

Iliass Ben ammar  
Nithusan Sivakanthan

# Contrôle d'un perso platformer à l'aide d'une caméra type Wii

Licence 3 informatique  
Université Paris 8  
Avril 2021  
Tuteur : Adrien Revault d'Allonnes

# TABLE DES MATIÈRES

## I : L'état de l'art

<b>1</b>	<b>Définitions et contexte de la thèse</b>	<b>2</b>
1.1	Recherches . . . . .	2
1.2	Définition des termes . . . . .	2
<b>2</b>	<b>Contrôler un personnage de platformer</b>	<b>4</b>
2.1	Les différentes technologies permettant le contrôle d'un personnage platformer . . . . .	4
2.2	Fonctionnement de la Wiimote . . . . .	4
<b>3</b>	<b>Idée de conception</b>	<b>7</b>
3.1	Comment allons-nous implémenter nos mouvements au personnage ? . . . . .	7
<b>4</b>	<b>Rectification du Tir.</b>	<b>7</b>
4.1	But du jeu. . . . .	7
<b>5</b>	<b>Caméra</b>	<b>8</b>
5.1	Librairies . . . . .	9
5.2	Code . . . . .	9
<b>6</b>	<b>Platformer</b>	<b>10</b>
6.1	Librairies . . . . .	10
6.2	Code . . . . .	10
<b>7</b>	<b>Conclusion</b>	<b>12</b>

# 1 Définitions et contexte de la thèse

## 1.1 Recherches

Pour ce projet, nous avons commencé par faire des recherches sur les personnages de plateformers, pour cela nous avons fait des recherches sur Google Scholar mais nous n'avons pas trouvé d'articles traitant du sujet, donc nous avons fait des recherches concernant notre problématique et nous avons trouvé divers articles intéressants.

## 1.2 Définition des termes

Tout d'abord, nous allons commencer par expliquer ce qu'est un "platformer".

Un "platformer" a pour principal genre : Action, Aventure, Jeu de Rôle, Simulation, Stratégie et Sport. Un "platformer" est un type de jeu où le monde est composé principalement de plateformes sur lesquelles le personnage se déplace et d'obstacles auxquels il doit faire face. Le principe du "platformer" se repose sur le saut, en effet le monde du "platformer" possède du vide et si le personnage y tombe il perd la partie ou revient à son checkpoint.

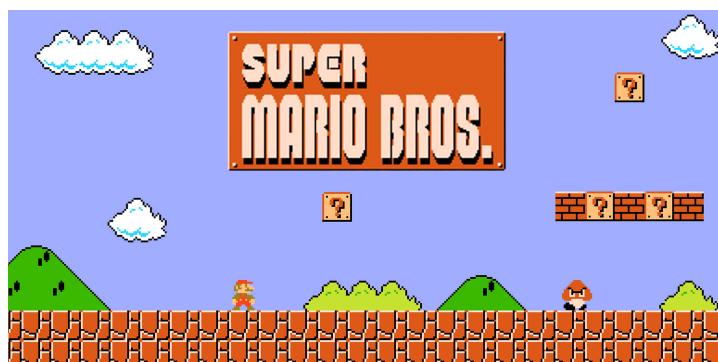
Il existe au moins deux types de "platformer", en premier nous avons les plateformers verticaux dits en "scrolling" avec comme exemple les jeux Doodle Jump, Jump King, etc.



source : Steam

Dans les "plateformers" en scrolling, tout se base sur le saut, la moindre erreur peut vous faire régresser dans votre parcours ou bien même vous faire recommencer depuis le début.

En second, nous avons les plateformers horizontaux avec des jeux tels que Mario, Celeste, etc.



source : Nintendo

**Voici quelques articles qui nous semble intéressant a vous présenter.**

- [https://www.theses.fi/bitstream/handle/10024/95047/Anttila\\_Joonas.pdf?sequence=1](https://www.theses.fi/bitstream/handle/10024/95047/Anttila_Joonas.pdf?sequence=1)

Cette article parle de créer une caméra adaptative pour les platformers en 3d et donc de contrôler le personnage à l'aide de cette caméra.

