



ภูมิไผท พันธิตพงษ์ 6031318421

นราวิชญ์ ชุติศิลป์ 6030319921

## 1. คำนำ

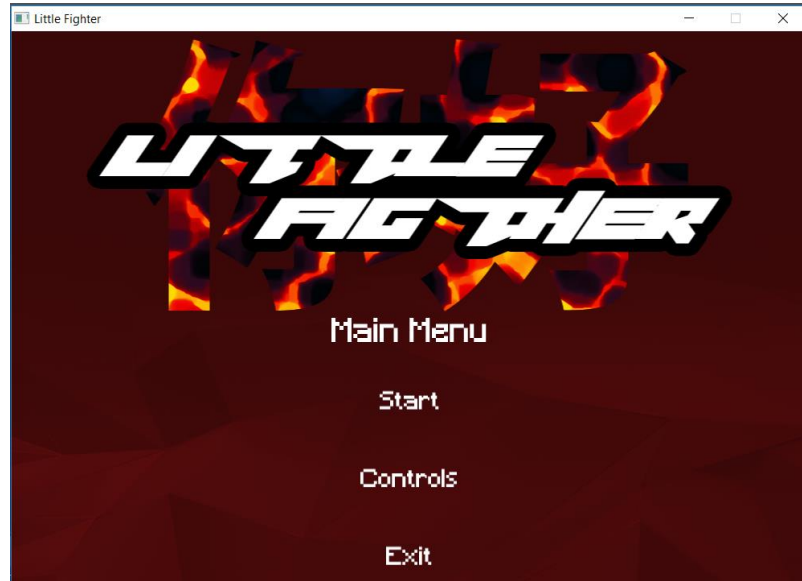
Little Fighter เป็นเกมต่อสู้ที่สามารถเล่นพร้อมกันได้สูงสุด 2 คน โดยผู้เล่นสามารถเลือกที่จะเล่นกับเพื่อนหรือเล่นเหงา ๆ คนเดียวก็ได้

เราจะได้เป็นตัวละครที่เลือกไว้ซึ่งแต่ละตัวจะมีสกิลและท่าทางที่แตกต่างกัน โดยเราสามารถเล่นเป็นทีมและเอาชนะทีมอื่น ๆ ได้ด้วยเช่นกัน

คุณพร้อมที่จะเอาชนะในสังเวียนแห่งศักดิ์ศรีแล้วหรือยัง? มาดูกัน!

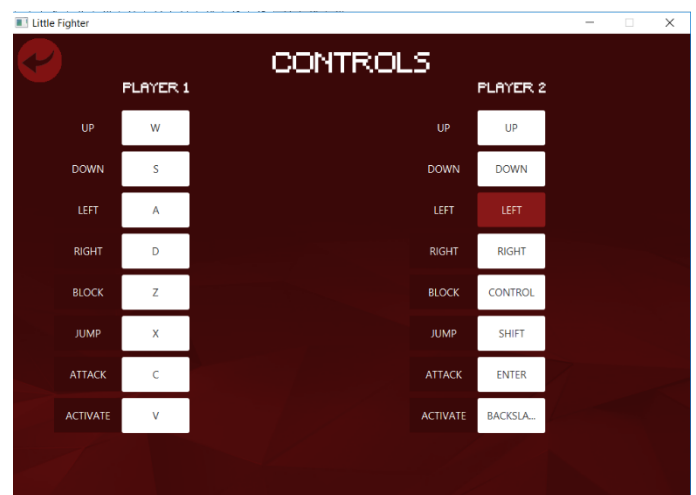
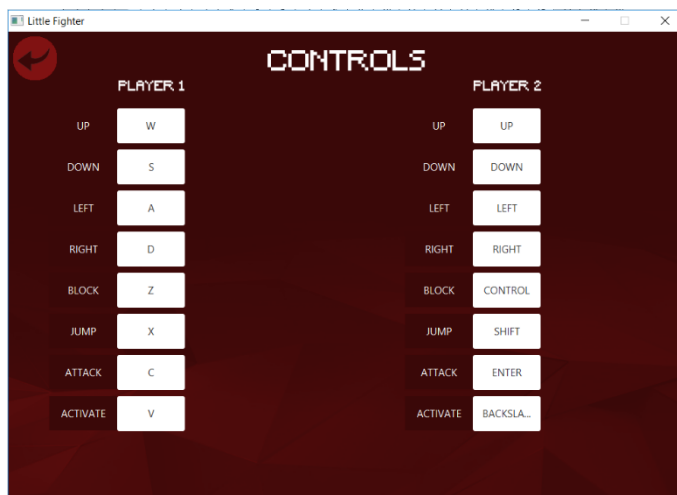
## 2. วิธีการเล่น

