KingKong Documentation

Created by

Jiratchaya Puvapaisankit 6230072521

Cholakorn Maturosjamnan 6230096621

2110215 Programming Methodology

Semester 1 Year 2010

Chulalongkorn University

Hold My Ball

Introduction

You have the mission from NASA to hold the ball that has the highest value in the universe by bouncing it. If you let the ball drop, the ground will break. And the brick that ball break can earn money!

Rules

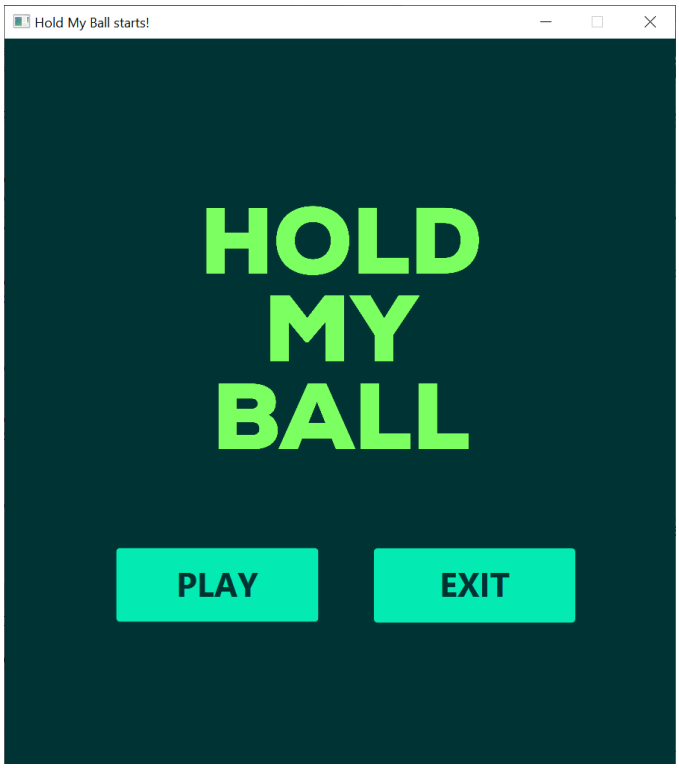
The player can move plane left and right to hold the ball that would bounce when the ball hit plane, brick or wall. Each time the ball hit plane, the ball will random its speed. Each time the ball hits the brick, brick will break and increase your score by 10. There are 6 types of bricks.

1. Normal brick
2. Extend brick – when the ball hit, the plane will extend 20 px.
3. Shrink brick – when the ball hit, the plane will shrink 20 px.
4. Multiple brick – when the ball hit, score will multiple 2 for 5 times break.
5. Strong brick – it breaks when the ball hit it 2 times.
6. Shooting brick – when the ball hit, the ball will change to the super red ball and it can through any brick until it hit the ceiling 1 time.

You will win when the ball breaks all bricks! and the game over when you cannot hold the ball.

Example

Start Scene



• When you are ready you click PLAY button.

Brick

This is a extend brick.

This is a shrink brick.

This is a multiple brick.

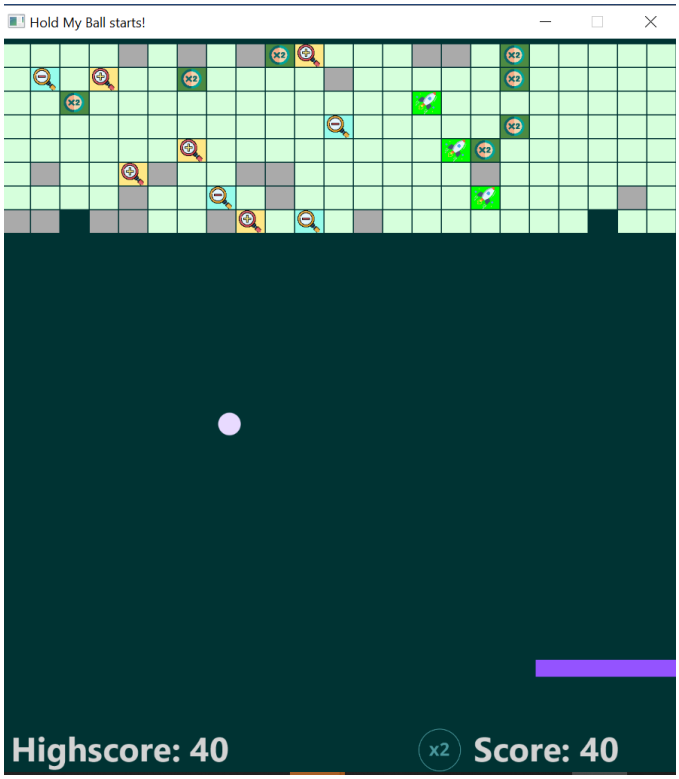
This is a normal brick.

