

Arrow Dome

Created by

Peemdej Chancharoen 6530313021

Ratchanon Thanatsang 6532147621

2110215 Programming Methodology

Semester 1 Year 2023

Chulalongkorn University

Arrow Dome

Introduction

Arrow Dome is a typing game featuring fewer keys, making it easier to play than other typing practice games. In this game, you play the role of an individual responsible for destroying exploding rockets that threaten to fall on Earth. To keep the city safe, you must enter the correct transmission code, sending a signal wave to destroy the rockets, ensuring they explode while still in the sky.

Rules

When you start the game, a rocket appears at the top of the screen. To destroy it, you need to enter the correct key as indicated by the transmission code at the bottom of the screen. Once you have entered the keys correctly, press space to submit your entry and cause the rocket to explode.

If you press the wrong key, the input will reset. Additionally, if you enter the correct keys but fail to press the spacebar, the input will also reset. Remember, pressing the spacebar is crucial.

Initially, only basic rockets will fall. As you continue playing, two additional types of rockets Advanced Rockets and Boss Rockets will appear. With each increment of ten points in your score, the game's difficulty increases.

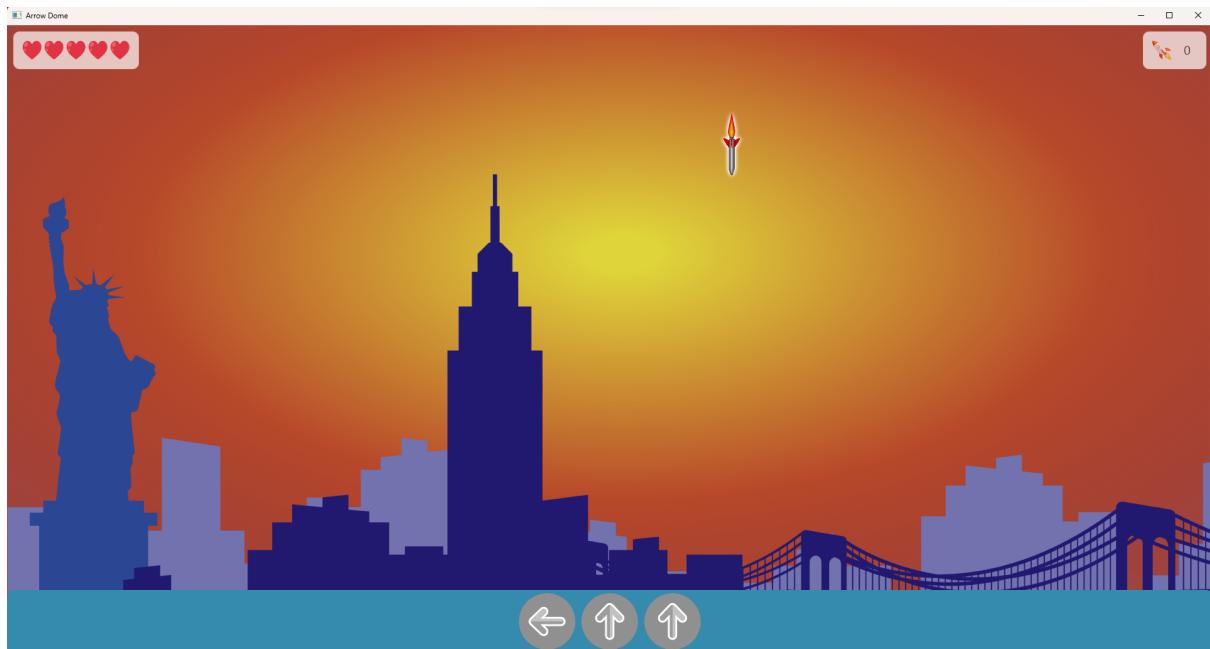
Scoring works as follows: Basic Rockets award 1 point. Advanced Rockets, which require more keys to be typed, give 3 points. Boss Rockets are worth 5 points and will destroy all rockets in the sky when destroyed.

You can switch the target rocket by pressing TAB.

