

คู่มือเกม Weeb Simulator 2020

โดย

นาย ภพธรรม เจริญแนว 6231346721

นาย ภู จongsarit 6231348021

2110215 Programming Methodology

ปีการศึกษา 2562 ภาคปลาย จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
1. Main menu	1
2. Main game	2
3. Help	5
4. Game Over	6
5. Class diagram	7
6. Packages	8
6.1. package main	8
6.2. package control	9
6.3. package base	26
6.4. package exception	28
6.5. package sprite.base	29
6.6. package sprite	31

1. Main menu



Picture 1.1 หน้าตัวเกมหลัก

หน้า Main menu จะมี 3 ตัวเลือกให้เลือกประกอบด้วย :

1. Start ใช้สำหรับเริ่มเกม
2. Help เพื่อดูวิธีเล่น
3. Quit เพื่อออกจากเกม

ซึ่งผู้เล่นจะสามารถเลือกตัวเลือกต่างๆ ด้วยการคลิกเมาส์

2. Main game

2.1. Game's Concept

โดยพื้นฐานการเล่นเกมนี้นี้จะต้องควบคุมตัวละครไปยังเป้าหมายของตัวละครนั้นๆ ภายในเวลาที่กำหนด โดยผู้เล่นจะควรหลีกเลี่ยงการเหยียบกับระเบิด (การเหยียบกับระเบิดจะทำให้ตัวละครเคลื่อนที่ช้าลง) ซึ่งการเก็บ item ที่ถูกต้อง ผู้เล่นจะได้เวลาในการเล่นเพิ่ม และ เมื่อเวลาการเล่น(เกจด้านบนสุด) หมดลง เกมจะจบ



รูป 2.1 เกจแสดงเวลาการเล่น

2.2. ตัวละคร, เป้าหมาย, กับระเบิด

ในเกมนี้นี้จะมีตัวละคร 3 ตัว

1. Frank the ninjutsu runner.



รูป 2.2 Frank the ninjutsu runner

2. The weeb boi.



รูป 2.3 The weeb boi

3. The mighty Thanos.



รูป 2.4 the Mighty Thanos

ซึ่งตัวละครแต่ละตัวจะมีเป้าหมายที่ต่างกันโดย

- เป้าหมายของ frank คือการปลดปล่อยมนุษย์ต่างดาวออกจาก Area 51



รูป 2.5 มนุษย์ต่างดาว

- เป้าหมายของ weeb boi คือการหาสาว 2D มาเพื่อที่จะคบกับเขา



รูป 2.6 สาว 2D

- เป้าหมายของ Thanos คือการครอบครอง Infinity gauntlet เพื่อจะสานต่อภารกิจของเขาให้จบ



รูป 2.7 Infinity gauntlet



รูป 2.8 ไดอะแกรมที่แสดงความเชื่อมโยงเป้าหมายต่างๆ กับตัวละครแต่ละตัว

อย่างไรก็ตามเพื่อที่จะทำให้เกมยากขึ้นไปอีกระดับ ผู้พัฒนาเกมได้ใส่กับระเบิดเป็นอุปสรรคภายในเกมโดยเมื่อผู้เล่นเหยียบกับระเบิด จะเคลื่อนที่ช้าลง



Picture 2.9 Landmine

2.3. การควบคุมตัวละคร

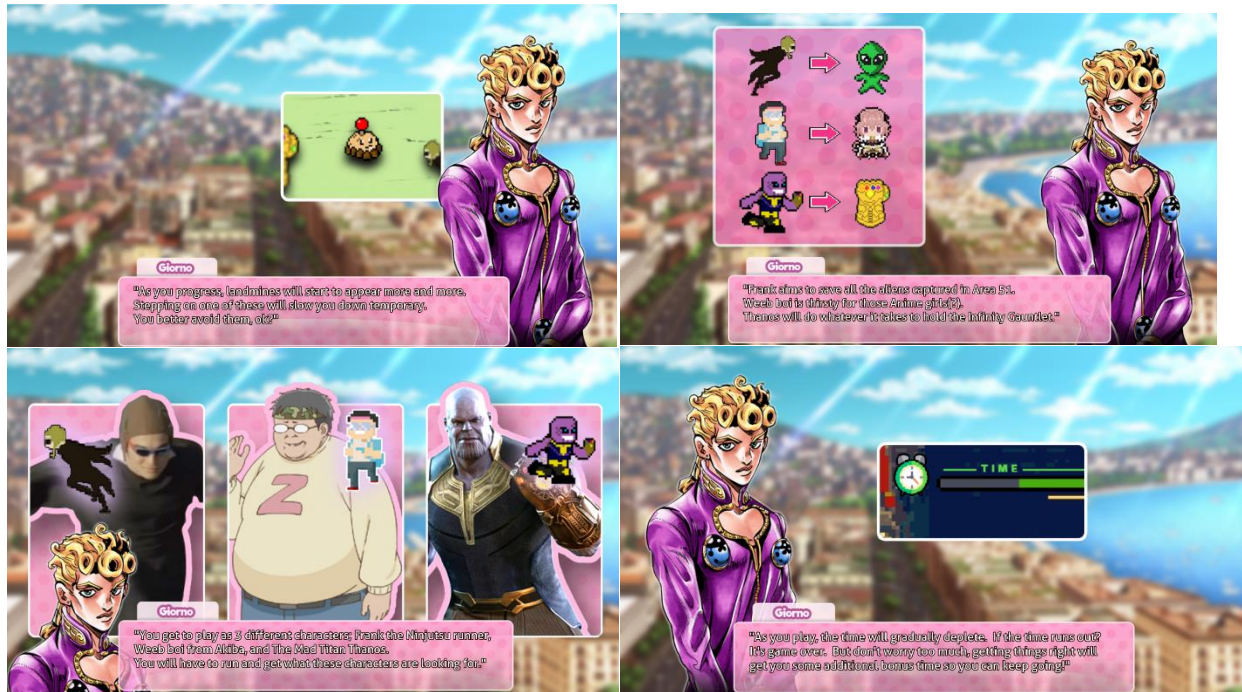
W / ลูกศรขึ้น	ตัวละครเคลื่อนที่ขึ้น
S / ลูกศรลง	ตัวละครเคลื่อนที่ลง
A / ลูกศรขวา	ตัวละครเคลื่อนที่ไปทางขวา
D / ลูกศรขวาซ้าย	ตัวละครเคลื่อนที่ไปทางซ้าย
LSHIFT / RSHIFT	เร่งความเร็วตัวละคร *** การเร่งความเร็วจะใช้ไม่ได้เมื่อเกจการ boost (เกจสีเหลือง) หมด



รูป 2.10 เกจสีเหลืองแสดงถึงความสามารถในการเร่งความเร็วที่เหลือ ณ ขณะใดๆ

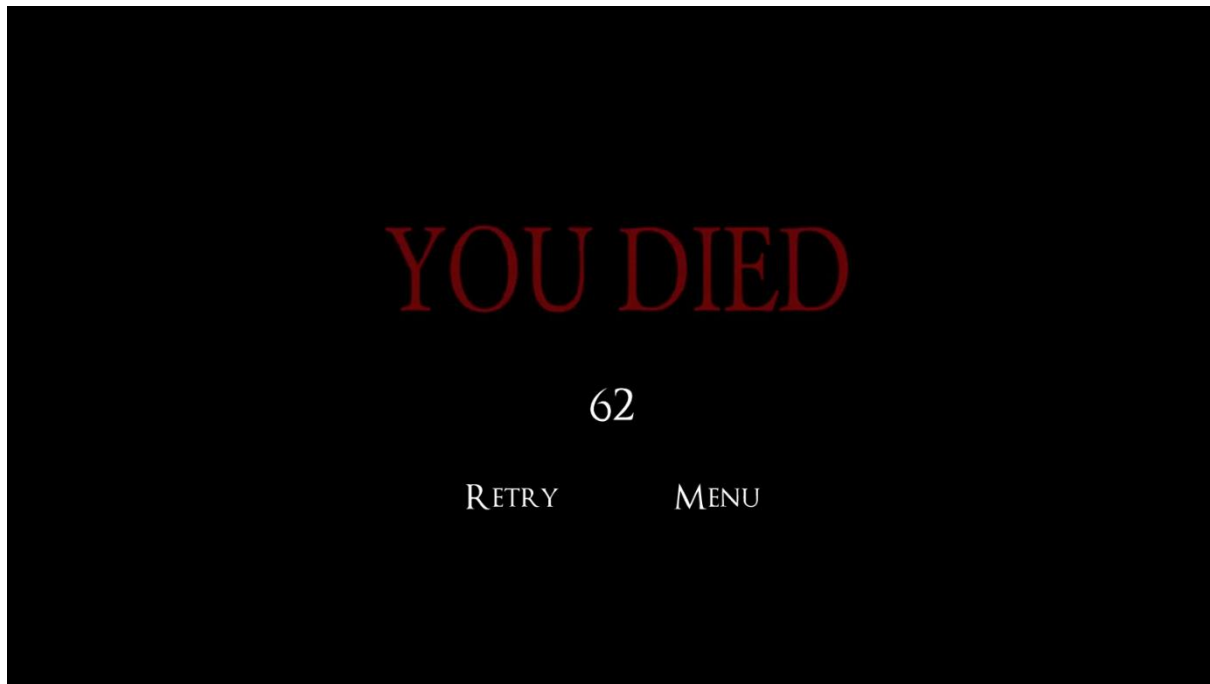
3. Help

ในส่วน Help นี้จะให้ข้อมูลผู้เล่นเกี่ยวกับวิธีการเล่น, กฎ, และ mechanic เล็กๆ ภายในเกม