This is a shooting brick.

This is a strong brick.

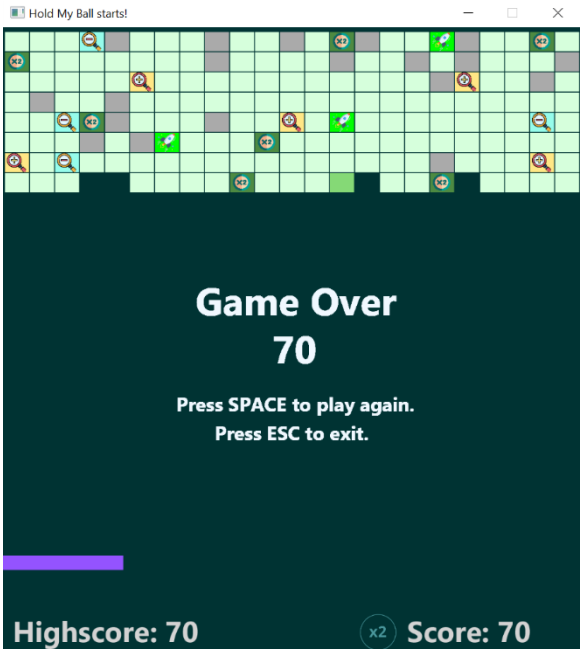
This is a strong brick after hit 1 time.

Game Scene

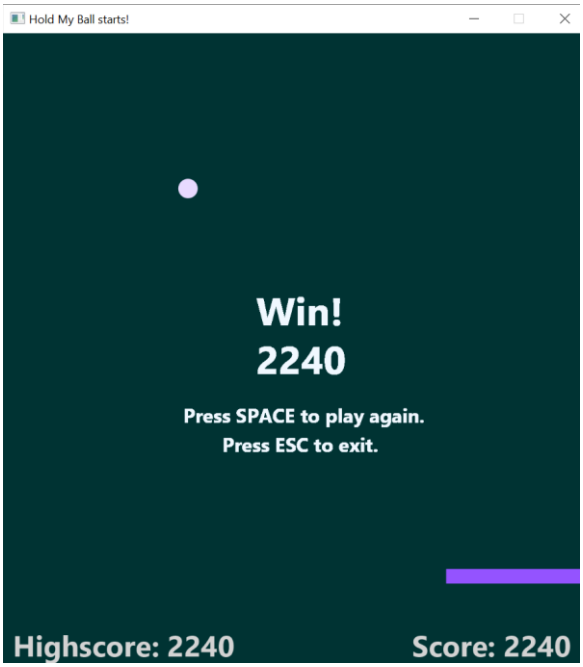


- You must hold your ball by keying LEFT or RIGHT to move plane and you will win when you can break all bricks.
- The different type of bricks has the different sounds when the ball hit it.
- Your score shows in the right bottom of the window.
- Your high score shows in the left bottom of the window.
- When you hit a multiple brick, **the icon x2** appear near score for 5 times the brick break.

End Scene



Game Over



Win!

- Score will show in the middle of window
- You can play again by pressing space bar.
- You can exit by pressing ESC

1. Package drawing

1.1 Class GameScene extends Scene

1.1.1 Fields

+ GraphicsContext gc	Graphics Context Field

1.1.2 Constructor

+ GameScene(StackPane stackPane)	- call super constructor. - create new canvas with width 600 and height 650, initialize gc with it and add it in stackpane. - set the Background with #003333 Color. - call setKey().