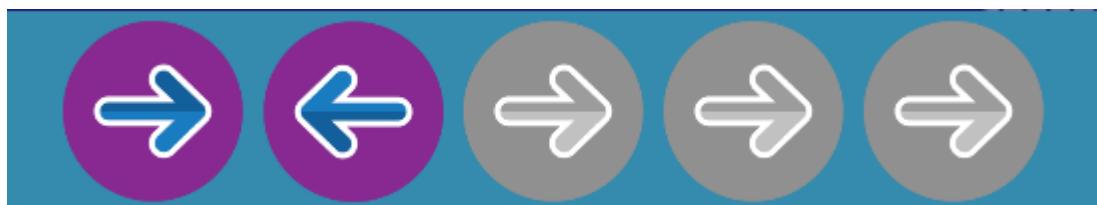
Your ability to let rockets hit the ground is represented by a heart in the top left corner of the screen. If you run out of hearts, the game is over

Example

At the start of the game. The rocket is spawned at the top screen.



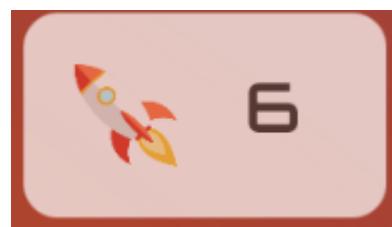
You must type the keys on the keyboard according to the key combination shown at the bottom of the screen. When the key set becomes color, it means you are typing correctly.



When you type accurately according to the key set, the rocket will explode. Your score from destroying the rocket will be displayed, and you can check your current score at the top left of the screen.



Score of that rocket .



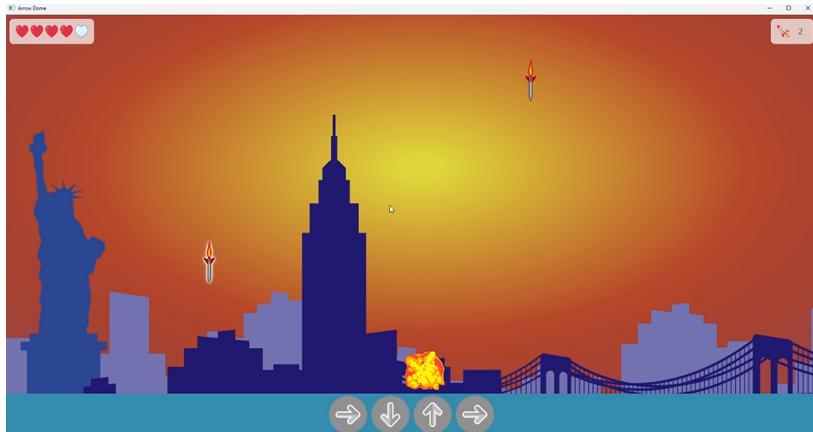
Your current score.

The target rocket, which will explode upon completion of the key set typing, is marked with a white highlight as an indicator. You can select a different target rocket by pressing the TAB key. As illustrated in the picture, the highlight shifts from one rocket to another.

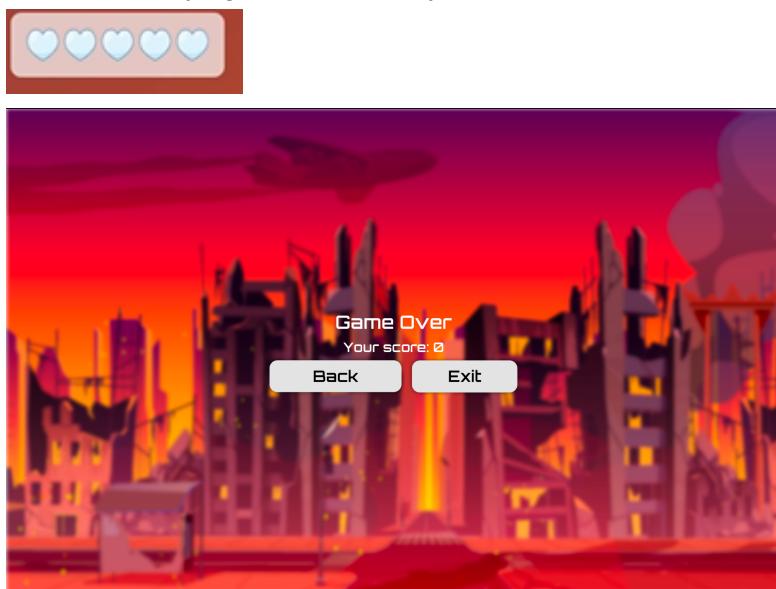
Please note that pressing TAB will reset the keys to a new key set.



