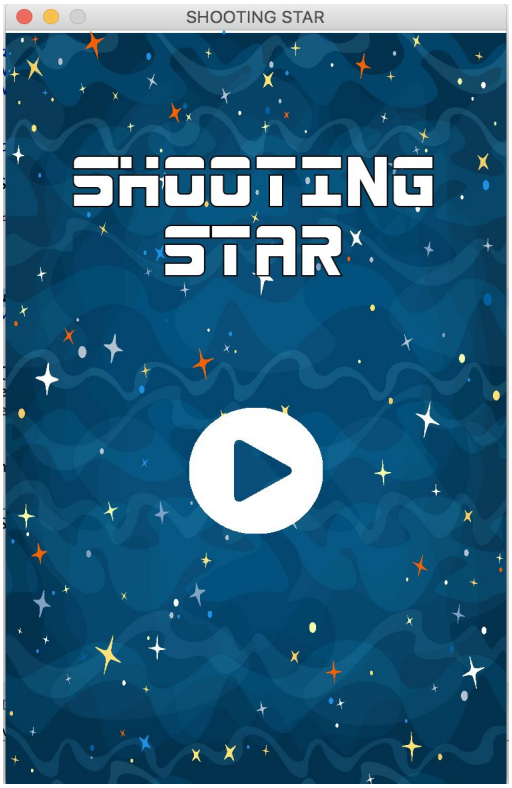
SHOOTING STAR

By Mild1
Natawat Kwanpum 5931017021
Warinthorn Chommanee 5931058821

About Game

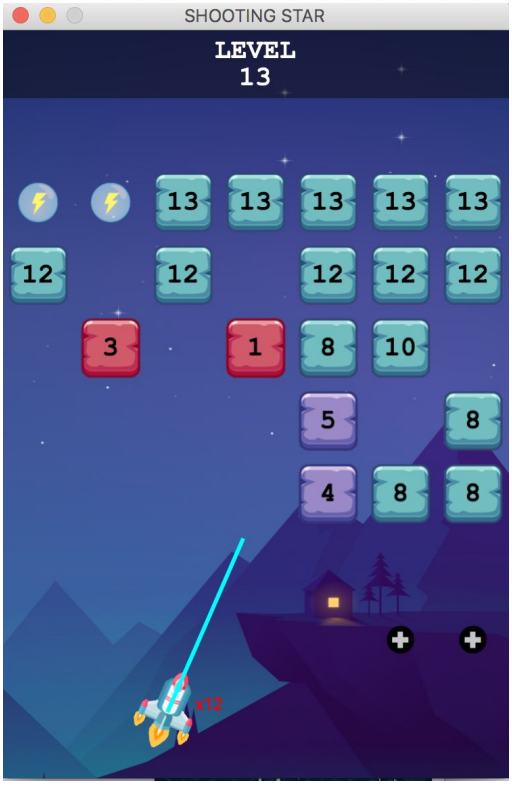
“Shooting Star” is a block breaker endless game made from JAVA FX . We are captain of teams and our mission is protect our earth. Purality block is coming to earth. We have to shoot it down. And this game have many item to help you. More level more difficult more fun. We will shoot ‘em all before it fall to earth. SHOOT ‘EM ALL!!!!!!!

How To Play



Main Menu

Click play button to start your game



Game Scene

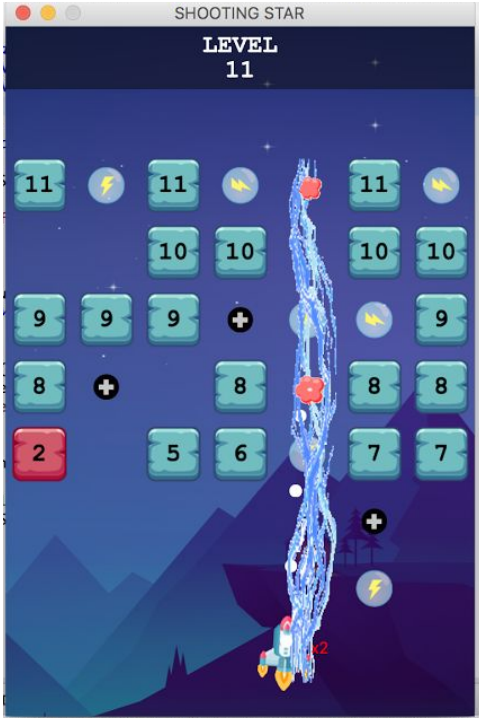
Top bar show the level you are
Each blocks will have number show life.
Shooter show number of bullet you have