- <https://dl.acm.org/doi/abs/10.1145/1324892.1324941>

Celui-ci parle de poser 5 sensorbars autour de soi afin de créer un monde virtuel grâce a la manette Wii.

- <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1094&context=cscsp>

L'article traite d'un jeu en 2d side-view contrôlable à l'aide de la commande de l'Apple TV.

- <https://www.ukessays.com/essays/information-technology/wii-mote-materials-and-design-methodology-information-technology-essay.php>

Cet article nous permet de comprendre comment fonctionne la manette Wii, son fonctionnement ainsi que son architecture.

Comme nous pouvons le voir après toutes nos recherches, nous n'avons pas trouvé beaucoup d'articles qui s'avéraient être utile pour notre problématique.

## 2 Contrôler un personnage de platformer

### 2.1 Les différentes technologies permettant le contrôle d'un personnage plate-former

Il existe plusieurs façons de contrôler un personnage, mais tout d'abord commençons par voir les technologies classiques utilisées par la majorité de la population mondiale permettant de contrôler un personnage.

Pour commencer nous avons les claviers et souris qui sont les moyens de déplacements utilisés sur des jeux plate-former sur ordinateur, puis nous avons les manettes filaires (Playstation, Xbox, Gamecube) permettant de contrôler un personnage grâce aux joysticks et flèches directionnelles sur des jeux de console.

Puis, nous avons les webcams d'ordinateur comme moyen d'effectuer des mouvements au personnage, en effet grâce à un script permettant de retranscrire les mouvements effectués par la personne en face de la webcam au personnage de plate-former.

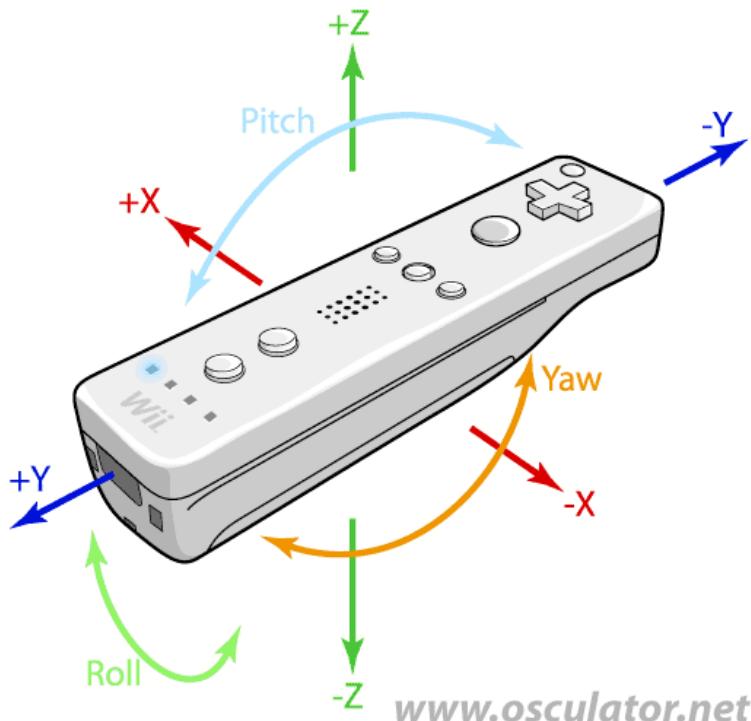
Enfin nous avons les manettes sans fil, ici on s'intéresse particulièrement à la manette de Wii (Wiimote) qui grâce à un capteur infrarouge permet de manipuler un curseur dans le menu de la Wii mais peut aussi être utilisé dans le mouvement d'un personnage.