When a rocket hits the ground, the number of hearts will decrease.

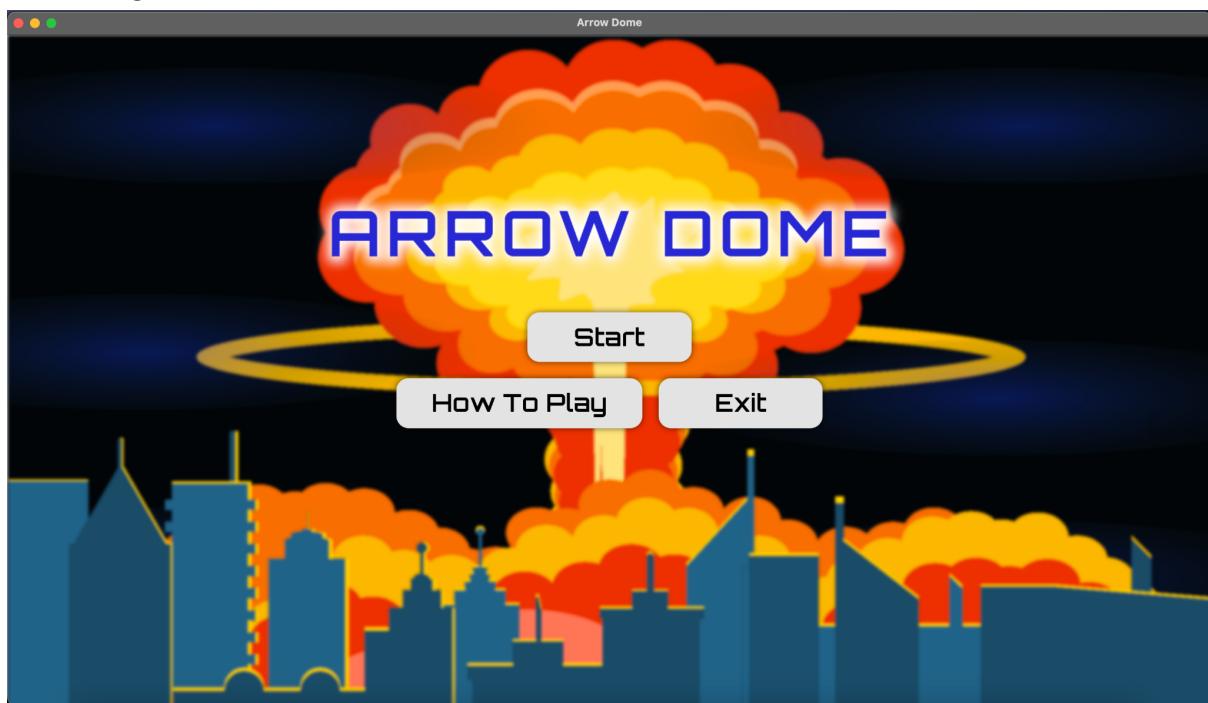


When you run out of hearts, the game is over. At that point, the game over screen will appear, displaying all the scores you have earned.