ในหน้าแรก ผู้เล่นสามารถเลือกที่เริ่มเล่นเลย ตั้งค่าปุ่ม หรือออกเกมได้



### 2.1 Controls

ในการเลือกแก้ไขปุ่ม ให้คลิกที่ปุ่มที่ต้องการแก้ไข ปุ่มนั้นจะเปลี่ยนเป็นสี ให้กดปุ่มใดในคีย์บอร์ดก็ได้ หากเป็นคีย์ที่ใช้กับอันอื่นอยู่แล้วหรือไม่ใช่คีย์ที่ต้องการ ระบบจะไม่ทำอะไรและรอจนกว่าผู้เล่นจะใส่คีย์ที่ต้องการหรือคีย์เดิมเข้าไป



## 2.2 Exit

เกมจะปิดเพลงและ Threads อื่น ๆ ที่เปิดอยู่

## 2.3 Start

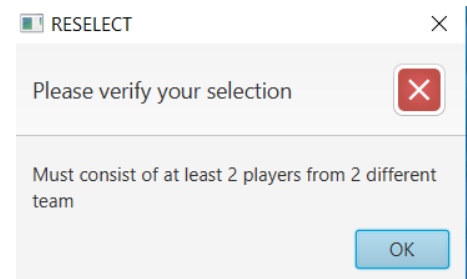
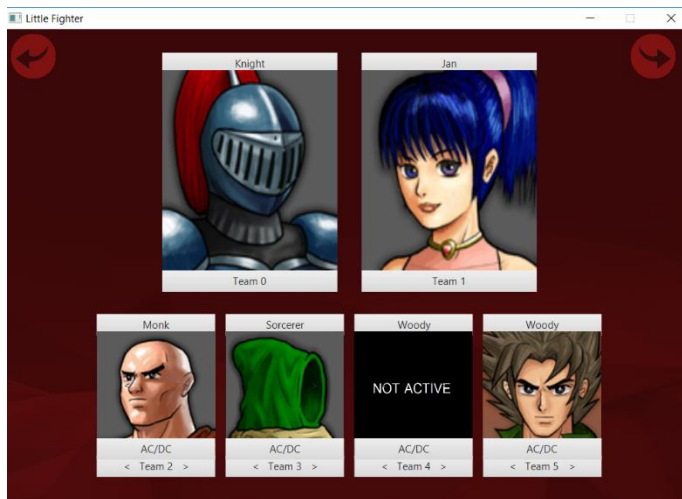
ผู้เล่นจะเข้ามาที่หน้าเลือกตัวละคร โดย PlayerCard สองอันแทนผู้เล่นที่ 1 และ 2 ตามลำดับ โดยเลือกด้วยการใช้ปุ่ม Controls ต่าง ๆ ที่ตั้งค่าไว้

[Activate] เพื่อเลือกเล่น

[Left] / [Right] เพื่อเลือกตัวละคร

[Up] / [Down] เพื่อเลือกทีม

นอกจากนี้ยังสามารถเลือกบอทที่จะเล่นด้วยโดยกด AC/DC และตัวละครและทีมได้ เมื่อเลือกเรียบร้อยแล้วกดปุ่มบนขวาเพื่อเริ่มการต่อสู้ระหว่างทีม!

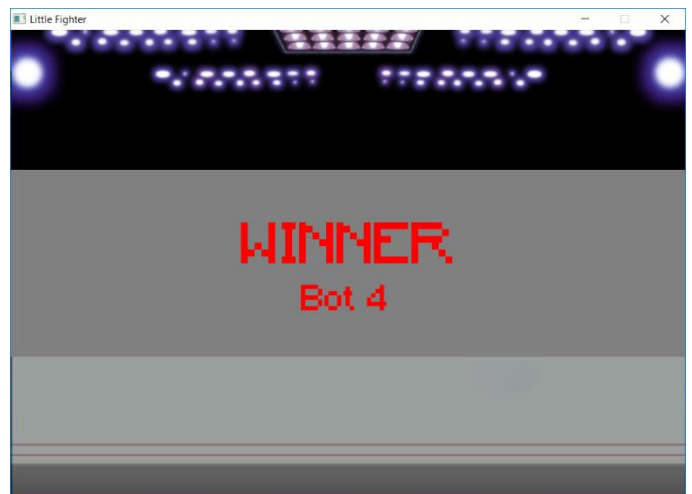


แต่! ถ้าหากมีผู้เล่นหรือทีมไม่เพียงพอ (น้อยกว่า 2) เกมจะขึ้นแจ้งเตือนจากการ Catch Exception ที่สร้างขึ้นเองอย่างละเอียดละเมียดละไม

เมื่อเริ่มเกมแล้ว ผู้เล่นสามารถควบคุมตัวละครได้ตามที่ตั้งค่าไว้

การเรียกใช้สกิลทำได้สองอย่าง

- Projectile = [Activate] + [Attack]
- Point Target = [Activate] + [Jump]



เมื่อจบเกม จะมีแถบสรุปผลการต่อสู้ ผู้เล่นสามารถคลิกที่แถบนี้เพื่อกลับไปยังเมนูหลัก