รูป 3.1 ตัวอย่างข้อมูลในหน้า Help

4. Game Over



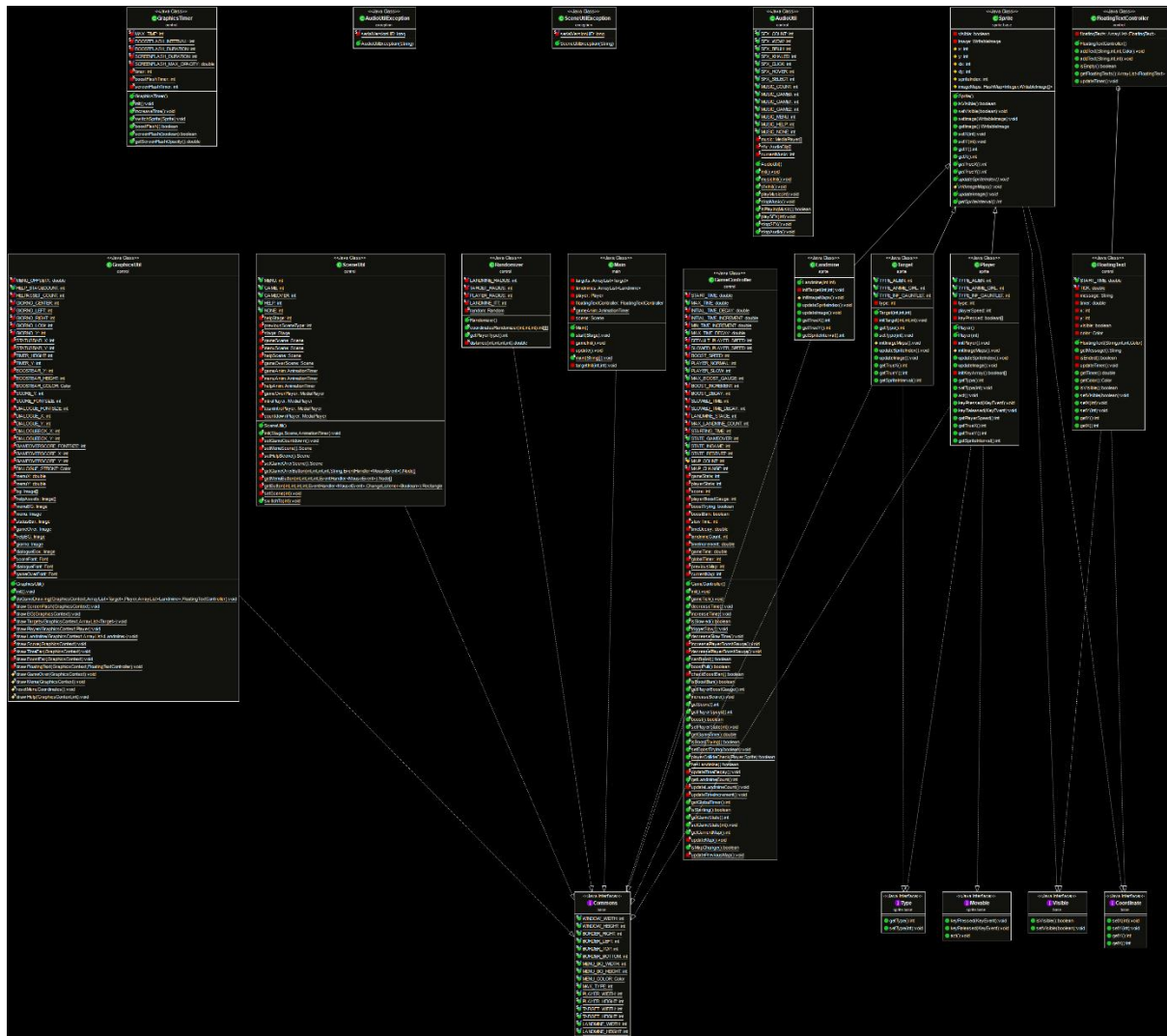
รูป 4.1 หน้า Game Over

เมื่อเวลาการเล่นหมดลง ตัวเกมจะเล่น animation เพื่อแสดงตัวอักษร “YOU DIED” ซึ่งผู้เล่นสามารถเลือกที่จะข้าม animation ได้ด้วยการ คลิกเมาส์ หรือกดคีย์ใดๆ บนคีย์บอร์ด หลังจาก animation นี้ จะมีการแสดงคะแนนที่ผู้เล่นทำได้ระหว่างเล่น และจะมีตัวเลือกให้ผู้เล่น 2 ตัวเลือก คือ Retry และ Menu

โดย

- ถ้าผู้เล่นเลือกตัวเลือก Retry จะเป็นการเล่นเกมหลักอีกครั้ง
- ถ้าผู้เล่นเลือกตัวเลือก Menu จะพาผู้เล่นกลับไปหน้า Main Menu

5. Class diagram



6. Method and Field

6.1. package Main

6.1.1. Class Main extends Applcation

6.1.1.1. Fields

- ArrayList<Target> targets	ArrayList ที่บรรจุ target
- ArrayList<Landmine> landmines	ArrayList ที่บรรจุ landmine
- Player player	Player ในเกม
- FloatingTextController floatingTextController	Controller ของ FloatingText
- AnimationTimer <u>gameAnim</u>	AnimationTimer ควบคุมการเล่นเกม
- scene	Scene ของเกมหลัก

6.1.1.2. Methods

+void Start(Stage stage)	ตั้งค่า stage คือ - สร้าง scene - รัน GraphicsUtil.init() - รัน AudioUtil.init() - กำหนดให้ gameAnim คอยตรวจสอบสถานะของเกม คือ ○ ถ้ายังอยู่ในเกมให้ อัปเดตเกมต่อไปเรื่อยๆ ○ การ restart เกม จะถูกรัน Trigger ผ่าน Methods นี้ ○ ในกรณีอื่นๆ ให้ จะสั่งให้เกมจบ (game over) ○ รัน SceneUtil.init() และ แสดงผล
- void gameInit()	- สร้าง player - รัน gamecontroller()
- void update()	อัปเดตสถานะของเกมและองค์ประกอบอื่นๆ เช่น Player ถูกลดความเร็วหรือไม่ Sprite Player มีการซ้อนทับกับ Sprite อื่นๆ หรือไม่ เป็นต้น
- void main(String[] args)	รัน launch() เพื่อเริ่มต้น application
- void targetInit(int playerX, int playerY)	สร้าง target ภายในเกม

6.2. package Control

6.2.1. Class AudioUtil

6.2.1.1. Fields

+ int <u>SFX_COUNT</u>	- จำนวนเสียงประกอบทั้งหมดที่ใช้ - มีค่าเท่ากับ 6
+ int <u>SFX_WOW</u>	- แสดงถึงประเภทของเสียงประกอบ “wow” - มีค่าเท่ากับ 0
+ int <u>SFX_BRUH</u>	- แสดงถึงเสียงประกอบ “BRUH” - มีค่าเท่ากับ 1
+ int <u>SFX_KHALED</u>	- แสดงถึงเสียงประกอบ จังหวะเหยียบระเบิด - มีค่าเท่ากับ 2
+ int <u>SFX_CLICK</u>	- แสดงถึงเสียง “Click” ที่หน้า Main menu - มีค่าเท่ากับ 3
+ int <u>SFX_HOVER</u>	- แสดงถึงเสียงประกอบการเล่นเมาส์บนตัวเลือกในหน้า Main Menu - มีค่าเท่ากับ 4
+ int <u>SFX_CLICK</u>	- แสดงถึงเสียงประกอบการเล่นเมาส์บนตัวเลือกบนหน้า Game Over - มีค่าเท่ากับ 5
+ int <u>MUSIC_COUNT</u>	- แสดงถึงจำนวนเพลงที่ใช้ในเกม - มีค่าเท่ากับ 5
+ int <u>MUSIC_GAME0</u>	- แสดงถึงเพลงที่ใช้ในด้านที่ 1 ในส่วนตัวเกมหลัก - มีค่าเท่ากับ 0
+ int <u>MUSIC_GAME1</u>	- แสดงถึงเพลงที่ใช้ในด้านที่ 2 ในส่วนตัวเกมหลัก - มีค่าเท่ากับ 1
+ int <u>MUSIC_GAME2</u>	- แสดงถึงเพลงที่ใช้ในด้านที่ 3 ในส่วนตัวเกมหลัก - มีค่าเท่ากับ 2
+ int <u>MUSIC_MENU</u>	- แสดงถึงเพลงที่เล่นบนหน้า Main Menu - มีค่าเท่ากับ 3

