Garden Escape

Ducumentation

created by

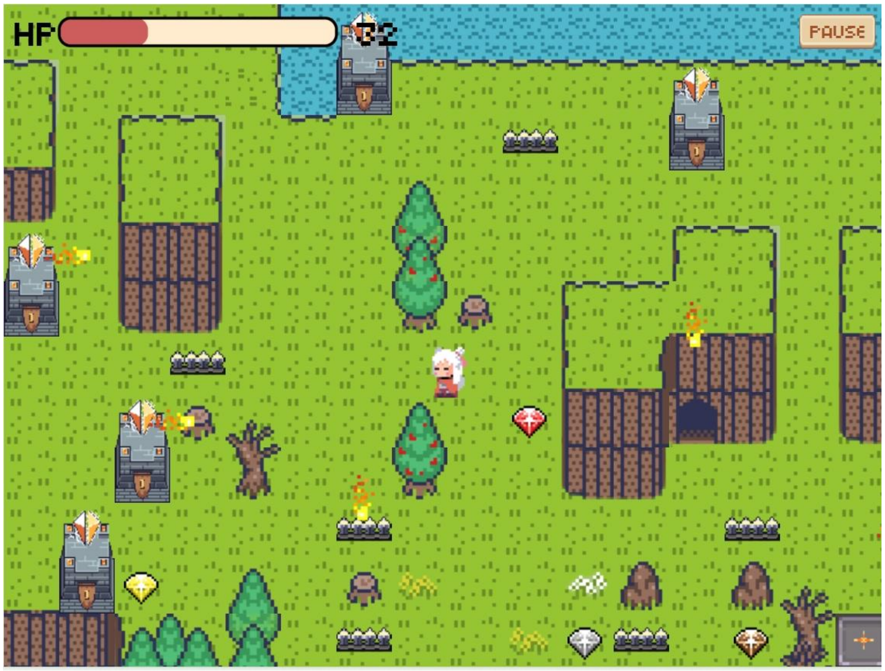
Sataporn Kruatejah	6430368521
Sarita Taveemunputtakarn	6432181821

2110215 Programming Methodology

Semester 1 year 2022

Chulalongkorn University

Garden Escape



You are lost in this mysterious garden.

Hmm Where am I? Whoa! What is that? That little red thing is coming this way. It’s so gorgeous, maybe I can touch it. Ahhhhhh My hand is burning!!! This place is insane. I have to get out.

This Game is an 2D Role-playing game (RPG) in which the player has to go to the endpoint at the bottom-right by walking through the garden. But It’s not that simple. Let’s read this following rules.

The rule :

1. Walk wisely cause **the more you walk the more HP decrease**.
2. Beware of **the fireball**. Even if it’is magnificent, It can **reduce your HP by 30**.
3. Watch your step! If be careless, Your HP can instantly **decrease by 40 from stepping on the trap.**

It’s that been too hard?

This garden still have some kindness

✧ *IF U SEE THE TWIKLE LITTLE STAR* ✧

*THAT’S A **DIAMOND!!!!***

It can +15 HP

— You can move by using the arrow keyboard —

Overview :

