Treason Documentation

Created By

Poom Suchao-in 6231349621

Napat Maneeratpongsuk 6231318121

Treason

Introduction

Treason is a game that inspired by contra, run and gun video game. The objective of the treason game is to win the enemy along the game and defeat the final boss at the end.

Rules

You must use all yours weapon to fight along the incoming enemy with your limit health. If your health is zero or your character fells off the map the game is over. It will be a helping item that can give you some assisting. at the very end if you can defeat the final boss, then you win the game.

Example

- your main character will start at the left of the map. You can move your main character by pressed W to jump, A to move left, D to move right and pressed SPACEBAR to attack.
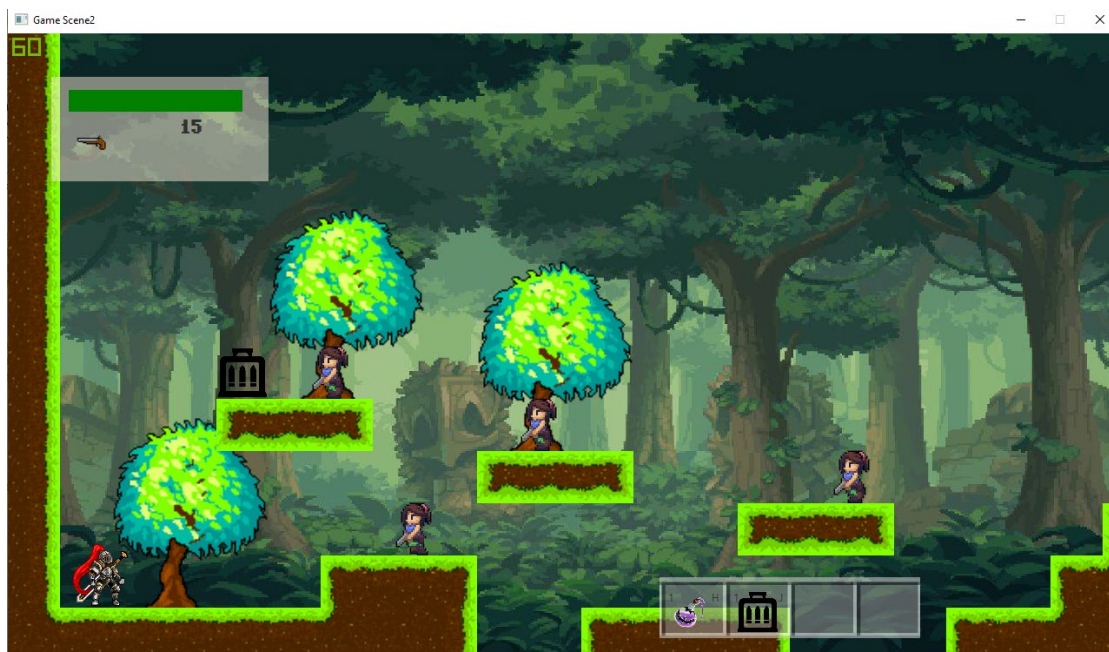
- you should avoid your character form enemy bullet. If bullet hit your main character, then your Hp will decrease.
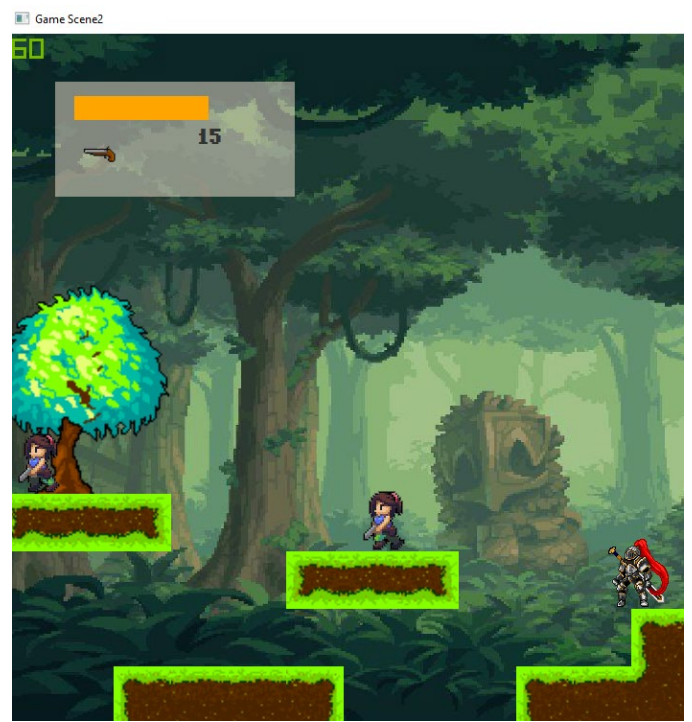
- use your weapon to attack the enemy. Specific weapon will deal different damage with enemy. In order to go to the next level, you must pass through the enemies in the map level.
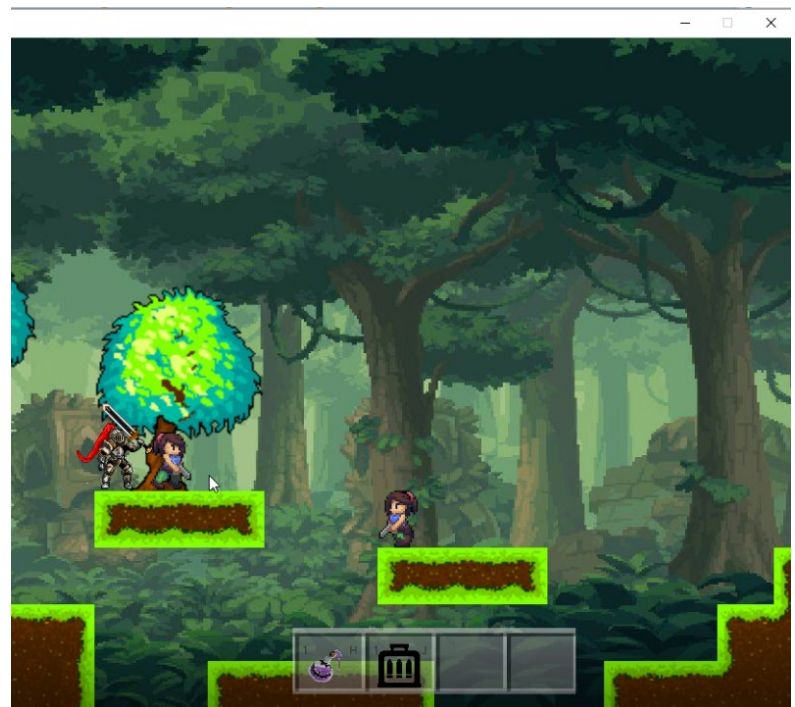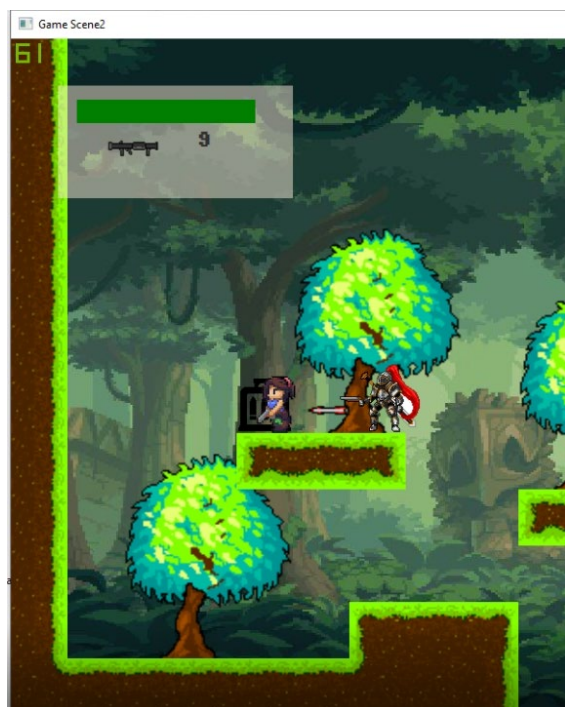
- move to the very right of the map to find a portal. You can go to the next level by pressing ENTER key.