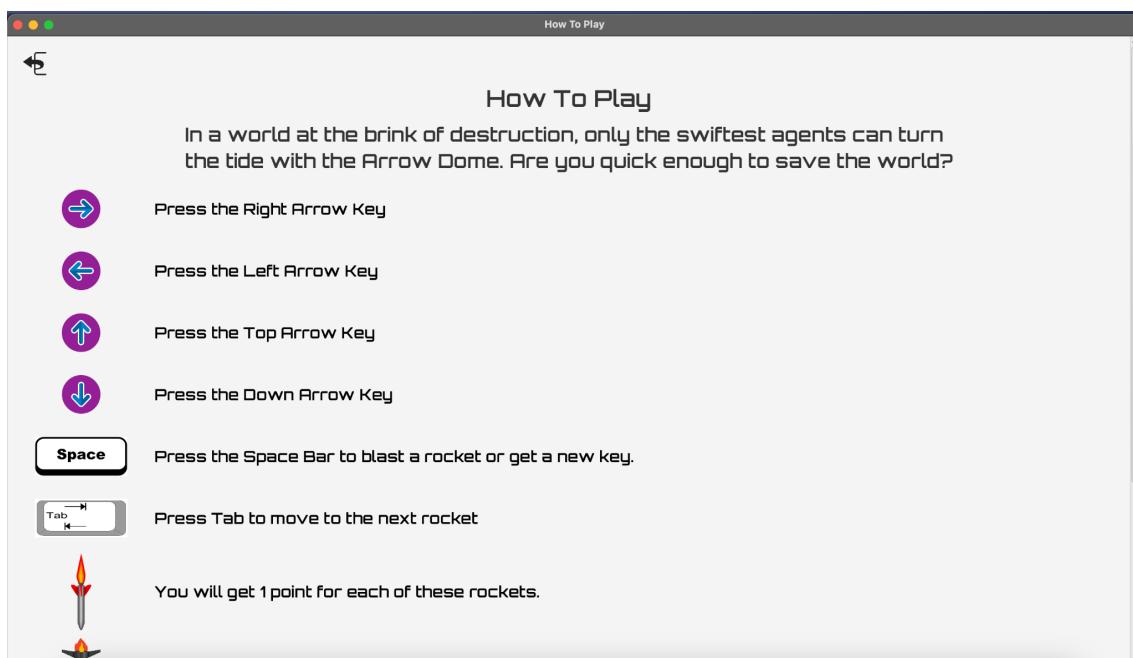


Main Manu Scene

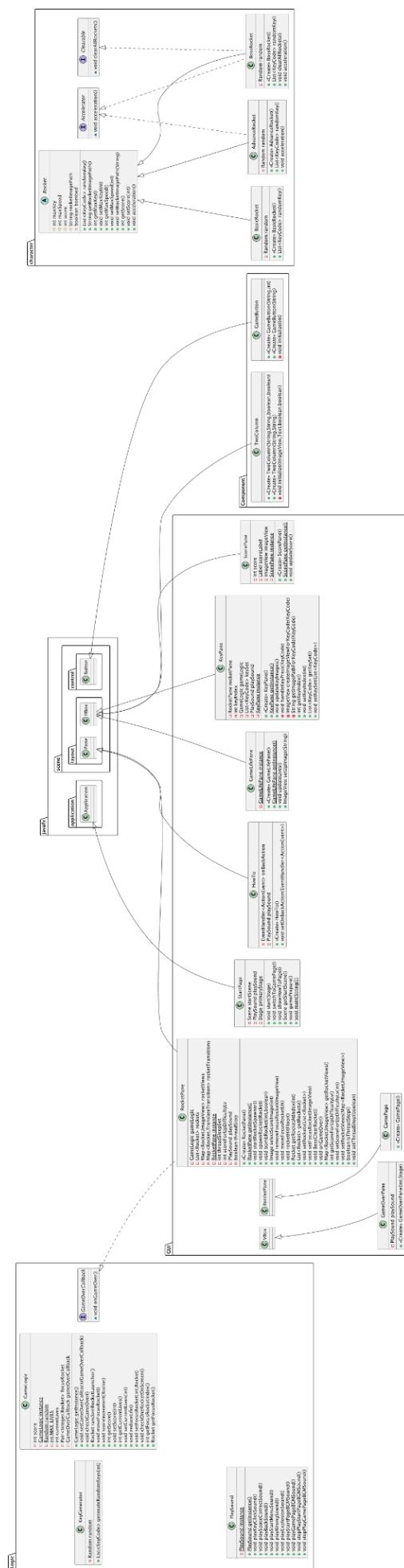
Start Page



When you click the “How To Play” button, it will show you the page similar to the one shown below.



Class Diagram



1. Package character

1.1 public abstract class **Rocket**

Field

Name	Description
# int maxKey	Number of key in rockets
# int maxSpeed	Number of speed in rockets
# int score	Score of rockets
# String rocketImagePath	Image path of rockets
- boolean bombed	State of rockets already bomb or not

Method

Name	Description
+ abstract List<KeyCode> randomKey()	Random key for eliminated this rocket
getter/setters	

1.2 public class **BasicRocket** extends **Rocket**

Field

Name	Description
- Random random	Use for generate random variable

Constructor

Name	Description
+ BasicRocket	Initialize field

Method

Name	Description
+ List<KeyCode> randomKey()	Random key for eliminated this rocket