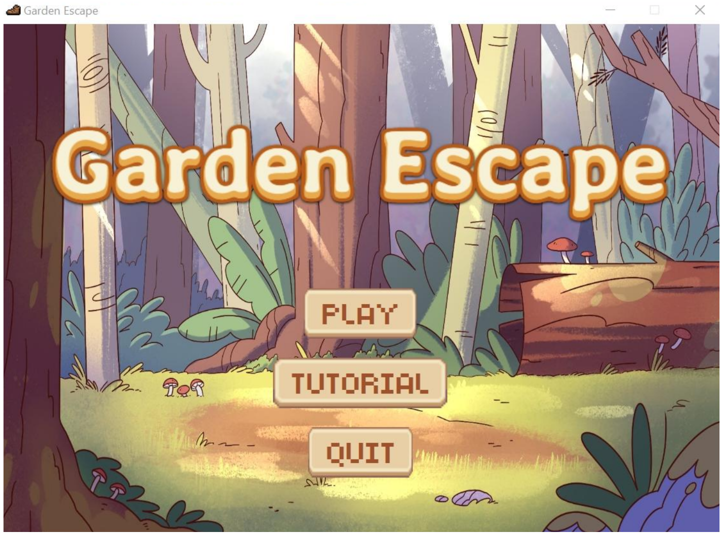
1. Main Menu Screen

- When open the game, It will show the main menu screen.
- There are 3 buttons :

  PLAY button -> click to switch to Stage Select Screen [3]

  TUTORIAL button -> click to switch to Tutorial Screen [2]

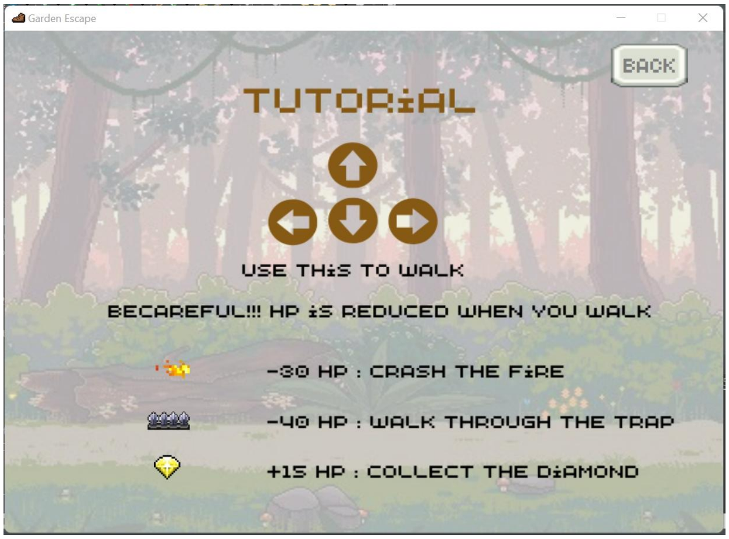
  QUIT button -> click to close the game



Main Menu Screen

2. Tutorial Screen

- Show the rule of this game
- There are BACK button -> click to switch to Main Menu Screen [1]



