épuisants physiquement comme la course. Néanmoins, c'est tout à fait le but de ce projet. En effet, ce jeu n'a pas pour simple but de jouer à qui veut gagner des millions, mais c'est plutôt un jeu crétin qui repose sur le hasard des mini-jeux et de la dextérité des utilisateurs.

#### 4 MODALITÉS UTILISÉES



Le jeu Marathon des Millions utilise deux modalités : la voix et le geste.

Le geste est utilisé pour jouer aux différents mini-jeux. On utilise cette modalité via les wiimotes. On l'utilise de trois façons différentes. Premièrement et généralement, comme outil de pointage, ensuite comme gestionnaire de vitesse de déplacement via l'accéléromètre et enfin via les boutons de la wiimote.

La voix est utilisée principalement pour répondre aux questions posées dans le jeu, pour cela, on utilise la reconnaissance vocale pour identifier quelle réponse l'utilisateur a indiqué. Nous utilisons également le niveau de voix de l'utilisateur pour le grand jeu de tir sur marmottes. En plus de cela, la voix est également utilisée comme modalité de sortie pour indiquer des actions à faire par l'utilisateur pour ce qui est des questions.

En regard du modèle CASE, les 2 modalités sont utilisées de manière concurrente durant la durée des mini-jeux. Les 2 modalités peuvent arriver en parallèle, mais ne sont pas utilisées ensemble pour définir une action. Lorsque l'utilisateur joue au grand jeu de marmottes, les 2 modalités sont utilisées de manière synergétique, elles arrivent toujours en parallèle, mais sont utilisés conjointement pour définir l'action à effectuer.

En regard du modèle CARE, toutes les modalités sont assignées. Il n'y a pas deux moyens d'atteindre un même état. L'utilisateur ne peut donc pas choisir la modalité qu'il préfère pour effectuer une action.

#### 5 CHOIX DE CONCEPTION

La première chose que nous avons faite a été de définir les bibliothèques utilisables pour la wiimote et la voix. Ainsi, une fois que nous avons découvert les principales bibliothèques, nous avons choisi un langage de programmation, qui nous permettait l'utilisation des deux bibliothèques. C'est pourquoi nous avons choisi le langage de programmation C#.

Une fois que nous avons choisi le langage de programmation, il a fallu qu'on choisisse une bibliothèque graphique pour effectuer le dessin des mini-jeux. Nous avons commencé par utiliser WinForms pour effectuer le dessin des mini-jeux. WinForms est la bibliothèque de base de C# pour le dessin d'interfaces graphiques. Nous avons donc commencé à développer les mini-jeux en utilisant cela, mais au fur et à mesure que les mini-jeux évoluaient, nous nous sommes rendus comptes des grosses limitations de Windows du surtout à son système de dessin



basé sur GDI+. Les jeux n'étaient absolument pas fluides. Il n'était même pas possible de suivre à l'écran le pointeur de la wiimote.

Nous avons donc dû nous orienter vers une autre librairie graphique plus centrée sur les jeux pour avoir des performances suffisantes. Nous avons trouvé deux librairies qui pourraient convenir en C# :

- XNA : Il s'agit de la plateforme de Microsoft pour le développement de jeux vidéo.
- SFML : Il s'agit d'une librairie graphique d'abord créée pour C++ qui a été portée en C#.

Nous avons testés les deux plateformes en parallèle. Les deux répondaient à nos besoins, mais il était plus simple d'intégrer XNA au code déjà existant plutôt que SFML. C'est pourquoi nous avons choisi XNA.

## 6 LIBRAIRIES UTILISÉES

Il nous a fallu trouver 2 librairies : une pour gérer la wiimote et une autre pour gérer la reconnaissance vocale ainsi que la synthèse vocale.

### 6.1 GESTION DE LA WIIMOTE

Pour la gestion de la Wiimote, nous avons utilisé WiimoteLib. C'est une librairie C# développé par Microsoft.

Cette librairie nous permet de gérer tous les aspects de la Wiimote :

- Coordonnées de pointage
- Accéléromètre
- Extensions Wii (Guitar Hero, WiiMotion Plus)
- Boutons de la Wiimote
- LEDs de la Wiimote

Dans notre projet, nous n'avons pas utilisé les extensions Wii, mais nous avons utilisés toutes les autres fonctions de la Wiimote.

Nous avons eu des problèmes pour comprendre le fonctionnement de l'accéléromètre de la wiimote qui n'est pas un concept facile à prendre en main. Nous avons également eu des soucis pour calibrer les coordonnées normalisés de la wiimote à l'écran. En effet, tous les écrans n'ont pas la même taille et il a fallu trouver des formules permettant de calibrer les coordonnées pour pouvoir se déplacer sur tout l'écran.