### 2.2 Fonctionnement de la Wiimote

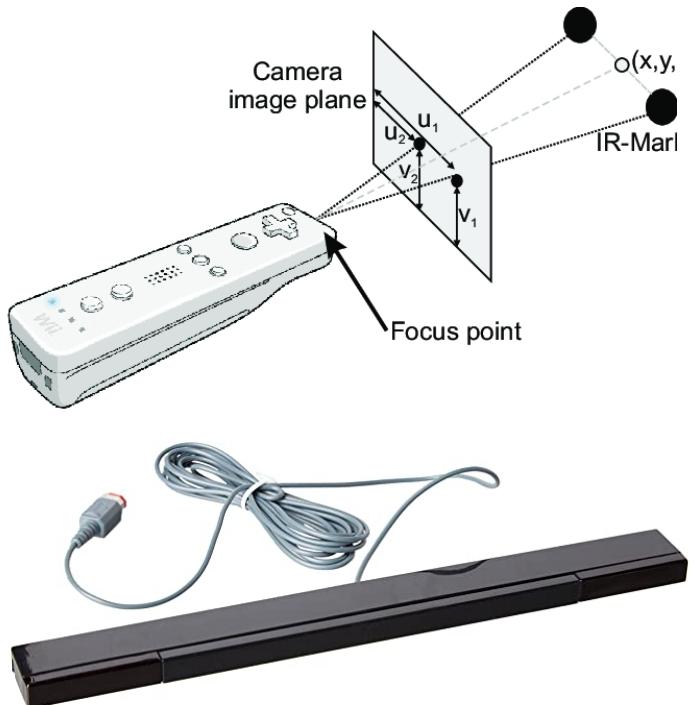
Maintenant qu'on a trouvé la technologie sur laquelle se baser, nous allons voir comment fonctionne une manette de Wii.

Tout d'abord de quoi est composé une manette de Wii ?

Dans cette manette de jeu, on trouve un accéléromètre trois axes qui a une fréquence d'échantillonnage de l'ordre de 35 Hz.



Ensuite nous avons la caméra infrarouge qui se trouve à l'avant de la manette. Elle va repérer dans son champs de vision les différentes sources infra-rouges. Le processeur du module de la caméra va ensuite analyser ces points et le placer sur un plan de résolution 1024\*768 pixels. Elle va pouvoir suivre 4 sources infra-rouges en même temps Et c'est ces coordonnées que la Wiimote va transmettre à la Wii qui va s'occuper des calculs en s'a aidant de la sensor bar.

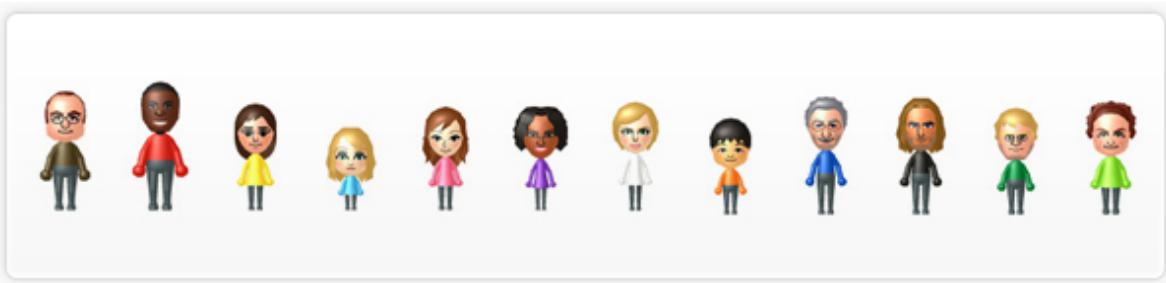


source : researchgate

Le grand public a souvent tendance à penser que la sensor bar est le capteur principal, mais pas du tout car ce n'est autre qu'une barre avec 5 leds infrarouges sur le coté droit et gauche, on pourrait très bien posé 2 bougies côté à côté ça fonctionnerait tout aussi bien.

Il y a ensuite le haut-parleur intégré 21mm piezo-électrique de basse qualité qui permet une meilleure immersion dans certains jeux en faisant, par exemple, parler la Wiimote.

