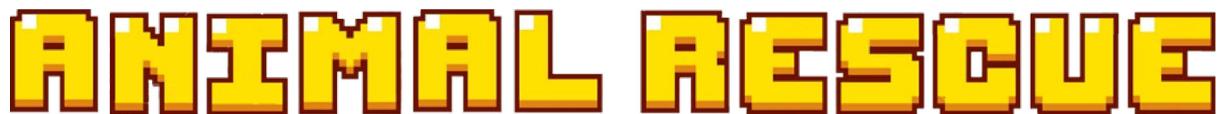


# Project Document : Animal Rescue

Pollawat Hongwimol 6030400021  
Peeranuth Kehasukcharoen 6030416021

## 1. Introduction

ในปี คศ. 3019 บ้านเมืองเจริญขึ้น พื้นที่ป่าลดน้อยลง ประกอบด้วยมลพิษของเสียที่มนุษย์ทิ้ง อุกมา ทำให้สัตว์ป่าล้มตายลงไปจนเกลื่อน รัฐบาลในสมัยนั้นได้วางแผนการขยายนายสัตว์ป่าที่ใกล้สูญพันธุ์จากในป่าเข้าไปในสวนสัตว์ที่เตรียมไว้โดยใช้เครื่องบินขนาดใหญ่ ในขณะกำลังขยับได้มีกลุ่มโจรมาปล้น ซึ่งโจรมีกลุ่มนั้นต้องการนำสัตว์ป่าเหล่านั้นไปขาย เพราะชิ้นส่วนของมันแต่ละชิ้นเป็นของหายาก และมีราคาแพง โจร์ได้ทำการปล่อยดอนเทนเนอร์สัตว์ทั้งหมดลงจากเครื่องบินขนส่ง และจึงได้ร่วมลงมาตาม ผู้เล่นจะได้รับบทเป็นพลทหารที่มีทักษะในการยิงปืนที่แม่นยำเป็นอย่างมาก มีภารกิจรับผิดชอบ การขยายนายสัตว์ป่าครั้งนี้ และคุณผู้อ่านจะได้เห็นการกระโดดและเคลื่อนไหวของตัวละคร ผู้เล่นมีหน้าที่กำจัดโจร์ และช่วยเหลือสัตว์ป่า ซึ่งเป็นภารกิจที่จะต้องทำให้สำเร็จ เพื่อรักษาสัตว์ป่าเหล่านั้นไว้ไม่ให้สูญพันธุ์ไป



ภาพที่ 1: Logo ของเกม Animal Rescue

เนื่องจากเกมนี้จำลองบทบาทการช่วยเหลือสัตว์จากการถูกโจรมีภารกิจล่ามานาน ดังนั้น เกมนี้จึงถูกตั้งชื่อว่า “Animal Rescue” และถูกออกแบบมาให้เป็นเกมที่มีรูปแบบ 8 บิต (8 bit) เพื่อให้คล้ายคลึงกับเกมส่วนใหญ่ที่ถูกสร้างมาในยุคแรก ๆ สัญลักษณ์ของเกมนี้จึงถูกสร้างออกแบบในรูปแบบ 8 บิตด้วยเช่นกัน ตั้งภาพที่ 1 นอกเหนือจากนี้ ตัวอักษร “A” ตัวแรกของชื่อยังถูกนำมาตั้งเป็น Icon ของเกม อีกด้วย

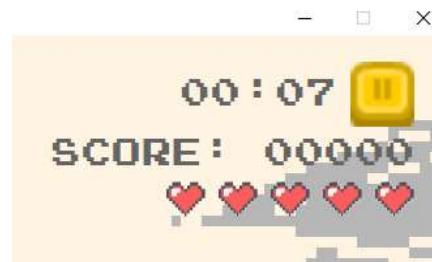
## 2. How to play

ให้ผู้เล่นกดปุ่มบนคีย์บอร์ดโดยไม่สนใจตัวอักษรใหญ่-เล็ก (case-insensitive) เพื่อกำจัดวัตถุ (unit) ที่ตกลงมาจากนฟ้า (ภาพที่ 2) ประกอบไปด้วย นายพรา (hunter) และ สัตว์ (animal) โดยวัตถุที่ตกลงมาจะมีตัวอักษรกำกับอยู่บนร่มพาราชาต (parachute) ซึ่งเมื่อผู้เล่นกำจัดนายพรา ผู้เล่นจะได้รับคะแนน (score) แต่ถ้าผู้เล่นกำจัดสัตว์ ผู้เล่นจะถูกหักคะแนน และผู้เล่นไม่สามารถกำจัดสัตว์มากเกินกว่าที่กำหนดได้ โดยมีรายละเอียดดังนี้

- เมื่อกำจัดนายพราสำเร็จ ได้รับคะแนน 8 คะแนน
- เมื่อกำจัดสัตว์ จะถูกหักคะแนน 5 คะแนน
- เมื่อกำจัดสัตว์ครบ 5 ครั้ง จะถือว่าภารกิจล้มเหลว และจบเกม
- เมื่อนายพราลงถึงพื้น จะถือว่าภารกิจล้มเหลว และจบเกม



ภาพที่ 2: units ประกอบไปด้วยนายพรา(ซ้าย) และสัตว์(ขวา)



ภาพที่ 3: มุมด้านขวาบนสุดของหน้าจอในขณะเล่นเกม

อย่างไรก็ตาม นาฬิกาจับเวลา (timer) จะแสดงอยู่ด้านบนขวาของหน้าจอ พร้อมกับคะแนนและ จำนวนชีวิต (life points) จำนวน 5 ชีวิตในรูปของหัวใจจำนวน 5 ดวงตามลำดับ (ดูภาพที่ 3 ประกอบ) ซึ่งเมื่อผู้เล่นกำจัดสัตว์ จำนวนของหัวใจจะลดลงตามจำนวนครั้ง ผู้เล่นจะยังสามารถเล่นเกมได้ต่อถึงแม้ว่าจะไม่มีจำนวนชีวิตเหลืออยู่ก็ตาม แต่ถ้าผู้เล่นกดปุ่มพลาดหรือจงใจเพื่อทำการกำจัดสัตว์ในครั้งถัดไป เกมจะจบลงทันที

### 3. Application flow

เมื่อเข้าสู่ตัวเกม หน้าจอจะแสดงหน้าจอเกมขึ้นมาในลักษณะดังภาพที่ 4 ซึ่งประกอบไปด้วย สัญลักษณ์ (logo) ของเกม ปุ่มเริ่มเล่นเกม (play) ปุ่มเครดิต (credit) ปุ่มตั้งค่า (option) ปุ่มออกจาก เกม (exit) และภาพพื้นหลัง (background) ซึ่งทั้งหมดถูกออกแบบมาให้อยู่ในรูปแบบ 8 bit



ภาพที่ 4: หน้าจอหลักเมื่อเริ่มต้นเข้าสู่ตัวเกม

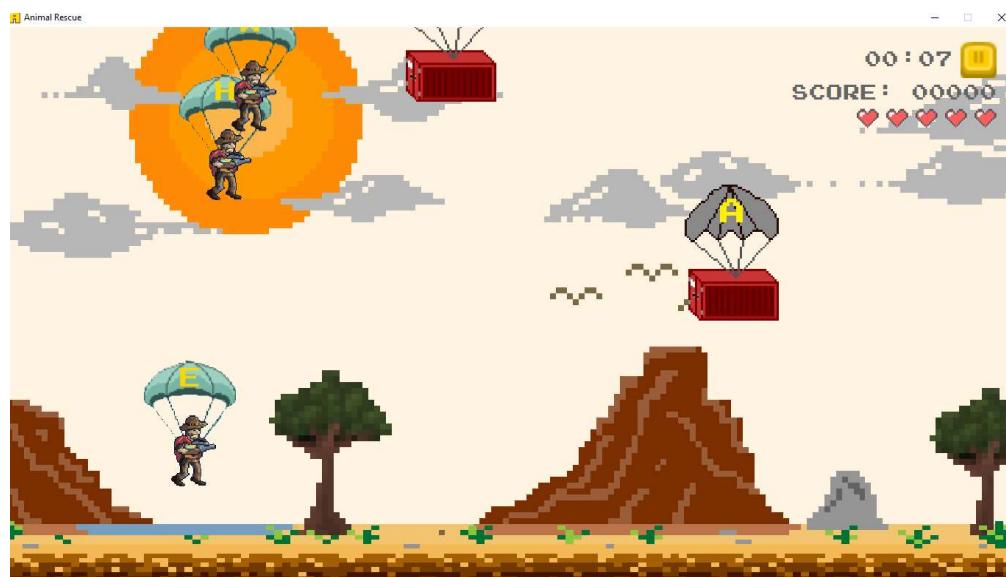
หลังจากเข้าสู่ตัวเกม การเล่นเกมนั้นจะทำได้โดยการกดปุ่มเริ่มเล่นเกม จากนั้นจะมีหน้าต่างย่อย (subscene) เลื่อนเข้ามายังทางด้านซ้ายดังภาพที่ 5 ซึ่งเป็นการแนะนำวิธีการเล่นเกม จากนั้นกดปุ่ม OK เพื่อทำการเริ่มเล่นเกม