1.3 public class **AdvanceRocket** extends **Rocket** implements **Accelerator**

Field

Name	Description
- Random random	Use for generate random variable

Constructor

Name	Description
+ AdvanceRocket	Initialize field

Method

Name	Description
+ List<KeyCode> randomKey()	Random key for eliminated this rocket
+ void acceleration()	increase the speed of the rocket

1.4 public class **BossRocket** extends **Rocket** implements **Clearable,Accelerator**

Field

Name	Description
- Random random	Use for generate random variable

Constructor

Name	Description
+ BossRocket	Initialize field

Method

Name	Description
+ List<KeyCode> randomKey()	Random key for eliminated this rocket
+ void clearAllRockets()	Eliminate all rockets when BossRocket got eliminated
+ void acceleration()	increase the speed of the rocket

1.5 public interface **Clearable**

Method

Name	Description
void clearAllRockets()	Clear all rockets

1.6 public interface **Accelerator**

Method

Name	Description
void acceleration()	increase the speed of the rocket

2. Package GUI

2.1 public class **StartPage** extends **Application**

Field

Name	Description
- Scene startScene	Scene of start page
- Playsound playsound	To access method of Playsound
- Stage primaryStage	Variable of primaryStage

Method

Name	Description
+ void start	Set start page
+ void switchToGamePage()	go to the GamePage
- void showHowToPage	go to the How to page.
+ void gamePrepare	set score to 0 and live to 5

2.2 public class **HowTo** extends **Pane**

Field

Name	Description
- EventHandler<ActionEvent> onBackAction	Go back to start page action
- PlaySound playSound	To access method of Playsound

Constructor

Name	Description
+ Howto	set the Howto page

Method

Name	Description
+void setOnBackAction(EventHandler<ActionEvent> onBackAction)	setter for onBackAction

2.3 public class **GamePage** extends **BorderPane**

Constructor

Name	Description
+ GamePage()	Set the GamePage. Set RocketPane at center. Set KeyPane at bottom. Set GameLifePane at top left. Set ScorePane at top right.

2.4 public class **KeyPane** extends **HBox**

Field

Name	Description
- RocketPane rocketPane	Use as instance of RocketPane
- int keyIndex	Index of key
- GameLogic gameLogic	Use as instance of GameLogic
- List<KeyCode> keyset	List of key
- Playsound playsound	To access method of playsound
- static Keypane instance	Instance of KeyPane

Constructor

Name	Description
+ KeyPane()	initialze KeyPane

Method

Name	Description
+ static KeyPane getInstance()	Get Instance of KeyPane
+ void updateKeyImages()	Update image of key set

- void handleKeyPress(KeyCode pressedKey)	Deal with typing input. If typing is correct, eliminate rockets. If typing is wrong, change to a new key set.
- ImageView createImageViewForKeyCode(KeyCode keycode)	Create an ImageView that shows the symbol of the key set.
- String getImagePathForKeyCode(KeyCode keyCode)	Get path of key set image
getter/setters	

2.5 public class **RocketPane** extends **Pane** implements **GameOverCallback**

Field

Name	Description
- GameLogic gamelogic	Instance of GameLogic
- List<Rocket> rockets	List of rockets show in GamePage
- Map<Rocket, ImageView> rocketviews	Map of rockets ImageView
- Map<Rocket, TranslateTransition> rocketTransitions	Map that store transition of rockets
- static RocketPane instance	Instance of RocketPane
- int threadSleepSet	Define how long a thread sleeps.
- int scoreForUpDifficultyLv	Score used to check if it hit the next difficulty.
- Playsound playsound	To access method of Playsound
- boolean threadStop	To set state of the thread

Constructor

Name	Description
+ RocketPane	Initialize RocketPane

Method

Name	Description
+ static RocketPane getInstance()	get Instance of RocketPane

- void startRocketSpawner()	Start a spawn rocket from the top edge of the screen.
+ void spawnRocket(Rocket rocket)	Make rocket spawn and set the transition of the rocket.
+ void bombRocket(int index,boolean hitfloor)	Choose a rocket to bomb at the indicated index.
- Image selectScoreImage(int score)	Select what image of score will show up when a rocket bomb.
+ Rocket getRocketAtIndex(int index)	Get a rocket at the indicated index.
+ void setFocusRocket(ImageView imageView)	Set a rocket that will explode next.
+ void removeFocusRocket(ImageView imageView)	Delete Highlight and focus state from focus rocket.
+ void moveFocusRocketUI()	Move Highlight to the new current focus rocket.
+ void rocketHitFloor()	Dealing with the situation when the rocket hit the floor.
+ void BossClearRocket()	When BossRocket explodes, explore all of the other rockets.
+ void onGameOver()	Dealing with the situation when Life hit 0 and game over.
getter/setters	