1.1.3 Methods

+ void setKey()	If onGame, do • If key left, call Logic.getPlane().moveLeft(), set LeftClick to true and set RightClick to false. • If key right, call Logic.getPlane().moveRight() , set LeftClick to false and set RightClick to betrue. If not, • If key space bar, set onStart and onEnd to false, set onGame to true. • If key esc, set onStart, onGame and onEnd to false.
+ void painComponent()	If onGame, do • Clear the window. • Call draw() for all Entity. • Draw text “Highscore: ” + String.valueOf(Logic.getHighScore()) in position (10,640) and "Score: " + String.valueOf(Logic.getScore()) in

	position (420,640) with Swiss 721 Black Condensed BT, **BOLD** FontWeight, size 30 and set it to be filled with LIGHTGREY color. If onEnd, do • If win, Draw text “Win!” in position (260,300) with “Swiss 721 Black Condensed BT”, **BOLD** FontWeight, size 40 and set it to be filled with ALICEBLUE color. • if not win, Draw text “Game Over” in position (200,300) with “Swiss 721 Black Condensed BT”, **BOLD** FontWeight, size 40 and set it to be filled with ALICEBLUE color. • Draw text "Press SPACE to play again." in position (185,400) and "Press ESC to exit." in position (225,430) with “Swiss 721 Black Condensed BT”, size 20 and set it to be filled with ALICEBLUE color. • Draw text score get in that game with y position is 350 and x position adapt by value of score - If the score less than 10 x position is 290. - If the score less than 100 x position is 280. - If the score less than 1000 x position is 270 - If the score exceed 1000 x position is 260.

1.2 Class StrartScene extends Scene

1.2.1 Constructor

+ StartScene(VBox vbox)	- Call super constructor. - Create an image (“HMB.PNG”) and set it fit width and height to 250. - Create **Exit Button** with the text “Exit” • Set it preferred width to 180. • Set preferred height 40. • Set style to - Text fill: #003333 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #03EAB3 - Font size 30 • Set on mouse clicked to set onGame, onStart and onEnd to false. • Set on mouse moved to set style - Text fill: #D30316 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #A2EAE1 - Font size 30 • Set on mouse exited to set style - Text fill: #003333 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #03EAB3 - Font size 30

	- Create **Play Button** with the text "Play" • Set it preferred width to 180. • Set preferred height 40. • Set style to - Text fill: #003333 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #03EAB3 - Font size 30 • Set on mouse clicked to call set onStart, onEnd to false and set onGame to true. • Set on mouse moved to set style - Text fill: #D30316 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #A2EAE1 - Font size 30 • Set on mouse exited to set style - Text fill: #003333 - Font-weight: **bold** - font-family: "Swiss 721 Black Condensed BT" - Background color: #03EAB3 - Font size 30 - Create HBox and add Play button and Exit Button. - Set HBox's margin from Play Button with Insets(5, 50, 0, 20) - Set HBox's margin from Exit Button with Insets(5, 0, 0, 0) - Add the image and the HBox in vBox - Set vBox's the inset padding of 80. - Set vBox's fill width to true.

	- Set vBox's spacing to 70. - Set the alignment of vBox to the CENTER. - Set background of vBox with #003333 color

2. Package Logic

2.1 Abstract Class Entity implements IRenderable

2.1.1 Fields

- double x	Position x of the entity
- double y	Position y of the entity
- boolean destroyed	A Boolean to keep track if the entity has been destroy or not.
- int width	Width of the entity
- int height	Height of the entity
- Color color	Color of the entity

2.1.2 Constructor

+ Entity(int x, int y)	Initialize x and y Set destroyed to false

2.1.3 Methods

setter/getter of all fields	

2.2 Class Ball extends Entity

2.2.1 Fields

- Double xSpeed	Speed x of the ball
- Double ySpeed	Speed y of the ball

2.2.2 Constructor

+ Ball(int x, int y)	- Call super constructor. - Set xSpeed to 8 and ySpeed to 7.2. - Set width and height to 20. - Set color to #E8DAFF Color.

2.2.3 Method

+ draw(GraphicsContext gc)	Draw Oval in start position (*x*,*y*) with *width* and *height* and set it filled with its Color
setter/getter of all fields	

2.3 Class Plane extends Entity.

2.3.1 Fields

- int planeSpeed	Speed of the plane

2.3.2 Constructor

+ Plane(int x, int y, int width)	-Call super constructor - Set *width* to width. - Set height to 15. - Set color to #9453FF Color. - Set planeSpeed to 0.