ภาพที่ 5: หน้าต่างย่อยหลังจากกดปุ่มเริ่มเล่นเกม (play)

ในหน้าต่างย่อของปุ่มเริ่มเล่นเกมนั้นจะประกอบไปด้วยคำแนะนำในการเล่นเกมว่าควรกดปุ่มเพื่อกำจัดนายพาน และไม่ควรกดปุ่มเพื่อกำจัดสัตว์ อีกทั้งยังอธิบายคร่าว ๆ เกี่ยวกับจำนวนชีวิต ซึ่งเป็นฟังก์ชันหนึ่งในเกมอีกด้วย

หลังจากกดปุ่ม OK ในหน้าต่างย่อของปุ่มเริ่มเล่นเกม เกมจะสับหน้าจอมาอย่างหน้าจอเล่นเกมดังที่แสดงให้เห็นดังภาพที่ 6 ซึ่งผู้เล่นจะสามารถเล่นเกมและเพลิดเพลินไปกับเกมอันสนุกสนานได้ที่หน้าจอนี้ ยิ่งไปกว่านั้น ผู้เล่นสามารถหยุดชั่วคราว (pause) และเล่นเกมต่อจากเดิม (unpause) ได้ที่ปุ่มทางด้านขวาบน ถัดจากนาฬิกาจับเวลา



ภาพที่ 6: หน้าจอระหว่างการเล่นเกม (game stage)

ในส่วนของการพักชั่วคราว เมื่อผู้เล่นกดปุ่มพักชั่วคราว เกมจะแสดงหน้าต่างย่อๆ ซึ่งภายในหน้าต่างย่อจะมีปุ่มเล่นใหม่อีกรอบ (replay) ปุ่มเมนู (menu) ซึ่งหมายถึงกลับไปยังหน้าจอดังที่แสดงในภาพที่ 4 ปุ่มออกจากเกม (exit) และแอบบปรับระดับความดังเสียง (volume) ดังภาพที่ 7



ภาพที่ 7: หน้าต่างย่อหลังจากกดปุ่มพักชั่วคราว (pause)

ภาพที่ 8 แสดงถึงหน้าจอเกมหลังจากเข้าสู่ตัวเกมและกดปุ่มเครดิต ซึ่งแสดงรูปภาพของผู้จัดทำ ทั้งสองคน ชื่อ-นามสกุล และรหัสประจำตัวนิสิต พร้อมข้อความติดผลกระทบด้านหลังว่า Withdrawn ใน ลักษณะเดียวกัน ฯ เพื่อให้สอดคล้องกับชื่อทีม (Withdrawers) และเป็นส่วนหนึ่งในการนำชื่อทีมที่ผู้จัดทำได้ใช้เวลาค่อนข้างมากในการรังสรรค์ออกแบบ มาใส่ลงในตัวเกม



ภาพที่ 8: หน้าจอหลังจากกดปุ่มเครดิต (credit)



ภาพที่ 9: หน้าต่างย่อยหลังจากกดปุ่ม Option

ในส่วนของหน้าต่างย่อยของปุ่ม Option นั้น จะมีแบบที่สามารถเลื่อนได้ (slider bar) ดังภาพที่ 9 ซึ่งสามารถใช้ปรับระดับความดังของเสียงเพลงพื้นหลังภายในเกม รวมถึงเสียงเอฟเฟกต์ (effect sound) ภายในเกมได้ ซึ่งระดับความดังจะเริ่มต้นที่ 50 มีค่าน้อยที่สุด (minimum) เท่ากับ 0 และค่ามากที่สุด (maximum) เท่ากับ 100

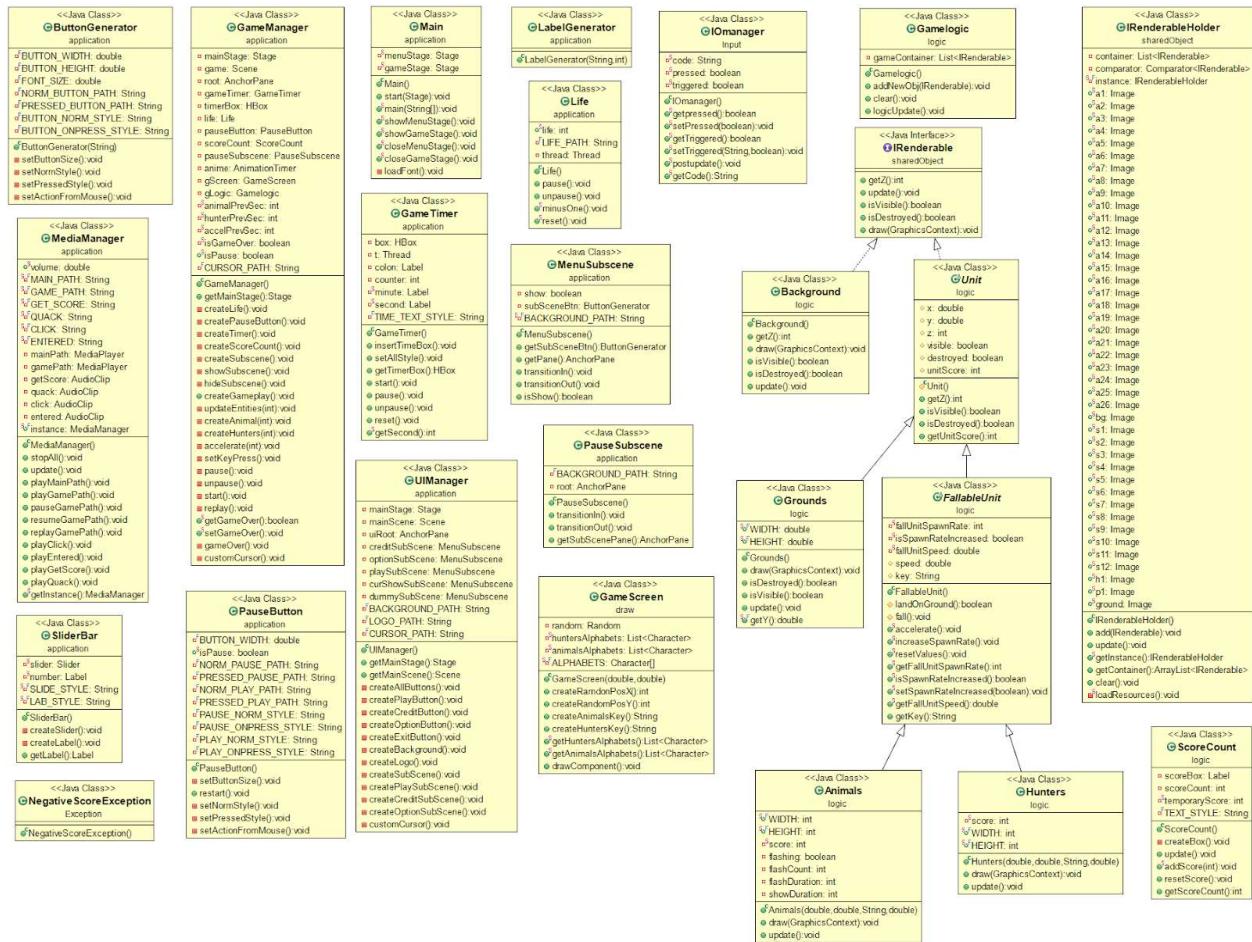
หลังจากที่ผู้เล่นกดปุ่มเริ่มเล่นเกม ผู้เล่นจะเข้าสู่หน้าจอเล่นเกมดังที่แสดงให้เห็นในภาพที่ 6 จากนั้นผู้เล่นมีหน้าที่กดปุ่มเพื่อทำลายสิ่งที่ตกลงมาจากด้านบนของตัวเกม ดังข้อกำหนดใน [2. How to play](#) หลังจากนั้น เมื่อนายพرانลงถึงพื้นหรือจำนวนชีวิตเหลือต่ำกว่า 0 เกมจะจบลง และจะแสดงหน้าจอสรุปคะแนนดังภาพที่ 10 ซึ่งประกอบไปด้วยข้อความ “Game Over” และคะแนนที่สามารถทำได้ เป็นตัวเลขตามลำดับ ในบรรทัดถัดลงมา จะมีปุ่มเล่นเกมอีกรอบ (again) และปุ่มกลับไปหน้าเมนู (menu) แสดงอยู่ เพื่อให้ผู้เล่นสามารถเลือกได้ว่าจะเล่นต่อหรือไม่