La Wiimote possède aussi une mémoire de 16kb qui va permettre de stocker les données sur les Mii(personages de la Wii) mais cette mémoire va pouvoir également stocker des informations qui vont servir pour le calibrage de l'accéléromètre et la sensibilité de la caméra infra-rouge.



source : Nintendo

**La Wiimote possède un port d'extension qui permet de rajouter des accessoires.**

- L'accessoire le plus connu est la Nunchuck qui apporte un deuxième joystick ainsi qu'un deuxième accéléromètre.



source : Amazon

- Il y a ensuite l'accessoire de la manette classique qui ressemble à celle de ses concurrents(Playstation et Xbox).



source : Amazon

- Il y a aussi le Wii Motion Plus qui apporte un gyroscope en plus et qui permet donc d'avoir une précision plus accrue dans les jeux.



source : Darty

- Il y a aussi plusieurs accessoires qui permettent une meilleure immersion dans les jeux sous forme de guitare, volant, arme, raquette, etc.



source : Amazon

**La Wiimote possède un module Bluetooth BroadCom BCM2042. Ce module va permettre d'envoyer et de recevoir des données sans fil. Les données que la manette envoie ne sont pas cryptées, cela veut dire que l'on peut lire ce qu'elle transmet et donc que l'on peut utiliser ces données sur un ordinateur.**

## 3 Idée de conception

### 3.1 Comment allons-nous implémenter nos mouvements au personnage ?

Pour répondre à la problématique on compte utiliser une librairie du nom de `Wiiuse` qui va nous permettre de récupérer les données de la `Wiimote` et donc de créer notre programme pour déplacer notre personnage.

Librairie : <https://github.com/wiiuse/wiiuse>

## 4 Rectification du Tir.

Nous avions débuté par une mauvaise base en raison d'un malentendu avec l'énoncé. On a donc modifié certains points. Premièrement, la caméra de type `Wii` correspond à une `webcam`, nous avons donc utilisé les caméras telles que la `webcam` intégrée à l'ordinateur portable ou bien une `webcam` classique, connectée à l'aide d'un port `USB`. Puis, nous avons refait nos recherches sur la façon de faire notre projet. Nous avons utilisé le langage `Python` pour ce projet.

### 4.1 But du jeu.

Le but de notre jeu est d'esquiver des obstacles afin de récupérer un objet. Nous incarnons un personnage qui court sans s'arrêter contrôlable à l'aide de la tête. Il peut monter sur des plates-formes et doit esquiver les trous et les piques parsemés partout sur la carte de récupérer le coffre final pour mettre fin à la partie.

## 5 Caméra

Au niveau de la caméra, nous avions le choix d'utiliser une Kinect ou bien une webcam classique.

La Kinect a été conçu par Microsoft en septembre 2008, à l'origine appelée Project Natal, Microsoft a conçu deux versions : l'une destinée à la Xbox 360. Puis la V2 destinée à la Xbox One et Windows. La Kinect permet de contrôler une interface sans utiliser de contrôleur à l'aide d'une caméra.



source : wikipedia

La webcam, parfois appelée cybercaméra ou webcaméra, est une caméra conçue pour être utilisée sur un ordinateur et permet de créer une vidéo dont le but n'est pas d'atteindre une haute qualité mais de pouvoir transmettre en direct à travers un réseau : le flux vidéo, par exemple Internet.



source : logitech

## 5.1 Librairies

Pour pouvoir implémenter la caméra dans le code, nous avions besoin d'une librairie en Python. Nous avons trouvé plusieurs librairies différentes dans le but d'implémenter une caméra.

En premier, nous avons la librairie openKinect.

Cette librairie permet d'utiliser la caméra kinect sur Windows, Mac et Linux. Cependant, nous n'avions pas de caméra kinect pour pouvoir utiliser cette librairie.



source : openkinect

Deuxièmement, nous avons la librairie OpenCV.

Cette librairie permet d'utiliser une webcam intégrée ou connectée à l'ordinateur.



source : opencv

```
import cv2
captureVideo = cv2.VideoCapture(0)
```

Ainsi, pour pouvoir avoir de la reconnaissance faciale, nous avions le choix entre une librairie Face\_recognition et un document XML fourni par le site d'OpenCV. Nous avons donc choisi d'utiliser le XML fourni car nous le trouvons plus simple d'utilisation.

```
import cv2
teteCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

## 5.2 Code

La caméra sera utilisée pour pouvoir déplacer le personnage à l'aide du visage. Pour se faire, on prend le rendu vidéo de la caméra et on le coupe en quatre parties. Chaque partie correspondant à un mouvement du personnage : Saut à droite, Saut à gauche, déplacement à droite et déplacement à gauche. Ainsi, lorsque notre visage sera reconnu dans une des quatre parties, le personnage se déplacera sur le jeu dans la direction choisie.



Après avoir implémenté la caméra, on transforme la vidéo en gris afin de faciliter la reconnaissance faciale.