- your game will over if you fall off the map or HP below zero.
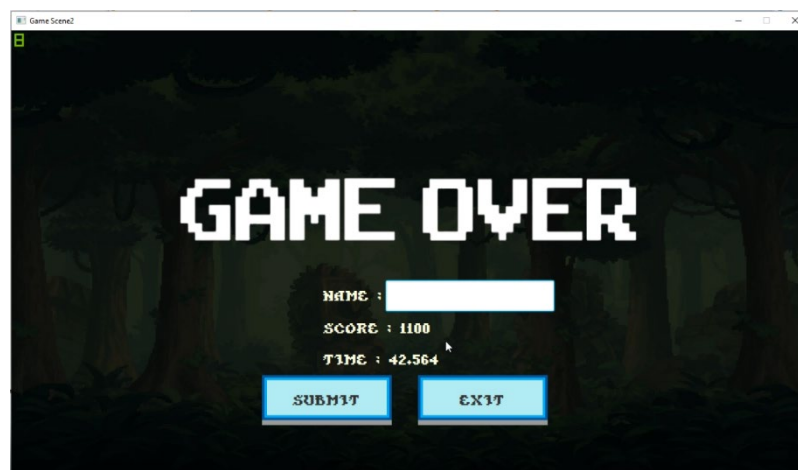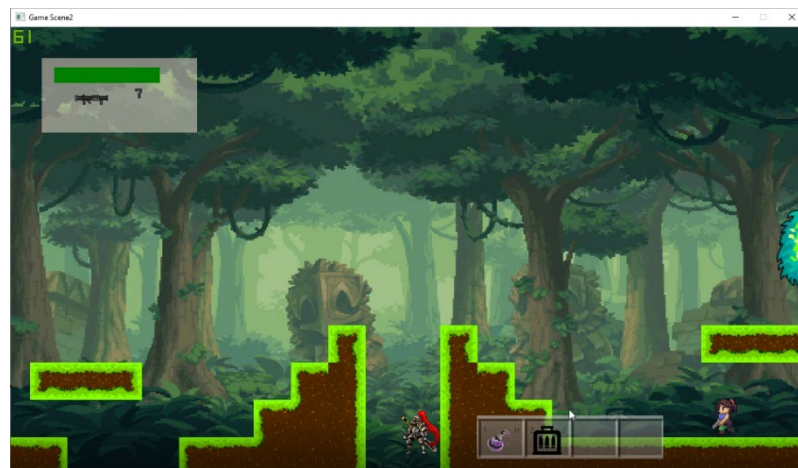
- You must kill the final boss to win the game. In the boss stage the boss will attack you with a bomb. While the boss is attacking, there will have boss pensioner spawn at the floor and try to attack your main character.
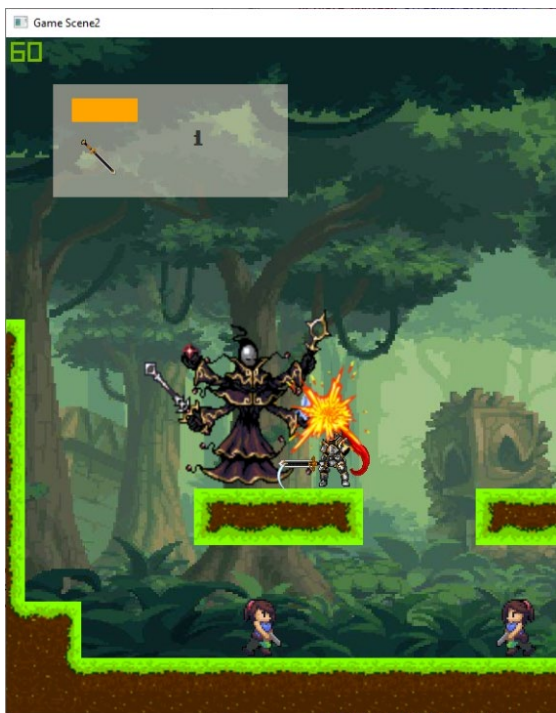
- If you kill the boss, you will win the game. The score and used time will show at the end. You can submit your name or not. then, you will return to main menu.