Controlling

Use mouse to aim
Left Click to shoot
Right Click to retrieve

Item

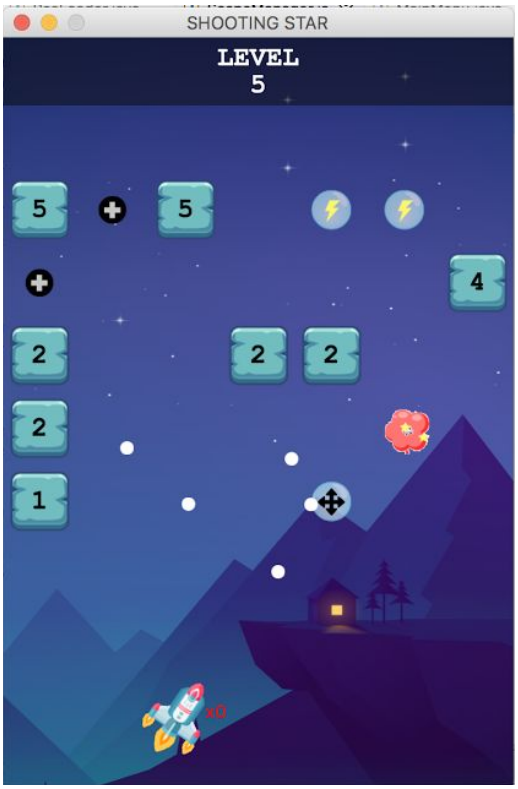
Ball Plus : Increase max capacity of you weapon

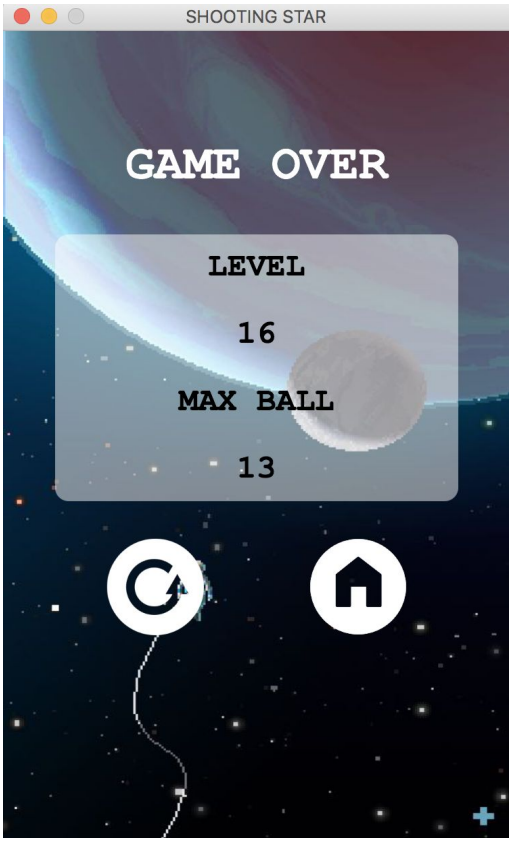


Lightning (Vectical) :
It’s will help you to
destroy block in that column

Lightning (Horizontal) :
Powerful Lightning destroy
all block in that row!!

Random Reflector :
Amplify your bullet damage
any block in the area.