TUTORIAL

USE THIS TO WALK

BECAREFUL!!! HP IS REDUCED WHEN YOU WALK

-30 HP : CRASH THE FIRE

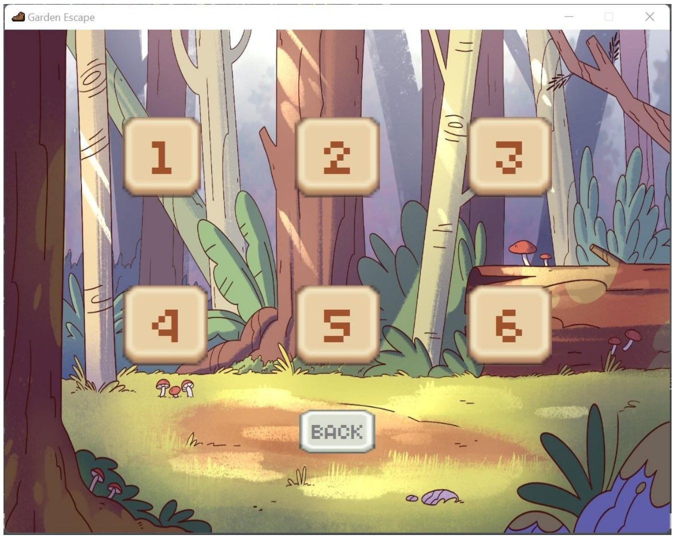
-40 HP : WALK THROUGH THE TRAP

+15 HP : COLLECT THE DIAMOND

Tutorial Scree

3. Stage Select Screen

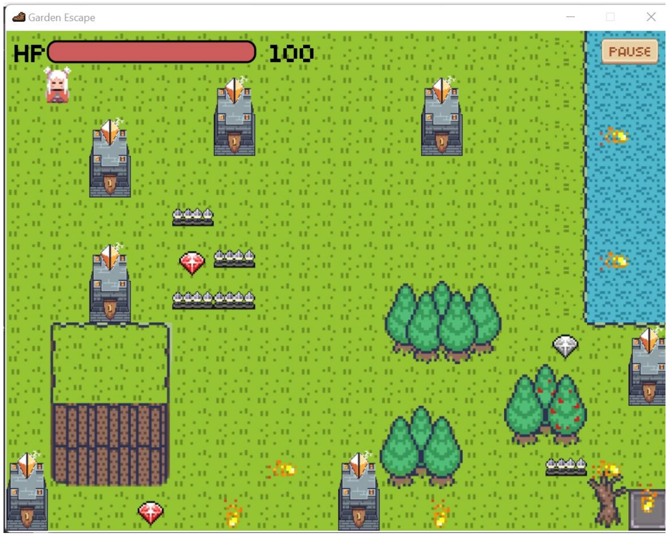
- There are buttons of 6 Stage -> click to switch to Main Menu Screen [4]
- There are BACK button -> click to switch to Main Menu Screen [1]



Stage Select Screen

4. Stage Game Screen

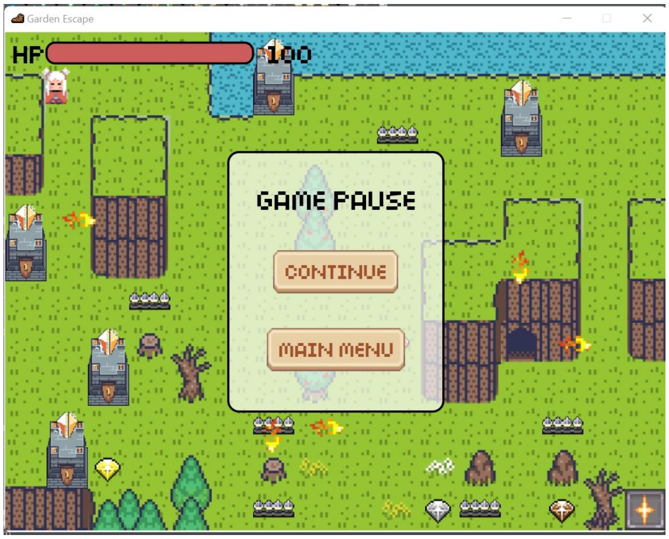
- Show the Game Screen [YOU CAN PLAY NOW]
- There are PAUSE button -> click to pause game and show Option Window [5]
- If your HP = 0 : You DIE -> show Die Window [6]
- If you reach the endpoint : Game Success -> show Success Window [7]
- If you collide with fireball or diamond, it will dissapear



Stage Game 4 Screen

5. **Obtion Window**

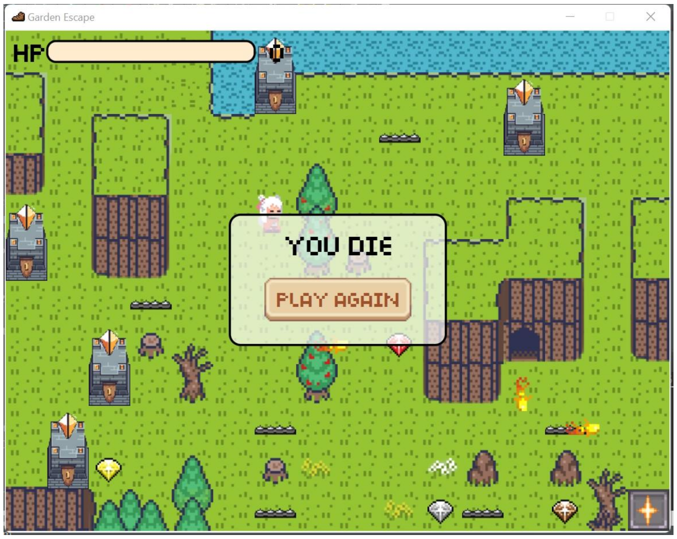
- Game pause now : Show the Obtion Window and the PAUSE button disapear
- There are 2 button
    - CONTINUE button -> click to continue the game and the obtion window will disappear and PAUSE button will comeback
    - MAIN MENU button -> click to switch to Main Menu Screen [1]