2.6 public class **GameLifePane** extends **HBox**

Field

Name	Description
- static GameLifePane instance	instance of GameLifePane

Constructor

Name	Description
+ GameLifePane()	Initialize GameLifePane

Method

Name	Description

+ static GameLifePane getInstance()	get instance of GameLifePane
+ void updateLives()	update display of heart
+ ImageView setUpImage(String imagePath)	set up ImageView of heart

2.7 public class **GameOverPane** extends **VBox**

Field

Name	Description
- Playsound playSound	To access method of Playsound

Constructor

Name	Description
+ GameOverPane(int finalScore, Stage primalyStage)	Set the pane when game over

2.8 public class **ScorePane** extends **HBox**

Field

Name	Description
- static ScorePane instance	Instance of ScorePane
- Label scoreLabel	Label of score
- int score	number of score

Constructor

Name	Description
- ScorePane	Initialize score pane

Method

Name	Description
+ static ScorePane getInstance()	get Instance of ScorePane
+ void updateScore()	Update the score

3. Package Component

3.1 public class **GameButton** extends **Button**

Constructor

Name	Description
+ GameButton(String text, int width)	initialize GameButton and set text of GameButton
+ GameButton(String text)	initialize GameButton and set text of GameButton

Method

Name	Description
- void initialize()	set appearance of button in StartPage, AboutPage, HowtoPage, OverPage

3.2 public class **TwoColumn** extends **HBox**

Constructor

Name	Description
+ TwoColumn(String imagePath, String text, boolean isSquare, boolean isRocket)	Set TwoColumn in HowtoPage. initialize ImageView and text
+ TwoColumn(String imagePath, String text)	Set TwoColumn in HowtoPage. initialize ImageView and text

Method

Name	Description
- void initialize(ImageView imageView, Text text, boolean isSquare, boolean isRocket)	Set ImageView to locate at left column and Text at right column.

4. Package Logic

4.1 public class **GameLogic**

Field

Name	Description
- int score	Score of game
- static GameLogic instance	Instance of game
- static final Random random	Use to generate random variable
- static final int MAX_LIVES	Number of heart in game

- int currentLives = MAX_LIVES	Current heart of the game. At the beginning, it was set to max number of hearts.
- Pair<Integer, Rocket> focusRocket	Index of rocket that will bomb when finish typing all keyset
- GameOverCallback gameOverCallback	Variable to deal with situation when game over

method

Name	Description
+ static GameLogic getInstance	Get Instance of GameLogic
+ Rocket randomRocketLauncher	Randomize the type of rocket that will land next.
+ void scoreIncrement(Rocket rocket)	Increment the score of game
+ void checkGameOver()	Checking if game over or not
+ void reduceLife()	Decrease the number of life
+ void checkOneRocketOnScreen()	Checking if there are one rocket on screen.
+ void moveFocusRocket()	Change focus rocket to the next rocket
getter/setters	

4.2 public class **KeyGenerator**

Field

Name	Description
- static final Random random	Use for generate random variable

method

Name	Description
+staticList<KeyCode> generateRandomKeys(int maxKey)	generate random keys set.

4.3 public class **PlaySound**

method

Name	Description
+ void playKeyClickSound	Playing sound when type key correct
+ void playSpaceCorrectSound	Playing sound when type space to submit whole correct keys
+ void playBackSound	Playing sound when hit back button
+ void playStartMenuSound	Playing sound when hit button in start page
+ void playWrongSound	Playing sound when type wrong key
+ void playExplosionSound	Playing sound when rocket is explosion
+ void playStartPageBGMSound	BGM of StartPage
+ void playGamePageBGMSound	BGM of GamePage
+ void stopPlayStartPageBGMSound	Stop playing BGM of StartPage
+ void stopPlayGamePageBGMSound	Stop playing BGM of Gamepage

4.4 public interface GameOverCallback

method

Name	Description
void onGameOver()	Reset game and show game over page