

1. Game Summary

PenBall เป็นเกมที่จะให้ผู้เล่นสวมบทบาทเป็นเพนกวิน ที่จะต้องผ่านเลเวลต่างๆ เพื่อที่จะพิชิตหัวหน้าของฝ่ายชั่วร้ายในชั่วโลกได้ แต่ในแต่ละเลเวลนั้น เพนกวินจะต้องเผชิญกับสัตว์ร้ายต่างๆ ที่คอยจะเข้ามาขัดขวางการเดินทางของเพนกวินในครั้งนี้ โดยผู้เล่นจะต้องควบคุมเพนกวินโจมตีคู่ต่อสู้ โดยเกมนี้จะใช้แค่เมาส์อย่างเดียวในการเล่น ในส่วนของหน้าต่างหลักของเกมจะมีสามตัวเลือกให้เลือก คือเล่นเกม ดูคะแนนสูงสุด และออกจากเกม



Figure 1: Main menu of the “PenBall” game.

โดยในการเล่นเกมนี้นี้ ผู้เล่นจะต้องใช้เมาส์ลากที่ตัวเพนกวิน เพื่อเปรียบเสมือนเป็นการยิงให้ตัวเพนกวินไปชนกับสัตว์ร้าย เป็นการโจมตี ส่วนสัตว์ร้ายต่างๆ ก็จะมีการยิงกระสุนออกมา เป็นการโจมตีเพนกวิน ซึ่งผู้เล่นก็ต้องคอยหลบหลีกไม่ให้โดน เพราะถ้าโดนยิง เลือดก็จะลด และถ้าเลือดลดจนหมด ก็ถือว่าจบเกม โดยเลือดของเพนกวินจะขึ้นอยู่กับตำแหน่งที่ด้านล่างของหน้าจอเกม และด้านล่างยังมีคะแนนของผู้เล่นอีกด้วย



Figure 2: Battle screen of the “PenBall” game.

ถ้าเพนกวีนสามารถโจมตีจนสัตว์ร้ายตายหมดในด่านนั้นๆได้ เพนกวีนก็จะสามารถที่จะไปเล่นในด่านต่อไป และทุกๆ 5 ด่านที่ผ่านไป ก็จะมีบอสหรือหัวหน้าออกมาขัดขวางเจ้าเพนกวีน โดยตัวหัวหน้าจะมีความสามารถพิเศษคือ สามารถยิงกระสุนได้หลายอันพร้อมๆกัน ก็จะทำให้การเล่นในด่านนี้ยากขึ้นไปอีก



Figure 3: Boss Battle screen of the “PenBall” game.

โดยเกมนี้จะให้ผู้เล่นได้เล่นไปเรื่อยๆ จนกว่าเพนกวินจะตาย ไม่มีการสิ้นสุดที่เลเวลใดเลเวลหนึ่ง และโดยหลังจากที่ฆ่าตัวหัวหน้าได้สำเร็จ เลเวลถัดไปก็จะเหมือนการเปลี่ยนไปโซนถัดไป ด้วยหน้าตาของแมฟในเกมที่เปลี่ยนไป