### 3. Implementation

#### 3.1 Package entity.character

##### 3.1.1 Class AllList

##### 3.1.1.1 Field

+ <u>Character[] charList</u>	List ของตัวละครในเกมทั้งหมด
+ <u>Players[] playerList</u>	ลิสต์เก็บผู้เล่นในสังเวียนรวมถึงบอท สูงสุด 6
+ <u>String[] teamcol</u>	เก็บรหัสสีของทีม
+ <u>PlayCard[] playCardList</u>	เก็บ PlayCard สำหรับหน้าเลือกตัวละคร
+ <u>TreeSet&lt;KeyCode&gt; kList</u>	สำหรับรับค่าคีย์ที่ผู้เล่นกด
+ <u>ArrayList&lt;Integer&gt; al</u>	เก็บ index ของ Player ที่ยัง Active อยู่
+ <u>ArrayList&lt;HMbar&gt; hm</u>	เก็บ HMBar ของตัวละคร
+ <u>ArrayList&lt;ArenaChar&gt; acc</u>	เก็บ Player ใน Arena ในรูปแบบของ ArenaChar
+ <u>ArrayList&lt;Skill&gt; skill</u>	เก็บ Skill ใน Arena

##### 3.1.1.3 Constructor

--	--

##### 3.1.1.3 Method

+ <u>void initAC</u>	เริ่มสร้าง List ต่าง ๆ สำหรับแสดงผลใน Arena
+ <u>void initPlayers</u>	เก็บ Player ไว้ใน playerList

+ <u>void initPlayCard</u>	สร้าง PlayCard สำหรับหน้าเลือกตัวละคร โดย ของเรียน initPlayers ก่อนเพื่อนำข้อมูลต่าง ๆ เข้ามาแสดงผล
+ <u>int hasWinner</u>	เช็คว่ามีผู้ชนะหรือยัง
+ <u>void ProceedArena</u>	เช็คว่ามี Players หรือ Teams พอที่จะเล่น หรือไม่
+ <u>ArrayList&lt;Players&gt; checkWinner</u>	คืนค่า List ผู้ชนะ

### 3.1.2 Class Character

#### 3.1.2.1 Field

- <u>ClassLoader CLASS_LOADER</u>	สำหรับ Load Resource ใน Class นี้
- String name	ชื่อ Character
- Image image	รูป Thumbnail
- Image jump	รูปตอนกระโดด
- Image block	รูปตอนบล็อก
- Image attack	รูปตอนโจมตี
- Image s1	รูปตอนใช้ Skill 1
- Image s2	รูปตอนใช้ Skill 2
- Image dead	รูปตอนตาย
- Image stand	รูปตอนยืน
- SkillDetail atk	SkillDetail ของการโจมตี
- SkillDetail skill1	SkillDetail ของ Skill 1
- SkillDetail skill2	SkillDetail ของ Skill 2

#### 3.1.2.2 Constructor

+ Character(String name, String url)	เก็บชื่อ และ url ของภาพไว้ เพื่อใช้ในการ Load รูปภาพท่าทางของ Character
--------------------------------------	---

### 3.1.2.3 Method

+ SkillDetail getATK()	Getter of atk
+ SkillDetail getSKILL1()	Getter of skill1
+ SkillDetail getSKILL2()	Getter of skill2
+ Image getJump()	Getter of jump
+ String getName()	Getter of name
+ Image getImage()	Getter of image
+ Image getBlock()	Getter of block
+ Image getAttack()	Getter of attack
+ Image getSk1()	Getter of sk1
+ Image getSk2()	Getter of sk2
+ Image getDead()	Getter of dead
+ Image getStand()	Getter of stand

### 3.1.3 Class Players

#### 3.1.3.1 Field

- String name	ชื่อของ Player
- int cindex	Index ของ Character ใน AllList.charList
- int team	หมายเลขทีม
- int pNum	หมายเลข Player (1 2 คือคน)
- int jump	สถานะของการกระโดด



- boolean active	สถานะว่าเล่นได้ (ต่างกับสถานการณ์ตาย)
- int hp	จำนวนเลือด
- int mp	จำนวนmana
- int posx	ตำแหน่งแกน x
- int posy	ตำแหน่งแกน y
- int posz	ตำแหน่งแกน z (เปลี่ยนแปลงตามการกระโดด)
- int bcd	ค่า Block cool down เมื่อ Player Block จะอยู่ได้นาน 10 ticks และ cool down 30 ticks
- int turn	สถานะหันขวา 1 ซ้าย -1
- int status	0 = normal 1 = blocked 2 = attack 3 = skill1 4 = skill2 5 = dead
- int timeout	จำนวน tick ที่ผู้เล่นจะทำอะไรไม่ได้

### 3.1.3.2 Constructor

+ Players(String name, int pNum, int team)	เก็บชื่อตัวละคร index และเลขทีม จากนั้นตั้งค่าต่าง ๆ เริ่มต้น โดย hp=500, mp=100 สุ่มค่า posx และ posy เพิ่มเป็นตำแหน่งเริ่มต้น posz=0, bcd=0, turn=1, jump=0, status=0, timeout=0
--	--