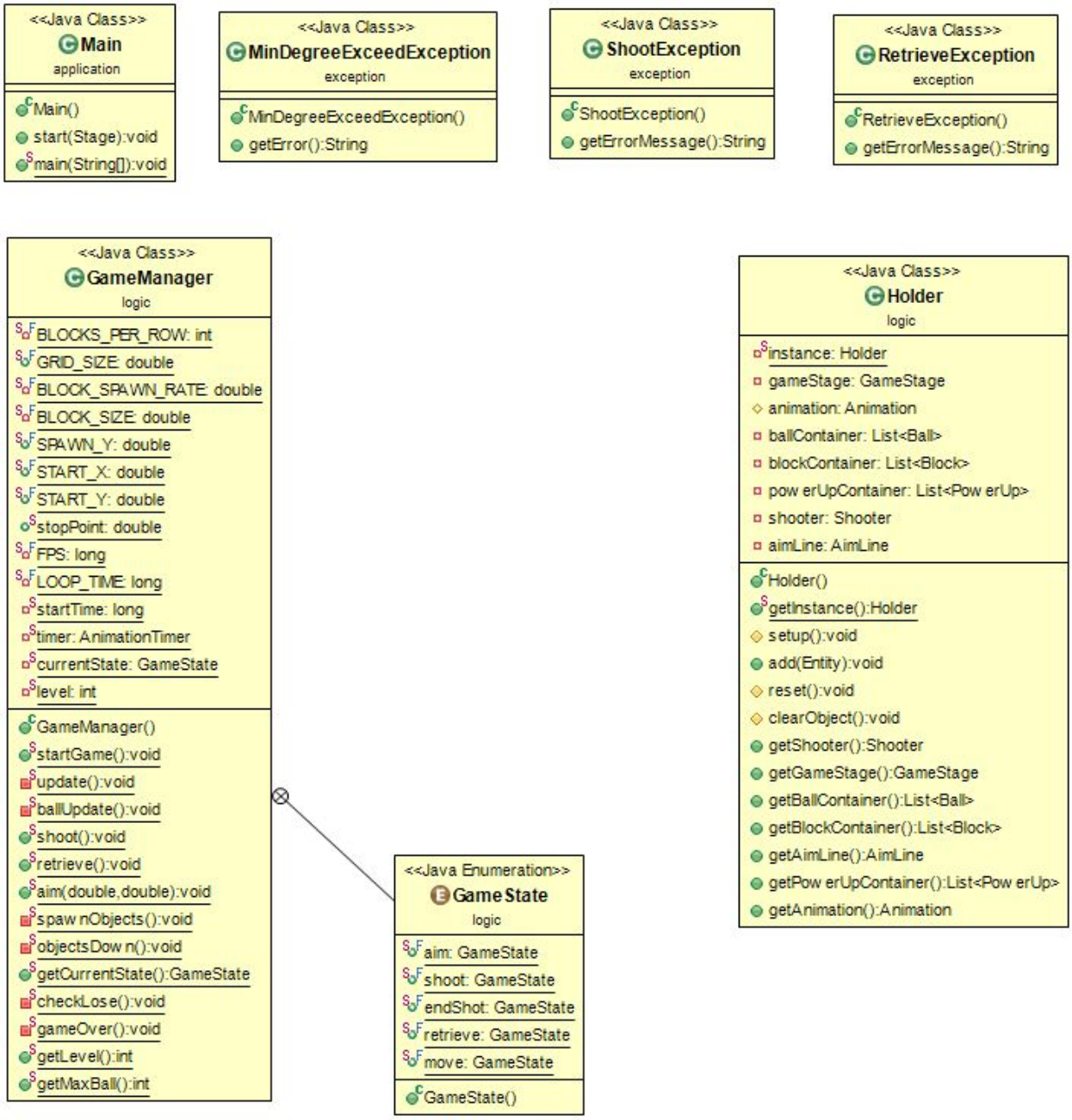


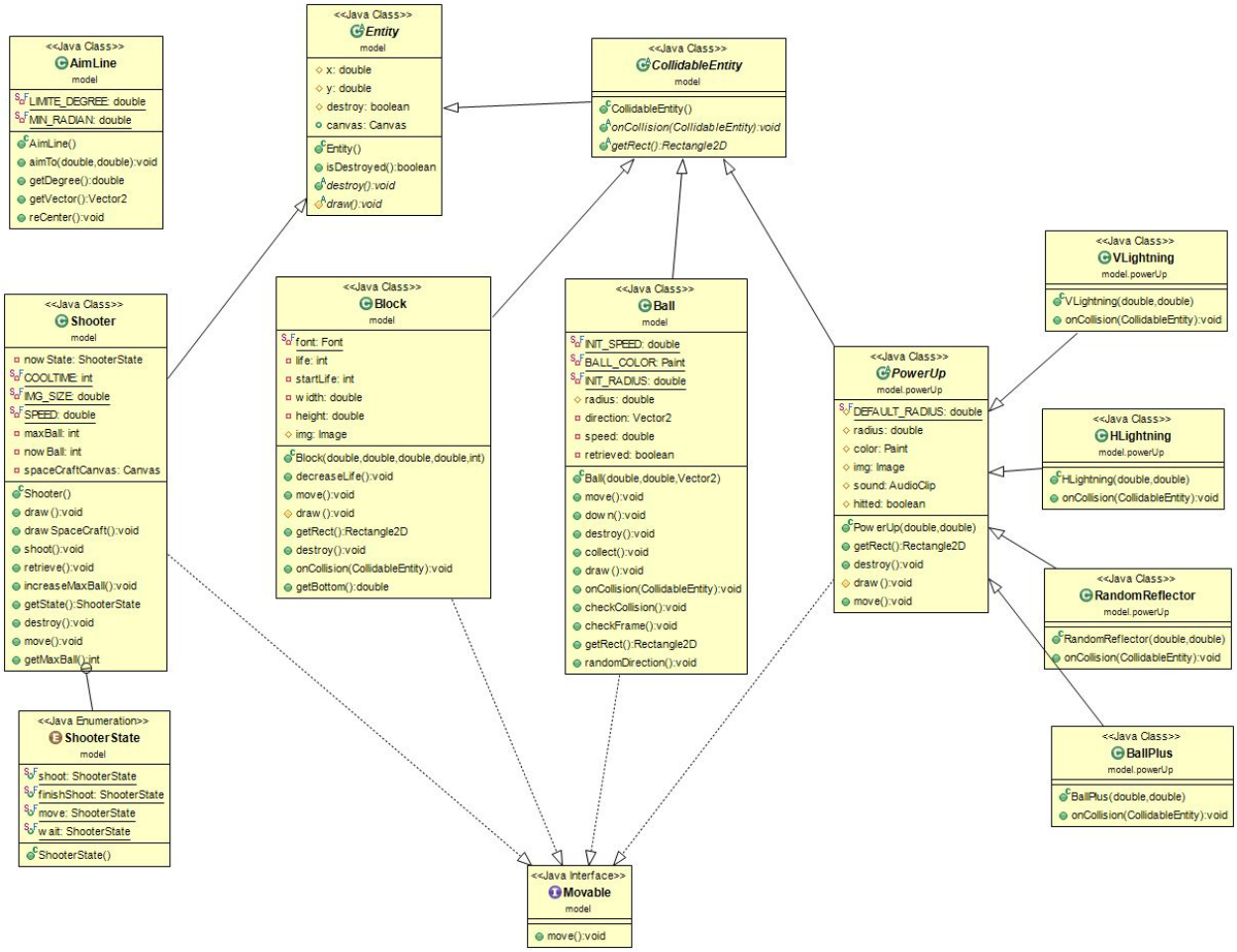


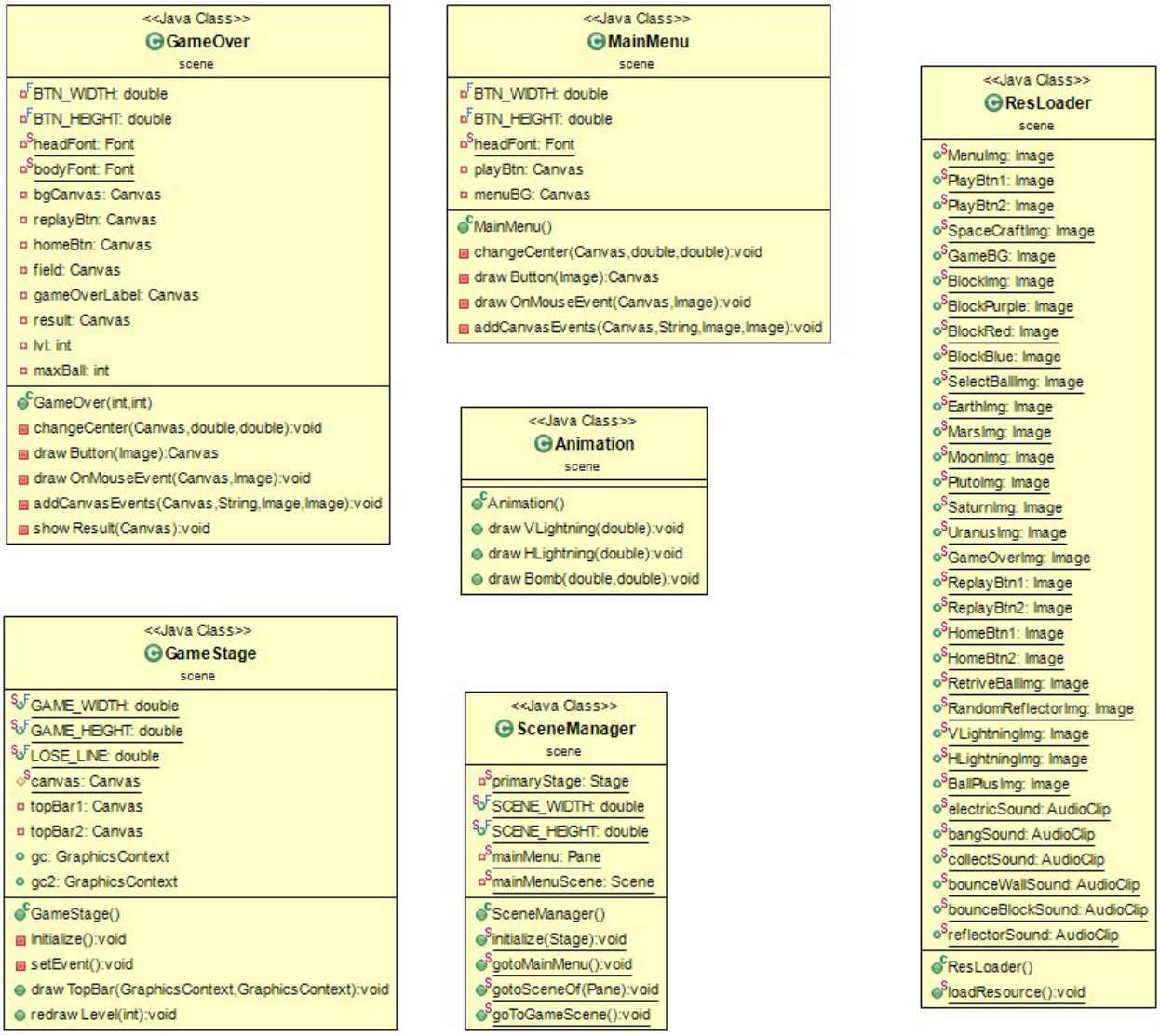
Game over Scene

You can try again or go back home

Implementation Details




<<Java Class>>
AimLine
model
LIMITE_DEGREE: double
MIN_RADIAN: double
AimLine()
aimTo(double,double):void
getDegree():double
getVector():Vector2
reCenter():void
<<Java Class>>
Entity
model
x: double
y: double
destroy: boolean
canvas: Canvas
Entity()
isDestroyed():boolean
destroy():void
draw():void
<<Java Class>>
CollidableEntity
model
CollidableEntity()
onCollision(CollidableEntity):void
getRect():Rectangle2D
<<Java Class>>
Shooter
model
now State: ShooterState
COOLTIME: int
IMG_SIZE: double
SPEED: double
maxBall: int
now Ball: int
spaceCraftCanvas: Canvas
Shooter()
draw ():void
draw SpaceCraft():void
shoot():void
retrieve():void
increaseMaxBall():void
getState():ShooterState
destroy():void
move():void
getMaxBall():int
<<Java Class>>
Block
model
font: Font
life: int
startLife: int
width: double
height: double
img: Image
Block(double,double,double,double,int)
decreaseLife():void
move():void
draw ():void
getRect():Rectangle2D
destroy():void
onCollision(CollidableEntity):void
getBottom():double
<<Java Class>>
Ball
model
INIT_SPEED: double
BALL_COLOR: Paint
INIT_RADIUS: double
radius: double
direction: Vector2
speed: double
retrieved: boolean
Ball(double,double,Vector2)
move():void
down():void
destroy():void
collect():void
draw ():void
onCollision(CollidableEntity):void
checkCollision():void
checkFrame():void
getRect():Rectangle2D
randomDirection():void
<<Java Class>>
PowerUp