2. Implementation Details :

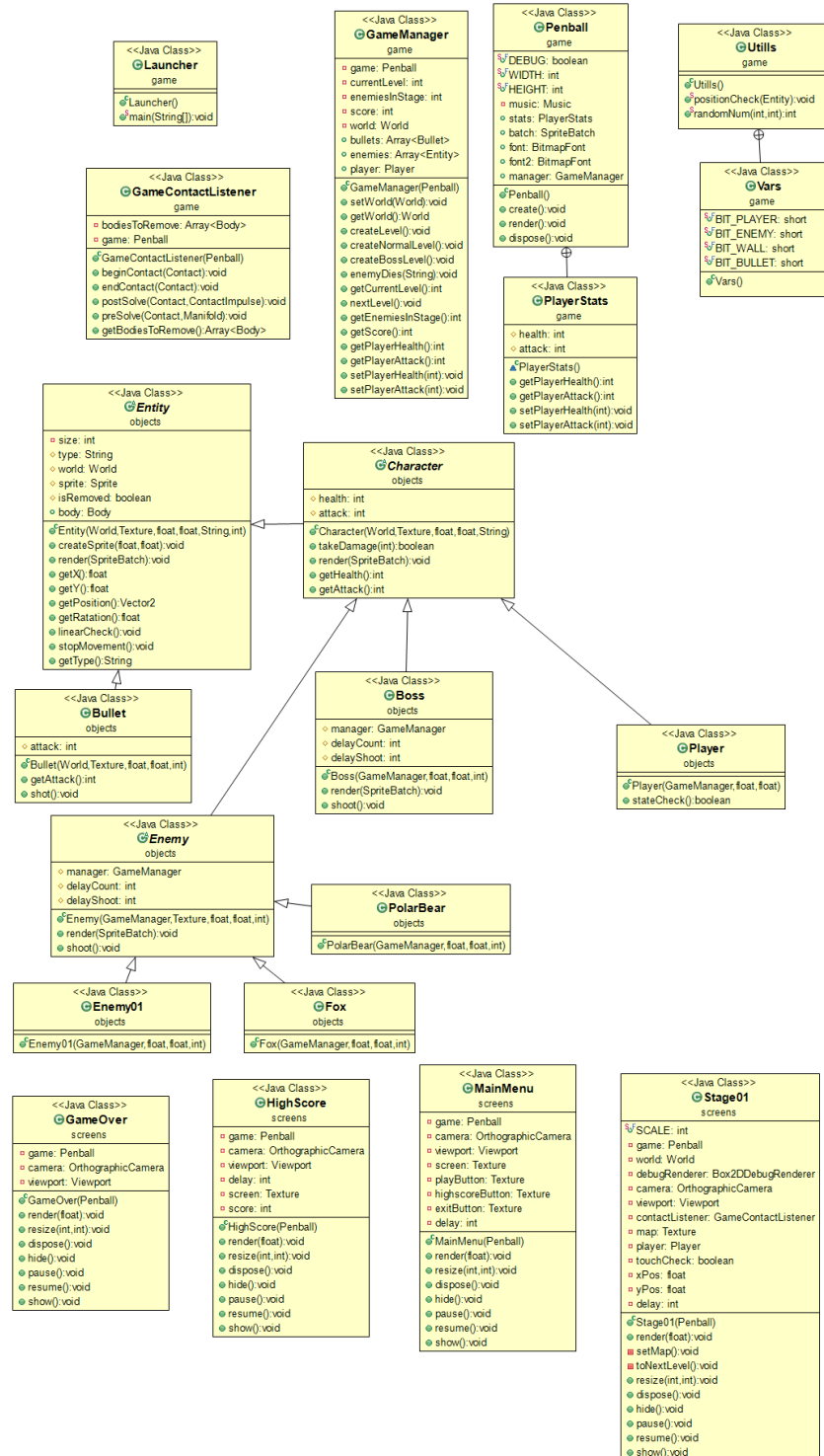


Figure 4: The UML diagram of the “PenBall” game.

* Noted that Access Modifier Notations can be listed as + (public), # (protected), - (private) and static will be underlined.

2.1 Class GameContactListener implements ContactListener

2.1.1 Field

- Array<Body> bodiesToRemove	อาเรย์ใช้เก็บ Body ของตัวละครที่ตายหลังจากการชน เพื่อส่งไปทำลายในคลาสหลัก
- Penball game	ตัวเกมหลัก

2.1.2 Constructor

+ GameContactListener(Penball game)	ContactListener เป็นคลาสของไลบรารี Box2d ที่จะรับตัวแปรเมื่อมี Fixture สองชิ้นชนกัน Constructor นี้รับตัวเกมหลักเพื่อนำค่ามาใช้
-------------------------------------	---