### 3.1.3.3 Method

+ void takeDamage(int dmg)	ลดเลือด ถ้าเลือดน้อยกว่าเท่ากับ 0 คือสิ้นชีพ
+ void toggleActive()	ใช้เปลี่ยนค่า active ตอนเลือกตัวละคร
+ void changeChar(int way)	เปลี่ยน Character
+ void changeTeam(int way)	เปลี่ยน team
+ void handleKey(KeyCode x)	Handle คีย์หน้าเลือกตัวละคร
+ void move(int way,int wayy)	เปลี่ยนตำแหน่งตัวละคร
+ int botMove()	เลือก action ของบอท
+ void handleCommand()	Handle คีย์ใน Arena
+ int getJump()	Getter of jump
+ int getStatus()	Getter of status
+ int getTimeout()	Getter of timeout
+ void gameUpdate()	Update ค่า mp timeout posz jump และ bcd
+ int getHp()	Getter of hp
+ int getMp()	Getter of mp
+ int getPosx()	Getter of posx
+ int getPosy()	Getter of posy
+ int getPosz()	Getter of posz
+ int getTurn()	Getter of turn
+ String getName()	Getter of name
+ void setName(String name)	Setter of name
+ int getCindex()	Getter of cindex
+void setCindex(int cindex)	Setter of cindex
+ int getTeam()	Getter of team
+ void setTeam(int team)	Setter of team

+ int getPNum()	Getter of pNum
+ void setpNum(int pNum)	Setter of pNum
+ boolean isActive()	Getter of active
+ void setActive(boolean active)	Setter of active

### 3.2 Package entity.control

#### 3.2.1 Class Control

##### 3.2.1.1 Field

+ <u>KeyCode[][] allkey</u>	เก็บการคีย์ต่าง ๆ
<u>int UP</u>	ตำแหน่งของ UP
<u>int DOWN</u>	ตำแหน่งของ DOWN
<u>int LEFT</u>	ตำแหน่งของ LEFT
<u>int RIGHT</u>	ตำแหน่งของ RIGHT
<u>int BLOCK</u>	ตำแหน่งของ BLOCK
<u>int JUMP</u>	ตำแหน่งของ JUMP
<u>int ATTACK</u>	ตำแหน่งของ ATTACK
<u>int ACTIVATE</u>	ตำแหน่งของ ACTIVATE
<u>int SETTING_PLAYER</u>	สถานะเลือกตัวละคร
<u>int SETTING_KEY</u>	สถานะตั้งค่า

### 3.3 Package main

#### 3.3.1 Class Main extends Application

App Entry Point

##### 3.3.1.1Field

+ <u>int SCREEN_WIDTH</u>	ความกว้างจอ
+ <u>int SCREEN_HEIGHT</u>	ความสูงจอ

### 3.3.1.2 Method

+ <u>void main(String[] args)</u>	Entry Point
+ <u>void start(Stage primaryStage)</u>	ตั้งค่าเริ่มต้นพร้อมไปที่หน้า Menu หลัก

### 3.3.2 Class SceneManager

จัดการหน้าต่างๆ

#### 3.3.2.1 Field

- <u>Stage primaryStage</u>	Stage หลัก
- <u>Scene menuScene</u>	หน้าเมนู
- <u>Scene controlScene</u>	หน้า set control
- <u>Scene charScene</u>	หน้าเลือกตัวละคร
- <u>Scene arenaScene</u>	สมรภูมิรบ

#### 3.3.2.2 Method

+ <u>void setMenuScene()</u>	เปลี่ยนไปหน้าเมนูและ Initialize(สร้างใหม่ถ้ายังไม่มี)
+ <u>void setArena()</u>	เปลี่ยนไปหน้าสมรภูมิและInitialize(สร้างใหม่ถ้ายังไม่มี)
+ <u>void setControlScene()</u>	เปลี่ยนไปหน้า Set Control และInitialize(สร้างใหม่ถ้ายังไม่มี)

+ <u>void setCharSelectScene()</u>	เปลี่ยนไปหน้าเลือกตัวละคร และInitialize(สร้างใหม่ถ้ายังไม่มี)
+ <u>void setStage(Stage primaryStage)</u>	เปลี่ยน Stage
+ <u>Scene getControlScene()</u>	เอา ControlScene ปัจจุบัน
+ <u>Scene getCharScene()</u>	เอา CharScene ปัจจุบัน

### 3.4 Package main.arena

#### 3.4.1 Class Arena implements BackGroundMusic

สมรภูมิรบ

##### 3.4.1.1 Field

- <u>ClassLoader CLASS_LOADER</u>	สำหรับ Load Resource ใน Class นี้
+ <u>Arena instance</u>	สำหรับเรียก Instance ปัจจุบัน ใน global scope
<u>MediaPlayer startMP</u>	เครื่องเล่นเสียง Background
<u>MediaPlayer transitionMP</u>	เครื่องเล่นเสียง Transition
- Media START_SOUND	เสียง Background
+ <u>int arenaWidth</u>	ความกว้างสนาม
+ <u>int arenaHeight</u>	ความสูงสนาม
- Pane backgroundPane	Container หลัก
- ImageView iv	รูปพื้นหลัง
- Canvas resultCanvas	แสดงผู้ชนะ
- GraphicsContext gc	ใช้ร่วมกับ resultCanvas