## 6.2 RECONNAISSANCE DE LA VOIX ET SYNTHÈSE VOCALE

Pour la reconnaissance vocale, nous avons utilisé la librairie Windows Speech de Windows Seven. Cette librairie est directement intégré dans Windows et en plus de cela, elle est directement disponible dans le langage C#.

Cette librairie nous fournit les fonctionnalités suivantes :

- Reconnaissance vocale avec ou sans vocabulaire. En plus de la reconnaissance, la librairie nous donne également des informations sur les hypothèses faites ainsi que sur le niveau sonore de la voix de l'utilisateur. Cette reconnaissance vocale ne nécessite pas d'entraînement.
- Synthèse vocale. Le moteur TTS de la librairie est très puissant et permet d'utiliser la voix configuré dans Windows comme voix.

Il faut noter que ces deux librairies sont dépendantes de la langue configurée dans le système Windows.

## 7 MATÉRIEL UTILISÉ

Au niveau du matériel nécessaire pour ce jeu, voici ce que nous avons utilisé :

- Une wiimote : Manette de Wii connectée en Bluetooth au PC sur lequel le jeu est lancé
- Oreillette Bluetooth : Utilisé comme micro

A noter qu'au vu des nombreux problèmes causés par l'oreillette, nous avons fini par utiliser le micro intégré à nos ordinateurs plutôt que l'oreillette qui n'était pas loin d'être inutilisable.





## 8 CAPTURES D'ÉCRAN

### 8.1 QUESTIONS

Voici la vue des questions :

Qu'est-ce qui tourne autour du soleil ?

1. La terre
2. La lune
3. Un haricot
4. Le soleil

19

En rouge, on peut voir le décompte de temps, c'est-à-dire le nombre de secondes qu'il reste à l'utilisateur pour répondre à la question. Une fois la réponse choisie, elle est dessinée en rouge.



## 8.2 RUN

Voici à quoi ressemble le jeu de course :



L'utilisateur (voiture verte) doit atteindre la ligne noire avant que l'ordinateur (voiture rouge) l'atteigne.