2.1.3 Method

+ void beginContact(Contact c)	ถ้า Fixture สองชิ้นชนกันให้เช็คว่าเป็นระหว่างผู้เล่นกับศัตรู หรือเปล่าถ้าใช่ให้ทำความเสียหายแก่ศัตรู และ ถ้าเป็นระหว่างผู้เล่นกับกระสุนให้ทำดาเมจกับผู้เล่นและลบกระสุน
+ void endContact(Contact c)	เป็นเมธอดที่ติดมากับ ContactListener ไม่ได้นำมาใช้
+ void postSolve(Contact c, ContactImpulse ci)	เป็นเมธอดที่ติดมากับ ContactListener ไม่ได้นำมาใช้
+ void preSolve(Contact c, Manifold m)	เป็นเมธอดที่ติดมากับ ContactListener ไม่ได้นำมาใช้
+ Array<Body> getBodiesToRemove()	Getter ของ getBodiesToRemove

2.2 Class GameManager

2.2.1 Field

- Penball game	ตัวเกมหลัก
- int currentLevel	Level ของเกมในขณะนี้
- int enemiesInStage	จำนวนศัตรูในด่าน
- int score	คะแนนของผู้เล่น
- World world	World จากไลบรารี Box2d ใช้เรนเดอร์ฟิสิกส์ในเกม
- PlayerStats stats	ค่าสถานะของผู้เล่น
+ Array<Entity> enemies	อาเรย์ที่ใช้เก็บศัตรูทั้งหมดในด่าน
+ Array<Entity> bullets	อาเรย์ที่ใช้เก็บข้อมูลกระสุนทั้งหมดในด่าน
+ Player player	ผู้เล่น

2.2.2 Constructor

+ GameManager(Penball game)	คลาส GameManager สร้างขึ้นมาเพื่อสร้างด่านให้ผู้เล่นในแต่ละเลเวล และยังเป็นตัวเก็บคะแนนของผู้เล่นอีก
-----------------------------	--

2.2.3 Method

+ void setWorld(World world)	Setter ของ world
+ World getWorld()	Getter ของ world
+ void createLevel()	ใช้สร้างด่าน โดยเรียก createNormalLevel() แต่ทุกๆ 5 ด่านจะสร้างด่านบอสแทน โดยใช้ createBossLevel()
+ void createNormalLevel()	สร้างด่านธรรมดาโดยเริ่มจากการสุ่มตำแหน่งของผู้เล่น แล้วสุ่มจำนวนศัตรูในด่าน และหลังจากนั้นจะสุ่มประเภทของศัตรูตัวนั้น แล้วสร้าง Entity นำไปเพิ่มใน enemies
+ void createBossLevel()	สร้างด่านบอส ให้บอสอยู่ตำแหน่งบนกลางของด่าน ให้ผู้เล่นอยู่ฝั่งตรงข้าม
+ void enemyDies(String type)	เช็คศัตรูประเภทใดตาย ให้เพิ่มแต้มไปตามประเภทนั้น แล้วลดจำนวนศัตรูในด่าน
+ int getCurrentLevel()	Getter ของ currentLevel
+ void nextLevel()	ไปด่านถัดไป ล้างอาเรย์ bullets และ enemies เพื่อใช้เก็บข้อมูลในด่านถัดไป เพิ่ม currentLevel อีกหนึ่ง
+ int getEnemiesInStage()	Getter ของ enemiesInStage
+ int getScore()	Getter ของ score
+ int getPlayerHealth()	Getter ของ health จาก game.stats
+ int getPlayerAttack()	Getter ของ attack จาก game.stats
+ void setPlayerHealth(int health)	Setter ของ health จาก game.stats
+ void setPlayerAttack(int attack)	Setter ของ attack จาก game.stats

2.3 Class Launcher

2.3.1 Method

+ void main(String[] args)	ใช้รันแอปพลิเคชัน
----------------------------	-------------------

