Type	**Jeu**
Nom du projet	**Ping Pong**
Commentaire	
Auteur	**Mayalale Clément KAFWIMBI**
Version	**1.0**
Date	**10/0**4**/2**022

Tabes des matières

3.3 Background.py

2. Principes des solutions techniques 1.1

Langage

Le langage utilisé c'est le langage python 3.10

1.2 Architecture du logiciel

1.3 Interface utilisateur

L'interface utilisateur se fera dans le terminal linux Ubuntu

1.3.1 Boucle de simulation

Le programme mettra en œuvre une boucle de simulation qui gèrera l'affichage et les événements clavier.

2.Analyse

2.1 Analyse noms/verbes :

Nom :Table, image

Verbes :change, collision, show, live

2.2 Types de données Type

: Ball =structure

Position (x, y) : réel

Vitesse (vx, vy) : réel

Color : entier

Type : Racket

Position (x, y) : entier

Vitesse (vx, vy) : réel

Color : entier

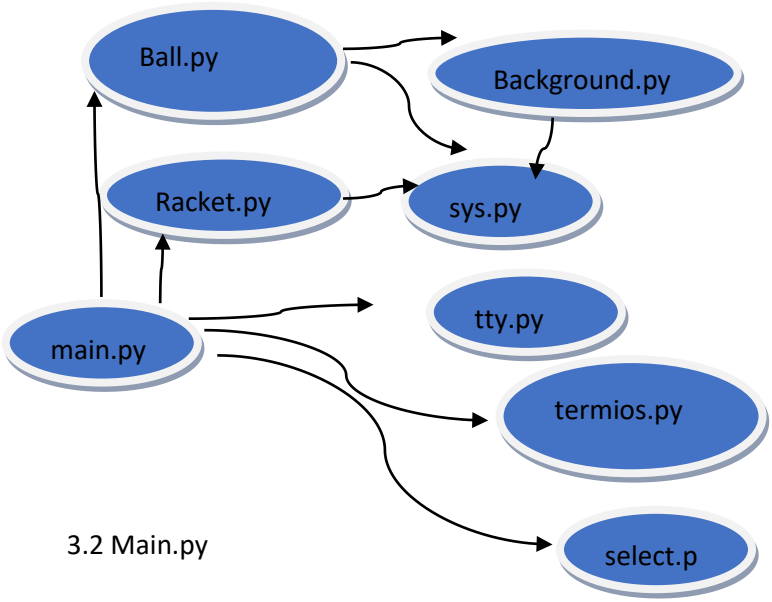
Type : Background

Width :entier

Heigth :entier

Mytable :fichier

2.3 Dépendance entre modules



3.2 Main.py

Main.main()->rien
Description : fonction principale du jeu
Paramètres : aucun
Valeur de retour :aucune
Main.init()->rien
Description : initialisation du jeu
Paramètres : aucun
Valeur de retour :aucune
Main.run()->rien
Description : boucle de simulation
Paramètres : aucun
Valeur de retour :aucune
Main.show()->rien
Description : fonction principale du jeu
Paramètres : aucun
Valeur de retour :aucune
Main.interact()->rien
Description : gère les événements clavier
Paramètres : aucun
Valeur de retour :aucune

3.2 Racket.py

•Racket.create(x,y,vx,vy,color)

• Racket.get_x(racket)

• Racket.set_x(racket,x)

• Racket.get_y(g)

• Racket.set_y(racket,y)

• Racket.get_vx(racket)

• Racket.set_vx(racket,vx)

• Racket.get_vy(racket)

• Racket.set_vy(racket,vy)

• Racket.live(racket,dt,img)

• Racket.show(racket)

•

Racket.create(x,y,vx,vy,color)->racket

Description : crée une nouvelle partie

Paramètres : x: réel y: réel vx :réel vy

:réel color : entier

Racket.live(racket,dt,img)->rien

Description: fait bouger la raquette

Paramètres : racket: racket

Racket.show(g)->rien

Description : positionne la raquette dans le terminale

Paramètres :racket racket : racket

Valeur de retour: rien

Racket.test_collision(t1,t2)->rien

Description : collisionne la raquette est la balle , mais pour le moment , elle reste inactive

Paramètres : g : Game

Valeur de retour: booleens

3.3 Ball.py

•Ball.create(x,y,vx,vy,color)

• Ball.get_x(ball)

• Ball.set_x(ball,x)

• Ball.get_y(ball)

• Ball.set_y(ball,y)
• Ball.get_vx(ball)

• Ball.set_vx(ball,vx)
• Ball.get_vy(ball)
• Ball.set_vy(ball,vy)

• Ball.live(ball,dt,img)

• Ball.show(ball)
• Ball.change_position(t)

Ball.create(x,y,vx,vy,color)->ball
Description: Crée une balle
Paramètres: rien
Valeur de retour: ball

• Ball.live(ball,dt,img)→ rien Description: position la balle Paramètres: g:racket dt:(entier)
Valeur de retour: rien
Ball.show(ball)

Description: renvoie la case sélectionnée Paramètres:
ball: ball
Valeur de retour: rien

Ball.test_collison(b1, b2)

Description : collisionne la raquette est la balle, mais pour le moment, elle reste inactive pour le moment
Paramètres :
ball : ball
Valeur de retour : booléens

Ball.change_position(t)

Description : elle change la position la balle lors de la collision
Paramètres :

Valeur de retour : rien

3.3 Background.py

- Background.create(mytable)
- Background.get_width(table)
- Background.set_width(table,width)
- Background.get_heigth(table)
- Background.set_heigth(table,heigth)
- Background.get_char(table,line_index, colum_index)
- Background.set_char(table,line_index, colum_index,c)
- Background.show(table)

Background.create(table)->table Description:
ouvre le fichier image.txt Paramètres:
Valeur de retour:table