- boolean fighting	Suppress Thread ที่ใช้ทำงานในสนามรบหากไม่ได้ทำสงครามอยู่
--------------------	---

#### 3.4.1.2 Constructor

+ Arena()	วาดทุกอย่าง และจัดการสร้าง Thread ที่ไว้ Update และทำงานหลัก รวมถึงการ Hook Event
-----------	---

#### 3.4.1.3 Method

+ void hh(KeyCode k)	เพิ่มปุ่มที่กดเข้าไปใน Pool ของปุ่ม
+ void ho(KeyCode k)	เอาปุ่มออกจาก Pool
+ void newArena()	Reset ทุกอย่างและหยุดการ Suppress
+ void addChild(Node e)	เพิ่ม Node ใหม่ใน Arena เช่น Skill
+ void remChild(Node e)	ลบ Node ออกจาก Arena
+ void startSound()	เล่นเพลง Background
+ void stopSound()	หยุดเพลง Background
+ <u>CharComponent getInstance()</u>	Return instance ปัจจุบัน
+ Pane getBackgroundPane()	Return Container หลัก

#### 3.4.2 Class ArenaChar extends Pane

เป็น Component แสดงผลหลักของ Character บน Arena

##### 3.4.2.1 Field

- Players pp	Player ที่ Link ด้วย
--------------	----------------------

- ImageView iv	แสดงภาพตัวละคร
- Button name	แสดงชื่อ Player

#### 3.4.2.2 Constructor

+ ArenaChar(Players p)	สร้างตัวละครบน Arena และใส่รูป
------------------------	--------------------------------

#### 3.4.2.3 Method

+ void gameUpdate()	Update การเคลื่อนไหวของตัวละคร ทุกอย่างทุกการกระทำ การหัน การ Block หรือการปล่อย Skill
---------------------	--

### 3.4.3 Class HMBar extends Pane

แถบผู้เล่น ที่ระบุ HP/MP ด้านบน Arena

#### 3.4.3.1 Field

- Players pp	Player ที่ Link ด้วย
- ProgressBar hp	แถบแสดง HP ที่เหลือ
- ProgressBar mp	แถบแสดง MP ที่เหลือ
- ImageView iv	แสดงภาพตัวละคร
- Button name	แสดงชื่อ Player

#### 3.4.3.2 Constructor

+ HMbar(Players p,int pos)	จัดการค่าต่างๆและ Set ตำแหน่งของ Component
----------------------------	--

#### 3.4.3.3 Method

+ void gameUpdate()	การ Update ในแต่ละ Frame
+ void hpUpdateColor()	ปรับสีแถบ HP ตามปริมาณ
+ void mpUpdateColor()	ปรับสีแถบ MP ตามปริมาณ

### 3.5 Package main.charac

#### 3.5.1 Class CharComponent implements BackGroundMusic

หน้าเลือกตัวละคร

##### 3.5.1.1 Field

- <u>ClassLoader CLASS_LOADER</u>	สำหรับ Load Resource ใน Class นี้
+ <u>CharComponent instance</u>	สำหรับเรียก Instance ปัจจุบัน ใน global scope
<u>MediaPlayer startMP</u>	เครื่องเล่นเสียง Background
<u>MediaPlayer transitionMP</u>	เครื่องเล่นเสียง Transition
- Media START_SOUND	เสียง Background
- Media TRANSITION_SOUND	เสียง Transition
- Pane backgroundPane	Container หลัก
- boolean active	Suppress Thread หากหน้าปัจจุบันไม่ใช่ Charcomponent



### 3.5.1.2 Constructor

+ CharComponent()	วาดหน้าเลือกตัวละครรวมทั้งสร้าง Thread ที่ไว้ Handle การกดปุ่มและเปลี่ยนรูปตัวละครต่างๆ
-------------------	---

### 3.5.1.3 Method

+ void newRound()	เคลียร์ค่าเก่าให้พร้อมรับการเลือกตัวละครรอบใหม่ และหยุดการ Suppress Thread
+ void hh(KeyCode k)	ผ่านการกดปุ่มลงไปให้Player จัดการ
- void drawBackground()	วาด Layout
+ void startSound()	เล่นเพลง Background
+ void stopSound()	หยุดเพลง Background
+ void startTransitionSound()	เล่นเสียง Transition
+ <u>CharComponent getInstance()</u>	Return instance ปัจจุบัน
+ Pane getBackgroundPane()	Return Container หลัก