+ int <u>MUSIC_HELP</u>	- แสดงถึงเพลงที่เล่นบนหน้า Help - มีค่าเท่ากับ 4
+ int <u>MUSIC_NONE</u>	- แสดงถึงการไม่เล่นเพลงใดๆ - มีค่าเท่ากับ -1
- MediaPlayer[] music	Array ที่จะเก็บเพลงที่ใช้ทั้งหมดในเกม
- AudioClip[] sfx	Array ที่เก็บเสียงประกอบทั้งหมดที่ใช้ในเกม
- int currentMusic	ตัวแปรระบุเพลงที่เล่นอยู่ ณ ขณะนั้น

6.2.1.2 Methods

+ void <u>init()</u>	รัน method sfxInit(), musicInit() และตั้งค่า currentMusic เป็น MUSIC_NONE (-1)
+ void <u>musicInit()</u>	- โหลดเพลงทั้งหมดที่ใช้ภายในเกมจาก res - บรรจุเพลงที่โหลดลง Array music และตั้งค่าระดับเสียงเป็น 0.5
+ void <u>sfxInit()</u>	โหลดเสียงประกอบทั้งหมดที่ใช้ภายในเกมจาก Folder res และบรรจุลง Array sfx และตั้งค่าระดับเสียงเป็น 1
+ void <u>playMusic(int type)</u>	เล่นเพลงโดยอ้างอิงจากค่า type โดยในกรณีที่ ค่า type มีค่าเท่ากับ MUSIC_MENU เมื่อเพลงจบจะวนกลับมาที่วินาทีที่ 22 (เพื่อการดูที่ดีขึ้น) และในกรณีอื่นจะวนกลับไปวินาทีที่ 0
+ void <u>stopMusic(int type)</u>	หยุดเพลงใดๆ ที่เล่นอยู่ และ ตั้งค่าเป็น MUSIC_NONE
+ Boolean <u>isPlayingMusic()</u> throw AudioUtilException	- ตรวจสอบว่าในขณะนี้มีการเล่นเพลงหรือไม่ - จะมีการ throw AudioUtilException ในกรณีที่ currentMusic มีค่านอกเหนือจากช่วง [0, MUSIC_COUNT)
+ void <u>playSFX(int type)</u>	เล่นเสียงประกอบโดยอ้างอิงจากค่า type
+ void <u>stopSFX()</u>	หยุดเสียงประกอบทั้งหมด
+ void <u>stopAudio()</u>	รัน stopMusic() และ stopSFX() เพื่อหยุดเสียงทั้งหมด

6.2.2. Class FloatingTextController

6.2.2.1. Fields

- ArrayList<FloatingText> floatingTexts	ArrayList ที่ใช้บรรจุ FloatingText ที่ใช้ในขณะนั้น
---	--

6.2.2.2. Constructor

+ FloatingTextController()	กำหนดให้ floatingTexts เป็น ArrayList ว่าง
----------------------------	--

6.2.2.3. Methods

+ void addText(String message, int x, int y, Color color)	สร้าง floatingText ที่มีข้อความ message มีพิกัดเป็น (x,y) มีสีตามตัวแปร color และ ใส่ลง Array floatingTexts
+ void addText(String message, int x, int y)	สร้าง floatingText ที่มีข้อความ message มีพิกัดเป็น (x,y) มีสีขาว และ ใส่ลง floatingTexts
+ boolean isEmpty()	ตรวจสอบว่า Array floatingTexts ว่างหรือไม่
+ ArrayList<FloatingText> getFloatingTexts()	return ตัวแปร floatingTexts
+ void updateTimer()	ตรวจสอบว่า floatingText ตัวใดเล่นจบหรือไม่ ถ้าเล่นจบค่า visible จะถูกกำหนดเป็น false และจะถูก remove ออกจาก floatingTexts

6.2.2.4. Inner Class FloatingText implements Visible, Coordinate

6.2.2.4.1. Fields

+ double <u>START_TIME</u>	แสดงถึงเวลาที่ floatingText แสดงบนหน้าจอ
- String message	ข้อความของ floatingText
- int x	พิกัดในแนวแกน X ของ floatingText
- int y	พิกัดในแนวแกน Y ของ floatingText
- boolean visible	ระบุสถานะการถูกเห็นของ floatingText
- double timer	เวลาที่เหลืออยู่ของ floatingText
- Color color	สีของข้อความของ floatingText
- double <u>TICK</u>	แสดงถึงเวลาต่อการเลื่อน floatingText 1 ครั้ง

6.2.2.4.2. Constructor

+ FloatingText(String message, int x, int y, Color color)	<ul style="list-style-type: none">- ตั้งค่าตัวแปร timer ให้มีค่าเท่ากับ START_TIMER- ตั้งค่าให้ floatingText ให้ถูกเห็นได้ (visible = true)- ตั้งค่าให้ข้อความของ floatingText มีค่าเท่ากับ message- ตั้งค่าพิกัดในแกน X เป็น x- ตั้งค่าพิกัดในแกน Y เป็น y- ตั้งค่าสีของข้อความเป็น Color
---	---

6.2.2.4.3. Methods

+ String getMessage()	return ตัวแปร message
- boolean isEnded()	ตรวจสอบว่า floatingText เล่นจบหรือไม่
- void updateTimer()	ลดเวลาการเล่นของ floatingText และจะลดค่า y ลง 1 เมื่อค่า timer หายด้วย TICK ลงตัว
+ boolean isVisible()	return ตัวแปร floatingText
+ void setVisible(boolean visible)	ตั้งค่าตัวแปร visible ให้มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ double getTimer()	return ตัวแปร timer

+ void setX(int x)	ตั้งค่าตัวแปร x ให้มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ void setY(int y)	ตั้งค่าตัวแปร y ให้มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ double getX()	return ตัวแปร x
+ double getY()	return ตัวแปร y
+ double getColor()	return ตัวแปร color

6.2.3. Class GraphicUtil implements Commons

6.2.3.1. Fields

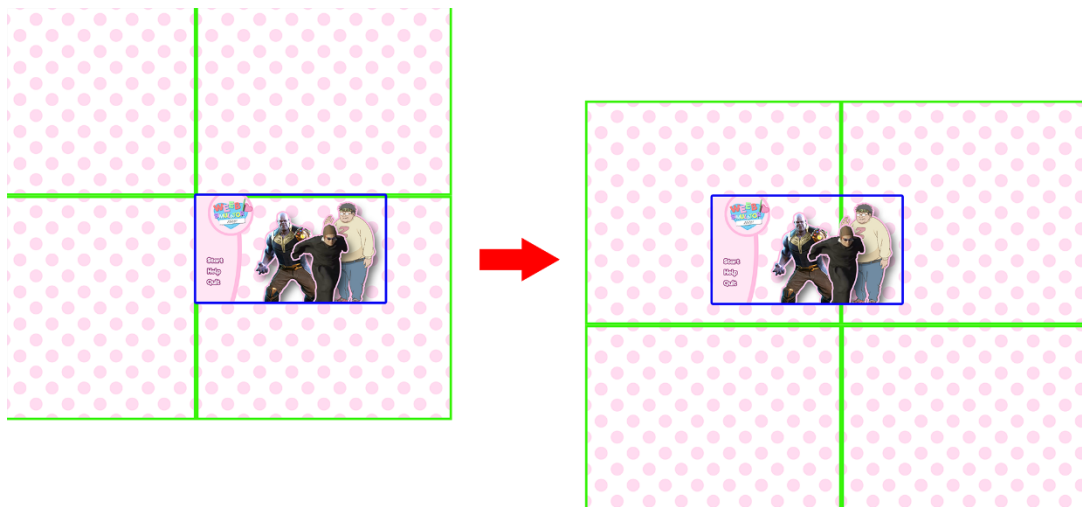
- double <u>MENU_OFFSETX</u>	- ค่า offset แกน X ของ menu - มีค่าเท่ากับ 242
- int <u>HELP_STAGECOUNT</u>	จำนวน stage ทั้งหมดของ help
- int <u>HELPASSET_COUNT</u>	จำนวนภาพประกอบของ help
- int <u>GIORNO_CENTER</u>	ตำแหน่งตามแกน X เมื่อ giorno อยู่ ณ กึ่งกลางหน้าจอ
- int <u>GIORNO_LEFT</u>	ตำแหน่งตามแกน X เมื่อ giorno อยู่ ณ ด้านซ้ายของหน้าจอ
- int <u>GIORNO_RIGHT</u>	ตำแหน่งตามแกน X เมื่อ giorno อยู่ ณ ด้านขวาของหน้าจอ
- int <u>GIORNO_LOW</u>	ตำแหน่งตามแกน Y เมื่อ giorno อยู่ ณ ด้านล่างของหน้าจอ
- int <u>GIORNO_Y</u>	ตำแหน่งตามแกน Y ของ giorno
- int <u>STATUSBAR_X</u>	ตำแหน่งตามแกน X ของ Status Bar
- int <u>STATUSBAR_Y</u>	ตำแหน่งตามแกน Y ของ Status Bar
- int <u>TIMER_HEIGHT</u>	ความสูง ของ Timer Bar
- int <u>TIMER_Y</u>	ตำแหน่งตามแกน Y ของ Timer Bar
- int <u>BOOSTBAR_Y</u>	ตำแหน่งตามแกน Y ของ Boost Bar
- int <u>BOOSTBAR_HEIGHT</u>	ความสูงของ Boost Bar
- int <u>Color BOOSTBAR_COLOR</u>	สีของ Boost Bar
- int <u>SCORE_Y</u>	ตำแหน่งตามแกน Y ของคะแนนที่แสดงในหน้าเกมหลัก