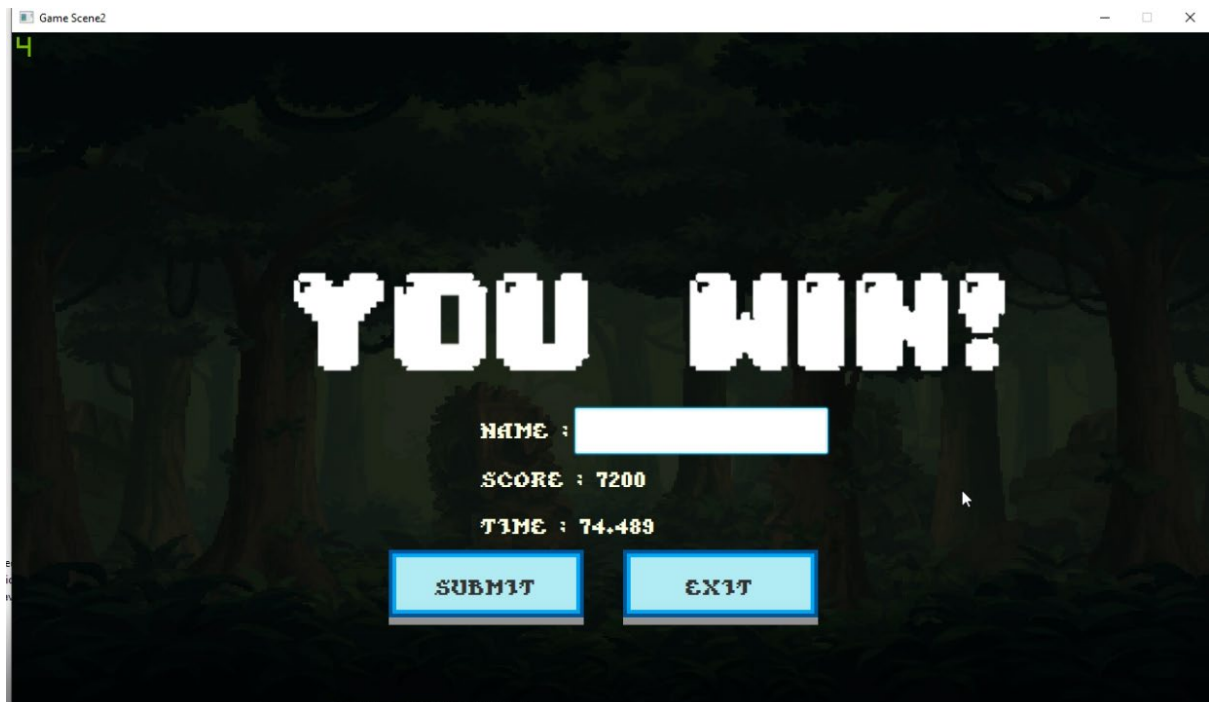
- The main menu pane will show the 5 buttons. You can start the game by click start new game or you can continue your game if you exit your game and still didn't close the window yet.
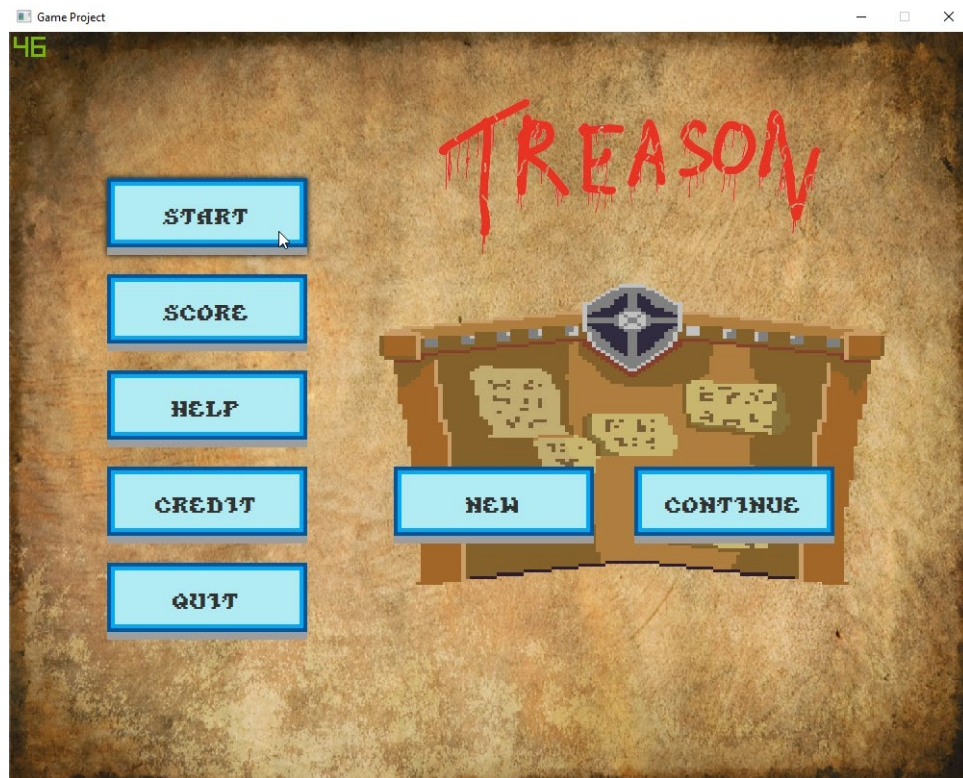
   - score button will show 10 highest score that have been submitted. Help button will give you some help. Credit button will show creator picture. Quit button will close the game window.