model.powerUp
DEFAULT_RADIUS: double
radius: double
color: Paint
img: Image
sound: AudioClip
hitted: boolean
PowerUp(double,double)
getRect():Rectangle2D
destroy():void
draw ():void
move():void
<<Java Class>>
VLightning
model.powerUp
VLightning(double,double)
onCollision(CollidableEntity):void
<<Java Class>>
HLightning
model.powerUp
HLightning(double,double)
onCollision(CollidableEntity):void
<<Java Class>>
RandomReflector
model.powerUp
RandomReflector(double,double)
onCollision(CollidableEntity):void
<<Java Class>>
BallPlus
model.powerUp
BallPlus(double,double)
onCollision(CollidableEntity):void
<<Java Enumeration>>
ShooterState
model
shoot: ShooterState
finishShoot: ShooterState
move: ShooterState
wait: ShooterState
ShooterState()
<<Java Interface>>
Movable
model
move():void



1. Package application

1.1 Class Main extends Application

1.1.1 Method

+ void start(Stage primaryStage)	The main entry point for the JavaFX applications.
+ void main(String[] args)	An entry point of the application.

2. Package exception

2.1 Class MinDegreeExceedException extends Exception

2.1.1 Method

+ String getError()	Return message "Min degree exceeded! Please aim higher."

2.2 Class RetrieveException extends Exception

2.2.1 Method

+ String getErrorMessage()	Return message "Can't retrieve when not shooted."

2.3 Class ShootException extends Exception

2.3.1 Method

+ String getErrorMessage()	Return message "Can't shoot now!!!"

3. Package logic

3.1 Class GameManager

3.1.1 Field

+ enum GameState	Define game states there are “aim, shoot, wait, endShot, retrieve, move”.