ภาพที่ 10: หน้าจอจบเกม (game over)

## 4. Implementation Details

ໄດ້ອາແກຣມ (diagram) ຂອງໂປຣແກຣມນີ້ແສດງໃຫ້ເຫັນດັ່ງການທີ 11 ດ້ວຍ ປະກອບໄປດ້ວຍ 5 ແພັກເກຈ (package) ຕົ້ນ application, draw, Exception, Input, logic ແລະ sharedObject ຜຶ້ນັບຈຳນວນຄລາສ (class) ລວມກັນທັງໝົດໄດ້ 25 ຄລາສ



ການທີ 11: UML Diagram ຂອງໂປຣແກຣມ

- Noted that Access Modifier Notations can be listed below

+ (**public**)

# (**protected**)

- (**private**)

**static** will be underlined.

**abstract** will be italic.

## 4.1 Class application. ButtonGenerator extends Button

### 4.1.1 Fields

- double BUTTON_WIDTH	ความกว้างของปุ่ม
- double BUTTON_HEIGHT	ความสูงของปุ่ม
- double FONT_SIZE	ขนาดของฟอนต์ด้านในปุ่ม
- String NORM_BOTTON_PATH	String url path ที่ระบุถึงภาพ Background ปุ่ม
- String PRESSED_BUTTON_PATH	String url path ที่ระบุถึงภาพ Background ปุ่มเมื่อถูกกด
- String BUTTON_NORM_STYLE	String ที่ใช้ตั้ง style ของปุ่มตอนปกติ
- String BUTTON_ONPRESS_STYLE	String ที่ใช้ตั้ง style ของปุ่มตอนโดนกด

### 4.1.2 Constructor

+ ButtonGenerator(String name)	รับ parameter เป็น String Constructor มีหน้าที่ <ul style="list-style-type: none"> <li>- ตั้งค่าด้านในปุ่ม ตาม parameter</li> <li>- ตั้งค่าขนาดปุ่มโดย setButtonSize()</li> <li>- ตั้งค่า style เเริ่มต้นของปุ่ม โดย setNormStyle()</li> <li>- ตั้งค่าการรับ input ต่างๆจาก mouse โดย setActionFromMouse()</li> </ul>
--------------------------------	--

### 4.1.3 Methods

- setButtonSize()	Method นี้มีหน้าที่ Set ค่าขนาดของปุ่มตามค่า BUTTON_WIDTH และ BUTTON_HEIGHT
- setNormStyle()	Method นี้มีหน้าที่ตั้งค่า Style ตาม BUTTON_NORM_STYLE และ เลื่อนตัวแหน่งของ ปุ่มลงไป 7 และเปลี่ยนความสูงของปุ่มเป็น 63
- setPressedStyle()	Method นี้มีหน้าที่ตั้งค่า Style ตาม BUTTON_ONPRESS_STYLE และ เลื่อนตัวแหน่ง ของปุ่มขึ้นไป 7 และเปลี่ยนความสูงของปุ่มเป็น 56
- setActionFromMouse()	Method นี้จะตั้งค่าการรับ input จากเมาส์ ดังนี้ <ul style="list-style-type: none"> <li>- เมื่อเมาส์เลื่อนเข้ามาวางบนปุ่ม : สร้าง effect เงา และ เล่นเสียง enterd จาก class MediaManager</li> <li>- เมื่อเมาส์เลื่อนออกจากปุ่ม : หยุด effect</li> </ul>

	<p>เงา</p> <ul style="list-style-type: none"><li>- เมื่อกดเมาส์ : ถ้าเป็นการคลิกซ้ายให้ตั้ง style ปุ่มเป็นตอนโดนกด และ เล่นเสียง clicked จาก MediaManager</li><li>- เมื่อปล่อยเมาส์ : ถ้าเป็นการปล่อยคลิกซ้ายให้ตั้ง style ปุ่มเป็นตอนปลดตัว</li></ul>
--	--

## 4.2 Class application. GameManager

### 4.2.1 Fields

- Stage mainStage	ค่าตัวแปรสำหรับ mainStage ของ game
- Scene game	ค่าตัวแปรสำหรับ Scene ของเกม
- AnchorPane root	Root ของ Scene
- GameTimer gameTimer	ค่าตัวแปร GameTimer
- HBox timerBox	Pane สำหรับแสดงเวลา
- Life life	ตัวแปร Life สำหรับแสดงชีวิตที่เหลือ
- PauseButton pauseButton	ตัวแปร PauseButton สำหรับปุ่ม pause
- ScoreCount scoreCount	ตัวแปร ScoreCount สำหรับแสดงคะแนน
- PauseSubscene pauseSubscene	ตัวแปร PauseSubScene สำหรับหน้า SubScene เวลา pause game
- AnimationTimer anime	ตัวแปร AnimationTimer สำหรับ Animation และ flow ของเกม
- GameScreen gScreen	ตัวแปร GameScreen สำหรับวิวภาพ graphic ต่างๆ
- GameLogic gLogic	ตัวแปร GameLogic สำหรับ flow ของเกม
- int animalPrevSec	ตัวแปรเก็บค่าวินาทีครั้งที่แล้วที่สร้างสัตว์
- int hunterPrevSec	ตัวแปรเก็บค่าวินาทีครั้งที่แล้วที่สร้างนายพราน
- int accelPrevSec	ตัวแปรเก็บค่าวินาทีที่แล้วที่เพิ่มความเร็วของ นายพรานและสัตว์
- boolean isGameOver	ตัวแปรที่เก็บค่าสถานะ game over
+ boolean isPause	ตัวแปรที่เก็บค่าสถานะการ pause game
- String CURSOR_PATH	String url Path ที่ระบุลักษณะของ cursor

### 4.2.2 Constructor

+ GameManager()	Constructor มีหน้าที่ดังนี้ - Initialize Scene, Stage GameScreen, GameLogic โดยให้
-----------------	---

	<ul style="list-style-type: none"> <li>- ขนาดเป็น 1366*768</li> <li>- Set title เป็น “Animal Rescue”</li> <li>- Set icon เกม</li> <li>- สร้าง GameTimer โดย createTimer()</li> <li>- สร้าง PauseButton โดย createPauseButton()</li> <li>- สร้าง ScoreCount โดย createScoreCount()</li> <li>- สร้าง PauseSubScene โดย createSubScene()</li> <li>- สร้าง Life โดย createLife()</li> <li>- Set Style ของ cursor</li> <li>- Set การรับ input ต่างๆ โดย setKeyPress()</li> <li>- เริ่ม Flow เกม</li> </ul>
--	---

#### 4.2.3 Methods

+ Stage getMainStage()	mainStage's getter
- void createLife()	Initialize life และ ตั้งตำแหน่งของ life เป็นที่ <ul style="list-style-type: none"> <li>- 30 ของ Anchor ขวา ของ Root</li> <li>- 110 ของ Anchor บน ของ Root</li> </ul>
- void createPauseButton()	Initialize pauseButton และ ตั้งตำแหน่งของ pauseButton เป็นที่ <ul style="list-style-type: none"> <li>- 30 ของ Anchor ขวา ของ Root</li> <li>- 20 ของ Anchor บน ของ Root</li> </ul>
- void createTimer()	Initialize gameTimer และ timerBox โดยรับค่ามาจาก gameTimer ตั้งตำแหน่งของ timerBox เป็นที่ <ul style="list-style-type: none"> <li>- 90 ของ Anchor ขวา ของ Root</li> <li>- 25 ของ Anchor บน ของ Root</li> </ul>
- void createScoreCount()	Initialize scoreCount และ ตั้งตำแหน่งของ scoreCount เป็นที่ <ul style="list-style-type: none"> <li>- 30 ของ Anchor ขวา ของ Root</li> <li>- 70 ของ Anchor บน ของ Root</li> </ul>
- void createSubScene()	Initialize pauseSubScene และ <ul style="list-style-type: none"> <li>- สร้างปุ่ม replay โดย ButtonGenerator โดย ตั้งให้มีอโคนกดจะข้อน subScene และเริ่มเกมใหม่</li> <li>- สร้างปุ่ม menu โดย ButtonGenerator โดยตั้งให้มีอโคนกดจะแสดงหน้า Stage</li> </ul>