Class Diagram
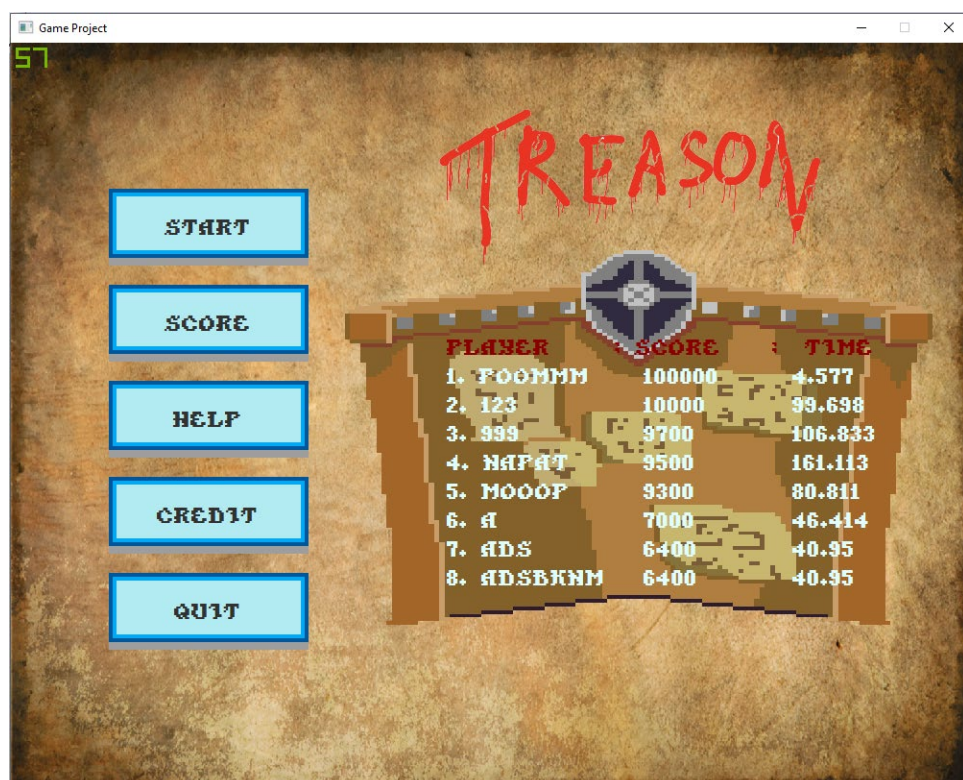
1 package main

1.1 Class Main extends Application

This class is the main application interface.

1.1.1 Methods

+void start(Stage primaryStage)	Starting the main application, put in Interfaces element.
+void main(String[] args)	Main application

2 package element

2.1 Class AudioLoader

2.1.1 Fields

+ static AudioClip Explosion_Sound	Explosion sound
+ static AudioClip Granade_Sound	Granade sound
+ static AudioClip Gun_Sound	Gun sound
+ static AudioClip Sword_Sound	Sword sound
+ static AudioClip Gameover_Sound	Game over sound
+ static AudioClip Mouse_Enter_Sound	Mouse enter sound
+ static AudioClip Mouse_Pressed_Sound	Mouse pressed sound
+ static AudioClip Off_Limits	BGM
+ static AudioClip Star_Commander	BGM

2.1.2 Methods

+ static void setVolume(Double volume)	Set the volume of every AudioClip

3 package exception

3.1 Class AddLeaderboardScoresFailed extends Exception

3.1.1 Fields

- static final long serialVersionUID	Set to 1L
+ String message	Error message

3.1.2 Constructor

+ AddLeaderboardScoresFailedException(String message)	Initialize all remaining fields

3.2 Class ConsumeItemFailedException extends Exception

3.2.1 Fields

- static final long serialVersionUID	Set to 1L
+ String message	Error message

3.2.2 Constructor

+ ConsumeItemFailedException(String message)	Initialize all remaining fields

3.3 Class FireBulletFailedException extends Exception

3.3.1 Fields

- static final long serialVersionUID	Set to 1L
+ String message	Error message

3.3.2 Constructor

+ FireBulletFailedException(String message)	Initialize all remaining fields

4 package gui

4.1 Class Gamebutton extends Button

4.1.1 Fields

- final String FONT_PATH	Path of font
- final String BUTTON_PRESSED_STYLE	Style of button when pressed
- final String BUTTON_FREE_STYLE	Style of button when not pressed

4.1.2 Constructor

+ GameButton(String text)	Initialize all remaining fields

4.1.1 Methods

- void setButtonFont()	Set the font of the button
- void setButtonPressedStyle()	Set the style of the button when pressed
- void setButtonFreeStyle()	Set the style of the button when not pressed
- void initializeButtonListerners()	Initialize all EventHandler

4.2 Class GameEndingScene extends AnchorPane

4.2.1 Fields

- Rectangle background	Background
- ImageView LogoImage	Status image
- Boolean isWin	Status
- int point	Point gained
- double time	Time spend
- TextField textField	Text field
- GameButton submitButton	Submit button
- GameButton exitButton	Exit button

4.2.2 Constructor

+ GameEndingScene(Boolean isWin, int point, double time)	Initialize all remaining fields

4.2.1 Methods

- void createBackground()	Create background
- void createStatusImage()	Create Status image if win show win picture else show game over
- void createTextBox()	Create 3 textboxes (Name,Score,Time)
- void createSubmitButton()	Create submit button
- void createExitButton()	Create exit button
Getter of fields	

4.3 Class GameSubScene extends SubScene

4.3.1 Fields

- static final int HEIGHT	Set to 400
- static final int WIDTH	Set to 600
- final String FONT_PATH	Path of font
- final String BACKGROUND_IMAGE	Background image path
- Boolean isHidden	Hidden status

4.3.2 Constructor

+ GameSubScene()	Initialize all remaining fields Set hidden status to true

4.3.1 Methods

- void moveSubScene ()	Show the sub scene if hidden status is false and vice versa
- AnchorPane getPane()	Return pane

4.4 Class PauseMenu extends AnchorPane

4.4.1 Fields

- Rectangle background	Background
- GameButton resume	Resume button
- GameButton restart	Restart button
- GameButton exit	Exit button
- ImageView sound	Sound icon
- Boolean isSoundON	Sound status

4.4.2 Constructor

+ PauseMenu	Initialize all remaining fields

4.4.3 Methods

- void createBackground()	Create background
- void createResumeButton()	Create resume button
- void createRestartButton()	Create restart button
- void createExitButton()	Create exit button
- void createSoundOn()	Create sound button
+ void changeSound()	Change the sound status
+ void setSoundOff()	Set sound status to false
+ void setSoundOn()	Set sound status to true
Getter of fields	

4.5 Class PlayerInfoBox extends AnchorPane

4.5.1 Fields

- Rectangle background	Background
- ImageView weaponImage	Weapon images
- Label number	number
- Rectangle playerHp	Hp of player

4.5.2 Constructor

+ PlayerInfoBox	Initialize all remaining fields

4.5.3 Methods

- void createBackground()	Create background
- void createWeaponImage()	Create weapon image
- void createNumber()	Create number

- void createPlayerHp	Create player Hp
+ void changeWeaponImage(ImageView imageView)	Change weapon image
Getter/Setter of fields	

4.6 Class PlayerInventoryBox extends AnchorPane

4.6.1 Fields

- MainCharacter player	Player
- ImageView inventory	Inventory image
- Label numberItem1	number
- Label numberItem2	number