- int BLOCKS_PER_ROW	Define number of blocks in a row to 7.
+double GRID_SIZE	Define size of one grid to GameStage.GAMEWIDTH divide by BLOCKS_PER_ROW.
- double BLOCK_SPAWN_RATE	Define rate of spawning blocks to 50 percentage.
- double BLOCK_SIZE	Define side of blocks to GRID_SIZE - 5.

+ double SPAWN_Y	Define level of y which use to spawn any object (that is 100).
+ double START_X	Define where x of shooter should be in the begin. (GameStage.GAME_WIDTH/2)
+ double START_Y	Define where y of shooter should be in the begin. (GameStage.GAME_HEIGHT - 50)
+ double stopPoint	Where x of first ball landed.
- long FPS	Refresh rate of the game in frame per second.
- long LOOP_TIME	Nanotime in one loop of animation timer.
- long startTime	Time when game start.
- AnimationTimer timer	Class AnimationTimer to handle all updates in game
- GameState currentState;	Current state of game.
- int level	Current level of game.

3.1.2 Method

+ void startGame()	Start the game by - Setup object in Holder - Initialize level of game to 1 - Spawn game object - Set current state to aim - And start AnimationTimer to update every loop time
+ void update()	Update game by following current state 1. Aim state : set AimLine visibility to true 2. Shoot state : set AimLine visibility to false and call ballUpdate() 3. Move state : move the shooter and if done set current state to aim 4. Endshot state : update level, slide objects down, spawn new objects, check if lose. When done set state to move;
- void ballUpdate()	Clear destroyed balls in ball container, move every ball and if have no ball in game, set state to “endShot”