	<p>Menu ขึ้นมาและปิด Stage ของเกม</p> <ul style="list-style-type: none"> <li>- สร้างปุ่ม exit โดย ButtonGenerator โดยเมื่อกดให้ปิด application ทันที</li> <li>- เก็บปุ่มทั้ง 3 ปุ่มใน VBox ที่มีขนาด spacing เป็น 10 และวางไว้ที่ ตำแหน่ง แกน x เป็น 400 แกน y เป็น 80</li> <li>- สร้าง SliderBar และวางที่ ตำแหน่งแกน x เป็น 120 แกน y เป็น 350</li> <li>- สร้าง LabelGenerator คำว่า "PAUSE" โดยมีขนาด 50 และตั้งที่ ตำแหน่งแกน x เป็น 60 แกน y เป็น 80</li> </ul>
- void showSubscene()	Show pauseSubScene โดยมี transition effect เลื่อนเข้ามาโดย method transitionIn() ของ pauseSubScene
- void hideSubscene()	ชื่อ pauseSubScene
- void createGameplay()	เริ่ม AnimationTimer เพื่อสร้าง flow ของ game เพื่อ update ค่าต่างๆ
- void updateEntities(int sec)	รับ parameter เวลาเป็นวินาทีเข้ามาเพื่อ <ul style="list-style-type: none"> <li>- สร้างสัตว์ โดย createAnimal(int sec)</li> <li>- สร้างนายพราน โดย createHunters(int sec)</li> <li>- ถ้า เวลาหารด้วย 25 ลงตัว และไม่เท่ากับ 0 ให้เพิ่มอัตราการสร้าง FallableUnit</li> <li>- ถ้าเวลาเท่ากับ 59 ให้ตั้งค่า animalPrevSec, hunterPrevSec, accelPrevSec เป็น 0</li> </ul>
- void createAnimal(int sec)	สร้างสัตว์ทุก 3 วินาที โดยสร้างจำนวนตามอัตราการสร้างของ FallableUnit
- void createHunters(int sec)	สร้างนายพรานทุก 2 วินาที โดยสร้างจำนวนตามอัตราการสร้างของ FallableUnit
- void accelerate(in sec)	เพิ่มความเร็วของ FallableUnit ทุก 5 วินาที
- void setKeyPress()	Set ค่าการรับ input ดังนี้ <ul style="list-style-type: none"> <li>- มีการกดคีย์บอร์ด : ถ้าไม่ได้กดค้าง ให้ set triggered ใน IOManager เป็น true</li> <li>- ปล่อยคีย์บอร์ด : set ค่า pressed ของ IOManager เป็น false</li> </ul>
- void pause()	Pause game หยุด flow เกมทุกอย่าง และปิด pauseSubScene

- void unpause()	Resume game กลับมาดำเนิน flow เกมทุกอย่างต่อ
- void start()	เริ่ม flow game โดย method createGamePlay()
- void replay()	เริ่มเกมใหม่โดยตั้งค่าตัวแปรต่างๆที่เกี่ยวกับ flow เกมกลับเป็นค่าเริ่มต้นทั้งหมด
+ boolean <u>getGameOver()</u>	isGameOver's getter
+ boolean <u>setGameOver()</u>	isGameOver's setter
- void gameOver()	<p>Method สำหรับเมื่อ game over โดยจะหยุด flow game ทั้งหมด และ disable ปุ่ม pause และแสดงหน้า SubScene game over</p> <p>โดยจะ Initialize PauseSubScene ในส่วน subSceneสำหรับหน้า game over โดยมีองค์ประกอบดังนี้</p> <ul style="list-style-type: none"> <li>- LabelGenerator คำว่า “GAMEOVER” ขนาด 48 ตั้งที่ ตำแหน่ง แกน y เป็น 49 มี Alignment เป็น Pos.CENTER มีความกว้างเท่ากับความกว้าง SubScene</li> <li>- LabelGenerator คำว่า “Your Score Is” ขนาด 40 ตั้งที่ ตำแหน่ง แกน y เป็น 140 มี Alignment เป็น Pos.CENTER มีความกว้างเท่ากับความกว้าง SubScene</li> <li>- LabelGenerator แสดงคะแนนขนาด 72 ตั้งที่ ตำแหน่ง แกน y เป็น 189 มี Alignment เป็น Pos.CENTER มีความกว้างเท่ากับความกว้าง SubScene</li> <li>- ButtonGenerator ปุ่ม “Again” ตั้งที่ ตำแหน่งแกน x เป็น ความกว้างของ SubScene หาร 2 และลับด้วย 272 แกน y เป็น 310 โดยตั้งให้มีอโคนกดปุ่มจะเริ่มเล่นเกมใหม่</li> <li>- ButtonGenerator ปุ่ม “Menu” ตั้งที่ ตำแหน่งแกน x เป็น ความกว้างของ SubScene หาร 2 และลับด้วย 20 แกน y เป็น 310 โดยตั้งให้มีอโคนกดปุ่มจะออกจาก Stage game และแสดง Stage menu</li> </ul>
- void customCursor()	ตั้งค่าของ cursor ตามรูปจาก CURSOR_PATH

## 4.3 Class application. GameTimer

### 4.3.1 Fields

- HBox box	HBox สำหรับแสดงค่าเวลา
- Thread t	Thread สำหรับการนับเวลา
- Label colon	Label สำหรับ ":"
- int counter	ตัวแปรสำหรับนับเวลา
- <u>Label minute</u>	ตัวแปรสำหรับแสดงเวลาในหน่วยนาที
- <u>Label second</u>	ตัวแปรสำหรับแสดงตัวแปรในหน่วยวินาที
- String TIME_TEXT_STYLE	String สำหรับใช้แต่ง style ของ Label ที่แสดง

### 4.3.2 Constructor

+ GameTimer()	Initialize Hbox และ Label ทั้งหมด โดยใช้ method insertTimeBox()
---------------	---

### 4.3.3 Methods

+ void insertTimeBox()	Initialize Hbox ที่มี spacing เป็น 1.4 Label ต่างๆ โดยตั้งค่าเริ่มต้นเป็น "00 : 00" และตั้งค่า style ของ Label ทั้งหมดโดย setAllStyle()
+ void setAllStyle()	ตั้งค่า style ของ Label ทั้งหมดตาม TIME_TEXT_STYLE
+ HBox getTimerBox()	Getter ของ box
+ void start()	Initialize thread สำหรับนับเวลา โดยใช้ counter นับวินาที และแสดงค่าเวลาโดยการเปลี่ยนตัวเลขที่ Label ที่ใช้แสดงเวลา
+ void pause()	หยุดการทำงานของ thread
+ void unpause()	Resume การทำงานของ thread
+ void reset()	Reset ค่าเวลากลับไปเป็นเวลาเริ่มต้น
+ int <u>getSecond()</u>	คืนเวลาที่ผ่านไปในหน่วยวินาที

## 4.4 Class application. LabelGenerator extends Label

### 4.4.1 Constructor

+ LabelGenerator(String text, int size)	รับ parameter เป็น String และ int และทำการตั้งค่าดังนี้ <ul style="list-style-type: none"><li>- Set text ตาม String ที่รับเข้ามา</li><li>- Warp text</li><li>- ตั้ง Position เป็น TOP_CENTER</li><li>- Set font ของ text เป็น Joystix Monospace และมีขนาดตามตัวแปร int ที่รับเข้ามา</li></ul>
---	---