## 3.5.2 Class PlayCard extends Pane

Character Card สำหรับ Player

### 3.5.2.1 Field

Players player	Player ที่ link ด้วยปัจจุบัน
- Button Cname	Label ชื่อตัวละคร
- Button Team	Label Team
- ImageView iv	ภาพตัวละคร

### 3.5.2.2 Constructor

- PlayCard(Players p,double posx,double posy,double width,double height)	Set ค่าต่างๆและตำแหน่งของ Card ตามที่ได้รับ
--	---

### 3.5.2.3 Method

+ void handleKey(KeyCode x)	ส่งค่าต่อไปให้ Player จัดการ
+ void update()	เปลี่ยนรูป / ทีม / Active ตาม player ในแต่ละ Frame

## 3.5.3 Class PlayCardBot extends Playcard

Character Card สำหรับ Bot สิ่งที่มีใน Playcard จะไม่เขียนที่นี่

### 3.5.3.1 Field

- Button Cplus	เลื่อนตัวละครเป็นทางด้านขวา
- Button Cminus	เลื่อนตัวละครเป็นทางด้านซ้าย
- Button Teamplus	เพิ่มเลขทีม
- Button Teamminus	ลดเลขทีม
- Button Activate	เปิด/ปิด Bot

### 3.5.3.2 Constructor

+ PlayCardBot(Players p,double posx,double posy,double width,double height)	สร้าง Card ตัวละครและวางตำแหน่งต่างๆตามที่ได้รับมา โดยโยนไปให้ Parent Class ด้วย
---	--

### 3.6 Package main.controls

#### 3.6.1 Class ControlsComponent implements BackgroundMusic

Control menu

##### 3.6.1.1 Field

- <u>ClassLoader CLASS_LOADER</u>	สำหรับ Load Resource ใน Class นี้
+ <u>ControlsComponent instance</u>	สำหรับเรียก Instance ปัจจุบัน ใน global scope
<u>MediaPlayer startMP</u>	เครื่องเล่นเสียง Background
<u>MediaPlayer transitionMP</u>	เครื่องเล่นเสียง Transition
- Media START_SOUND	เสียง Background
- Media TRANSITION_SOUND	เสียง Transition
- Pane backgroundPane	Container หลัก
- ArrayList<KeyComponent> list	เก็บตัวจัดการปุ่มทั้งหมด

##### 3.6.1.2 Constructor

+ ControlsComponent()	สร้าง Menu คุ่มปุ่ม , วาด Layout , สร้าง component คุ่มปุ่มทุกปุ่ม และสร้าง Thread เพื่อรองรับการ Update
-----------------------	--

### 3.6.1.3 Method

+ void startSound()	เล่นเพลง Background
+ void stopSound()	หยุดเพลง Background
+ void startTransitionSound()	เล่นเสียง Transition
+ ControlsComponent getInstance()	Return instance ปัจจุบัน
+ Pane getBackgroundPane()	Return Container หลัก

### 3.6.2 Class ControlsHandler

จัดการ action ต่างๆของ control menu

#### 3.6.2.1 Method

+ <u>void moveToMenuScene()</u>	เลื่อนกลับไปเมนูหลัก
---------------------------------	----------------------

### 3.6.3 Class KeyComponent extends Pane

Panel Set Control ของแต่ละปุ่ม

#### 3.6.3.1 Field

- Button Keyname	Label ที่บอกชื่อ Action(ใช้เป็น Button เพราะจัด Center ง่าย)
- Button Keyvalue	Label ที่บอกKeyCode(ใช้เป็น Button เพราะจัด Center ง่าย)
- int player	บอกว่าปุ่มนี้จัดการ Player ไหน
- int key_number	บอกว่าปุ่มนี้จัดการ Action ไหน

### 3.6.3.2 Constructor

+ KeyComponent(int player,int key_number,double width,double height,double posx,double posy,String lab)	สร้าง Component Set Control ต่อ 1 ปุ่ม โดยสามารถ Set ตำแหน่ง ผู้เล่น Label ได้เลย รวมถึง Hook event ต่างๆเสร็จในนี้ รวมทั้ง Thread ที่ไว้เปลี่ยน Control
---	---

### 3.6.3.3 Method

+ int GetPlayer()	คืนค่าผู้เล่นที่ปุ่มคิดอยู่
+ int GetKey()	คืนค่าว่า Control นี้คุม action อะไร
+ void update()	Update ทุกเฟรมเช่น โดนกดแล้วจะเป็นสีฟ้าจนกว่าจะมีคนมากด Cancel หรือปุ่มใหม่

## 3.7 Package main.exception

### 3.7.1 Class NotEnoughPlayersOrTeamsException

เป็น Exception ที่โยนหากเลือกผู้เล่นหรือทีมไม่ครบ

#### 3.7.1.1 Field

+ <u>long serialVersionUID</u>	Serial
--------------------------------	--------

#### 3.7.1.2 Constructor

+ NotEnoughPlayersOrTeamsException(String message)	ผลิต Exception ด้วย message
--	-----------------------------