2.4 Class Penball extends Game

2.4.1 Field

+ final boolean <u>DEBUG</u>	ใช้เปิด/ปิด debug renderer
+ final int <u>WIDTH</u>	ความกว้างของจอ
+ final int <u>HEIGHT</u>	ความสูงของจอ
+ PlayerStats stats	ใช้เก็บค่าสถานะของผู้เล่น ทำให้ค่าสถานะของผู้เล่นไม่เปลี่ยนเมื่อเปลี่ยนด่าน
+ SpriteBatch batch	SpriteBatch ของไลบรารี libgdx ใช้เพื่อวาด sprite ลงบนโปรแกรม
+ BitmapFont font	BitmapFont ของไลบรารี libgdx ใช้เพื่อวาดฟอนต์ลงบนโปรแกรม
+ BitmapFont font2;	BitmapFont ของไลบรารี libgdx ใช้เพื่อวาดฟอนต์ลงบนโปรแกรม
+ GameManager manager;	เป็นตัวจัดการเมื่อเปลี่ยนด่าน ทั้งใช้สร้างศัตรูในด่าน ส่งค่าระหว่างคลาส

2.4.2 Method

+ void create()	คลาส Game มาจากไลบรารี libgdx เมธอดนี้จะถูกเรียกจาก Launcher หลังจากนั้น Initializes ค่าในคลาส แล้วสั่งใช้เมธอด setScreen เพื่อเปลี่ยนจอไปยังจอเมนูในเกม
+ void render()	อัปเดตคลาส คล้ายกับ update() ของ javafx
+ void dispose()	เมธอดนี้จะถูกเรียกใช้เมื่อจบการทำงาน ใช้เพื่อทำลายทุกอย่างที่ทำลายได้เพื่อประหยัดพื้นที่

2.5 Class PlayerStats

2.5.1 Field

# int health	ค่าเลือดของผู้เล่น
# int attack	ค่าพลังโจมตีของผู้เล่น

2.5.2 Constructor

+ PlayerStats()	Initializes ค่าโดยตั้งต้นให้ health = 20 และ attack = 2
-----------------	---

2.5.3 Method

+ int getPlayerHealth()	Getter of health
+ int getPlayerAttack()	Getter of attack
+ void setPlayerHealth(int health)	Setter of health
+ void setPlayerAttack(int attack)	Setter of attack

2.6 Class Utils

2.6.1 Method

+ void <u>positionCheck</u> (Entity entity)	ใช้เพื่อเช็คตำแหน่งของ Entity เพื่อหาปัญหาต่างๆตอน Debug เกม
+ int <u>randomNum</u> (int low, int high)	ใช้สุ่มค่าตั้งแต่ low ถึง high

2.7 Class Vars

2.7.1 Field

+ final short <u>BIT_PLAYER</u>	ค่าเพื่อนำไปใช้ตั้งค่า Fixture ของ Player
+ final short <u>BIT_ENEMY</u>	ค่าเพื่อนำไปใช้ตั้งค่า Fixture ของ Enemy
+ final short <u>BIT_WALL</u>	ค่าเพื่อนำไปใช้ตั้งค่า Fixture ของ Wall
+ final short <u>BIT_BULLET</u>	ค่าเพื่อนำไปใช้ตั้งค่า Fixture ของ Bullet

2.8 Class Boss extends Character

2.8.1 Field

# GameManager manager	ใช้เพื่อจัดการข้อมูลในเกม นำมาเพื่อรับส่งข้อมูลต่างๆ
# int delayCount	ตัวนับค่าในตัวละครโจมตี ให้เพิ่ม 1 ทุกครั้งที่ render
# int delayShoot	มีค่าเท่ากับ 300 เพื่อให้บอสโจมตีทุกๆ 5 วินาที

2.8.2 Constructor

+ Boss(GameManager manager, float x, float y, int level)	Initializes ค่าต่างๆ ให้ delayCount = 150 เพื่อที่จะให้บอสโจมตีครั้งแรกเร็วกว่าปกติ
--	---