2.3.3 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in start position (x,y) with width *width* and height 15 and set it filled with *color*. - Increase x with plane speed.
+ moveLeft()	If x is less than 10, set plane speed to 0. If not, set plane speed to -10.
+ moveRight()	- If x more than 600 – plane's width – 10, set plane speed to 0. - If not, set plane speed to 10.
setter/getter of all fields	

2.4 Class Brick extends Entity

2.4.1 Constructor

+ Brick(int x, int y)	- Call super constructor. - Set width to 25. - Set height to 20.

2.4.2 Method

+ abstract void ability()	Use each Brick's type unique ability.

2.5 Class NormalBrick extends Brick

2.5.1 Constructor

+ NormalBrick(int x, int y)	- Call super constructor. - Set Color to #D6FFDC Color.

2.5.2 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in its start position (x, y) with its *width* and *height* and set it filled with its *color*.
+ void ability()	- if wallCount = 0, play sound *normalBrickBreak* in RenderableHolder - Set Destroyed to true.

2.6 Class ExtendBrick extends Brick

2.6.1 Constructor

+ ExtendBrick(int x, int y)	- Call super constructor. - Set Color to #FFE785 Color.

2.6.2 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in start position its (x,y) with its *width* and *height* and set it filled with its *color*. - Draw Image *extendPic* from RenderableHolder in start position (x+2, y) with width 20 and height 20.

+ void ability()	- Create new plane with 20 increased from current plane and set it as current plane. - remove old plane from entities in RenderableHolder. - if isLeftClick, call Logic.getPlane().moveLeft() - if isRightClick, call Logic.getPlane().moveRight() - if wallCount = 0, play sound *extendBrickBreak* in RenderableHolder - Set Destroyed to true.

2.7 Class ShrinkBrick extends Brick

2.7.1 Constructor

+ ShrinkBrick(int x, int y)	- Call super constructor. - Set Color to #96FCEB Color.

2.7.2 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in its start position (x, y) with its *width* and *height* and set it filled with its *color*. - Draw Image *shrinkPic* from RenderableHolder in start position (x+2, y) with width 20 and height 20.
+ void ability()	- Create new plane with 20 decreased from current plane and set it as current plane. - remove old plane from entities in RenderableHolder. - if isLeftClick, call Logic.getPlane().moveLeft() - if isRightClick, call Logic.getPlane().moveRight() - if wallCount = 0, play sound *shrinkBrickBreak* in RenderableHolder. - Set Destroyed to true.

2.8 Class MultipleBrick extends Brick

2.8.1 Constructor

+ MultipleBrick(int x, int y)	- Call super constructor. - Set Color to #4D8A42 Color.

2.8.2 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in its start position (x,y) with its *width* and *height* and set it filled with its *color*. - Draw Image *multiplePic* from RenderableHolder. in start position (x+2, y) with width 20 and height 20.
+ void ability()	- Increase 5 to multiple time. - if wall count = 0, play sound *multipleBrickBreak* in RenderableHolder - Set Destroyed to true.

2.9 Class ShootingBrick extends Brick

2.9.1 Constructor

+ ShootingBrick(int x, int y)	- Call super constructor. - Set Color to #00FF00 Color.

2.9.2 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in its start position (x,y) with its *width* and *height* and set it filled with its *color*. - Draw Image *shootingPic* from RenderableHolder. in start position (x+2, y) with width 20 and height 20.
+ void ability()	- Increase wall count 1. - play sound *shootingBrickBreak* in RenderableHolder - Set Destroyed to true.

2.10 Class StrongBrick extends Brick

2.10.1 Field

- int hp	Durability of the Brick

2.10.2 Constructor

+ StrongBrick(int x, int y)	- Call super constructor. - Set Color to #AAAAAA Color.

2.10.3 Methods

+ draw(GraphicsContext gc)	- Draw Rectangle in start position (x, y) with *width* and *height* and set it filled with *color*.
+ void ability()	if hp is more than 0, do • Decrease hp by 1. • Set Color to #85B976 Color. • play sound *strongBrickBreak* in RenderableHolder If wall count = 0. If not, play sound normal*BrickBreak* in RenderableHolder If wall count = 0.

2.11 Class GameMap

2.11.1 Fields

- static ArrayList<Integer> brickType	An ArrayList contains type of each brick that randomly generate by numberGenerate.