## 4.5 Class application. Life extends HBox

### 4.5.1 Fields

+ int life	จำนวนชีวิต
- String LIFE_PATH	String ของ url ที่ที่ระบุสิ่งไฟล์ภาพของรูป life
- Thread thread	Thread สำหรับคอยเช็ค status ของ life

### 4.5.2 Constructor

+ Life()	Initialize โดย <ul style="list-style-type: none"> <li>- ตั้ง spacing เป็น 10</li> <li>- Start thread สำหรับ check status ของ life ทุก 20 ms โดยถ้าชีวิตเหลือต่า กว่า 0 ให้ game Over ทันที</li> </ul>
----------	---

### 4.5.3 Methods

+ void pause()	หยุดการทำงานของ thread
+ void unpause()	Resume การทำงานของ thread
+ void minusOne()	ลดจำนวนชีวิตลง 1
+ void reset()	Reset จำนวนชีวิตกลับเป็นค่าเริ่มต้น

## 4.6 Class application. Main extends Application

### 4.6.1 Fields

<u>- Stage menuStage</u>	Stage ของหน้า Menu
<u>- Stage gameStage</u>	Stage ของหน้า Game

### 4.6.2 Methods

<u>+ void start(Stage primaryStage)</u>	ทำการโหลด Font ทั้งหมด และแสดงหน้า Menu โดยใช้ loadFont() และ showMenuStage()
<u>+ void main(String[] args)</u>	Launch application
<u>+ void showMenuStage()</u>	Initialize menuStage โดยรับค่าจาก UIManager และแสดงหน้า Menu โดยให้ขยายขนาด Stage ไม่ได้พร้อมกับเล่นเพลิงของหน้า menu
<u>+ void showGameStage()</u>	Initialize gameStage โดยรับค่าจาก GameManager และแสดงหน้า Game โดยให้ขยายขนาด Stage ไม่ได้พร้อมกับเล่นเพลิงของหน้า game
<u>+ void closeMenuStage()</u>	ปิดหน้า Menu ลง
<u>+ void closeGameStage()</u>	ปิดหน้า Game ลง
<u>- void loadFont()</u>	Load font ที่ใช้ทั้งหมดในเกม

## 4.7 Class application. MediaManager

### 4.7.1 Fields

<u>+ double volume</u>	ตัวแปรเก็บค่าความดังของเสียงในเกม
<u>- String MAIN_PATH</u>	String url ที่ระบุถึงเพลงสำหรับหน้าเมนู
<u>- String GAME_PATH</u>	String url ที่ระบุถึงเพลงสำหรับหน้าเกม
<u>- String GET_SCORE</u>	String url ที่ระบุถึงเสียงเวลาได้รับคะแนน
<u>- String QUACK</u>	String url ที่ระบุถึงเสียงเวลาโดนลงคะแนน
<u>- String CLICK</u>	String url ที่ระบุถึงเสียงเวลาคลิกปุ่ม
<u>- String ENTERED</u>	String url ที่ระบุถึงเสียงเวลาเลื่อนมาสู่ปุ่ม
<u>- MediaPlayer mainPath</u>	ตัวแปร MediaPlayer สำหรับเพลงหน้าเมนู
<u>- MediaPlayer gamePath</u>	ตัวแปร MediaPlayer สำหรับเพลงหน้าเกม
<u>- AudioClip getScore</u>	ตัวแปร AudioClip สำหรับเสียงเวลาได้รับคะแนน
<u>- AudioClip quack</u>	ตัวแปร AudioClip สำหรับเสียงเวลาโดนลงคะแนน
<u>- AudioClip click</u>	ตัวแปร AudioClip สำหรับเสียงเวลาคลิกปุ่ม
<u>- AudioClip entered</u>	ตัวแปร AudioClip สำหรับเสียงเวลาเลื่อนมาสู่ปุ่ม
<u>+ MediaManager instance</u>	ตัวแปร instance สำหรับใช้แบบ singleton

### 4.7.2 Constructor

<u>+ MediaManager()</u>	Initialize การเล่นวนของเพลงหน้าเมนูและหน้าเกม
-------------------------	---

### 4.7.3 Methods

<u>+ void stopAll()</u>	หยุดการเล่นเพลงหน้าเมนูและหน้าเกม
<u>+ void update()</u>	Update ความดังของเสียงตามค่าของ volume
<u>+ void playMainPath()</u>	หยุดเพลงทั้งหมดและเล่นเพลงหน้าเมนู
<u>+ void playGamePath()</u>	หยุดเพลงทั้งหมดเล่นเพลงหน้าเกม

+ void pauseGamePath()	Pause เพลงหน้าเกม
+ void resumeGamePath()	Resume เพลงหน้าเกม
+ void replayGamePath()	ตั้งค่าเพลงในหน้าเกมให้กลับมาเริ่มต้นใหม่
+ void playClick()	เล่นเสียง clicked
+ void playEntered()	เล่นเสียง entered
+ void playGetScore()	เล่นเสียง getScore
+ void playQuack()	เล่นเสียง quack
+ <u>MediaManager getInstance()</u>	คืนตัวแปร instance

## 4.8 Class application. MenuSubscene extends Subscene

### 4.8.1 Fields

- boolean show	ตัวแปร boolean สำหรับแสดงว่า subScene แสดงอยู่หรือไม่ โดยมีค่าเริ่มต้นเป็น false
- <u>ButtonGenerator</u> subSceneBtn	ตัวแปร ButtonGenerator สำหรับปุ่มใน subScene
- String BACKGROUND_PATH	String url สำหรับรับระบุถึงภาพ background ของหน้า subScene

### 4.8.2 Constructor

+ MenuSubscene()	Initialize SubScene ขนาดกว้าง 700 สูง 430.5 โดยมี root เป็น AnchorPane ที่ถูกตั้ง style เป็น BACKGROUND_PATH และหน้า SubScene มีตำแหน่งแกน x เป็น 1400 แกน y เป็น 245 โดยมีองค์ประกอบหน้า SubSceneดังนี้ <ul style="list-style-type: none"> <li>- ปุ่ม ButtonGenerator “Back” โดยมี ความสูง 31.5 กว้าง 210 และตำแหน่งแกน x เป็น 450 แกน y เป็น 330</li> </ul>
------------------	---

### 4.8.3 Methods

+ void transitionIn()	Show ตัว subscene โดยการเลื่อนเข้ามาในหน้า application โดยเลื่อน -875 และมีระยะเวลาการเล่น transition เป็น 300 ms
+ void transitionOut()	ช่อนตัว subscene โดยเลื่อนออกจากหน้า application โดยเลื่อน 875 และมีระยะเวลาการเล่น transition เป็น 300 ms
+ Gette & Setter	<ul style="list-style-type: none"> <li>- Getter ของ root</li> <li>- Getter ของ subSceneBtn</li> <li>- Getter ของ show</li> </ul>

## 4.9 Class application. PauseButton extends Button

### 4.9.1 Fields

- double BUTTON_WIDTH	ความกว้างปุ่ม
+ boolean isPause	ตัวแปรเก็บสถานะการ pause
- String NORM_PAUSE_PATH	String url ที่ระบุถึงไฟล์ภาพปุ่ม pause ตอนปกติ
- String PRESSED_PAUSE_PATH	String url ที่ระบุถึงไฟล์ภาพปุ่ม pause ตอนโดนกด
- String NORM_PLAY_PATH	String url ที่ระบุถึงไฟล์ภาพปุ่ม play ตอนปกติ
- String PRESSED_PLAY_PATH	String url ที่ระบุถึงไฟล์ภาพปุ่ม play ตอนโดนกด
- String PAUSE_NORM_STYLE	Style ของปุ่ม pause ตอนปกติ
- String PAUSE_ONPRESS_STYLE	Style ของปุ่ม pause ตอนโดนกด
- String PLAY_NORM_STYLE	Style ของปุ่ม play ตอนปกติ
- String PLAY_ONPRESS_STYLE	Style ของปุ่ม play ตอนโดนกด

### 4.9.2 Constructor

+ PauseButton()	Initialize ปุ่มและ <ul style="list-style-type: none"> <li>- ตั้งค่าขนาดปุ่มโดย setButtonSize()</li> <li>- ตั้งต่า style ปุ่มเริ่มต้น โดย setNormStyle()</li> <li>- Set การรับ input ต่างๆ โดย setActionFromMouse()</li> </ul>
-----------------	---