4.6.2 Constructor

+ PlayerInventoryBox(MainCharacter player)	Initialize all remaining fields

4.6.3 Methods

- void createInventoryImage()	Create inventory image
- void createItemImage()	Create item image
- void createItem1()	Create numberItem1
- void createItem2()	Create numberItem2
Getter of fields	

4.7 Class SpriteAnimation extends Transition

4.7.1 Fields

- final ImageView imageView	ImageView
- final int count	Count
- final int colums	Colums

- final int offsetX	X-offset
- final int offsetY	Y-offset
- final int width	width
- final int height	height
- final int lastIndex	Last index

4.7.2 Constructor

+ SpriteAnimation(ImageView imageView, Duration duration, int count, int colums, int offsetX, int offset, int width, int height)	Initialize all remaining fields

4.7.3 Methods

# void interpolate(double k)	Interpolate sprites

5 package implement

5.1 Interface Consumable

5.1.1 Methods

+ abstract void consumed(MainCharacter player) throws ConsumeItemFailedException	Check consuming item

5.2 Interface Explodable

5.2.1 Methods

+ abstract void explode()	explode

5.3 Interface Fireable

5.3.1 Methods

+ abstract void addBullet(int count)	Add bullet
+ abstract Bullet fireBullet(GameCharacter character, Boolean isRight) throws FireBulletFailedException	Fire bullet
+ abstract Bullet fireBulletIninite(GameCharacter character, Boolean isRight)	Fire without consuming bullet
+ abstract Bullet setPositionBullet(GameCharacter character, Boolean isRight, Bullet bullet)	Bullet position

5.4 Interface Interactable

5.4.1 Methods

+ abstract void interact(MainCharacter player)	Interact

6 package components

6.1 Abstract Class Entity

6.1.1 Fields

# int x	X position
# int y	Y position
# int width	Width

# int height	Height
# Rectangle box	Box
# Image image	Image
# ImageView imageView	ImageView
# Animation sprite	Sprite
# String image_Path	Path of image
# int initX	Initial x
# int initY	Initial y
# int disX	X distant entity can move
# int disY	Y distant entity can move
# int velocityX	X velocity
# int velocityY	Y velocity
# int boundX	X bound of image
# int boundY	Y bound of image
# int addOnX	Move entity in x direction on top
# int addOnY	Move entity in y direction on top
# double finalTime	Time before entity get deleted

6.1.2 Constructor

# Entity(String image Path,int initX,int initY,int width,int height)	Initialize all remaining fields

6.1.3 Methods

Getter/Setter of fields	

6.2 Abstract Class Item

6.2.1 Fields

# String image_Path	Path of image

# Image image	Image
# ImageView imageView	ImageView
# int width	Width
# int height	Height
# int boundX	X bound of image
# int boundY	Y bound of image

6.2.2 Constructor

+ Item(String image_Path, int width, int height)	Initialize all remaining fields

6.2.3 Methods

Getter/Setter of fields	

6.3 Package box

6.3.1 Abstract Class Box extends Entity

6.3.1.1 Constructor

+ Box(String image_path, int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.2 Class AmmoBox extends Box implements Interactable

6.3.2.1 Constructor

+ AmmoBox(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.2.2 Methods

+ void interact(MainCharacter player)	Add ammo box to player inventory when interacted

6.3.3 Class Grass0 extends Box

6.3.3.1 Constructor

+ Grass0(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.4 Class Grass1 extends Box

6.3.4.1 Constructor

+ Grass1(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.5 Class Grass2 extends Box

6.3.5.1 Constructor

+ Grass2(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.6 Class Grass3 extends Box

6.3.6.1 Constructor

+ Grass3(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.7 Class Grass4 extends Box

6.3.7.1 Constructor

+ Grass4(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.8 Class Grass5 extends Box

6.3.8.1 Constructor

+ Grass5(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.9 Class Oak extends Box

6.3.9.1 Constructor

+ Oak(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.10 Class Portal extends Box

6.3.10.1 Fields

- int level	Game level

6.3.10.2 Constructor

+ Portal(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.10.3 Methods

Getter/Setter of fields	

6.3.11 Class PotionBox extends Box implements Interactable

6.4.11.1 Constructor

+ PotionBox(int initX, int initY, int width, int height)	Initialize all remaining fields

6.3.11.1 Methods

+ void interact(MainCharacter player)	Add potion to player inventory when interacted

6.4 Package bullet

6.4.1 Class Bomb extends Bullet implements Exploadble

6.4.1.1 Constructor

+ Bomb(boolean isRight, int initX, int initY)	Initialize all remaining fields

 6.4.1.2 Methods

+ void explode()	Play explosion sound Run explosion animation

6.4.2 Abstract Class Bullet extends Entity

6.4.2.1 Fields

# boolean isRignt	Boolean
# int damage	Damage

 6.4.2.2 Constructor

+ Bullet(String image_Path, boolean isRight, int initX, int initY, int width, int height)	Initialize all remaining fields

 6.4.2.3 Methods

Getter/Setter of fields	

6.4.3 Class GunBullet extends Bullet

6.4.3.1 Constructor

+ GunBullet(boolean isRight, int initX, int initY)	Initialize all remaining fields

6.4.4 Class RocketBullet **extends Bullet implements Exploadble**

6.4.4.1 Constructor

+ RocketBullet(boolean isRight, int initX, int initY)	Initialize all remaining fields

6.4.4.2 Methods

+ void explode()	Play explosion sound
	Run explosion animation

6.4.5 Class SwordSlice **extends Bullet**

6.4.5.1 Constructor

+ SwordSlice(boolean isRight, int initX, int initY)	Initialize all remaining fields

6.4.5.2 Methods

+ void createAnimation()	Create sword animation

6.5 Package character

6.5.1 Abstract Class GameCharacter **extends Entity**

6.5.1.1 Fields

# Rectangle lowBox	A box below the character feet
# Rectangle highBox	A box above the character head
# ArrayList<Boolean>	Arraylist of all action
# boolean pensioner	Counter for character pensioner
# Duration turnTime	Turn duration time
# Duration walkTime	Walk duration time
# Duration fireTime	Fire duration time
# int pictureWidth	Image width
# int pictureHeight	Image height