```
gris = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = teteCascade.detectMultiScale(gris, 1.1, 4)
```

Ensuite, dès lors que l'on détecte une tête, on récupère la position x et y du cadre pour ensuite faire déplacer le personnage en fonction de x et y.

```
for (x, y, w, h) in faces:
    # Partir à Droite
    if (x<325 and y>250):
        self.personnage.change_x = vitesseDeplacement
```

## 6 Platformer

### 6.1 Librairies

Pour pouvoir contrôler un personnage platformer, il nous faut un jeu de type platformer. Nous avions donc le choix entre deux librairies, Pygame et Arcade.



source :Pygame,Arcade

Nous avons choisi d'utiliser la librairie Arcade car c'est une bibliothèque Python facile à prendre en main pour créer des jeux vidéo en 2D.

### 6.2 Code

Nous avons séparé le code avec des vues de la même façon qu'avec des slides.

```
JeuView() et victoireView()
```

"JeuView" correspond à la partie principale du jeu et "victoireView" va permettre d'afficher une image de victoire dès lors que l'on récupère le coffre.

Ensuite, nous avons commencé par créer une map sur Tiled, puis, nous l'avons chargée en Python couche par couche. Subséquemment, pour ce qui est du personnage, nous avons pris un sprite sheet d'un personnage avec huit positions différentes, dont quatre pour les mouvements de déplacement vers la gauche et quatre pour les mouvements de déplacement vers la droite.

**La physique du jeu est grandement facilitée grâce à la fonction "PhysicsEnginePlatformer()".** Elle permet de gérer les collisions avec les plates-formes et les déplacements du personnage.

Après avoir géré les déplacements du personnage, nous avons mis en place la caméra qui va permettre de suivre le personnage tout au long de la partie. Pour cela, nous avons utilisé "arcade.set\_viewport()". Elle permet de gérer le placement de la caméra.

Pour finir, on gère la collision du personnage avec le coffre et les piques à l'aide de la fonction.

```
arcade.check_for_collision_with_list()
```

## 7 Conclusion

Nous avons donc réussi à créer un jeu contrôlable à l'aide de la caméra. Ce dernier nous a tout d'abord permis d'apprendre à concevoir un jeu platformer à l'aide de librairies, de sa conception théorique jusque sa conception pratique. De plus, cela nous a donné la chance de découvrir des librairies ainsi que de renforcer nos connaissances sur ces dernières. Nous pouvons en particulier souligner le fort potentiel d'openCV qui permettra de faciliter la réalisation de lourdes tâches telles que la reconnaissance d'images.

## Références

- [1] Librairie : OpenCV,  
<https://opencv.org/>
- [2] Librairie : Arcade,  
<https://arcade.academy/>
- [3] Librairie : Pygame,  
<https://www.pygame.org/news>
- [4] Librairie : Openkinect,  
[https://openkinect.org/wiki/Main\\_Page](https://openkinect.org/wiki/Main_Page)
- [5] Librairie : Face\_recognition,  
[https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- [6] Librairie : Wiiuse,  
<https://github.com/wiiuse/wiiuse>
- [7] Images : Wikipedia,  
[https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil\\_principal](https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal)
- [8] Images : Steam,  
<https://store.steampowered.com/?l=french>
- [9] Images : Nintendo,  
<https://www.nintendo.fr/>
- [10] Images : Osculator,  
<https://osculator.net/>
- [11] Images : Researchgate,  
<https://www.researchgate.net/>
- [12] Images : Amazon,  
<https://www.amazon.fr/>
- [13] Images : Darty,  
<https://www.darty.com/>
- [14] Images : Logitech,  
<https://www.logitech.fr/fr-fr>