### 4.9.3 Methods

- void setButtonSize()	ตั้งค่าความกว้างปุ่มตาม BUTTON_WIDTH
+ void restart()	เลิก pause เกมและตั้งทุกอย่างกลับเป็นค่า และstyleเริ่มต้น และเลิก disable ปุ่ม
- void setNormStyle()	ทำการตั้ง style ปุ่มเป็นแบบตอนปกติ โดย <ul style="list-style-type: none"> <li>- ความสูงปุ่ม 49</li> <li>- เลื่อนปุ่มขึ้น 10</li> <li>- ถ้าเกม pause อยู่ตั้ง style เป็นปุ่ม play ตอนปกติ ถ้าไม่ pause ตั้ง style เป็นปุ่ม pause ตอนปกติ</li> </ul>

- void setPressedStyle()	ทำการตั้ง style ปุ่มเป็นแบบตอนโดนกด โดย - ความสูงปุ่ม 45 - เลื่อนปุ่มลง 10 - ถ้าเกม pause อุ่ตั้ง style เป็นปุ่ม play ตอนโดนกด ถ้าไม่ pause ตั้ง style เป็น ปุ่ม pause ตอนโดนกด
- void setActionFromMouse()	ตั้งค่าการรับ input ของปุ่มดังนี้ - เม้าส์เลื่อนมาวาง : สร้างเงาของปุ่ม - เม้าส์เลื่อนออก : ลบเงาของปุ่ม - เม้าส์กด : ตั้ง style ปุ่ม เป็นตอนโดนกด - เม้าส์ปล่อย : ถ้าเกม pause อุ่ให้เลิก และตั้งปุ่ม style เป็นตอนปกติ

## 4.10 Class application. PauseSubscene extends Subscene

### 4.10.1 Fields

- String BACKGROUND_PATH	String url ระบุถึงไฟล์ภาพ background ของ PauseSubScene
- AnchorPane root	PauseSubScene's root

### 4.10.2 Constructor

+ PauseSubscene()	Initialize SubScene โดยมี root เป็น AnchorPane และ <ul style="list-style-type: none"> <li>- ตั้งขนาด subScene กว้าง 730 สูง 449</li> <li>- ตั้ง style ของ SubScene ให้มี background เป็น ภาพตาม BACKGROUND_PATH</li> <li>- ตั้งตำแหน่งแกน x เป็น <math>(1366 - \text{ความกว้าง subScene})/2</math></li> <li>- ตั้งตำแหน่งแกน y เป็น <math>-(\text{ความสูง subScene})</math></li> </ul>
-------------------	--

### 4.10.3 Methods

+ void transitionIn()	สร้าง effect เลื่อนเข้ามาของ subScene โดยมี ระยะเวลา transition เป็น 300 ms
+ void transitionOut()	สร้าง effect เลื่อนออกของ subScene โดยมี ระยะเวลา transition เป็น 300 ms
+ Getter & Setter	Getter ของ root

## 4.11 Class application. SliderBar extends HBox

### 4.11.1 Fields

<u>- Slider slider</u>	SliderBar's slider
<u>- Label number</u>	SliderBar's number สำหรับแสดงค่าของตัวเลื่อน
<u>- String SLIDE_STYLE</u>	Style ของ slider
<u>- String LAB_STYLE</u>	Style ของ Label

### 4.11.2 Constructor

+ SliderBar()	Initialize Hbox โดย spacing เท่ากับ 10 และ <ul style="list-style-type: none"><li>- สร้าง label สำหรับคำว่า volume และ set style ตาม LAB_STYLE</li><li>- สร้าง slider โดย createSlider()</li><li>- สร้าง number โดย createLabel()</li></ul>
---------------	--

### 4.11.3 Methods

- void createSlider()	Initialize slider โดยมีค่าต่างๆดังนี้ <ul style="list-style-type: none"><li>- Min = 0, Max = 100</li><li>- ค่าเริ่มต้นเป็น MediaManager.volume*100</li><li>- ตั้งขนาด 296*16</li><li>- ตั้ง style ตาม SLIDE_STYLE</li></ul>
- void createLabel()	Initialize number โดยตั้งค่าดังนี้ <ul style="list-style-type: none"><li>- Style ตาม LAB_STYLE</li><li>- Bind ค่ากับ slider</li><li>- สร้าง thread มาโดย update ความดัง เสียงในเกม</li></ul>
+ Getter & Setter	Getter ของ number

## 4.12 Class application. UIManager

### 4.12.1 Fields

- Stage mainStage	UIManager's Stage
- Scene mainScene	UIManager's Scene
- AnchorPane uiRoot	UIManager's Root
- MenuSubscene creditSubscene	ตัวแปร MenuSubScene ของหน้า credit
- MenuSubscene optionSubscene	ตัวแปร MenuSubScene ของหน้า option
- MenuSubscene playSubscene	ตัวแปร MenuSubScene ของหน้า play
- MenuSubscene curShowSubscene	ตัวแปร MenuSubScene สำหรับเก็บค่าตัว SubScene ที่แสดงอยู่
- MenuSubscene dummySubscene	ตัวแปร MenuSubScene สำหรับเก็บค่าตัว SubScene ที่ใช้เป็นตัวสับไปมา
- String BACKGROUND_PATH	String url ที่ระบุถึงไฟล์ภาพ Background
- String LOGO_PATH	String url ที่ระบุถึงไฟล์ภาพ logo
- String CURSOR_PATH	String url ที่ระบุถึงไฟล์ภาพ cursor

### 4.12.2 Constructor

+ UIManager()	<b>Initialize</b> ตัวแปรต่างๆดังนี้ <ul style="list-style-type: none"> <li>- Initialize uiRoot, mainScene และ mainStage โดยมีขนาด 1366*768</li> <li>- ตั้งภาพ background โดย createBackground()</li> <li>- สร้างปุ่ม โดย createAllButton()</li> <li>- สร้าง logo โดย createLogo()</li> <li>- ตั้ง style cursor โดย customCursor()</li> <li>- สร้าง SubScene ต่างๆ โดย createSubScene()</li> </ul>
---------------	---

### 4.12.3 Methods

- void createAllButtons()	สร้างปุ่มทั้งหมด โดย <ul style="list-style-type: none"> <li>- ปุ่ม play โดย createPlayButton()</li> <li>- ปุ่ม credit โดย createCreditButton()</li> </ul>
---------------------------	---

	<ul style="list-style-type: none"> <li>- ปุ่ม option โดย createOptionButton()</li> <li>- ปุ่ม exit โดย createExitButton()</li> </ul>
- void createPlayButton()	<p>สร้างปุ่ม play โดย ButtonGenerator โดยตั้งค่าดังนี้</p> <ul style="list-style-type: none"> <li>- Text เป็น “PLAY”</li> <li>- ตำแหน่ง แกน x ที่ 122.5 แกน y ที่ 315</li> <li>- ตั้งการรับ input มาส์คลิกให้แสดง subScene นี้เมื่อคลิกซ้ายและ ถ้าแสดงอยู่แล้วให้ซ่อน subScene</li> </ul>
- void createCreditButton()	<p>สร้างปุ่ม credit โดย ButtonGenerator โดยตั้งค่าดังนี้</p> <ul style="list-style-type: none"> <li>- Text เป็น “CREDIT”</li> <li>- ตำแหน่ง แกน x ที่ 122.5 แกน y ที่ 385</li> <li>- ตั้งการรับ input มาส์คลิกให้แสดง subScene นี้เมื่อคลิกซ้ายและ ถ้าแสดงอยู่แล้วให้ซ่อน subScene</li> </ul>
- void createOptionButton()	<p>สร้างปุ่ม option โดย ButtonGenerator โดยตั้งค่าดังนี้</p> <ul style="list-style-type: none"> <li>- Text เป็น “OPTION”</li> <li>- ตำแหน่ง แกน x ที่ 122.5 แกน y ที่ 455</li> <li>- ตั้งการรับ input มาส์คลิกให้แสดง subScene นี้เมื่อคลิกซ้ายและ ถ้าแสดงอยู่แล้วให้ซ่อน subScene</li> </ul>
- void createExitButton()	<p>สร้างปุ่ม exit โดย ButtonGenerator โดยตั้งค่าดังนี้</p> <ul style="list-style-type: none"> <li>- Text เป็น “EXIT”</li> <li>- ตำแหน่ง แกน x ที่ 122.5 แกน y ที่ 525</li> <li>- ตั้งการรับ input มาส์คลิกซ้ายให้ออกเกมทันที</li> </ul>
- void createBackground()	Set background ตาม BACKGROUND_PATH
- void createLogo()	สร้าง ImageView ของ logo ตาม LOGO_PATH และตั้งตำแหน่งแกน x แกน y เป็น 100 และ 120 ตามลำดับ
- void createSubscene()	<p>Initialize SubScene ทั้งหมดโดย</p> <ul style="list-style-type: none"> <li>- playSubScene โดย createPlaySubScene()</li> <li>- creditSubScene โดย createPlaySubScene()</li> <li>- optionSubScene โดย createOptionSubScene()</li> </ul>