Obtion Window

6. **Die Window**

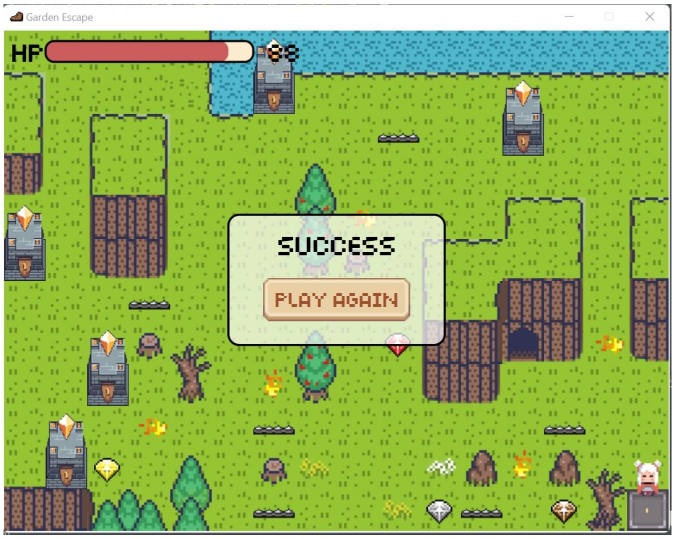
- You DIE now : Show the Die Window
- There are PLAY AGAIN button -> click to switch to Main Menu Screen [1]