### 3.8 Package interfaces

#### 3.8.1 Interface BackGroundMusic

เป็นเครื่องเตือนใจว่าห้ามลืม implement ระบบเสียง

##### 3.8.1.1 Method

+ void startSound()	บังคับว่าต้องมีฟังก์ชันเล่นเสียง
+ void stopSound()	บังคับว่าต้องมีฟังก์ชันหยุดเสียง

### 3.9 Package main.menu

#### 3.9.1 Class MenuComponent implements BackGroundMusic

เมนูหลัก

##### 3.9.1.1 Field

- <u>ClassLoader CLASS_LOADER</u>	สำหรับ Load Resource ใน Class นี้
+ <u>MenuComponent instance</u>	สำหรับเรียก Instance ปัจจุบัน ใน global scope
<u>MediaPlayer startMP</u>	เครื่องเล่นเสียง Background
<u>MediaPlayer transitionMP</u>	เครื่องเล่นเสียง Transition

- Media START_SOUND	เสียง Background
- Media TRANSITION_SOUND	เสียง Transition
- Pane backgroundPane	Container หลัก
- Canvas canvasVSMODE	ปุ่ม Start
- Canvas canvasControls	ปุ่มปรับ Control
- Canvas canvasExit	ปุ่ม Exit

### 3.9.1.2 Constructor

+ MenuComponent()	สร้าง Menu , Load เสียง และ วาดทุกอย่างบนเมนู
-------------------	--

### 3.9.1.3 Method

+ void startSound()	เล่นเพลง Background
+ void stopSound()	หยุดเพลง Background
+ void startTransitionSound()	เล่นเสียง Transition
+ <u>MenuComponent getInstance()</u>	Return instance ปัจจุบัน
+ Pane getBackgroundPane()	Return Container หลัก

## 3.9.2 Class MenuHandler

จัดการ Action ต่างๆของเมนู

### 3.9.2.1 Method

+ <u>void exit()</u>	ออกจากเกม
----------------------	-----------

+ <u>void moveToControlsScene()</u>	เลื่อนไปหน้า Set Control
+ <u>void moveToGameScene()</u>	เลื่อนไปหน้า เลือกตัวละคร
+ <u>void drawBorder(Canvas canvas, Color color)</u>	Effect เมื่อลากเมาส์ไปเหนือปุ่ม
+ <u>void undrawBorder(Canvas canvas, String text)</u>	เมื่อเอาเมาส์ออกให้กลับไปเหมือนเดิม

### 3.10 Package skills

#### 3.10.1 Abstract Class Skill extends Pane

เป็น Parent Class ของสกิลทั้ง 2 แบบด้านบน โดยเป็น Pane ในตัว คือเป็น instance ของ graphic Skill บนหน้าจอด้วย

##### 3.10.1.1 Field

# int team	บอกว่าเป็น Skill ของทีมไหน จะไม่โดนทีมเดียวกันเอง
# int damage	พลังการโจมตี
# int turn	การหันของ Skill ซ้าย,ขวา
# int mana	Mana ที่ใช้
# int posx	ตำแหน่งในสนามรบแกน X
# int posy	ตำแหน่งในสนามรบแกน Y
# int posz	ตำแหน่งในสนามรบแกน Z



# int rx	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเขีคว่าโดนไหม(สำหรับ FloatSkill) ในแนวแกน X
# int ry	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเขีคว่าโดนไหม(สำหรับ FloatSkill) ในแนวแกน Y
# int rz	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเขีคว่าโดนไหม(สำหรับ FloatSkill) ในแนวแกน Z
# int tilldead	บอกว่าแสดง Effect ได้อีกกี่ frame ก่อนโดนลบทิ้ง
# int effdu	ระยะเวลา Effect ทั้งหมดก่อนโดนลบทิ้ง
# boolean blockable	สามารถ Block ได้หรือไม่
# boolean dead	โดนสั่งว่า จะโดนลบในอีก tilldead frame หรือยัง
# boolean hit	โดนเป้าหมายหรือยัง(หาก Skill หลุดนอกรอบ จะโดนสั่ง Dead แต่ไม่สั่ง Hit ทำงานไม่ซ้ำซ้อนกัน)
# Image body	ภาพ Skill หลังร้าย
# Image hitPic	ภาพ Skill หลังโดนเป้าหมาย
# ImageView iv	Component ไว้อโฮลดภาพ
# Media startsound	เสียงหลังร้าย
# Media hitsound	เสียงหลังโดนเป้าหมาย

### 3.10.1.2 Constructor

+ Skill(int team,int damage,int turn,int mana,Image body,Image hitPic,int posx,int posy,int posz,int rx,int ry,int rz,boolean blockable,int effdu,Media startsound,Media endsound)	เก็บข้อมูลของ Skill ทั้งหมด พร้อมเริ่มเล่นเสียงขณะร้าย และ set ภาพเป็นภาพ Skill ตอนเริ่มร้าย
--	--