	<ul style="list-style-type: none"> <li>- curShowSubScene และ dummySubScene ให้ initialize ตามปกติ</li> </ul>
- void createPlaySubscene()	Initialize playSubScene โดย <ul style="list-style-type: none"> <li>- Set background ตามภาพ play_scene.png</li> <li>- สร้าง Label คำว่า “Tutorial” ขนาด 36 มี Alignment เป็น Pos.Center ความกว้าง ตามขนาด subScene ตำแหน่งแกน y เป็น 30</li> <li>- ตั้ง subSceneBtn เป็นคำว่า “ok” เมื่อโดนกดให้เริ่มเกม</li> </ul>
- void createCreditSubscene()	Initialize creditSubScene โดย <ul style="list-style-type: none"> <li>- Set background ตามภาพ credit_scene.png</li> <li>- ตั้ง subSceneBtn เมื่อโดนกดให้ช่อง subScene</li> </ul>
- void createOptionSubscene()	Initialize playSubScene โดย <ul style="list-style-type: none"> <li>- สร้าง Label คำว่า “Option” ขนาด 36 มี Alignment เป็น Pos.Center ความกว้าง ตามขนาด subScene ตำแหน่งแกน y เป็น 30</li> <li>- ตั้ง subSceneBtn เมื่อโดนกดให้ช่อง subScene</li> <li>- สร้าง SliderBar ไว้ที่ตำแหน่ง แกน x ที่ 100 แกน y ที่ 180</li> </ul>
- void customCursor()	ตั้ง style ของ cursor ตาม CURSOR_PATH
+ Getter & Setter	Getter ของ mainStage & mainScene

## 4.13 Class draw. GameScreen extends Canvas

### 4.13.1 Fields

- Random random	GameScreen's random สำหรับสุ่ม
- List<Character> huntersAlphabets	ArrayList ของตัวอักษรของนายพรานที่อยู่บนหน้าจอเกม
- List<Character> animalsAlphabets	ArrayList ของตัวอักษรของสัตว์ที่อยู่บนหน้าจอเกม
- Character[] ALPHABETS	อาร์เรย์เก็บตัวอักษรภาษาอังกฤษ A – Z

### 4.13.2 Constructor

+ GameScreen(double width, double height)	<b>Initialize ค่าต่างๆ</b> <ul style="list-style-type: none"> <li>- Initialize Canvas ขนาดตาม parameter ที่รับเข้ามา</li> <li>- Initialize random</li> <li>- Initialize huntersAlphabets และ animalsAlphabets</li> </ul>
---	--

### 4.13.3 Methods

+ int createRandomPosX()	สุ่มตำแหน่งในแกน x
+ int createRandomPosY()	สุ่มตำแหน่งในแกน y
+ String createAnimalsKey()	สุ่มสร้างตัวอักษรให้กับสัตว์ โดยจะไม่เอาตัวอักษรเดียวกับของนายพราน
+ String createHuntersKey()	สุ่มสร้างตัวอักษรให้กับนายพราน โดยจะไม่เอาตัวอักษรเดียวกับของสัตว์
+ void drawComponent()	เรียก method draw() ของตัวแปร IRenderable ใน container ของ IRenderableHolder ถ้ายังไม่ถูกทำลายและ visible
+ Getter & setter	Getter ของ huntersAlphabets และ animalsAlphabets

## 4.14 Class Exception. NegativeScoreException extends Exception

### 4.14.1 Constructor

+ NegativeScoreException()	Throw exception พร้อมข้อความ the score would not be negative!
----------------------------	---

## 4.15 Class Input. IOManager

### 4.15.1 Fields

<u>- String code</u>	ตัวแปรเก็บตัวอักษรที่โดนกด
<u>- boolean pressed</u>	ตัวแปรเก็บสถานะการ pressed
<u>- boolean triggered</u>	ตัวแปรเก็บสถานะการ triggered

### 4.15.2 Methods

<u>+ void postUpdate()</u>	ตั้งให้ triggered เป็น false
<u>+ Getter &amp; Setter</u>	<ul style="list-style-type: none"><li>- Getter &amp; Setter ของ pressed</li><li>- Getter &amp; Setter ของ triggered ถ้าโดน trigger ให้ เก็บตัวแปร code เป็น ตัวอักษรที่กดด้วย</li><li>- Getter ของ code</li></ul>

## 4.16 Class logic. Animals extends FallableUnit

### 4.16.1 Fields

<u>+ int WIDTH</u>	ความกว้างของรูปสัตว์
<u>+ int HEIGHT</u>	ความสูงของรูปสัตว์
<u>- int score</u>	คะแนนที่จะถูกหักเมื่อกำจัดสัตว์
<u>- boolean flashing</u>	ตัวแปรเก็บสถานะการกระพริบ
<u>- int flashCount</u>	จำนวนครั้งการกระพริบ
<u>- int flashDuration</u>	ระยะเวลาของการกระพริบ
<u>- int showDuration</u>	ระยะเวลาของการแสดงระหว่างการกระพริบ

### 4.16.2 Constructor

<u>+ Animals(double x, double y, String key, double speed)</u>	Initialize ตัวแปรทั้งหมดตาม parameters ที่รับเข้ามาและให้ค่า z ของสัตว์เป็น 8
--	---

### 4.16.3 Methods

<u>+ void draw(GraphicsContext gc)</u>	วาดภาพตาม key ของสัตว์
<u>+ void update()</u>	<ul style="list-style-type: none"> <li>- เรียก method fall()</li> <li>- ถ้ากด keyboard ตรงกับ key ของสัตว์ให้ทำลายสัตว์และลบคะแนนและชีวิต</li> <li>- ถ้าสัตว์ตกลงถึงพื้นให้ speed เป็น 0 และให้กระพริบ</li> <li>- ถ้า game Over ให้ทำลายสัตว์</li> </ul>

## 4.17 Class logic. Background implements IRenderable

### 4.17.1 Methods

+ int getZ()	คืนค่าเป็น -999
+ void draw(GraphicsContext gc)	วาดภาพตามภาพจาก IRenderableHolder.bg ที่ตำแหน่ง (0, 0) ขนาด 1366*768
+ boolean isVisible()	คืนค่า true
+ boolean isDestroyed()	คืนค่า false
+ void update()	

## 4.18 Class logic. FallableUnit (Abstract) extends Unit

### 4.18.1 Fields

<u>- int fallUnitSpawnRate</u>	อัตราการสร้างตัวละครที่เป็น FallableUnit
<u>- boolean isSpawnRateIncreased</u>	สถานะการเพิ่มอัตราการสร้างตัวละคร FallableUnit
<u>- double fallUnitSpeed</u>	ความเร็วของตัวละคร FallableUnit
# double speed	ความเร็วในการตก
# String key	ตัวแปรใช้เก็บค่าตัวอักษรของ Unit

### 4.18.2 Methods

# bool landOnGround()	คืนค่า true เมื่อความสูงของพื้นกับความสูงของ unit ห่างกันไม่เกิน 150 ไม่อย่างนั้นคืนค่า false
# void fall()	เพิ่มค่า y = speed
+ void accelerate()	fallUnitSpeed เพิ่มความเร็วเท่ากับ 5% ของความเร็ว fallUnitSpeed เดิม
+ void increaseSpawnRate	เพิ่ม fallableUnitSpawnRate 1 หน่วย แต่ fallableUnitSpawnRate ห้ามเกิน 5
+ void resetValues()	<ul style="list-style-type: none"> <li>- กำหนดให้ fallUnitGenerate เป็น 1</li> <li>- กำหนดให้ fallUnitSpeed เป็น 2</li> <li>- กำหนดให้ isGenRateIncreased เป็น false</li> </ul>
+ Getter & Setter	<ul style="list-style-type: none"> <li>- <u>Getter</u> ของ fallUnitSpawnRate</li> <li>- <u>Getter &amp; Setter</u> ของ isSpawnRateIncreased</li> <li>- <u>Getter</u> ของ fallableUnitSpeed</li> <li>- <u>Getter</u> ของ key</li> </ul>