2.8.3 Method

+ void render(SpriteBatch batch)	รับค่า SpriteBatch เพื่อส่งไป super.render แล้วเพิ่มค่า delayCount หนึ่ง ถ้าตัวละครยังไม่ถูกลบ (isRemoved เป็น false)
+ void shoot()	โจมตีผู้เล่น โดยที่จะโจมตี 6 ครั้ง ในทิศที่ต่างกัน เริ่มจากการสร้าง bullet แล้วนำไปใส่ในอาร์เรย์ bullets แล้วใช้เมธอด applyLinearImpulse ไปตามทิศเวเตอร์ของศัตรูถึงผู้เล่น (ได้จากเอาตำแหน่งของผู้เล่น - ตำแหน่งของศัตรู)

2.9 Class Bullet extends Entity

2.9.1 Field

# int attack	ค่าพลังโจมตีของตัวละครกระสุน
--------------	------------------------------

2.9.2 Constructor

+ Bullet(World world, Texture texture, float x, float y, int attack)	Initializes ค่าต่างๆ
--	----------------------

2.9.3 Method

+ int getAttack()	Getter of attack
+ void shot()	ใช้เมื่อยิงโดนผนังหรือผู้เล่น ให้ isRemoved เป็น true ทำเกมหยุดการเรนเดอร์กระสุนอันนี้

2.10 Class Character extends Entity

2.10.1 Field

# int health	ค่าเลือดของตัวละคร
# int attack	ค่าพลังโจมตีของตัวละคร

2.10.2 Constructor

+ Character(World world, Texture texture, float x, float y, String type)	Initializes ค่า
--	-----------------

2.10.3 Method

+ boolean takeDamage(int damage)	ลดเลือดตามดาเมจที่ได้รับ เลือดจะไม่น้อยกว่า 0 ถ้าเลือดเท่ากับ 0 ใน isRemoved เป็น true และ return true ไม่งั้นก็ return false
+ void render(SpriteBatch batch)	ส่งค่าไป render ใน Entity
+ int getHealth()	Getter of health
+ int getAttack()	Getter of attack

2.11 Class Enemy extends Character

2.11.1 Field

# GameManager manager	ใช้เพื่อจัดการข้อมูลในเกม นำมาเพื่อรับส่งข้อมูลต่างๆ
# int delayCount	ตัวนับค่าในตัวละครโจมตี ให้เพิ่ม 1 ทุกครั้งที่ render
# int delayShoot	มีไว้ใช้กำลังระยะเวลาดีเลย์การยิงของตัวละคร โดยทุกๆ 1 วินาที = 60

2.11.2 Constructor

+ Enemy(GameManager manager, Texture texture, float x, float y, int level)	Initializes ค่า
--	-----------------

2.11.3 Method

+ void render(SpriteBatch batch)	รับค่า spriteBatch เพื่อส่งไป super.render แล้วเพิ่มค่า delayCount หนึ่ง ถ้าตัวละครยังไม่ถูกลบ (isRemoved เป็น false)
+ void shoot()	โจมตีผู้เล่น เริ่มจากการสร้าง bullet แล้วนำไปใส่ในอาร์เรย์ bullets แล้วใช้เมธอด applyLinearImpulse ไปตามทิศเวเตอร์ของศัตรูถึงผู้เล่น (ได้จากเอาตำแหน่งของผู้เล่น - ตำแหน่งของศัตรู)

2.12 Class Enemy extends Character

2.12.1 Constructor

+ Enemy01(GameManager manager, float x, float y, int level)	Initializes ค่า
---	-----------------

2.13 Class Entity

2.13.1 Field

- int size	ขนาดของตัวละคร ใช้เพื่อเซตขนาดของ Fixture
# String type	ประเภทของตัวละคร
# World world	World จากไลบรารี Box2d ใช้เรนเดอร์ฟิสิกส์ในเกม
# Sprite sprite	Sprite จากไลบรารี libgdx ใช้เพื่อวาดตัวละคร
# boolean isRemoved	ใช้เช็คตัวละครถูกลบไปหรือยัง
+ Body body	Body จากไลบรารี Box2d ใช้เพื่อคำนวณค่าฟิสิกส์ของตัวละคร

2.13.2 Constructor

+ Entity(World world, Texture texture, float x, float y, String type, int size)	Initializes ค่า
---	-----------------