2.11.2 Methods

+ static void numberGenerate()	Initialize brickType and randomly generate 0,1,2,3,4,5 to each slot in brickType with size 184.
+ static void genMap()	Generate Map by - Called numberGenerate() to make brickType .

	- Generate each brick by called generateBrick(int slot ,int brickType) which slot are the index of ArrayList and brickType is data in each index.
+ static Brick generateBrick(int slot, int brickType)	Generate Brick in position x = 2 + 26 * (slot % 23) and y = 5 + 21 * (slot / 23) and select the type of Brick switch the brickType by 0 : NormalBrick 1 : ExtendBrick 2 : ShrinkBrick 3 : MultipleBrick 4 : ShootingBrick 5 : StrongBrick

2.12 Class Logic

2.12.1 Fields

- static Scene currentScene	Current Scene
- static boolean isWin = false	- A boolean to keep track if is win or not. - set isWin to false.
- static Ball ball	Ball of current game.
- static Plane plane	Plane of current game.
- static int score	Current score in that game.
- static int highScore	High Score at all time the application are still running
- static boolean leftClick = false	- A boolean to keep track if Left Click is being clicked or not. - set leftClick to false.
- static boolean rightClick = false	- A boolean to keep track if Right Click is being clicked or not. - set rightClick to false.
- static int multipleTime	Amount of time that next brick break and get the multiple score.
- static int wallCount	Time amount that the ball hit the ceiling.
- static int brickBreak	Amount of brick break

2.12.2 Constructor

+ Logic()	Call resetLogic().

2.12.3 Methods

+ static boolean checkIntersect(Entity entity1, Entity entity2)	- Create entity1 and entity2 Rectangle in its start position (*x*, *y*) with its *width* and *height*. - if Rectangle of them intersect, do • Set ball's y speed to be opposite if wall count = 0. • Return true. - if not, return false.
+ static void logicUpdate()	- Create ball Rectangle in its start position (*x*, *y*) with its *width* and *height* - Create Plane Rectangle in its start position (*x*, *y*) with its *width* and height 1. - Create Left Plane Rectangle with start position (*x*, *y*+1) with width 1 and height *height* – 1. - Create Right Plane Rectangle with start position (*x*+*width*–1, *y*+1) with width 1 and *height*-1. - If ball Rectangle and Plane Rectangle intersect, random the xSpeed between (5,10) in the same direction and random the ySpeed between (5,10) in opposite direction. - If ball Rectangle and Left Plane Rectangle intersect, random the xSpeed between (5,10) to left direction and random the ySpeed between (5,10) in opposite direction. - If ball Rectangle and Right Plane Rectangle intersect, random the xSpeed between (5,10) to right

	direction and random the ySpeed between (5,10) in opposite direction. - If ball ‘s x position more than 570 throw new BallOutofBoundException (1) - If ball ‘s x position lower than 0 throw new BallOutofBoundException(2) - If ball ‘s y position lower than 0 throw new BallOutofBoundException(3) - If ball ‘s y position more than 650 throw new BallOutofBoundException (4) - if number of brickBreak = 184, do • Set the isWin to true. • Play sound *gameOver* in RenderableHolder. • Set onGame and onStart to false and set onEnd to true. - If wallCount less than 0, set it to 0. - Increase the ball ‘s x position by the ball ‘s xSpeed. - Increase the ball ‘s y position by the ball ‘s ySpeed. - If the plane x position getting less than 0 or more than 600 throw new PlaneMoveFailedException - Find the Brick inside Entities which intersect with the ball by call checkIntersect method. If found, do • Using the Brick ability. • Check if the Brick was destroyed, if true do - Check if multipleTime is more than 0, do - Increase the score by 10. - Decrease the multiple Time count by 1.

	- Increase the score by 10. -Increase brickBreak by 1. - If the score more than highScore, set it as new highScore. • Break the loop so the Brick cannot be hit more than 1 Brick in 1 frame.
1+ static void addEntity(Entity entity)	Add entity in to entities that's in RenderableHolder.
+ static void resetLogic()	- Clear entities in RenderableHolder. - Set Score, Multiple Time, Wall Count and Brick Break to 0. - initialize ball and plane and add it in to entities. - Set Win, Left Click and Right Click to false. - Generate Map by calling GameMap. genmap()
getter/setter of all fields	