Die Window

7. **Success Window**

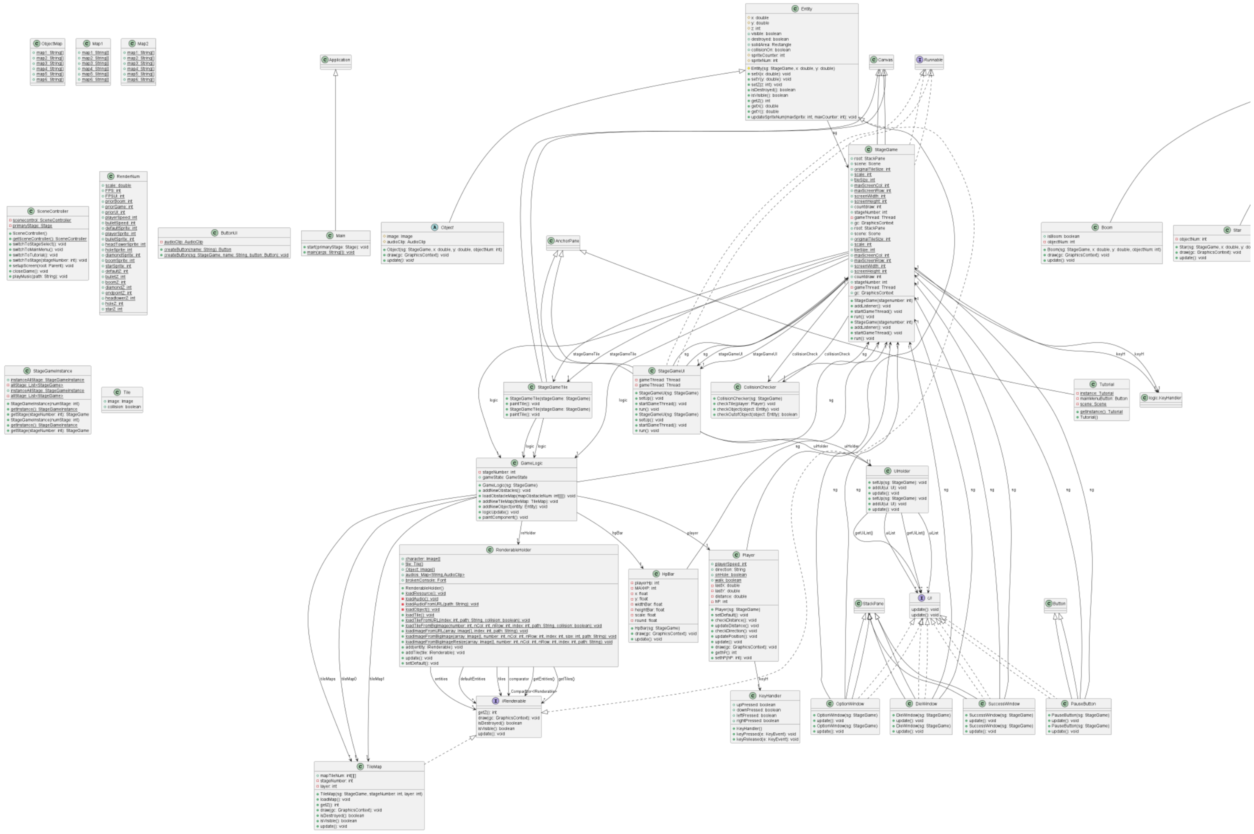
- Game success now : Show the Success Window
- There are PLAY AGAIN button -> click to switch to Main Menu Screen [1]



Success Window

Class Diagram

also include in GitHub Repositories



1. Package entity

1.1 Abstract class Entity implements IRenderable

This class is the base class of all Entities in this Game.

1.1.1 Fields

# StageGame sg	The stage game which is currently playing.
# double x	Position of the entity in X-axis
# double y	Position of the entity in Y-axis
# int z	Position of the entity in Z-axis
+ boolean visible	Represent that the entity is visible or not
+ boolean destroyed	Represent that the entity has been destroyed or not
+ Rectangle solidArea	The area in which the entity is solid (Use to detect collision)
+ boolean collisionOn	Represent that the entity is collided with others or not

# int spriteCounter	-The Counter for the sprite loop animation of the entity -Default is 0
# int spriteNum	-The Number of Sprite Image of the entity -Default is 1

1.1.2 Constructor

# Entity(StageGame sg, double x, double y)	Constructor method Initializes with the following specifications: Initialize this StageGame sg with sg Set the entity to be visible Set the entity to not be destroyed Set the entity to not be collided Initialize this x with x Initialize this y with y Initialize this z with z Assign solid area with new Rectangle() Set x and y of the solid area to 0 Set width and height to tile size using StageGame.tileSize

1.1.3 Methods

+ void updateSpriteNum(int maxsprite, int maxCounter)	Using to update the sprite number of the sprite image used for loop animation of the entity -count up the spriteCounter -If spriteCounter is more than maxCounter : If spriteNum equals to maxSprite, reset spriteNum to 1. If not, add up spriteNum.

	Reset spriteCounter to 0
+void setX(double x)	set x : between 0 and StageGame.screenWidth - StageGame.tileSize
+void setY(double y)	set y : between 0 and StageGame.screenHeight - StageGame.tileSize
+ Getters and setters for each field include the one that overrides.	