# int pictureOffsetX	Image start point in x axis
# int pictureOffsetY	Image start point in y axis
# int pictureColumn	Number of each image column in sprite
# boolean is OnFloor	Is character on floor
# int currentHP	Current Hp of character
# int maxHP	Max Hp of character
# int point	point
# int finalPositionX	Destination in x axis
# ArrayList<Weapon> weaponInventory	List of character's weapons
# int weaponKey	Current weapon key

6.5.1.2 Constructor

+ GameCharacter(String image_Path, int initX, int initY, int width, int height, int currentHP)	Initialize all remaining fields

6.5.1.3 Methods

+ void createAnimation()	Create animation using sprite
# void createAnimation()	Initialize all action and add to arraylist
+ void resetAction()	Reset all action Boolean to false
+ void doTurnRight()	Play sprite animation and set boolean turn right to true
+ void doTurnLeftt()	Play sprite animation and set boolean turn left to true
+ void doWalkLeft()	Play sprite animation and set boolean walk right to true
+ void doWalkRight()	Play sprite animation and set boolean walk left to true
+ void doFireLeft()	Play sprite animation and set boolean fire right to true
+ void doFireRight()	Play sprite animation and set boolean fire left to true

+ void doDieLeft()	Play sprite animation and set boolean die right to true
+ void doDieRight()	Play sprite animation and set boolean die left to true
+ void jump()	Jump
+ void decreaseCurrentHP(int damage)	Decrease character Hp
+ void increaseCurrentHP(int heal)	Heal character
+ boolean checkTurn(ArrayList<Entity> platforms)	Check if low box of character doesn't collide with any platform box then return true. Also check if high box collides with any platform then return true. Something else return false.
+ void changeWeaponLeft()	Decrease weaponKey
+ void changeWeaponRight()	Increase weaponKey
Getter/Setter of fields	

6.5.2 Class MainCharacter extends GameCharacter

6.5.2.1 Fields

- int point	Point
- boolean isBlink	Invincible state
# ArrayList<ConsumableItem> itemsInventory	Item inventory

6.5.2.2 Constructor

+ MainCharacter(int initX, int initY)	Initialize all remaining fields Create potion, ammo, gun, rocket gun, sword

6.5.2.3 Methods

- void createPotion()	Create potion and add to itemsInventory
- void createAmmo()	Create ammo and add to itemsInventory
- void createGun()	Create gun and add to weaponsInventory

- void createRocketGun()	Create rocket gun and add to weaponsInventory
- void createSword()	Create sword and add to weaponsInventory
+ void blink()	Set opacity
+ void setOpacityNormal()	Set opacity to 1
Getter/Setter of fields	

6.6 Package consumable

6.6.1 Class Ammo extends ConsumableItem implements Consumable

6.6.1.1 Constructor

+ Ammo(int amount)	Initialize all remaining fields Set hotkey

6.6.1.2 Methods

+ void consumed() throws ConsumeItemFailedException	Decrease item when used Add 10 bullet to the carried weapon If item is 0 throw exception If bullet is full throw exception

6.6.2 Abstract Class ConsumableItem extends Item

6.6.2.1 Fields

# int amount	Amount
# char hotKey	Hotkey

6.6.2.2 Constructor

+ ConsumableItem(String image_Path, int width, int height, int amount)	Initialize all remaining fields

6.6.2.3 Methods

+ void addAmount(int amount)	Increase item
Getter of fields	

6.6.3 Class Potion extends ConsumableItem implements Consumable

6.6.3.1 Constructor

+ Potion(int amount)	Initialize all remaining fields
	Set hotkey

6.6.3.2 Methods

+ void consumed()throws ConsumeItemFailedException	Decrease item when used
	Add 100 hp to game character
	If item is 0 throw exception
	If health is full throw exception

6.7 Package effect

6.7.1 Class Barrier extends Effect

6.7.1.1 Constructor

+ Barrier(int initX, initY)	Initialize all remaining fields
	Play sprite

6.7.2 Abstract Class Effect extends Entity

6.7.2.1 Fields

# boolean isInUsed	Status

6.7.2.2 Constructor

+ Effect(String image_Path, int initX, int initY, int width, int height, int amount)	Initialize all remaining fields
	IsInUsed = true

6.7.2.3 Methods

+ void addAmount(int amount)	Increase item
Getter/Setter of fields	

6.7.3 Class Light extends Effect

6.7.3.1 Constructor

+ Light(int initX, initY)	Initialize all remaining fields Play sprite

6.7.4 Class RecoveryLight extends Effect

6.7.4.1 Constructor

+ recoveryLight(int initX, initY)	Initialize all remaining fields Play sprite

6.7.5 Class Warning extends Effect

6.7.5.1 Constructor

+ Warning(int initX, initY)	Initialize all remaining fields

6.8 Package enemy

6.8.1 Class BossEnemy extends GameCharacter

6.8.1.1 Constructor

+ BossEnemy (int initX, initY)	Initialize all remaining fields Create gun and add to inventory Play sprite

6.8.2 Class ColliderEnemy extends GameCharacter

6.8.2.1 Constructor

+ BossEnemy (int initX, initY)	Initialize all remaining fields

6.8.3 Class GunEnemy extends GameCharacter

6.8.3.1 Constructor

+ GunEnemy (int initX, initY)	Initialize all remaining fields Create gun and add to inventory

6.9 Package weapon

6.9.1 Class BombGun extends Weapon implements Fireable

6.9.1.1 Constructor

+ BombGun(int bullet)	Initialize all remaining fields Set maximum bullet to 10 Add bullet

6.9.1.2 Methods

+ void addBullet(int count)	Increase bullet
+ Bullet setPositionBullet(GameCharacter, character, boolean isRight, Bullet bullet)	Set bullet position
+ Bullet fireBullet(GameCharacter, character, boolean isRight) throws FireBulletFailedException	Fire bullet Throws exception if there is not any bullet left in this weapon
+ Bullet fireBulletInfinite(GameCharacter, character, boolean isRight)	Fire bullet without using ammo

6.9.2 Class Gun extends Weapon implements Fireable

6.9.2.1 Constructor

+ Gun(int bullet)	Initialize all remaining fields Set maximum bullet to 15 Add bullet