+ void shoot()	Try to call shoot() in Shooter and set current state to shoot. (Catch ShootException)
+ void retrieve()	Try to call retrive() in Shooter. (catch RetrieveException)
+ void aim(double x, double y)	Try to call aimTo() in AimLine. (catch MinDegreeExceedException)
- void spawnObjects()	Spawn blocks with rate and powerups. (each 5% rate)
- void objectsDown()	Clear all destroy object, and move every blocks and powerups
- void checkLose()	Check if any blocks pass the LOSE_LINE
- void gameOver()	Stop AnimationTimer, reset objects in holder, and goto gameover scene.
+GameState getCurrentState()	Getter of currentState.
+ int getLevel()	Getter of level.
+ int getMaxBall()	Getter of max amount of ball in game

3.2 Class Holder

3.2.1 Field

- Holder instance	A singleton of holder.
- GameStage gameStage;	A GameStage object.
# Animation animation	Animation player for effect
- List<Ball> ballContainer	ArrayList of all balls in game.
- List<Block> blockContainer	List of all blocks in game.
- List<PowerUp> powerUpContainer	List of all power-up objects in game.
- Shooter shooter;	A Shooter object in game.
- AimLine aimLine	An AimLine object in game.

3.2.2 Constructor

+ Holder()	Initialize a ballContainer, blockContainer, powerUpContainer, and animation.

3.2.3 Method

+ Holder getInstance ()	Getter of instance (singleton)
# void setup()	Initialize gameStage, shooter, aimline, and animation.
+ void add(Entity entity)	Add entity to the appropriate list. (also add their canvas to gameStage)
# void reset()	Reset all field.
# void clearObject()	Clear all destroyed objects. (ball, block, powerup)

Generate all getter.

4. Package model

4.1 Class *Entity*

4.1.1 Field

# double x	X position in game.
# double y	Y position in game.
+ Canvas canvas	Canvas of the object.

4.1.2 Constructor

+ Entity()	Set destroy to false.

4.1.3 Method

+ boolean isDestroyed()	Return destroy.
+ *void destroy()*	Abstract method of destroy.
+ void *draw()*	Abtract mothod of render object.

4.2 Class *CollidableEntity* extends Entity

4.2.1 Method

+ void *onCollision(CollidableEntity other)*	Abstract method to do action when collide with others.
+ *Rectangle2D getRect()*	Abstract method to get collision bound of object.

4.3 Interface Moveable

4.3.1 Method

+ void move()	move object.

4.4 Class Shooter extends Entity implements Movable

4.4.1 Field

+ enum ShooterState	Enumulator of states of shooter (shoot, finishshoot, move, wait).
- ShooterState nowState	Current state of shooter.
- int COOLTIME	Cooldown of ball shooting.
- double IMG_SIZE	Size of shooter image
- double SPEED	Speed of shooter when move
- int maxBall	Current maximum number of balls.
- int nowBall	Current number of ball that shooter have.
- Canvas spaceCraftCanvas	Canvas for shooter sprite.

4.4.2 Constructor

+ Shooter()	Initialize shooter - set maxBall to 1 - Set nowBall to MaxBall - Set x, y to START_X, START_Y in GameManager - Initialize spaceCraftCanvas

	- Add two canvas to GameStage - Set initial state to wait - Draw details

4.4.3 Method

+ void draw()	Draw amount of balls, and space craft.
+ void drawSpaceCraft()	Draw spacecraft at current position and rotate to appropriate angle.
+ void shoot() throws ShootException	(Throw exception when this method called in aim game-state) - Reset stop point to -1. - Set shooter state to shoot. - Get shoot direction from aimline. - New thread to isntantiate ball. (after shoot all ball, set shooter state to finish shoot.)
+ void retrieve() throws RetrieveException	(Throw exception when this method called when shooter is shooting.) - Set state to finishshoot. - Retrieve (down) all balls.
+ void increaseMaxBall()	Increase maxBall by one.
+ void destroy()	Do nothing (must be override)
+ void move()	- Reset nowBall to maxBall - Set shooterState to move - Move to stopPoint in GameManager - Once reached stopPoint set shooterState to wait

Generate getter of shooterState and maxball

4.5 Class Block extends CollidableEntity implements Movable

4.5.1 Field

- Font font	Default font for show block life
- int life	Current life of block
- int startLife	Life of the block at the start

- double width	Width of block
- double height	Height of block
# Image img	Image of block

4.5.2 Constructor

+ Block(double x, double y, double width, double height, int life)	- Initialize field from given - Initialize canvas - Draw block

4.5.3 Method

+ void decreaseLife()	decrease block life by 1 and check if should destroy this block, redraw block
+ void move()	Drop down this block 1 step, translate canvas
# void draw()	- Translate canvas - Repaint color of block related to block life - Draw block life
- Rectangle2D getRect()	Return collision bound of block
+ void destroy()	- Play block effect - Set destroy to true - Remove canvas
+ void onCollision(CollidableEntity other)	Decrease block life and play sound
+ double getBottom()	Return bottom point of this block

4.6 Class Ball extends CollidableEntity implements Movable

(Use org.dyn4j.geometry.Vector2 [External JAR])

4.6.1 Field

- double INIT_SPEED	Default speed of ball
- Paint BALL_COLOR	Default ball color
- double INIT_RADIUS	Default radius of ball
# double radius	This radius

- Vector2 direction	This ball direction
- double speed	This ball speed
- boolean retrieved	Has this ball retrieved?