1.2 Class HpBar

This class is the Bar that shows the current HP of Player

1.2.1 Fields

- int playerHp	Represent current player's HP
- final int MAXHP	Represent player's Max HP
- StageGame sg	The stage game which is currently playing
- float x	Position of the HpBar in X-axis
- float y	Position of the HpBa in Y-axis
- float widthBar	Width of the HpBar
- float heightBar	Height of the HpBar
- float scale	Scale of HpBar calculated from widthBar/MAXHP
- float round	round of rectangle of the HpBar

1.2.2 Constructor

+ HpBar(StageGame sg)	Constructor method Initializes with the following specifications: Initialize this StageGame sg with sg

1.2.3 Methods

+ void draw(GraphicsContext gc)	-Fill Text of gc with "HP" : Position = (x - 40, y+heightBar)

	Font : RenderableHolder.brokenConsole
	Color : Color.BLACK
	-Fill RoundRect of gc :
	Position = (x,y)
	Size = (widthBar,heightBar)
	Round = (this.round,this.round)
	Color : Color.BLANCHEDALMOND
	-Fill RoundRect of gc :
	Position = (x,y)
	Size = (playerHp * scale,heightBar)
	Round = (this.round,this.round)
	Color : Color.INDIANRED
	-Stroke RoundRect of gc :
	Position = (x,y)
	Size = (widthBar,heightBar)
	Round = (this.round,this.round)
	LineWidth = 2.5
	Color : Color.INDIANRED
	-Fill Text of gc with playerHp :
	Position = (x + widthBar + 16, y+heightBar)
	Font : RenderableHolder.brokenConsole
	Color : Color.BLACK

+ void update()	Assign this.playerHp with currently player’ HP using sg.logic.player.gethP()

1.3 Abstract class Object extends Entity

This class is the entity that is the base class of all object in this Game.

1.3.1 Fields

# Image image	Image of this Object
# AudioClip audioClip	AudioClip which is sound effect of this Object

1.3.2 Constructor

+ Object(StageGame sg, double x, double y, int objectNum)	Constructor method Initializes with the following specifications: Initialize with super constructor assign image with RenderableHolder.Object[objectNum]

1.3.3 Methods

+ void draw(GraphicsContext gc)	drawImage in gc with this.image at (x, y)
+ void update()	this method should never be called with from here, it can only be called from subclasses.

1.4 Class Player extends Entity

This is the entity represent player of the game.

1.4.1 Fields

+ final int playerSpeed	player' speed equals to RenderNum.playerSpeed
+ String direction	represent direction that playing moving to
+ boolean onHole	state that this player is on hole or not
+ boolean walk	state that this player is walking or not
+ double lastX	previous position of player in X-Axis
+ double lastY	previous position of player in Y-Axis
- double distance	distance that this player walk
- int hP	hp of this the player

# KeyHandler keyH	Using for checking key input from keybord

1.4.2 Constructor

+ Player(StageGame sg)	Constructor method Initializes with the following specifications: Initialize with super constructor set this.keyH to sg.keyH setdefault() setsolidArea: position = (16,16) Size = (16,25) set z to 10

1.4.3 Methods

+ void setDefault()	set x = 35;
	set y = 35;
	set direction = "down"
	set hP = 100
	set onHole = false
	set walk = false
	set distance = 0
	set lastX = x
	set lastY = y

+ void checkDistance()	if distance >= StageGame.tileSize * 2 : decrease 1 hp reset distance to 0

+ void updateDistance()	if this player is walking update distance that increases from last position Note : this game can move in one direction each time
+ void checkDirection()	check what direction that player move using keyH
+ void updatePosition()	check whether the tile can move into or not if can, update position up to the direction

+ void update()	Update the player :
	-updateDistance()
	-checkDistance()
	if the keyboard is pressed:
	checkDirection()
	updatePosition()
	and if the plyer not walk, set walk = true
	if not be pressed:
	set walk = false

+ void draw(GraphicsContext gc)	reset image to null draw this player with differen image up to the direction
+ int gethP()	getter of hP
+ void sethP(int hP)	set hp : between 0 and 100

2. Package logic

2.1 Class CollisionChecker

2.1.1 Fields

- StageGame sg	The stage game which is currently playing.