- int <u>SCORE_FONTSIZE</u>	ขนาด Font ของคะแนนที่แสดงในหน้าเกมหลัก
- int <u>DIALOGUE_FONTSIZE</u>	ขนาด Font ในหน้า help
- int <u>DIALOGUE_X</u>	ตำแหน่งตามแกน X ของข้อความในหน้า Help
- int <u>DIALOGUE_Y</u>	ตำแหน่งตามแกน Y ของข้อความในหน้า Help
- int <u>DIALOGUEBOX_X</u>	ตำแหน่งตามแกน X ของกล่องข้อความในหน้า Help
- int <u>DIALOGUEBOX_Y</u>	ตำแหน่งตามแกน Y ของกล่องข้อความในหน้า Help
- int <u>GAMEOVERSCORE_FONTSIZE</u>	ขนาด Font ในหน้า GameOver
- int <u>GAMEOVERSCORE_X</u>	ตำแหน่งตามแกน X ของคะแนนที่แสดงในหน้าเกม GameOver
- int <u>GAMEOVERSCORE_Y</u>	ตำแหน่งตามแกน Y ของคะแนนที่แสดงในหน้าเกม GameOver
- Color <u>DIALOGUE_STROKE</u>	สีเส้นขอบของข้อความในหน้า Help
- double <u>menuX</u>	ตำแหน่ง background ตามแกน X ของ Menu
- double <u>menuY</u>	ตำแหน่ง background ตามแกน Y ของ Menu
- Image[] <u>bg</u>	ArrayList ที่ใช้บรรจุฉากหลัง
- Image[] <u>helpAssets</u>	ArrayList ภาพประกอบในหน้า Help
- Image <u>menuBG</u>	ฉากหลังของ Menu
- Image <u>menu</u>	ภาพฉากหน้า Menu
- Image <u>statusBar</u>	ภาพ statusBar
- Image <u>gameOver</u>	ภาพหน้า GameOver
- Image <u>helpBG</u>	ฉากหลังของ help
- Image <u>giorno</u>	รูปของ giorno (ตัวละครที่ใช้แนะนำเกมในหน้า Help)
- Image <u>dialogueBox</u>	รูปกล่องข้อความที่ใช้ในหน้า Help
- Font <u>scoreFont</u>	Font ของคะแนนที่แสดงในหน้าเกมหลัก
- Font <u>dialogueFont</u>	Font ของข้อความในหน้า Help
- Font <u>gameOverFont</u>	Font ในหน้า GameOver

6.2.3.2. Methods

+ void <u>init()</u>	โหลดภาพต่างๆ จากโฟลเดอร์ resource มายังตัวแปรที่กำหนดและบรรจุตัวแปรลง Array bg
+ void <u>doGameDrawing(GraphicsContext g, ArrayList<Target> targets, Player player, ArrayList<Landmine> landmines, FloatingTextController floatingTextController)</u>	<ul style="list-style-type: none"> - แสดงผล player และ target ทั้งหมดใน targets ลงบนหน้าจอ - player, target จะมีการแสดงผล effect dropShadow (ในกรณีที่มีการกด boost จะมีการแสดงผล InnerShadow สีขาวด้วย) - แสดงผล TimeBar, BoostBar, Score, Landmine (ในกรณีที่มี), floatingText (ในกรณีที่มี) ลงบนหน้าจอ - ในกรณีที่เกมเพิ่งเริ่ม (ตรวจสอบผ่าน method isStarting() ของ GameController) จะมีการรัน method drawScreenFlash(g)
- void <u>drawScreenFlash(GraphicsContext g)</u>	แสดงผล effect แพลชลงบนหน้าจอเกม
- void <u>drawBG(GraphicsContext g)</u>	แสดงผลฉากของฉากปัจจุบันหลังลงบนหน้าจอเกม
- void <u>drawTargets(GraphicsContext g, ArrayList<Target> targets)</u>	แสดงผล target ทั้งหมดที่มีค่า visible เป็น true ใน targets ลงบนหน้าจอเกม
- void <u>drawPlayer(GraphicsContext g, Player player)</u>	แสดงผล player ลงบนหน้าจอเกม
- void <u>drawLandmine(GraphicsContext g, ArrayList<Landmine> landmines)</u>	แสดงผล landmine ทั้งหมดที่มีค่า visible เป็น true ใน landmines ลงบนหน้าจอเกม
- void <u>drawScore(GraphicsContext g)</u>	แสดงผลคะแนนลงบนหน้าจอเกม
- void <u>drawBoostBar(GraphicsContext g)</u>	แสดงผล Boost Bar ลงบนหน้าจอเกม
- void <u>drawFloatingText(GraphicsContext g, FloatingTextController floatingTextController)</u>	แสดงผล floatingText ทั้งหมดที่เกิดขึ้นในขณะนั้น และมีค่า visible เป็น true ใน floatingTexts ลงบนหน้าจอเกม
# void <u>drawGameOver(GraphicsContext g)</u>	แสดงผล graphic หน้า GameOver ลงบนหน้าจอเกม
# void <u>drawMenu(GraphicsContext g)</u>	- แสดงผลหน้า menu ลงบน g

	<ul style="list-style-type: none"> - วาดฉากหลังของเมนู 4 ครั้งตามตำแหน่งดังรูป 5.2.3.2.1 - เมื่อถูกเรียกจะเลื่อนฉากหลังลง 1 หน่วย และไปทางขวา 1 หน่วย (ดูภาพประกอบเพื่อความเข้าใจมากขึ้น) - เมื่อตำแหน่ง x, y ของฉากหลังเลื่อนจนสุด จะถูกรีเซ็ตกลับเป็นตำแหน่งเริ่มต้น - เมื่อวาดฉากหลังแล้วจึงวาดกรอบเมนูทับ
--	---



รูป 5.2.3.2.1 แสดงการทำงานของ method drawMenu(GraphicsContext g)

# void <u>resetMenuCoordinates()</u>	<ul style="list-style-type: none"> - ตั้งค่า menuX ให้เป็น MENU_OFFSETX - ตั้งค่า menuY ให้เป็น 0
# void <u>drawHelp(GraphicsContext g, int stage)</u>	<p>แสดงผล graphic ของหน้า Help ลงบน g โดย parameter stage จะถูกใช้เพื่อระบุหน้าที่ต้องการแสดงของ Help เช่น เมื่อค่า stage = 1 method จะแสดงผลหน้าที่ 1 ของ Help ลงบน g และเมื่อ ค่า stage มีค่าเท่ากับ 2 method ก็ จะแสดงผลหน้าที่ 2 ของ Help ลงบน g ตามลำดับ</p>

6.2.4. Class GameController implements Commons

6.2.4.1. Fields

- double <u>START_TIME</u>	- เวลาในการเล่น ณ ตอนเริ่มเกม - มีค่าเท่ากับ 800
+ double <u>MAX_TIME</u>	- เวลาในการเล่นสูงสุด - มีค่าเท่ากับ 1200
- double <u>INITIAL_TIME_DECAY</u>	- อัตราการลดของเวลาการเล่น ณ ตอนเริ่มเกม - มีค่าเท่ากับ 2.5
- double <u>INITIAL_TIME_INCREMENT</u>	- เวลาที่เพิ่มต่อการไปถึง target ที่ถูกต้อง 1 ครั้ง ณ ตอนเริ่มเกม - มีค่าเท่ากับ 400
- double <u>MIN_TIME_INCREMENT</u>	- ขอบล่างของเวลาที่เพิ่มต่อการไปถึง target ที่ ถูกต้อง 1 ครั้ง - มีค่าเท่ากับ 200
+ double <u>MAX_TIME_DECAY</u>	- ขอบบนอัตราการลดของเวลาการเล่น - มีค่าเท่ากับ 300
- int <u>DEFAULT_PLAYER_SPEED</u>	- ความเร็วของ player ในสถานะปกติ - มีค่าเท่ากับ 10
- int <u>SLOWED_PLAYER_SPEED</u>	- ความเร็วของ player เมื่อเทียบกับระเบิด - มีค่าเท่ากับ 4
- int <u>BOOST_SPEED</u>	- ความเร็วที่เพิ่มขึ้น ของ player เมื่อกด boost - มีค่าเท่ากับ 4
+ int <u>PLAYER_NORMAL</u>	- แสดงถึง player ในสถานะปกติ - มีค่าเท่ากับ 0
+ int <u>PLAYER_SLOW</u>	- แสดงถึง player เมื่อเทียบกับระเบิด - มีค่าเท่ากับ 1
+ int <u>MAX_BOOST_GAUGE</u>	- ค่าการ playerBoostGauge สูงสุดของ player - มีค่าเท่ากับ 500
- int <u>BOOST_INCREMENT</u>	- อัตราการเพิ่มของค่า playerBoostGauge - มีค่าเท่ากับ 2
- int <u>BOOST_DECAY</u>	- อัตราการลดของค่า playerBoostGauge - มีค่าเท่ากับ 4