3. Package main

3.1 Class Main extends Application

3.1.1 Fields

+ static final int WIDTH	Width of the window = 600
+ static final int HEIGHT	Height of the window = 650
+ static boolean onStart	- A boolean to keep track if start scene has been started or not. - Set onStart to true
+ static boolean onGame	- A boolean to keep track if game has been started or not. - Set onGame to false
+ static boolean onEnd	- A boolean to keep track if game has been ended or not. - Set onEnd to false
- GameScene startScene	Start Scene

- GameScene gameScene	Game Scene

3.1.2 Methods

+ void start(Stage primaryStage) throws PlaneMoveFailedException, BallOutofBoundException	- Create new VBox() and initialize startScene with it. - Create new StackPane() and initalize gameScene with it. - Set primaryStage to not resizable. - Set primaryStage with title "Hold My Ball starts!". - Show the primaryStage. - Create AnimationTimer • If onStart, set currentScene to be startScene and play sound *startScene* in RenderableHolder. • If onGame, stop sound *startScene* and *gameOver,* set currentScene to be gameScene and play sound game*Scene* in RenderableHolder. - Try Logic.logicUpdate() which could catch PlaneMoveFailedException and BallOutofBoundException but do nothing. - Update *instance*. • If onEnd, call gameScene.paintComponent • If not onGame,not onStart and onEnd, close primaryStage.
static void main(String[] args)	main application

4. Package sharedObject

4.1 interface IRenderable

4.1.1 Method

+ void draw(GraphicsContext gc)	Draw Graphics

4.2 Class RenderableHolder

4.2.1 Fields

- static final RenderableHolder instance	Initialize instance
- ArrayList<Entity> entities	ArrayList which contain all entity in Game Scene
+ static AudioClip normalBrickBreak	Sound play when Normal Brick Break
+ static AudioClip extendBrickBreak	Sound play when Extend Brick Break
+ static AudioClip multipleBrickBreak	Sound play when Multiple Brick Break
+ static AudioClip shootingBrickBreak	Sound play when Shooting Brick Break
+ static AudioClip shrinkBrickBreak	Sound play when Shrink Brick Break
+ static AudioClip strongBrickBreak	Sound play when Strong Brick Break
+ static AudioClip gameOver	Sound play when Game Over
+ static AudioClip startScene	Sound play in Start window
+ static Image shrinkPic	Image in Shrink Brick
+ static Image extendPic	Image in Extend Brick
+ static Image multiplePic	Image in Multiple Brick
+ static Image shootingPic	Image in Shooting Brick

4.2.2 Constructor

+ RenderableHolder()	Initalize entities

4.2.3 Methods

+ static void loadResource()	- Initialize normalBrickBreak with "NormalBrickBreak.wav" - Initialize extendBrickBreak with " ExtendBrickBreak.wav" - Initialize multipleBrickBreak with " MultipleBrickBreak.wav" - Initialize shrinkBrickBreak with " ShrinkBrickBreak.wav" - Initialize shootingBrickBreak with " ShootingBrickBreak.wav" - Initialize strongBrickBreak with " StrongBrickBreak.wav"

	- Initialize gameOver with " GameOver.wav" - Initialize startScene with " StartScene.wav" - Initialize shrinkPic with " shrink.png" - Initialize extendPic with " extend.png" - Initialize multiplePic with " multiple.png" - Initialize shootingPic with " Rocket.png"
+ void update()	Remove all destroyed entity in entities
+ static RenderableHolder getInstance	Get instance
+ ArrayList<Entity> getEntities()	Get entities

5. Package Exception

5.1 Class BallOutofBoundException extends Exception

5.1.1 Constructor

+ BallOutofBound(int bound)	Change the speed direction for the ball by case • 1 : Set the ball x position to 560 and reverse the xSpeed. • 2 : Set the ball x position to 0 and reverse the xSpeed. • 3 : Set the ball y position to 10 and reverse the ySpeed. • 4 : The game lose ,do this below - Set onGame to false. - Set onStart to false. - Set onEnd to true. - Play gameOver sound in RenderableHolder.

5.2 Class PlaneMoveFailedException extends Exception

5.2.1 Constructor

+PlaneMoveFailedException	- Set current Plane's speed to 0. - Set leftClick to false. - Set rightClick to false. - Do this below by case • If the plane x position less than 0 set it to 0. • If the plane x position more than 600-plane's width set the x position to 600-plane's width.