## 4.19 Class logic. Gamelogic

### 4.19.1 Field

- List<IRenderable> gameContainer	ตัวแปรที่ใช้เก็บ Object ที่เป็น IRenderable
-----------------------------------	---

### 4.19.2 Constructor

+ GameLogic()	<ul style="list-style-type: none"><li>- ประกาศ gameContainer เป็น ArrayList</li><li>- เพิ่ม background และ ground ลงใน IRenderableHolder</li></ul>
---------------	--

### 4.19.3 Methods

+ addNewObj(IRenderable r)	เพิ่ม Object r ลงใน gameContainer และ IRenderableHolder
+ void clear()	ลบทุกสมาชิกใน gameContainer
+ logicUpdate()	ลบสมาชิกทุกตัวใน gameContainer ที่มีค่า isDestryed เป็น true และ update() ที่เหลือ

## 4.20 Class logic. Grounds extends Unit

### 4.20.1 Fields

+ double <u>WIDTH</u>	ค่าความกว้าง
+ double <u>HEIGHT</u>	ค่าความสูง

### 4.20.2 Constructor

+ Grounds()	<ul style="list-style-type: none"><li>- กำหนดค่า z เป็น -998 คืออยู่หลังสุดของภาพ background</li><li>- กำหนดค่า x เป็น 0</li><li>- กำหนดค่า y เป็น 668</li></ul>
-------------	--

### 4.20.3 Methods

+ void draw(GraphicsContext gc)	วาดภาพ ground ที่ตำแหน่ง x, y
+ bool isDestroyed()	คืนค่า false
+ bool isVisible()	ค่าค่า true
+ void update()	
+ double <u>getY()</u>	คืนค่า 668 (ค่าของ y)

## 4.21 Class logic. Hunters extends FallableUnit

### 4.21.1 Fields

<u>- int score</u>	คะแนนที่ได้จากการกำจัดนายพ_ratio_n
<u>+ int WIDTH</u>	ความกว้างของนายพ_ratio_n
<u>+ int HEIGHT</u>	ความสูงของนายพ_ratio_n

### 4.21.2 Constructor

<u>+ Hunters(double x, double y, String key, double speed)</u>	Set ค่าตัวแปร x, y, key, speed และ ให้ z เป็น 9
--	---

### 4.21.3 Methods

<u>+ void draw(GraphicsContext gc)</u>	วาดภาพนายพ_ratio_n ซึ่งมีตัวอักษร key แปะอยู่บนร่ม
<u>+ void update()</u>	ค่อยรับค่า key ที่ใส่เข้ามาจาก IOmanager ถ้า ตรงกับ key ของนายพ_ratio_n ให้ทำลาย พร้อมกับ getScore และเล่นเสียง

## 4.22 Class logic. scoreCount extends HBox

### 4.22.1 Fields

- Label scoreBox	คะแนนที่แสดงทางหน้าจอ
- int scoreCount	ตัวนับคะแนน
- int temporaryScore	ตัวนับการเปลี่ยนแปลงของคะแนน
- String TEXT_STYLE	String ที่ใช้ setStyle ของข้อความ

### 4.22.2 Constructor

+ ScoreCount()	กำหนด scoreCount เป็น 0 เรียก createBox()
----------------	--

### 4.22.3 Methods

- void createBox()	<ul style="list-style-type: none"> <li>- สร้าง Label ค่าว่า “SCORE: ”</li> <li>- สร้าง scoreBox โดยให้ค่าเริ่มต้นเป็น 00000</li> <li>- add Label และ scoreBox ลงในคลาส</li> </ul>
+ void update() throws NegativeScoreException()	<ul style="list-style-type: none"> <li>- นำ scoreCount และ temporaryScore มาบวกกัน และให้ temporaryScore เป็น 0</li> <li>- ให้ text ของ scoreBox เป็น scoreCount ที่เป็น String 5 หลัก นำหน้าด้วย 0</li> </ul>
+ void addScore(int score)	เพิ่ม score ลงใน temporaryScore
+ void resetScore()	ให้ scoreCount มีค่าเป็น 0
+ int getScoreCount()	คืนค่า scoreCount

## 4.23 Class logic. Unit (Abstract) implements IRenderable

### 4.23.1 Fields

# double x	ตำแหน่งแนวแกน x
# double y	ตำแหน่งแนวแกน y
# int z	ความตื้น-ลึกของ
# boolean visible	เก็บค่าสถานะการแสดง
# boolean destroyed	เก็บค่าสถานะการถูกทำลาย
# int UnitScore	คะแนนที่จะได้หรือถูกหักจาก unit

### 4.23.2 Constructor

# Unit()	set visible = true set destroyed = false
----------	---

### 4.23.3 Methods

+ int getZ()	คืนค่า z
+ boolean isVisible()	คืนสถานะ visible
+ boolean isDestroyed()	คืนสถานะ destroyed
+ int getUnitScore()	คืน unitScore

## 4.24 Interface sharedObject. IRenderable

### 4.24.1 Methods

+ int getZ()	
+ void update()	
+ boolean isVisible()	
+ boolean isDestroyed()	
+ void draw(GraphicsContext gc)	

## 4.25 Class sharedObject. IRenderableHolder

### 4.25.1 Fields

- List<IRenderable> container	List ที่เก็บตัวแปร IRenderable
- Comparator<IRenderable> comparator	ตัวแปรที่เก็บสถานะการเปรียบเทียบค่า z เพื่อจัดลำดับการแสดงข้างหน้าข้างหลังบนหน้าจอ
- <u>IRenderableHolder instance</u>	เก็บค่าคงที่ของคลาส IRenderableHolder
+ <u>Image a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, a16, a17, a18, a19, a20, a21, a22, a23, a24, a25, a26</u>	รูปภาพของตัว container ของสัตว์ ที่มีตัวอักษรภาษาอังกฤษ 1 ตัวอยู่ในร่ม จำนวน 26 รูปครบถ้วน
+ <u>Image bg1</u>	รูปภาพพื้นหลังของเกม
+ <u>Image s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12</u>	รูปภาพของตัว container ของสัตว์ที่ไม่มีตัวอักษรใด ๆ ห้างหมัด 12 สถานะตั้งแต่ร่มกลางจนถึงร่มหนูและหายไป
+ <u>Image h1</u>	รูปนายพราน
+ <u>Image p1</u>	รูปร่มของนายพราน
+ <u>Image ground</u>	รูปพื้นของเกมใน GameStage

### 4.25.2 Constructor

+ IRenderableHolder()	<ul style="list-style-type: none"> <li>- ประกาศตัวแปร container ชื่นมาเป็น ArrayList&lt;IRenderable&gt;</li> <li>- ประกาศตัวแปร comparator ให้เปรียบเทียบค่า z ของ Object IRenderable 2 ตัว โดยที่คืนค่าเป็น 1 เมื่อตัวแรกมีค่า z มากกว่าตัวที่สอง และคืนที่ -1 เมื่อตัวแรกมีค่าน้อยกว่าตัวที่สอง และถ้าทั้งสองตัวเป็นนายพรานหรือสัตว์ให้เรียงจาก Z ไป A เพื่อให้เวลา draw() ตัวอักษรที่น้อยกว่าจะอยู่ด้านหน้ากว่า</li> </ul>
-----------------------	---

### 4.25.3 Methods

+ void add(IRenderable unit)	เพิ่ม Object IRenderable ลงใน container และทำการเรียงใหม่
+ void update()	เช็ค Object ใน container ที่ลະตัวว่า isDestroy() ถ้ามีสถานะเป็น true ให้ลบออกจาก

	container
+ IRenderableHolder getInstance()	คืนค่าตัวแปร instance
+ ArrayList<IRenderable> getContainer()	คืนค่าตัวแปร container เป็น ArrayList<IRenderable>
+ void clear()	ลบ Object ใน container ทุกตัวที่เป็นสมาชิกของ class Animals หรือ Hunters
- void loadResource()	LoadResource ของรูปภาพทักรูปในคลาส IRenderableHolder โดยใช้ ClassLoader