- int <u>SLOWED_TIME</u>	- ขอบบนของเวลาที่ player เคลื่อนที่ช้าลงเมื่อเหยียบกับระเบิด - มีค่าเท่ากับ 200
- int <u>SLOWED_TIME_DECAY</u>	- อัตราการลดของเวลาที่ player เคลื่อนที่ช้าลง - มีค่าเท่ากับ 1
- int <u>LANDMINE_STAGE</u>	- คะแนนที่ต้องการต่อการเพิ่มกับระเบิด 1 ลูก - มีค่าเท่ากับ 12
- int <u>MAX_LANDMINE_COUNT</u>	- จำนวนกับระเบิดที่มีมากที่สุดในเกม - มีค่าเท่ากับ 5
- int <u>STARTING_TIME</u>	- ช่วงเวลา 10 หน่วยแรกขณะเริ่มเกม
- int <u>STATE_GAMEOVER</u>	- ตัวแปรที่แสดงว่า เกมจบแล้ว - มีค่าเท่ากับ 0
- int <u>STATE_INGAME</u>	- ตัวแปรที่แสดงว่า เกมยังดำเนินอยู่ - มีค่าเท่ากับ 1
- int <u>STATE_RESTART</u>	- ตัวแปรที่แสดงว่า เกมถูก restart - มีค่าเท่ากับ 2
- int <u>MAP_COUNT</u>	- จำนวนฉากหลังทั้งหมดในส่วนตัวเกมหลัก
- int <u>MAP_CHANGE</u>	- คะแนนที่ต้องการต่อการเปลี่ยนฉากหลังของตัวเกมหลัก - มีค่าเท่ากับ 30
- int <u>gameState</u>	แสดงถึงสถานะของเกม
- int <u>playerState</u>	แสดงถึงสถานะของ player
- int <u>playerBoostGauge</u>	แสดงถึงค่าการ boost ของ player ณ ขณะใดๆ
- boolean <u>boostTrying</u>	ตัวแปรแสดงว่า ณ ขณะนั้นมีความพยายามที่จะกด boost หรือไม่
- boolean <u>boostBan</u>	ตัวแปรแสดงว่า ณ ช่วงเวลานั้นการ boost ถูกระงับหรือไม่
- int <u>slowTime</u>	ช่วงเวลาที่ player เคลื่อนที่ช้าลงจากปกติ ณ ขณะใดๆ
- double <u>timeDecay</u>	อัตราการลดของเวลาการเล่น ณ ขณะใดๆ
- int <u>landmineCount</u>	จำนวน landmine ณ ขณะนั้น

- double <u>timeIncrement</u>	เวลาที่เพิ่มต่อการไปถึง target ที่ถูกต้อง 1 ครั้ง ณ ขณะนั้น
- double <u>gameTime</u>	เวลาการเล่นที่เหลือ
- int <u>globalTime</u>	เวลาที่เริ่มนับตั้งแต่เริ่มตัวเกมหลัก
- int <u>previousMap</u>	แสดงถึงฉากหลังก่อนหน้าฉากปัจจุบัน
- int <u>currentMap</u>	ฉากหลังปัจจุบัน

6.2.4.2. Methods

+ void <u>init()</u>	<ul style="list-style-type: none"> - ตั้งค่าตัวแปร gametime ให้เท่ากับ START_TIME - ตั้งค่า สถานะ player ให้เป็นสถานะปกติ (ความเร็วปกติ) - ตั้งค่าคะแนนเป็น 0 - ตั้งค่า playerBoostGauge ให้เท่ากับ MAX_BOOST_GAUGE - ตั้งค่า boostBan เป็น false - ตั้งค่า timeDecay ให้เท่ากับ INITIAL_TIME_DECAY - ตั้งค่า landmineCount, globalTimer, currentMap ให้เท่ากับ 0 - รัน GraphicsTimer.init()
+ void <u>gameTick()</u>	<ul style="list-style-type: none"> - เพิ่มค่า globalTimer 1 ต่อการเรียก 1 ครั้ง - ลดเวลาการเล่น - ตรวจสอบว่าการใช้ boost ถูกกระบังหรือไม่ - ในกรณีที่ มีความพยายามที่จะกด boost และ boost สามารถใช้ได้ จะลดค่าการ boost ของ player - ในกรณีที่ ค่า boost น้อยกว่า MAX_BOOST_GUAGE และ ไม่ได้มีความพยายามที่จะกด boost ให้เพิ่มค่า boost - รัน GraphicsTimer.increaseTime() - รัน updateMap()
+ void <u>decreaseTime()</u>	ลด gameTime ลงด้วยค่า timeDecay ต่อการเรียก method 1 ครั้ง โดยถ้าในขณะนั้น player ถูกทำให้ช้าลง จะรัน decreaseSlowTime() ด้วย
+ void <u>increaseTime()</u>	เพิ่ม gameTime ด้วยค่า timeIncrement
+ void <u>isSlowed()</u>	ตรวจสอบว่า player ถูกทำให้ช้าลงหรือไม่

+ void <u>triggerSlow()</u>	ทำให้ player ช้าลงโดย slowTime มีค่าเท่ากับ SLOWED_TIME
+ void <u>decreaseSlowTime()</u>	ลดค่า slowTime ลงด้วยค่า timeDecay ต่อการเรียก method 1 ครั้ง โดยค่า slowTime จะไม่สามารถน้อยกว่า 0
- void <u>increasePlayerBoostGauge()</u>	เพิ่มค่า playerBoostGauge ของ player ขึ้นเท่ากับ BOOST_INCREMENT โดยค่า boost จะไม่สามารถมีค่ามากกว่า MAX_BOOST_GAUGE
- void <u>decreasePlayerBoostGauge()</u>	ลดค่า playerBoostGauge ของ player ลงเท่ากับ BOOST_DECAY
+ boolean <u>canBoost()</u>	ตรวจสอบว่ากด boost ได้หรือไม่
+ boolean <u>boostFull()</u>	ตรวจสอบว่าค่า playerBoostGauge เต็มหรือไม่
- boolean <u>checkBoostBan()</u>	ตรวจสอบว่าการใช้ boost ถูกกระจับหรือไม่
+ boolean <u>isBoostBan()</u>	return ตัวแปร boostBan
+ int <u>getPlayerBoostGauge()</u>	return ตัวแปร playerBoostGauge
+ void <u>increaseScore()</u>	<ul style="list-style-type: none"> - เพิ่มคะแนน 1 คะแนน - เพิ่มอัตราการลดของเวลาการเล่น - ลดเวลาที่จะได้เพิ่มต่อการไปถึงเป้าหมายที่ถูกต้อง - update จำนวนกับระเบิด
+ int <u>getScore()</u>	return ตัวแปร score
+ int <u>getPlayerSpeed()</u>	return ตัวแปร playerSpeed (เป็น local variable) โดยค่า playerSpeed ขึ้นอยู่กับสถานะของ playerState - ถ้า playerState มีค่าเท่ากับ PLAYER_NORMAL playerSpeed จะมีค่าเท่ากับDEFAULT_PLAYER_SPEED - ถ้า playerState มีค่าเท่ากับ PLAYER_SLOW playerSpeed จะมีค่าเท่ากับ SLOWED_PLAYER_SPEED - ในกรณีอื่นๆ playerSpeed จะมีค่าเท่ากับ DEFAULT_PLAYER_SPEED - ในกรณีที่กำลัง boost playerSpeed จะถูกเพิ่มเท่ากับ BOOST_SPEED
+ int boolean <u>boost()</u>	ตรวจสอบว่า boost อยู่หรือไม่