2.1.2 Constructor

+ CollisionChecker(StageGame sg)	Constructor method

	Initializes with the following specifications: assign this.sg with sg

2.1.3 Methods

+ void checkTile(Player player)	check that the tile that player moving to is collisionOn or not
+ void checkObject(Entity object)	check that the object is collide with player or not
+ boolean checkOutofObject(Entity object)	check that the player moved out of the object or not

2.2 Class GameLogic

2.2.1 Fields

- StageGame sg	The stage game which is currently playing.
- int stageNumber	The number of stage which is currently playing.
+ HpBar hpBar	player HP
+ Player player	player of this game
+ RenderableHolder reHolder	RenderableHolder for current game stage
+ GameState gameState	current game state
+ enum GameState	enum class of game state contains : PLAY, PAUSE, STOP, DIE, SUCCESS
+ ArrayList<TileMap> tileMaps	array that keep all layer of map
+ TileMap tileMap0	map layer 0
+ TileMap tileMap1	map layer 1

2.2.2 Constructor

+ GameLogic(StageGame sg)	Constructor method Initializes with the following specifications:

	intitialize reHolder intialize tileMaps assign this.sg with sg initialize tileMap0 then addNewTileMap intitialize tileMaps1 then addNewTileMap initialize player with this.sg initialize hPBar with this.sg

2.2.3 Methods

+ void addNewObstacles()	create new obstacles from the loadObstacleMap() and addNewObject() for each
+ void loadObstacleMap(int[][] mapObstacleNum)	load obstacles from the array map which is in Class ObjectMap
+ void addNewTileMap(TileMap tileMap)	add tileMap to the array tileMaps and add this tile to this.reHolder
+ void addNewObject(Entity entity)	add new object to this.reHolder
+ void logicUpdate()	-if HP equals to 0: play sound effect then set gamestate to DIE -if gamestate equals to PLAY: update player and reHolder
+ void paintComponent()	draw each entity in reHolder in gc draw player

2.3 Class KeyHandler

2.3.1 Fields

+ boolean upPressed	state that press up key or not
+ boolean downPressed	state that press down key or not

+ boolean leftPressed	state that press left key or not
+ boolean rightPressed	state that press right key or not

2.3.2 Constructor

+ KeyHandler()	Do nothing

2.3.3 Methods

+ void keyPressed(KeyEvent e)	update the field up to the key that pressed to be true
+ void keyReleased(KeyEvent e)	update the field up to the key that released to be false

3. Package map

array of map for the game

3.1 Class Map1

array tile map layer 0

3.1.1 Fields

+ String[] map1	array map layer 0 of stage1
+ String[] map2	array map layer 0 of stage2
+ String[] map3	array map layer 0 of stage3
+ String[] map4	array map layer 0 of stage4
+ String[] map5	array map layer 0 of stage5
+ String[] map6	array map layer 0 of stage6

3.2 Class Map2

array tile map layer 1

3.2.1 Fields

+ String[] map1	array map layer 1 of stage1
+ String[] map2	array map layer 1 of stage2
+ String[] map3	array map layer 1 of stage3
+ String[] map4	array map layer 1 of stage4
+ String[] map5	array map layer 1 of stage5
+ String[] map6	array map layer 1 of stage6

3.3 Class ObjectMap

array object map

3.3.1 Felds

+ String[] map1	object map of stage1
+ String[] map2	object map of stage2
+ String[] map3	object map of stage3
+ String[] map4	object map of stage4
+ String[] map5	object map of stage5
+ String[] map6	object map of stage6

4. Package object

4.1 Class Boom extends entity.Object

This is the object that will draw if the bullet collide with player. but we don’t use it cause it make our game lag.