2.13.3 Method

+ void createSprite(float x, float y)	ใช้เพื่อการสร้างตัวละคร เริ่มจากสร้าง BodyDef มาเก็บค่าคุณสมบัติของ Body เมื่อตั้งค่าเสร็จแล้ว นำไปใส่ใน world ด้วยเมธอด createBody แล้วสร้าง Fixture ขึ้นมาเพื่อกำหนดรูปร่าง และกำหนดว่า Entity type ไหนคืออะไรด้วย categoryBits และสามารถชนกับ Entity ไหนได้บ้างด้วย maskBits
+ void render(SpriteBatch batch)	ถ้า isRemoved เป็น false ให้ใช้เมธอด draw ของ sprite เพื่อวาด sprite ลงบนจอ โดยให้ตำแหน่ง sprite นั้นตรงกับตำแหน่งของ Body (ยกเว้นว่า Entity นั้นเป็นบอส ให้เลื่อนตำแหน่งมาที่เท้าของ sprite แทน) โดเมนการเอาตำแหน่งสามารถทำได้โดยเมธอด getX, getY
+ float getX()	คืนค่าตำแหน่ง x ของ Body
+ float getY()	คืนค่าตำแหน่ง y ของ Body
+ Vector2 getPosition()	คืนค่าตำแหน่งเป็น vector ของ Body
+ float getRotation()	คืนค่าตำแหน่งที่เป็นมุมในการหมุนของ Body

+ void linearCheck()	ใช้เพื่อเช็คความเร็วของ Body ถึงกำหนดหรือยังถ้าถึงให้หยุดการเคลื่อนไหวกโดยใช้ stopMovement() ที่ให้หยุดการเคลื่อนที่ เพราะว่า บาง Body หลังถูกชนแล้วจะไม่หยุดเคลื่อนที่ แต่จะเคลื่อนที่อย่างช้าๆทำให้เปลืองทรัพยากรการประมวลผลเป็นอย่างมาก
+ void stopMovement()	ใช้เพื่อหยุดการเคลื่อนไหวกของ Body
+ String getType()	คืนค่าประเภทของ Entity นั้นๆ

2.14 Class Fox extends Enemy

2.14.1 Constructor

+ Fox(GameManager manager, float x, float y, int level)	Initializes ค่า
---	-----------------

2.15 Class Player extends Character

2.15.1 Constructor

+ Player(GameManager manager, float x, float y)	Initializes ค่า
---	-----------------

2.15.2 Method

+ boolean stateCheck()	เช็คค่าผู้เล่นเลือด 0 รีเปลา ถ้าถึงให้ return false แล้วหลังจากนั้นเช็คความเร็วด้วย linearCheck() แล้ว return true
------------------------	--

2.16 Class PolarBear extends Enemy

2.16.1 Constructor

+ PolarBear(GameManager manager, float x, float y, int level)	Initializes ค่า
---	-----------------

2.17 Class GameOver implements Screen

2.17.1 Field

- Penball game	ตัวเกมหลัก
----------------	------------

- OrthographicCamera camera	OrthographicCamera จาก libgdx ใช้เพื่อเรนเดอร์ตัวเกม
- Viewport viewport	Viewport จาก libgdx ใช้เพื่อปรับขนาดหน้าจอของตัวเกม เวลาย่อขยาย

2.17.2 Constructor

+ GameOver(Penball game)	Initializes ค่า
--------------------------	-----------------

2.17.3 Method

+ void render(float arg0)	เรนเดอร์หน้าจอ 60 ครั้ง/วินาที ทำงานคล้าย update() ของ javafx ภายในเมธอด จะทำสิ่งต่างๆดังนี้ - เคลียร์หน้าจอ - อัปเดตทกกล่อง - วาด text บนหน้าจอ ภายใน SpriteBatch - ให้ออกจากเกมเมื่อมีการคลิก
+ void resize(int arg0, int arg1)	จะสั่งทำงานเมื่อ ย่อ/ขยาย หน้าจอ ใช้ update viewport ทำให้เกมยังคงสัดส่วนเดิม
+ void dispose()	จะสั่งทำงานเมื่อจบการใช้งาน ภายในเมธอดนี้จะเซฟค่า score ที่เล่นมาลงใน highscore.txt ถ้าเซฟไม่ได้จะขึ้น Error แล้วปิดโปรแกรม