+ void <u>setPlayerState</u> (int playerState)	ตั้งค่าให้ค่า playerState มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ double <u>getTime</u> ()	return ตัวแปร gameTime
+ boolean <u>isBoostTrying</u> ()	return ตัวแปร boostTrying
+ void <u>setBoostTrying</u> (boolean boostTrying)	ตั้งค่าให้ค่า boostTrying มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ boolean <u>playerCollideCheck</u> (Player a, Sprite b)	ตรวจสอบว่ามีการซ้อนทับกันของ Player a และ Sprite b หรือไม่
+ boolean <u>hasLandmine</u> ()	ตรวจสอบสถานะเกมปัจจุบันว่ามี landmine หรือไม่
+ void <u>updateTimeDecay</u> ()	คูณค่า timeDecay ด้วย 1.01 โดย ค่า timeDecay จะไม่มากกว่า MAX_TIME_DECAY
+ int <u>getLandmineCount</u> ()	return ตัวแปร landmineCount
- void <u>updateLandmineCount</u> ()	จะอัปเดตค่า landmine โดยเพิ่มค่า landmineCount ขึ้น 1 โดยค่า landmineCount จะไม่มากกว่า MAX_LANDMINE_COUNT ทุกครั้งที่ค่า score ทหารด้วย LANDMINE_STAGE ลงตัว
+ void <u>updateTimeIncrement</u> ()	หารค่า timeDecay ด้วย 1.002 โดย ค่า timeIncrement จะไม่น้อยกว่า MIN_TIME_INCREMENT
+ int <u>getGlobalTimer</u> ()	return ตัวแปร globalTimer
+ boolean <u>isStarting</u> ()	ตรวจสอบว่าเกมอยู่ในช่วง 10 หน่วยแรกหรือไม่ (globalTime < STARTING_TIME)
+ int <u>getGameState</u> ()	return ตัวแปร gameState
+ void <u>setGameState</u> (int gameState)	ตั้งค่าให้ค่า gameState มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ int <u>getCurrentMap</u> ()	return ตัวแปร currentMap
- void <u>updateMap</u> ()	Update map ให้เป็นปัจจุบัน
+ boolean <u>isMapChange</u> ()	ตรวจสอบว่ามีการเปลี่ยน map ณ ขณะนั้นหรือไม่
- void <u>updatePreviousMap</u> ()	ตั้งค่าให้ค่า previousMap ให้เท่ากับ currentMap

6.2.5. Class SceneUtil implements Commons

6.2.5.1. Fields

+ int <u>MENU</u>	- รหัสแสดงถึง scene ของ Menu - มีค่าเท่ากับ 0
+ int <u>GAME</u>	- รหัสแสดงถึง scene ของตัวเกมหลัก - มีค่าเท่ากับ 1
+ int <u>GAMEOVER</u>	- รหัสแสดงถึง scene ของ GameOver - มีค่าเท่ากับ 2
+ int <u>HELP</u>	- รหัสแสดงถึง scene ของ Help - มีค่าเท่ากับ 3
+ int <u>NONE</u>	- รหัสแสดงถึง scene เปล่า - มีค่าเท่ากับ -1
- int <u>helpStage</u>	แสดงถึงค่า stage ณ ขณะนั้นของ help
- int <u>previousSceneType</u>	แสดงถึงค่าของ scene ก่อนหน้า scene ปัจจุบัน
- Stage <u>stage</u>	แสดงถึง stage ณ ขณะปัจจุบัน
- Scene <u>gameScene</u>	Scene ของตัวเกมหลัก
- Scene <u>menuScene</u>	Scene ของ Menu
- Scene <u>helpScene</u>	Scene ของ Help
- Scene <u>gameOverScene</u>	Scene ของ GameOver
- AnimationTimer <u>gameAnim</u>	ตัวแปรที่ใช้ update สถานะของเกม (โดยจะถูกสร้างใน main ในที่นี้เพียงแค่ดึงมาใช้)
- AnimationTimer <u>menuAnim</u>	ตัวแปรที่ใช้ควบคุม animation บนหน้า menu โดยจะเป็นตัวเรียก method GraphicUtil.drawMenu() เพื่ออัปเดตฉากหลัง
- AnimationTimer <u>helpAnim</u>	ตัวแปรที่ใช้ควบคุมการเปลี่ยน stage ย่อยของ help
- MediaPlayer <u>gameOverPlayer</u>	MediaPlayer ที่เล่น VDO บนหน้า GameOver
- MediaPlayer <u>introPlayer</u>	MediaPlayer ที่เล่น VDO Intro หน้า Menu
- MediaPlayer <u>teamIntroPlayer</u>	MediaPlayer ที่เล่น VDO ที่แสดง - Logo - ชื่อ Team “Pepelaugh” - คำเตือน
- MediaPlayer <u>countdownPlayer</u>	MediaPlayer ที่เล่น VDO countdown เริ่มเกม

6.2.5.2. Methods

+ void <u>init</u> (Stage stage, Scene gameScene, AnimationTimer gameAnim)	- สร้าง Scene ทั้งหมดที่ใช้งานภายในเกม - ตั้งค่า previousSceneType ให้มีค่าเท่ากับ NONE
- void <u>setGameCountDown</u> ()	โหลด VDO ใส่ countdownPlayer เก็บไว้สำหรับใช้งาน
- void Scene <u>setMenuScene</u> ()	สร้าง scene ของ Menu เก็บไว้เตรียมใช้งาน
- void Scene <u>setHelpScene</u> ()	สร้าง scene ของ Help เก็บไว้เตรียมใช้งาน
- void Scene <u>setGameOverScene</u> ()	สร้าง scene ของ GameOver เก็บไว้เตรียมใช้งาน
- void Node[] <u>getGameOverButton</u> (int x, int y, int width , int height, String message, EventHandler<MouseEvent> eventHandler)	Return Array ของ Node ที่แทนปุ่มในหน้า GameOver
- void Node[] <u>getMenuButton</u> (int x, int y, int width , int height, String message, EventHandler<MouseEvent> eventHandler)	Return Array ของ Node ที่แทนปุ่มในหน้า Menu
- Rectangle <u>getButton</u> (int x, int y, int width, int height, EventHandler<MouseEvent> eventHandler, ChangeListener<Boolean> changeListener)	ตั้งค่าคุณสมบัติพื้นฐานให้ปุ่มที่จะใช้ในหน้า GameOver และ Menu
- void <u>setScene</u> (int type) throws Exception	- แสดง scene ที่กำหนดโดยอ้างอิงจากค่า parameter type - ในกรณีที่มีค่า type นอกเหนือจาก {-1,0,1,2,3} แสดงว่า type เป็นรหัส scene ที่ไม่มีอยู่จริงและจะทำการ throw SceneUtilException()
+ void <u>switchTo</u> (int type)	- เปลี่ยน scene โดยอ้างอิง scene ที่ต้องการจะเปลี่ยนจาก parameter type - ถ้าเปลี่ยนไม่ได้ จะมีการพยายามกลับไปเล่นหน้า Menu และ จะรายงานผู้ใช้งานผ่าน terminal ว่าปัญหาเกิดจากส่วนไหน

6.2.6. Class Randomizer implements Commons

6.2.6.1. Fields

- int <u>LANDMINE_RADIUS</u>	- ค่ารัศมีของ landmine จะนำไปใช้บวกกับ ค่ารัศมีของ sprite อื่นๆ เพื่อกำหนด ระยะห่างอย่างต่ำระหว่าง landmine และ sprite นั้นๆ - มีค่าเท่ากับ 60
- int <u>TARGET_RADIUS</u>	- ค่ารัศมีของ target จะนำไปใช้บวกกับ ค่ารัศมีของ sprite อื่นๆ เพื่อกำหนดระยะห่าง อย่างต่ำระหว่าง target และ sprite นั้นๆ - มีค่าเท่ากับ 90
- int <u>PLAYER_RADIUS</u>	- ค่ารัศมีของ player จะนำไปใช้บวกกับ ค่ารัศมีของ sprite อื่นๆ เพื่อกำหนดระยะห่าง อย่างต่ำระหว่าง target และ player นั้นๆ - มีค่าเท่ากับ 140
- int <u>LANDMINE_ITI</u>	- ค่า index เริ่มต้นของ landmine ใน array บรรจุพิกัดของ sprites ต่างๆ - มีค่าเท่ากับ 4
- Random <u>random</u>	ตัวแปร random ใช้ในการสุ่มค่าต่างๆ เพื่อ ในมาใช้ภายในคลาส

6.2.6.2. Methods

+ int[][] coordinatesRandomizer (int playerX, int playerY, int landmineCount)	สุ่มตำแหน่งของ sprite ต่างๆ ยกเว้น player โดยตำแหน่ง sprite ต่างๆจะไม่ซ้อนทับกัน
+ int <u>getPlayerType()</u>	สุ่ม type ของ player ใหม่
- double distance(int x1, int x2, int y1, int y2)	คำนวณระยะห่างระหว่าง (x1,y1) และ (x2,y2)

6.2.7. Class GraphicTimer

6.2.7.1. Fields

- int <u>MAX_TIME</u>	- ค่าสูงสุดของหน่วยเวลา
- int <u>BOOSTFLASH_INTERVAL</u>	- จำนวน frame ต่อ 1 รอบการกระพริบ - มีค่าเท่ากับ 20
- int <u>BOOSTFLASH_DURATION</u>	- จำนวน frame ที่สว่างในกระบวนการกระพริบ - มีค่าเท่ากับ 11
- int <u>SCREENFLASH_DURATION</u>	- ระยะเวลาที่หน้าจอแสดง effect แพลช
- int <u>SCREENFLASH_MAX_OPACITY</u>	- ค่าความทึบแสงสูงสุดของ effect แพลช
- int timer	- เวลา ณ ขณะใดๆ
- int boostFlashTimer	- เวลา ณ ขณะใดๆ ของการกระพริบ
- int screenFlashTimer	- เวลา ณ ขณะใดๆ ของการแสดง effect แพลช