### 8.3 CLICKME

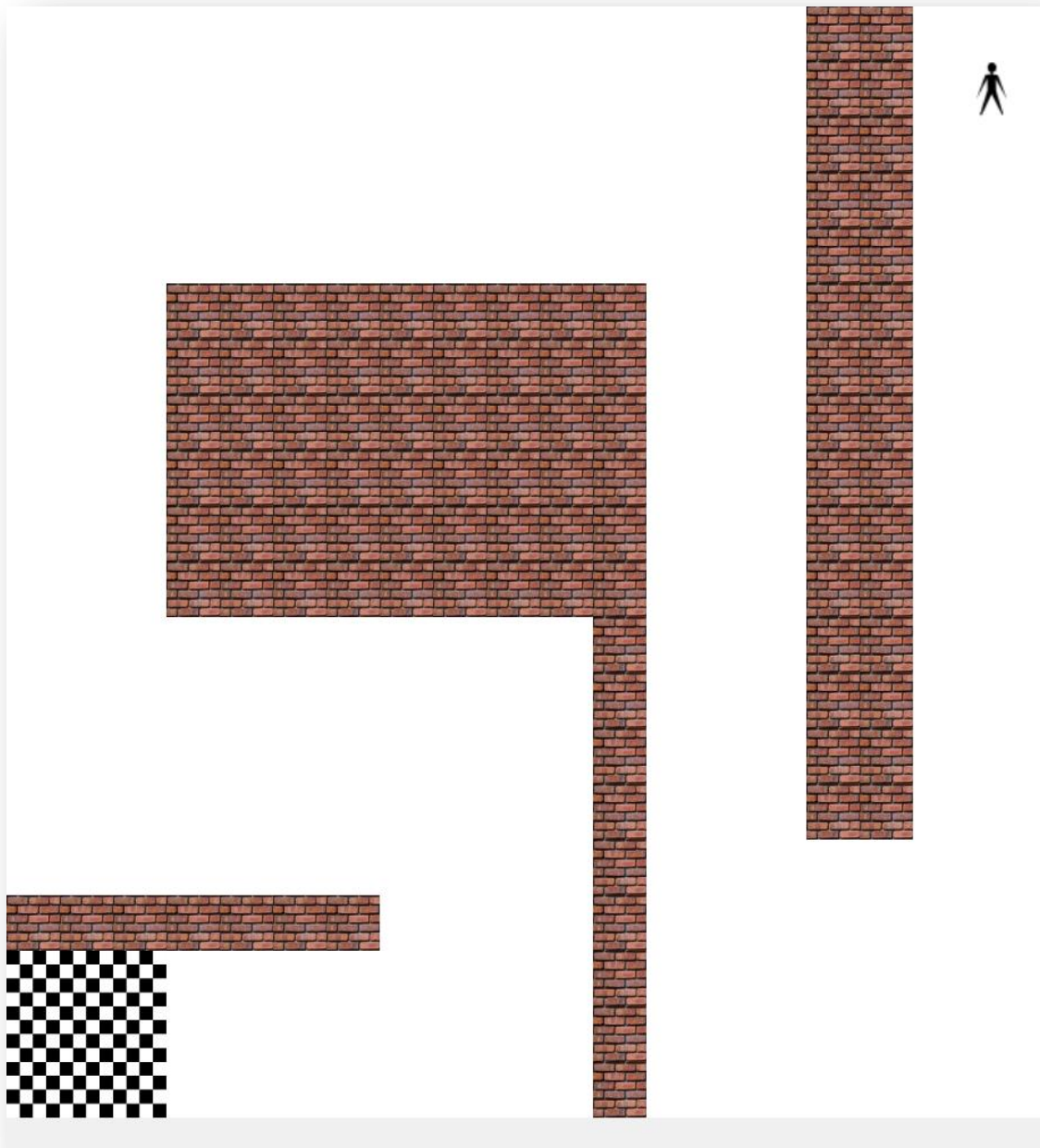


En haut à gauche, on peut voir l'image de la marmotte sur laquelle il faut tirer. En haut à droite, on peut voir le temps restant ainsi que le nombre de marmottes à attraper avant la fin du temps.





#### 8.4 LABYRINTH



En haut à droite se trouve Bryan, qu'il faut faire sortir du labyrinthe pour arriver à l'arrivée, qu'on peut voir en bas à gauche.



## 8.5 BUTTONS



A + B

Ce jeu affiche simplement la combinaison qu'il faut faire avec les boutons. Une fois que les boutons sont pressés, la combinaison est affichée en bleu, en rouge sinon.



## 9 PROCÉDURE INSTALLATION

Ce jeu peut être utilisé uniquement sous Windows. Il ne nécessite pas d'installation particulière.

Pour y jouer, il suffit de copier le dossier Executable présent sur le CD attaché à ce rapport et lancer l'application WindowsGame1. Ensuite, il suffit de cliquer sur Start pour démarrer le jeu.

Avant de pouvoir jouer, il faut préalablement avoir connecté une wiimote à l'ordinateur. Sans cela les jeux ne pourront pas se lancer. Pour cela, il faut vous rendre dans la liste des périphériques de votre ordinateur puis cliquer sur « Ajouter un nouveau périphérique ». Pour que la manette soit détectée dans la liste, il faut cliquer simultanément sur 1 et 2. Ensuite, il faut la choisir sur la liste Windows et choisir de ne pas utiliser de code pour la connexion et enfin atteindre que l'installation se termine sur Windows.



## 10 AMÉLIORATIONS POSSIBLES

Au niveau de ce projet, au vu du peu de temps à disposition, nous nous sommes concentrés sur les fonctionnalités de base du jeu. Il est donc possible de faire de nouvelles améliorations sur le projet :

- Il faudrait améliorer le design des questions, par exemple en intégrant une image de qui veut gagner des millions et afficher les questions à l'intérieur de l'image.
- Il faudrait également améliorer le BigGame en lui faisant un design plus original.
- Il serait également intéressant d'ajouter de nouveaux minijeux, on peut imaginer un jeu genre guitar hero ou d'autres minijeux inspirés des lapins crétins.
- Il serait bien d'adapter le jeu pour qu'il marche sur un système en anglais et sur un système en français alors qu'actuellement il ne fonctionne que sur des systèmes anglais. Sur un système français, il est possible de l'utiliser, mais il faut prononcer les chiffres des réponses en français, mais tout le texte parlé sera en anglais.
- Il faudrait tenir compte des hypothèses de la machine pour ajouter un meilleur niveau d'interaction avec l'utilisateur dans le cas où la reconnaissance vocale se passe mal.

## 11 ANNEXES

En annexes de ce rapport, vous trouverez les documents suivants :

- La présentation au format PowerPoint du projet
- L'application compilée dans le dossier Executables
- Les sources de l'application dans le dossier MarathonMillions
- Ce document au format PDF ainsi que ses sources au format DOCX
- Une fiche descriptive de la vidéo au format PDF et ses sources au format DOCX
- La vidéo de présentation du projet

## 12 CONCLUSION

Nous avons trouvé très amusant de développer ce projet. Nous avons pris beaucoup de plaisir à développer cette application, malgré le peu de temps imparti pour ce faire. Le fait de développer un jeu vidéo ajoute forcément un côté ludique à la chose. Nous avons également appris plusieurs choses sur les techniques utilisées, notamment au niveau de la gestion de la wiimote et de la librairie Windows de reconnaissance vocale.