4.6.2 Constructor

+ Ball(double x, double y , Vector2 direction)	- Set speed, radius to default - Set x,y,direction to given - Initialize canvas - Draw object

4.6.3 Method

+ void move()	- Set new x,y related to speed and direction - Check colliosion if not retrieved - Translate canvas to new x,y
+ void down()	- Set retrieved to true - Set direction to downward - Set new speed
+ void destroy()	Set speed to zero, set destroy to true and remove canvas from GameStage
+ void collect()	If first ball not landed, set stop point to this x Move ball to stop point
+ void draw()	Draw object
+ void onCollision(CollidableEntity other)	(Override) change direction when collide with block
+ void checkCollision()	Check if this ball collide with others object and take action
+ void checkFrame()	Collect a ball when go out of game scene and redirection of ball when bounce with border
+ Rectangle2D getRect()	Get collision bound of this ball
+ void randomDirection()	Random new direction of ball

4.7 Class AimLine extends Line

4.7.1 Field

- double LIMITE_DEGREE	Degree of limitation of shooting
- double MIN_RADIAN	Convert LIMIT_DEGREE to radian

4.7.2 Constructor

+ AimLine()	- Set style of this line - Set start x,y relate to shooter - Set stop x,y to default - Add canvas to gameStage

4.7.3 Method

+ void aimTo(double xx,double yy) throws MinDegreeExceedException	(throw exception if degree less than MIN_DEGREE) - Set end point to xx,yy - And if GameState is aim rotate the spacecraft
+ double getDegree()	Return degree of this aimline
+ Vector2 getVector()	Return vector of this aimline
+ void reCenter()	Set start x,y related to shooter

5. Package model.powerup

5.1 Class PowerUp extends CollidableEntity implements Movable

5.1.1 Field

# double DEFAULT_RADIUS	Default radius of all power up.
# double radius	Radius of this object.
# Image img	Image sprite of this object.
# AudioClip sound	Sound then ball collided.
# boolean hitted	Check if this object is hitted.

5.1.2 Constructor

+ PowerUp(double x, double y)	Initialize - Set x, y to given. - Set radius to default. - Initial canvas and set translate to position.

5.1.3 Method

+ Rectangle2D getRect()	(Override) Return rect of upper-left x lower-right bound of this object.
+ void destroy()	(Override) set destroy to true and remove this canvas from GameStage
# void draw()	Draw this sprite.
+ void move()	If hitted destroy(), else set new y of this object, and translate canvas. Also check if object reach LOSE_LINE, destroy them.

5.2 Class BallPlus extends PowerUp

5.2.1 Constructor

+ BallPlus(double x, double y)	- Call super constructor - Set radius to 70% of default - Set canvas translate to this position - Set image and sound - Draw this object

5.2.2 Method

+ void onCollision(CollidableEntity other)	(Override) increase shooter maxball, play sound, and destroy this object.

5.3 Class RandomReflector extends PowerUp

5.3.1 Constructor

+ RandomReflector(double x, double y)	- Call super constructor - Set image and sound - Draw object

5.3.2 Method

+ void onCollision(CollidableEntity other)	(Override) set hitted to true, random direction of ball which collide with, play sound

5.4 Class VLightning extends PowerUp

5.4.1 Constructor

+ VLightning(double x, double y)	- Call super constructor - Set image - Draw object

5.4.2 Method

+ void onCollision(CollidableEntity other)	(Override) - Set hitted to true - Play sound and effect - Decrease a blocklife in same column of this object.

5.5 Class HLightning extends PowerUp

5.5.1 Constructor

+ HLightning(double x, double y)	- Call super constructor - Set image - Draw object

5.5.2 Method

+ void onCollision(CollidableEntity other)	- Set hitted to true - Decrease blocks that lay in the same row of this object - Play effect and sound

6. Package scene

6.1 Class Animation

6.1.1 Method

+ void drawVLightning(double x)	Draw Vertical Lightning Animation as shown in Figure
+ void drawHLightning(double y)	Draw Horizontal Lightning as shown in Figure
+ void drawBomb(double x, double y)	Draw Bomb Animation as shown in Figure

6.2 Class GameOver extends Pane

6.2.1 Field

- double BTN_WIDTH	Width of the button
- double BTN_HEIGHT	Width of the button
- Font headFont	Font of the head part
- Font bodyFont	Font of the body part
- Canvas bgCanvas	Canvas of background
- Canvas replayBtn	Canvas of replay button
- Canvas homeBtn	Canvas of home button
- Canvas field	Canvas of text field
- Canvas gameOverLabel	Canvas of gameover label
- Canvas result	Canvas of result field
- int lvl	Level you get before the game end
- int maxBall	maxBall you get before the game end