### 3.10.1.3 Method

+ abstract boolean checkhit(Players p)	เช็ควาสกิลโดนเป้าหมายหรือไม่
+ abstract void update()	การเปลี่ยนแปลงของสกิลทุก Frame
+ boolean des()	เรียกเช็คได้ว่าควรจะลบ Skill นี้ออกจากหน้าจอและระบบ
+ void sigDead()	บอก Skill ว่าโดนเป้าหมาย และให้เริ่มแสดง effect ตอนโดนเป้าหมาย
+ boolean isdead()	เรียกเช็คว่าได้โดนเป้าหมายไปหรือยัง

### 3.10.2 Class FloatSkill extends Skill

Skill ที่พุ่งไปข้างหน้าเรื่อยๆจนกว่าจะชนเป้าหมายหรือหมดระยะ ทุกอย่างที่มีใน Class Skill จะไม่ถูกเขียนที่นี่

#### 3.10.2.1 Field

- int range	ระยะมากที่สุดของ Skill
- int velo	ความเร็วในการพุ่งของ Skill

#### 3.10.2.2 Constructor

+ FloatSkill(int team,int damage,int turn,int mana,Image body,Image hitPic,int posx,int posy,int posz,int rx,int ry,int rz,boolean blockable,int range,int velo,int effdu,Media startsound,Media endsound)	เก็บข้อมูลโดยเรียก Parent Constructor + 2 field ด้านบน และเซตตำแหน่งเริ่มต้นของ Skill บนจอ
--	--

### 3.10.2.3 Method

+ boolean checkhit(Players p)	เช็คว่าเป็น Frame นั้น Skill โดน Player p หรือไม่
+ void update()	เซตเหตุการณ์ที่ควรเกิดในแต่ละ Frame เช่นตำแหน่ง ลดเวลา effect เป็นต้น

### 3.10.3 Class PointSkill extends Skill

Skill ที่อยู่กับที่ แต่โดนได้หลายผู้เล่นพร้อมกัน ทุกอย่างที่มีใน Class Skill จะไม่ถูกเขียนที่นี่

#### 3.10.3.1 Constructor

PointSkill(int team,int damage,int turn,int mana,Image body,Image hitPic,int posx,int posy,int posz,int rx,int ry,int rz,boolean blockable,int effdu,Media startsound,Media endsound)	เก็บข้อมูลโดยเรียก Parent Constructor และเซตตำแหน่งเริ่มต้นของ Skill บนจอ และเริ่มมาให้แสดง effect พร้อมรายละเอียดว่าโดนเป้าหมายทันที
---	---

#### 3.10.3.2 Method

+ boolean checkhit(Players p)	เช็คว่าเป็น Frame นั้น Skill โดน Player p หรือไม่
+ void update()	เช็คเหตุการณ์ที่ควรที่จะเกิดในแต่ละ Frame เช่นตำแหน่ง ลดเวลา effect เป็นต้น

### 3.10.4 Class SkillDetail

เก็บข้อมูลของ Skill ที่กำหนด เพื่อให้ง่ายต่อการนำไปปล่อยใน Arena ขณะเล่น โดย Class นี้ไม่ใช่ Getters  
 เนื่องจากตัวแปรเยอะจะทำให้ Code ดูรก และยากต่อการใช้งานมากกว่าการ Access โดยตรง

#### 3.10.4.1 Field

+ <u>ClassLoader CLASS_LOADER</u>	ใช้ Load Resource สำหรับ Class นี้
+ int damage	พลังการโจมตี
+ int mana	Mana ที่ใช้
+ int rx	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเช็คว่าเป็นโดนไหม(สำหรับ FloatSkill) ในแนวแกน X
+ int ry	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเช็คว่าเป็นโดนไหม(สำหรับ FloatSkill) ในแนวแกน Y
+ int rz	รัศมีการทำลายล้าง(สำหรับ PointSkill) หรือรัศมีการเช็คว่าเป็นโดนไหม(สำหรับ FloatSkill) ในแนวแกน Z
+ int range	ศูนย์กลาง Skill (สำหรับ PointSkill) หรือ ระยะทางมากที่สุดที่เคลื่อนที่ได้(สำหรับ FloatSkill)
+ int velo	ความเร็วในการเคลื่อนที่ของ FloatSkill

+ int effdu	ระยะเวลาของภาพ effect หลังจาก Skill โดนเป้าหมาย
+ boolean blockable	สามารถ Block ได้หรือไม่
+ Image body	ภาพสกิลตอนเริ่มร้าย
+ Image hit	ภาพ effect เมื่อโดนเป้าหมาย
+ Media startsound	เสียงขณะร้าย
+ Media hitsound	เสียงขณะโดนเป้าหมาย

#### 3.10.4.2 Constructor

SkillDetail(int damage,int mana,int rx,int ry,int rz,int range,int velo,boolean blockable,String folder,int effdu,String sfolder)	เก็บข้อมูลของ Skill ที่ได้รับ โดยหากเป็นภาพและเสียงจะ Load จาก Resource ไว้ใน Class นี้เลย
---	--