6.9.2.2 Methods

+ void addBullet(int count)	Increase bullet
+ Bullet setPositionBullet(GameCharacter, character, boolean isRight, Bullet bullet)	Set bullet position
+ Bullet fireBullet(GameCharacter, character, boolean isRight) throws FireBulletFailedException	Fire bullet Throws exception if there is not any bullet left in this weapon
+ Bullet fireBulletInfinite(GameCharacter, character, boolean isRight)	Fire bullet without using ammo

6.9.3 Class RocketGun extends Weapon implements Fireable

6.9.3.1 Constructor

+ RocketGun(int bullet)	Initialize all remaining fields Set maximum bullet to 10 Add bullet

6.9.3.2 Methods

+ void addBullet(int count)	Increase bullet
+ Bullet setPositionBullet(GameCharacter, character, boolean isRight, Bullet bullet)	Set bullet position
+ Bullet fireBullet(GameCharacter, character, boolean isRight) throws FireBulletFailedException	Fire bullet Throws exception if there is not any bullet left in this weapon
+ Bullet fireBulletInfinite(GameCharacter, character, boolean isRight)	Fire bullet without using ammo

6.9.4 Class Sword extends Weapon implements Fireable

6.9.4.1 Constructor

+ Sword(int bullet)	Initialize all remaining fields
	Set maximum bullet to 1
	Add bullet

6.9.4.2 Methods

+ void addBullet(int count)	Increase bullet
+ Bullet setPositionBullet(GameCharacter, character, boolean isRight, Bullet bullet)	Set bullet position
+ Bullet fireBullet(GameCharacter, character, boolean isRight) throws FireBulletFailedException	Fire bullet
	Throws exception if there is not any bullet left in this weapon
+ Bullet fireBulletInfinite(GameCharacter, character, boolean isRight)	Fire bullet without using ammo

6.9.5 Abstract Class Weapon extends Item

6.9.5.1 Fields

# int maxBullet	Maximum bullet
# int currentBullet	Current bullet
# char bulletType	Bullet type
# ArrayList<Bullet> bullets	ArrrayList for bullet

6.9.5.2 Constructor

+ Weapon(String image_Path, int width, int height, int amount)	Initialize all remaining fields

6.9.5.3 Methods

Getter/Setter of fields	

7 package logic

7.1 Class LevelData

7.1.1 Fields

- static final String[] LEVEL1	Array of string represent game map for level 1
- static final String[] LEVEL2	Array of string represent game map for level 2
- static final String[] LEVEL3	Array of string represent game map for level 3
- static final String[][] ALLLEVELMAP	Array of all array map
- static LevelData levelD	LevelData data
- Pane gameRoot	Game pane for component
- ArrayList<Entity> platforms	ArrayList of all game box object
- ArrayList<Portal> portalList	ArrayList of all portal object
- BossEnemy boss	Boss enemy
- ArrayList<GameCharacter> enemieList	ArrayList of all enemy object
- ArrayList<Entity> playerInteractEntity	ArrayList of all object that can interact with player
- Rectangle bg	background
- int levelWidth	Level width
- int levelHeight	Level height

7.1.2 Constuctor

+ LevelData()	Initialize all field

7.1.3 Methods

+ LevelData getInstance()	Initialize all data
+ void loadLevel(int level)	turn all string data into object form specific level
Generate getter of all fields	Level height

7.2 Class Leaderboards

7.2.1 Fields

- static Leaderboards lBoard	leaderboard
- String filePath	Leaderboard data File path
- String highScores	File name
- ArrayList<String> topPlayer	Arraylist of 10 top player names
- ArrayList<Integer> topScores	Arraylist of 10 top player scores
- ArrayList<Double> topTimes	Arraylist of 10 top player time records

7.2.2 Constuctor

+ Leaderboards()	Initialize all field

7.2.3 Methods

+ Leaderboard getInstance()	Initialize all data
+void addPlayerScore(String name, int score, Double time) throws AddLeaderboardScoresFailedException	Add player name, score, time to data arraylist Name must be 1-9 characters or number. If name is more than 9 throw exception. If name contains marks or spaces throw exception. If name is the same with old one throw exception

+ void loadScore()	Load data
+ void saveScores()	Save data
- void createSaveData()	Create save data incase it doesn't have old data
Generate getter of all fields	

7.3 Class GameController

7.3.1 Fields

- HashMap<KeyCode, Boolean> keys	HashMap of boolean keycode that pressed
- MainCharacter player	Main character
- Pane gameRoot	Game pane for component
- Double time	time
- LevelData levelD	map from LevelData
- ArrayList<Entity> platforms	Arraylist of box
- ArrayList<Portal> portalList	Arraylist of portal
- ArrayList<Entity> playerInteractEntity	Arraylist of entity that can interact with main character
- ArrayList<Bullet> playerBullets	Arraylist of bullet that fired from main character
- ArrayList<Entity> timedEntityList	Arraylist of entity that has been timed to remove
- ArrayList<Bullet> enemyBulletsX	Arraylist of bullet that fired from enemy character
- ArrayList<Bullet> enemyBulletsY	Arraylist of bullet that fired from enemy character
- Rectangle bg	background
- Boolean isWin	Check that is player win the game
- int counter	Loop counter
- BossEnemy boss	Boss character

- ArrayList<GameCharacter> enemieList	Arraylist of enemy character
- int levelWidth	Level map width
- int levelHeight	Level map height
- boolean alreadyPressedQ	Check that is player pressed already pressed Q
- boolean alreadyPressedE	Check that is player pressed already pressed E
- boolean alreadyPressedW	Check that is player pressed already pressed W
- boolean alreadyPressedSPACE	Check that is player pressed already pressed Spacebar
- boolean alreadyPressedH	Check that is player pressed already pressed H
- boolean alreadyPressedJ	Check that is player pressed already pressed J
- boolean isEnemyFire	Check that is enemy already fire bullet
- int offsetX	Player position in X axis
- int offsetY	Player position in Y axis
- boolean isBarrierOpen	Check that is game character open barrier
- boolean islight	Check that is game character open light
- boolean isRecoverylightOpen	Check that is game character open recoverylight
- Light light	light
- RecoveryLight recoveryLight	recoverylight
- Barrier barrier	barrier
- boolean isBossStart	Check that is boss start
- boolean isBossMove	Check that is boss move
- int pensioner	Number of boss pensioner
- boolean isGameEnd	Check that is the game end