2.18 Class HighScore implements Screen

2.18.1 Field

- Penball game	ตัวเกมหลัก
- OrthographicCamera camera	OrthographicCamera จาก libgdx ใช้เพื่อเรนเดอร์ตัวเกม
- Viewport viewport	Viewport จาก libgdx ใช้เพื่อปรับขนาดหน้าจอของตัวเกม เวลาต่อขยาย
- int delay	ดีเลย์ไม่ให้หน้าจอเปลี่ยนเลยทันที
- Texture screen	Texture จาก libgdx ใช้โหลดภาพพื้นหลัง ทำงานคล้าย classLoader
- int score	คะแนนที่โหลดมาจากไฟล์ txt

2.18.2 Constructor

+ HighScore(Penball game)	Initializes ค่า และเปิดไฟล์ highscore.txt เพื่อหาค่าสูงสุด แล้วนำมาเก็บในตัวแปร score ถ้าเปิดหรือเซฟไม่ได้จะขึ้น Error แล้วปิดโปรแกรม
---------------------------	---

2.18.3 Method

+ void render(float arg0)	เรนเดอร์หน้าจอ 60 ครั้ง/วินาที ทำงานคล้าย update() ของ javafx ภายในเมธอด จะทำสิ่งต่างๆดังนี้ - เคลียร์หน้าจอ - อัปเดตเทกสตรู - วาด text บนหน้าจอ ภายใน SpriteBatch - ให้กลับไปหน้า MainMenu ถ้ามีการคลิก
+ void resize(int arg0, int arg1)	จะสั่งทำงานเมื่อ ย่อ/ขยาย หน้าจอ ใช้ update viewport ทำให้เกมยังคงสัดส่วนเดิม
+ void dispose()	จะสั่งทำงานเมื่อจบการใช้งาน ภายในเมธอดนี้จะเซฟค่า score ที่เล่นมาลงใน highscore.txt ถ้าเซฟไม่ได้จะขึ้น Error แล้วปิดโปรแกรม

2.19 Class MainMenu implements Screen

2.19.1 Field

- Penball game	ตัวเกมหลัก
- OrthographicCamera camera	OrthographicCamera จาก libgdx ใช้เพื่อเรนเดอร์ตัวเกม
- Viewport viewport	Viewport จาก libgdx ใช้เพื่อปรับขนาดหน้าจอของตัวเกม เวลาต่อขยาย
- Texture screen	Texture ของหน้าจอ
- Texture playButton	Texture ของปุ่ม Play

- Texture highscoreButton	Texture ของปุ่ม Highscore
- Texture exitButton	Texture ของปุ่ม Exit
- int delay	ดีเลย์ไม่ให้หน้าจอเปลี่ยนเลยทันที

2.19.2 Constructor

+ MainMenu(Penball game)	Initializes ค่า
--------------------------	-----------------

2.19.3 Method

+ void render(float arg0)	เรนเดอร์หน้าจอ 60 ครั้ง/วินาที ทำงานคล้าย update() ของ javafx ภายในเมธอด จะทำสิ่งต่าง ๆ ดังนี้ - เคลียร์หน้าจอ - อัปเดตกล้อง - วาดปุ่มบนหน้าจอ ภายใน SpriteBatch - เช็คควมมีการคลิกบนปุ่มไหม ถ้าคลิกปุ่ม play ให้เข้าสู่เกม ถ้าคลิก highscore ให้ไปหน้า highscore ถ้าคลิก exit ให้ออกจากเกม
+ void resize(int arg0, int arg1)	จะสั่งทำงานเมื่อ ย่อ/ขยาย หน้าจอ ใช้ update viewport ทำให้เกมยังคงสัดส่วนเดิม
+ void dispose()	ทำลายค่าที่สามารถทำลายได้เพื่อประหยัดทรัพยากร