6.2.2 Constructor

+ GameOver(int lvl, int maxBall)	Initialize lvl and maxBall as given values Initialize all buttons and Draw all canvases

6.2.3 Method

- void changeCenter (Canvas canvas, double x, double y)	Change the center of canvas
- Canvas drawButton (Image img)	Draw a button as given image
- void drawOnMouseEvent (Canvas canvas, Image img)	Change the image of the button when mouse enter
- void addCanvasEvents (Canvas canvas, String buttonName, Image img1, Image img2)	- If you click on replay button, it will go to GameScene - If you click on home button, it will go to MainMenuScene - If your mouse enter replay button or home button, it will change the button image - If your mouse exit replay button or home button, it will change back the button image
- void showResult(Canvas canvas)	Show the result of the game includes level and maxball

6.3 Class GameStage extends Pane

6.3.1 Field

+ double GAME_WIDTH	Width of game stage scene
+ double GAME_HEIGHT	Height of game stage scene
+ double LOSE_LINE	Line of ground that when your ball touch this line, it will disappear
# Canvas canvas	A canvas
- Canvas topBar1	Canvas of field of topBar
- Canvas topBar2	Canvas of level of game

+ GraphicsContext gc	Draw field of topBar
+ GraphicsContext gc2	Draw level of game

6.3.2 Constructor

+ GameStage()	Initialize and set event

6.3.3 Method

- void initialize()	Draw game stage background
- void setEvent()	- When you move your mouse it will store your (x,y) on scene - If you left click, it will shoot balls - If you right click, it will retrieve all balls
+ void drawTopBar (GraphicsContext gc, GraphicsContext gc2)	Draw topbar by given canvas
+ void redrawLevel(int lvl)	Update level by redraw on canvas

6.4 Class MainMenu extends Pane

- double BTN_WIDTH	Width of button
- double BTN_HEIGHT	Height of button
- Font headFont	Font of head part
- Canvas playBtn	Canvas of play button
- Canvas menuBG	Canvas of main menu scene background

6.4.2 Constructor

+ MainMenu()	Initialize all buttons and Draw all Canvases

6.4.3 Method

- void changeCenter(Canvas canvas,double x, double y)	Change the canter of canvas
- Canvas drawButton(Image img)	Draw a button by given image
- void drawOnMouseEvent (Canvas canvas, Image img)	Change the image of the button when mouse enter
- void addCanvasEvents (Canvas canvas, String buttonName, Image img1, Image img2)	- If you click on play button, it will go to GameScene - If your mouse enter play button, it will change the button image - If your mouse exit play button, it will change back the button image

6.5 Class ResLoader

6.5.1 Field

+ Image MenuImg	Image of main menu background
+ Image PlayBtn1	Image of play button when your mouse enter
+ Image PlayBtn2	Image of play button when your mouse exit
+ Image SpaceCraftimg	Image of spacecraft
+ Image GameBG	Image of game stage scene
+ Image BlockImg	Image of grey block
+ Image BlockPurple	Image of purple block
+ Image BlockRed	Image of red block
+ Image BlockBlue	Image of blue block
+ Image GameOverImg	Image of game over scene image
+ Image ReplayBtn1	Image of replay button when your mouse enter
+ Image ReplayBtn2	Image of replay button when your mouse exit
+ Image HomeBtn1	Image of home button when your mouse enter

+ Image HomeBtn2	Image of home button when your mouse exit
+ Image RandomReflectorImg	Image of random reflector
+ Image VLightning	Image of vertical lightning
+ Image HLightning	Image of horizontal lightning
+ Image BallPlusImg	Image of ball plus item
+ Image HLightningAniImg	Image of horizontal lightning animation
+ Image VLightningAniImg	Image of vertical lightning animation
+ Image ExplosionAniImg	Image of explosion animation
+ AudioClip electricSound	Sound of electric
+ AudioClip bangSound	Sound of bang
+ AudioClip collectSound	Sound of collecting item
+ AudioClip bounceWallSound	Sound of bouncing wall
+ AudioClip bounceBlockSound	Sound of bouncing block
+ AudioClip reflectorSound	Sound of bouncing random reflector

6.5.2 Method

+ void loadResource()	Load all resource in res folder

6.6 Class SceneManager

6.6.1 Field

- Stage primaryStage	Primary stage
+ double SCENE_WIDTH	Width of scene
+ double SCENE_HEIGHT	Height of scene
- Pane mainMenu	Pane of main menu

- Scene mainMenuScene	Scene of main menu

6.6.2 Method

+ void initialize(Stage stage)	Set primaryStage to given stage Set primaryStage resizable to be false Show primaryStage
+ void gotoMainMenu()	Set scene to mainMenuScene
+ void gotoSceneOf(Pane pane)	Set scene by given Pane
+ void goToGameScene()	Start game and Set scene to GameScene