7.3.2 Constuctor

+ GameController()	Initialize all field

7.3.3 Methods

- void createLevel(int level)	Load map from level data
+ void getControl()	Call all the movement method and keyboard handler
- void createPlayer()	Create main character
- void check()	Check all the check collision method
- void gravity()	Move some item in y axis like gravity
- void fireBullet(boolean)	Call the weapon fire method to fire bullet from main character
- void allEnemyFireBullet()	fire bullet from all enemy character if game character enters near enemy
- void bossFireBullet()	fire bullet from boss if boss mode start is start
- Bullet enemyFireMethod(GameCharacter enemy)	Call the weapon fire bullet method for specific enemy
- void checkPortal()	Check if main character collides with portal
- void moveBullets()	Call movePlayerBullet if the bullet is in playerBullets. Call moveEntityX if the bullet is in EnemyBulletsX.
- void moveEnemies()	Call moveEnemyX for all the enemy in enemy arraylist
- boolean isPressed(KeyCode keycode)	Return true if specific keycode is pressed
- boolean isKeyboardPressed()	Return false if all the keycode is not pressed. Return true if has one keycode pressed.
- void movePlayerX(int value)	Move player in right if value is positive.

	Move player to left if value is negative.
- void movePlayerBulletX(Entity item, int value)	Move all bullet in a fired direction (if fired right move right, if fired left move left) with specific velocity value If enemy bullet collide with main character then decrease main character health.if enemy bullet collide with platform then remove. If the bullet can explode then explode the bullet when hit something.
- void moveEnemybulletX(Entity item, int value)	Move all bullet in a fired direction (if fired right move right, if fired left move left) with specific velocity value. If enemy bullet collide with main character then decrease main character health.if enemy bullet collide with platform then remove. If the bullet can explode then explode the bullet when hit something.
- void moveEnemyX(GameCharacter enemy, int value)	Move specific enemy in a specific velocity value.
- void moveBossX()	move boss to specific location if hit the ground or arrive at the final position stop boss moving
- void moveBossTo(int, int)	Jump the boss and set the final position
- void moveEntityY(Entity entity, int value)	Move specific entity with specific velocity value
- boolean checkEnemyCollision(Entity entity)	Check is enemy collided with specific entity
- boolean checkBossCollision(Entity entity)	Check is boss collided with specific entity
- void checkPlayerCollision()	Check for all entity in playerInteractEntity

- void characterDie(GameCharacter gameCharacter)	Make a character die .then, remove it.
- void timedEntity(Entity entity, Double duration)	Timed the entity with a specific duration
- void checkTime()	Check all entity in timed list to remove it
- void randomGunEnemyInRange(int X1, int X2, int Y1, int Y2)	Spawn enemy in range of specific location
- void removeEntity(Entity entity)	Remove specific entity from game root
- void checkBossCollision(Entity entity)	Check that is boss collide with main character bullet
- void removeAll()	Remove all field
- void createBarrier()	Create barrier around main character
Generate getter of all fields	

8 package view

8.1 Class ViewManager

8.1.1 Fields

- static final int HEIGHT	Height of main scene
- static final int WIDTH	Width of main scene
- AnchorPane mainPane	Main pane

- Scene mainScene	Main scene
- int MENU_BUTTONS_START_X	Layout of first button in x axis
- int MENU_BUTTONS_START_Y	Layout of first button in y axis
- GameSubScene HelpSubScene	Help Subscene show help infomation
- GameSubScene CreditSubScene	credit Subscene show creator name
- GameSubScene ScoreSubScene	score Subscene show 10 highest score player
- GameSubScene StartSubScene	start Subscene show start menu
- GameSubScene NowShowing	Now showing Subscene
- boolean hasGameStage	Check that player already start the game
- GameViewManager gameManager	Main gamemanager
- Stage gameStage	Game stage
- AudioClip MenuThemeSong	Game theme song (starcommander)

8.1.2 Constructor

+ ViewManager()	Initialize all remaining fields

8.1.3 Methods

+stage getMainStage()	Return mainStage
- void keyboardListener()	Initialize keyboard listener
- void CreateSubScenes()	Create all subscene field

- void ShowSubScene(GameSubScene gameSubscene)	Show new subscene if there are other one that already show hide it before show the new one.
- void addMenuButton(GameButton button)	Add all button to menuButtons arraylist
- void createButtons()	Call all create button method
- void createStartButton()	Create start button
- void createScoreButton()	Create score button
- void createHelpButton()	Create help button
- void createCreditButton()	Create credit button
- void createQuitButton()	Create quit button
- void createLogo()	Create logo image
- void createBackground()	Create background image

8.2 Class GameViewManager

8.2.1 Fields

- long startTime	Time when start the animation timer loop
- DoubleProperty time	Time from animation timer loop
- static final int HEIGHT	Height of scene
- static final int WIDTH	Width of scene
- AnchorPane gamePane	Game pane

- Scene gameScene	Game scene
- Stage gameStage	Game stage
- Stage menuStage	Start menu stage
- AnimationTimer gameTimer	Game timer
- boolean alreadyPressedESCAPE	Check that is keyboard pressed escape button
- boolean isPause	Check that is game timer stop
- HashMap<KeyCode, Boolean> keys	Hash map of keycode from keyboard
- Boolean isGameover	Check that is Game end
- AudioClip gameThemeSong	Game theme song
- AudioClip menuThemeSong	start menu theme song
- Pane gameRoot	Game pane for component
- GameController player1Controller	Game controller
- PauseMenu pauseMenu	Pause menu pane
- PlayerInfoBox playerInfoBox	Player info pane
- PlayerInventoryBox playerInventory	Player inventory pane
- GameEndingScene gameEndingScene	Game ending pane

8.2.2 Constructor

+ GameViewManager()	Call inititializeStage()

8.2.3 Methods

+ void createNewGame(Stage)	Show game stage and hide start menu stage
+ void createGameLoop()	Start the animation timer loop
- void update()	Update all the field
- void keyboardListener()	Keyboard handler
- Boolean isPressed(KeyCode keycode)	Return true if keycode is pressed Return false if keycode isn't pressed
- void createPlayerInfo()	Create player information Box and player inventory
- void createMenu()	Create game pauseMenu
- void createEndSubScene()	Create ending pane
+ void removeAll()	Clear all field
+ void continueThemeSong()	If sound in pauseMenu is close set game theme song stop. If sound is on set play game theme song.
Generate getter of all fields	