2.20 Class Stage01 implements Screen

2.20.1 Field

+ final int <u>SCALE</u>	สเกลย่อขนาดที่ใช้กับไลบรารี Box2d
- Penball game	ตัวเกมหลัก
- World world	World จากไลบรารี Box2d ใช้เรนเดอร์ฟิสิกส์ในเกม
- Box2DDebugRenderer debugRenderer	ใช้เพื่อเรนเดอร์ให้เห็นขอบของ Fixture จะเปิดเมื่อ DEBUG = true เท่านั้น
- OrthographicCamera camera	OrthographicCamera จาก libgdx ใช้เพื่อเรนเดอร์ตัวเกม

- Viewport viewport	Viewport จาก libgdx ใช้เพื่อปรับขนาดหน้าจอของตัวเกม เวลาต่อขยาย
- GameContactListener contactListener	ใช้เพื่อรับค่าเมื่อมีการชนกันของ Fixture เกิดขึ้น
- Texture map	Texture ของ map นั้น
- Player player	ผู้เล่น
- boolean touchCheck	ใช้เช็คว่าการคลิกบนหน้าจออยู่หรือเปล่า
- float xPos	ใช้เก็บตำแหน่งแกน x ของเมาส์
- float yPos	ใช้เก็บตำแหน่งแกน y ของเมาส์
- int delay	ดีเลย์ไม่ให้หน้าจอเปลี่ยนเลย์ทันทันที

2.20.2 Constructor

+ Stage01(Penball game)	Initializes ค่า และ setMap() เพื่อตั้งภาพพื้นหลังของเกม และสร้างผนังของห้องโดยการสร้าง Body และ Fixture 4 ข้าง
-------------------------	--

2.20.3 Method

+ void render(float arg0)	<p>เรนเดอร์หน้าจอ 60 ครั้ง/วินาที ทำงานคล้าย update() ของ javafx ภายในเมธอด จะทำสิ่งต่างๆดังนี้</p> <ul style="list-style-type: none"> - เคลียร์หน้าจอ - อัปเดตทล้อง - เช็คว่าการคลิกที่ตัวผู้เล่นหรือเปล่า ถ้ามีการคลิกที่ตัวผู้เล่น แล้วปล่อยในจุดที่ห่างออกมา ให้ผู้เล่นเคลื่อนที่ไปทิศทางตรงกันข้ามของทิศทางของเมาส์ - วาดพื้นหลังใน SpriteBatch - วาดตัวละครทั้งหมดในเกมใน SpriteBatch - โดยเริ่มจากผู้เล่น ศัตรู (จาก อาเรย์ enemies ของ GameManager) และ กระสุน (จาก อาเรย์ bullets ของ GameManager)
---------------------------	---

	<ul style="list-style-type: none"> - เช็คค่า health ของผู้เล่นเหลือ 0 หรือเปล่า ถ้าเหลือ 0 ให้ไปฉาก GameOver - เช็คค่าศัตรูในด้านเหลือ 0 หรือเปล่า ถ้าใช่ให้ไปด้านถัดไป
+ void resize(int arg0, int arg1)	จะสั่งทำงานเมื่อ ย่อ/ขยาย หน้าจอ ใช้ update viewport ทำให้เกมยังคงสัดส่วนเดิม
+ void dispose()	จะสั่งทำงานเมื่อจบการใช้งาน ภายในเมธอดนี้จะเซฟค่า score ที่เล่นมาลงใน highscore.txt ถ้าเซฟไม่ได้จะขึ้น Error แล้วปิดโปรแกรม
- void toNextLevel()	บอก GameManager ให้ไปด้านถัดไป และอัปเดตค่า health กับ attack ในคลาสเกมหลักเพื่อนำไปใช้ในด้านถัดไป ใช้เมธอด setScreen เพื่อเปลี่ยนด่าน
- void setMap()	ตั้งรูปพื้นหลังของเกม โดยให้สีพื้นหลังเปลี่ยนทุกๆ 5 เลเวล