6.2.7.2. Methods

+ void <u>init()</u>	เริ่มต้นการนับเวลา
+ void <u>increaseTime()</u>	เพิ่มเวลา
+ void <u>switchSprite(Sprite sprite)</u>	แสดง animation ของ sprite ต่างๆ คือ player -> แสดงท่าเดิน target -> ขยับซ้าย - ขวา landmine -> กระพริบ ซึ่งใน sprite ที่ต่างกันเวลาในการแสดงแต่ละเฟรมก็จะต่างกันด้วย
+ boolean <u>boostFlash()</u>	- ลดเวลาการกระพริบ - return ว่า boostFlashTime หมดหรือไม่
+ boolean <u>screenFlash()</u>	- ลดเวลาการแสดง effect แพลช - return ว่า screenFlashTimer หมดหรือไม่
+ double <u>getScreenFlashOpacity()</u>	- return ค่าความทึบแสงของ effect แพลช ณ ขณะใดๆ โดยคำนวณจาก $\frac{\text{screenFlashTimer}}{\text{screenFlashDuration}} \times \text{SCREENFLASH_MAX_OPACITY}$

6.3. package base

6.3.1. Interface Common

6.3.1.1. Fields

+ int <u>WINDOW_WIDTH</u>	- ความกว้างของหน้าจอเกม - มีค่าเป็น 1280
+ int <u>WINDOW_HEIGHT</u>	- ความสูงของหน้าจอเกม - มีค่าเป็น 720
+ int <u>BORDER_RIGHT</u>	- ระยะห่างจากขอบขวาของหน้าจอ ซึ่งเว้นไว้เพื่อกำหนดขอบเขตของ play area - มีค่าเท่ากับ 30
+ int <u>BORDER_LEFT</u>	- ระยะห่างจากขอบซ้ายของหน้าจอ ซึ่งเว้นไว้เพื่อกำหนดขอบเขตของ play area - มีค่าเท่ากับ 30
+ int <u>BORDER_TOP</u>	- ระยะห่างที่เว้นจากขอบบนของหน้าจอเกม ซึ่งเว้นไว้เพื่อกำหนดขอบเขตของ play area - มีค่าเท่ากับ 100
+ int <u>BORDER_BOTTOM</u>	- ระยะห่างที่เว้นจากขอบล่างของหน้าจอเกม ซึ่งเว้นไว้เพื่อกำหนดขอบเขตของ play area - มีค่าเท่ากับ 40
+ int <u>MENU_BG_WIDTH</u>	- ความกว้างของ Background หน้า Menu - มีค่าเท่ากับ 1740
+ int <u>MENU_BG_HEIGHT</u>	- ความสูงของ Background หน้า Menu - มีค่าเท่ากับ 1522
+ Color <u>MENU_COLOR</u>	- ค่าสี RGB ของ effect ที่จะแสดงเมื่อเลื่อนเมาส์ไปบน ตัวเลือกต่างๆ บนหน้า Menu - มีค่าสีเป็น #9F4985
+ int <u>MAX_TYPE</u>	- จำนวนชนิดทั้งหมดของ Player และ Target
+ int <u>PLAYER_WIDTH</u>	- ความกว้างของ player - มีค่าเท่ากับ 100
+ int <u>PLAYER_HEIGHT</u>	- ความสูงของ player - มีค่าเท่ากับ 150

+ int <u>TARGET_WIDTH</u>	- ความกว้างของ target - มีค่าเท่ากับ 100
+ int <u>TARGET_HEIGHT</u>	- ความสูงของ target - มีค่าเท่ากับ 150
+ int <u>LANDMINE_WIDTH</u>	- ความกว้างของ landmine - มีค่าเท่ากับ 80
+ int <u>LANDMINE_HEIGHT</u>	- ความสูงของ landmine - มีค่าเท่ากับ 80

6.3.2. Interface Coordinate

6.3.2.1. Methods

+ <i>void setX(int x)</i>	ตั้งค่าตำแหน่งแกน x (จะถูก Override)
+ <i>void setY(int y)</i>	ตั้งค่าตำแหน่งแกน y (จะถูก Override)
+ <i>int getX()</i>	return ค่าตำแหน่งแกน x (จะถูก Override)
+ <i>int getY()</i>	return ค่าตำแหน่งแกน y (จะถูก Override)

6.3.3. Interface Visible

6.3.3.1. Methods

+ <i>Boolean isVisible()</i>	return ค่า visible (จะถูก Override)
+ <i>void setVisible(boolean visible);</i>	ตั้งค่า visible ให้มีค่าเท่ากับ parameter ที่ป้อนเข้ามา (จะถูก Override)

6.4. package exception

6.4.1. Class AudioUtilException extends Exception

6.4.1.1. Fields

- long <u>SERIALVERSIONUID</u>	- ค่า serialVersionUID ประจำ exception - มีค่า 1
--------------------------------	---

6.4.1.2. Constructor

+ AudioUtilException(String message)	ตั้งค่าให้ message ของ exception มีค่าเท่ากับ Parameter เข้ามา
--------------------------------------	---

6.4.2. Class SceneUtilException extends Exception

6.4.2.1. Fields

- long <u>SERIALVERSIONUID</u>	- ค่า serialVersionUID ประจำ exception - มีค่า 1
--------------------------------	---

6.4.2.2. Constructor

+ SceneUtilException(String message)	ตั้งค่าให้ message ของ exception มีค่าเท่ากับ Parameter เข้ามา
--------------------------------------	---

6.5. package sprite.base

6.5.1. Interface Type

+ <i>int getType()</i>	return ค่า type (จะถูก override)
+ <i>int setType()</i>	ตั้งค่า type ให้เท่ากับ parameter ที่ป้อนเข้ามา (จะถูก override)

6.5.2. Interface Movable

+ <i>void KeyPressed(KeyEvent)</i>	เป็น method ที่ใช้ตอบสนองต่อการกดปุ่มที่กำหนด (จะถูก override)
+ <i>void KeyRelease(KeyEvent)</i>	เป็น method ที่ใช้ตอบสนองต่อการปล่อยปุ่มที่กำหนด (จะถูก override)
+ <i>void act()</i>	เป็น method ที่ใช้สั่งให้ class ที่ขยายไปแสดงการกระทำใดๆ บนหน้าจอ (จะถูก override)

6.5.3 Abstract Class Sprite implement Coordinate, Visible

6.5.3.1. Fields

- boolean visible	ค่า visible
- WritableImage image	ภาพ ณ ปัจจุบันของ sprite
# int x	พิกัดตามแกน x ของ sprite
# int y	พิกัดตามแกน y ของ sprite
# int dx	ความเร็วตามแกน x ของ sprite
# int dy	ความเร็วตามแกน y ของ sprite
# int spriteIndex	แสดงถึงรหัสของภาพแต่ละภาพของ sprite
# HashMap<Integer, WritableImage[]> imageMaps	Hashmap ที่ใช้เชื่อมโยงรหัส spriteIndex และ รูปแต่ละรูปของ sprite เข้าด้วยกัน

6.5.3.2. Constructor

+ Sprite()	<ul style="list-style-type: none"> - ตั้งค่า spriteIndex เป็น 0 - ตั้งค่า visible เป็น true - รัน initImageMaps()
------------	--

6.5.3.3. Methods

+ boolean isVisible()	return ค่า visible
+ setVisible(boolean visible)	ตั้งค่า visible ให้เท่ากับ parameter ที่ป้อนเข้ามา
+ void setImage(WritableImage image)	ตั้งค่าให้ image มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ Writable getImage()	return ตัวแปร image
+ void setX(int x)	ตั้งค่าให้ x มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ void setY(int y)	ตั้งค่าให้ y มีค่าเท่ากับ parameter ที่ป้อนเข้ามา
+ int getX()	return ค่า x
+ int getY()	return ค่า y
+ int <i>getTrueX()</i>	return พิกัด x ของจุดกึ่งกลาง sprite
+ int <i>getTrueY()</i>	return พิกัด y ของจุดกึ่งกลาง sprite
+ void <i>updateSpriteIndex()</i>	update ค่า spriteIndex
+ void <i>initImageMaps()</i>	สร้างความเชื่อมโยงระหว่าง รหัส spriteIndex และ frame แต่ละ frame ของ sprite
+ void <i>updateImage()</i>	update รูปที่แสดงบนหน้าจอเกมของ sprite
+ int <i>getSpriteInterval()</i>	return จำนวน frame ที่แสดง sprite ต่อจำนวน frame ที่แสดงทั้งหมดต่อ

6.6. package sprite

6.6.1. Class Player extends Sprite implements Commons, Type, Movable

6.6.1.1. Fields

+ int <u>TYPE_ALIEN</u>	- แสดงถึง target Alien - มีค่าเท่ากับ 0
+ int <u>TYPE_ANIME_GIRL</u>	- แสดงถึง target Anime girl - มีค่าเท่ากับ 1
+ int <u>TYPE_INF_GAUNTLET</u>	- แสดงถึง target Infinity Gauntlet - มีค่าเท่ากับ 2
- type	ประเภทของ player เมื่อ: type = 0 จะแสดงถึง Frank the ninjutsu runner type = 1 จะแสดงถึง Weebboi และเมื่อ type = 2 จะแสดงถึง Thanos
- int playerSpeed	ความเร็วในการเคลื่อนที่ของ player
- Boolean[] keyPressed;	Array ที่เก็บว่ามีปุ่มใดถูกกดอยู่ ณ ขณะนั้น

6.6.1.2. Constructors

+ Player()	- รัน constructor ของ class Sprite เพื่อสร้างตัวแปรพื้นฐาน - ตั้งค่า ให้ keyPressed มีค่าเป็น {false, false, false, false} - รัน initPlayer() - ตั้งค่า type เท่ากับ 0
+ Player(int type)	- รัน constructor ของ class sprite เพื่อสร้างตัวแปรพื้นฐาน - ตั้งค่า ให้ keyPressed มีค่าเป็น {false, false, false, false} - รัน initPlayer() - ตั้งค่า type เท่ากับ parameter ที่ป้อนเข้ามา

6.6.1.3. Methods

- initPlayer()	- ตั้งค่าให้พิกัดของ player อยู่กลางหน้าจอเกม - ตั้งค่าความเร็วของ player ให้เป็นค่าความเร็วปกติ
# void initImageMaps()	- สร้างความเชื่อมโยงระหว่าง spriteIndex กับ frame ต่างๆ ของ player
+ updateSpriteIndex()	- update spriteIndex เพื่อให้ตัว player มี animation การขยับ โดยค่า spriteIndex จะมีค่าอยู่ในช่วง [0,เฟรมทั้งหมดของ player) - รัน updateImage()
+ updateImage()	เปลี่ยน frame ที่แสดงอยู่
- Boolean[] initKeyArray()	Return array boolean ที่มีลักษณะเป็น {false, false, false, false} ซึ่งค่า boolean แต่ละตัวในแสดงถึงสถานะการกดปุ่มเคลื่อนที่ 4 ปุ่ม ({ซ้าย, ขวา, ขึ้น, ลง} ซึ่งในตอนแรกจะยังไม่มีการกดปุ่มใดๆ จึงมีค่าเป็น false ทั้งหมด)
+ int getType()	return ค่า type ของ player
+ int setType(int type)	- ตั้งค่า type ให้เท่ากับ parameter ที่ป้อนเข้ามา - ตั้งค่า frame ปัจจุบันให้เป็น frame ที่ 0 - เปลี่ยนเฟรมของ player
+ void act()	- เป็น method ที่สั่งให้ player ขยับไปในทิศทางที่สอดคล้องกับปุ่มที่ถูกกดภายในหน้าจอเกม

+ void keyPressed(KeyEvent e)	<p>เป็น method ที่ใช้ตอบสนองต่อการกดปุ่มที่กำหนดโดย</p> <ul style="list-style-type: none"> - เมื่อกดปุ่ม SHIFT ถูกกด Player จะถูกเร่งความเร็ว - เมื่อกดปุ่ม a หรือ ลูกศรซ้ายถูกกด ค่า keyPressed[0] จะเป็น true - เมื่อกดปุ่ม d หรือ ลูกศรขวาถูกกด ค่า keyPressed[1] จะเป็น true - เมื่อกดปุ่ม w หรือ ลูกศรขึ้นถูกกด ค่า keyPressed[2] จะเป็น true - เมื่อกดปุ่ม s หรือ ลูกศรลงถูกกด ค่า keyPressed[3] จะเป็น true <p>เพื่อนำไปใช้อ้างอิงในการควบคุมการเคลื่อนที่ใน method Act() ต่อไป</p>
+ void keyReleased(KeyEvent e)	<p>เป็น method ที่ใช้ตอบสนองต่อการกดปล่อยที่กำหนดโดย</p> <ul style="list-style-type: none"> - เมื่อกดปุ่ม SHIFT ถูกปล่อย Player จะยุติการเร่งความเร็ว - เมื่อกดปุ่ม a หรือ ลูกศรซ้ายถูกปล่อย ค่า keyPressed[0] จะเป็น false - เมื่อกดปุ่ม d หรือ ลูกศรขวาถูกปล่อย ค่า keyPressed[1] จะเป็น false - เมื่อกดปุ่ม w หรือ ลูกศรขึ้นถูกปล่อย ค่า keyPressed[2] จะเป็น false - เมื่อกดปุ่ม s หรือ ลูกศรลงถูกปล่อย ค่า keyPressed[3] จะเป็น false <p>เพื่อนำไปใช้อ้างอิงในการควบคุมการเคลื่อนที่ใน method Act() ต่อไป</p>
+ int getPlayerSpeed()	return ค่า playerSpeed
+ int getTrueX()	return พิกัดจุดกึ่งกลางในแกน X ของ player
+ int getTrueY()	return พิกัดจุดกึ่งกลางในแกน Y ของ player
+ int getSpriteInterval()	return เวลาในการแสดงแต่ละ frame ของ target ซึ่งมีค่าเท่ากับ 10

6.6.2. Class Target extends Sprite implements Commons

6.6.2.1. Fields

+ int <u>TYPE_ALIEN</u>	- แสดงถึง target Alien - มีค่าเท่ากับ 0
+ int <u>TYPE_ANIME_GIRL</u>	- แสดงถึง target Anime girl - มีค่าเท่ากับ 1
+ int <u>TYPE_INF_GAUNTLET</u>	- แสดงถึง target Infinity Gauntlet - มีค่าเท่ากับ 2
+ int type()	ประเภทของ target

6.6.2.2. Constructor

+ Target(int x, int y, int type)	- รัน constructor ของ class Sprite เพื่อสร้างตัวแปรพื้นฐาน - รัน initTarget(x, y, type) เพื่อตั้งค่าพิกัดและประเภทของ target ให้ตรงกับ parameter ที่ป้อนเข้ามา
----------------------------------	---

6.6.2.3. Methods

+ void initTarget(int x, int y, int type)	- ตั้งค่าพิกัดและประเภทของ target ให้ตรงกับ parameter ที่ป้อนเข้ามา - เปลี่ยน frame เป็น frame เริ่มต้นของ target
+ int getType()	return ค่า type
+ void setType(int type)	- ตั้งค่า type ให้เท่ากับ parameter ที่ป้อนเข้ามา - ตั้งค่า spriteIndex เป็น 0 - เปลี่ยนเฟรมของ target
# void initImageMaps()	- สร้างความเชื่อมโยงระหว่าง spriteIndex กับ frame ต่างๆ ของ target

+ updateSpriteIndex()	- update spriteIndex เพื่อให้ตัว target มี animation การขยับ โดยค่า spriteIndex จะมีค่าอยู่ในช่วง [0, เฟรมทั้งหมดของ target) - รัน updateImage()
+ updateImage()	เปลี่ยน frame ที่แสดงอยู่
+ int getTrueX()	return พิกัดจุดกึ่งกลางในแกน X ของ target
+ int getTrueY()	return พิกัดจุดกึ่งกลางในแกน Y ของ target
+ int getSpriteInterval()	return เวลาในการแสดงแต่ละ frame ของ target ซึ่งมีค่าเท่ากับ 30

6.6.3. Class Landmine extends Sprite implements Common

6.6.3.1. Constructor

+ Landmine(int x, int y)	<ul style="list-style-type: none">- รัน constructor ของ class Sprite เพื่อสร้างตัวแปรพื้นฐาน- รัน initTarget(x, y) เพื่อตั้งค่าพิกัด ให้ตรงกับ parameter ที่ป้อนเข้ามา
--------------------------	---

6.6.3.2. Methods

+ void initTarget(int x, int y)	<ul style="list-style-type: none">- ตั้งค่าพิกัดของ landmine ให้ตรงกับ parameter ที่ป้อนเข้ามา- เปลี่ยน frame เป็น frame เริ่มต้นของ landmine
# void initImageMaps()	- สร้างความเชื่อมโยงระหว่าง spriteIndex กับ frame ต่างๆ ของ landmine
+ updateSpriteIndex()	<ul style="list-style-type: none">- update spriteIndex เพื่อให้ตัว landmine มี animation การขยับ โดยค่า spriteIndex จะมีค่าอยู่ในช่วง [0,เฟรมทั้งหมดของ landmine)- รัน updateImage()
+ updateImage()	เปลี่ยน frame ที่แสดงอยู่
+ int getTrueX()	return พิกัดจุดกึ่งกลางในแกน X ของ landmine
+ int getTrueY()	return พิกัดจุดกึ่งกลางในแกน Y ของ landmine
+ int getSpriteInterval()	return เวลาในการแสดงแต่ละ frame ของ landmine ซึ่งมีค่าเท่ากับ 5