4.1.1 Fields

+ boolean isBoom	state that boom or not
- int objectNum	number of object

4.1.2 Constructor

+ Boom(StageGame sg, double x, double y, int objectNum)	Constructor method

	Initializes with the following specifications: assign this.sg with sg set this.x with x set this.y with y initialize image with RenderableHolder.Object[objectNum] initialize this.visible with true set this.z with RenderNum.boomZ

4.1.3 Methods

+ void draw(GraphicsContext gc)	Initialize image with RenderableHolder.Object[objectNum + spriteNum - 1] draw image with x,y in gc
+ void update()	call updateSpriteNum(12, RenderNum.boomSprite) if spriteNum = 12, set this.visible with false

4.2 Class Bullet extends entity.Object

This is the object that is one of the obstacles that will decrease HP.

4.2.1 Fields

+ int speed	Speed of bullet
- double xStart	start point of bullet that shot in x-Axis
- double yStart	start point of bullet that shot in y-Axis
- int timer	timer for bullet collision thread
- int objectNum	number of object

4.2.2 Constructor

+ Bullet(StageGame sg, int x, int y, int objectNum, int speed)	Constructor method Initializes with the following specifications:

	assign this.sg with sg set this.x with x set this.y with y initialize image with RenderableHolder.Object[objectNum] intialize audioClip with RenderableHolder.audios.get("effect/hit1") set this.speed to speed set this.xStart to x set this.yStart to y set this.sg to sg set this.z with RenderNum.bulletZ set this.objectNum to objectNum new HeadTower(sg, x, y, 422)

4.2.3 Methods

+ void moveY()	set y with y+this.speed if y > 576: set collisionOn with false set y with yStart set visible with true
+ void moveX()	set x to x+this.speed if y > 768: set collisionOn with false set x with xStart set visible with true
+ void move()	if objectNum = 379, call moveY().

	if objectNum = 365, call moveX().
+ void draw(GraphicsContext gc)	Initialize image with RenderableHolder.Object[objectNum + spriteNum - 1]. draw image with x,y in gc.
+ void update()	-create new thread that checks whether if the bullet has already pass. If passed collisionOn will be set to false. -call move() to update bullet position -if collisionOn is false: check that the object is collide or not if collisionOn is true, call audioClip.play() if isVible(), decrease hp by 30 set visible to false run thread -call updateSpriteNum

4.3 Class Diamond extends entity.Object

This is the object that will increase HP.

4.3.1 Constructor

+ Diamond(StageGame sg, int x, int y, int objectNum)	Constructor method Initializes with the following specifications: call super constructor set audioClip setZ to RenderNum.diamondZ

4.3.3 Methods

+ void draw(GraphicsContext gc)	draw image at (x + 8, y + 8 + (spriteNum * 2))
+ void update()	This method update the diamond by

	-Check this object is collide with player or not -if collisionOn : play audio increase player’s HP with 15 set destroyed to true set collisionOn to false -if not : set visible = true set destroyed = false -updateSpriteNum

4.4 Class Endpoint extends entity.Object

This is the object is the destination of each stageGame.

4.4.1 Constructor

+ Endpoint(StageGame sg, int x, int y, int objectNum)	Constructor method Initializes with the following specifications: call super constructor set audioClip create new Star setZ to RenderNum.endpointZ

4.4.2 Methods

+ void update()	This method update the endpoint by -Check this object is collide with player or not -if collisionOn : play audio

	set state game to SUCCESS set collisionOn to false

4.5 Class HeadTower extends entity.Object

This object is the source of bullet object.

4.5.1 Fields

- int objectNum	number of the image of this object

4.5.2 Constructor

+ HeadTower(StageGame sg, double x, double y, int objectNum)	Constructor method Initializes with the following specifications: call super constructor set this objectNum to objectNum add this as NewObject setZ to RenderNum.headtowerZ

4.5.3 Methods

+ void draw(GraphicsContext gc)	assign image with RenderableHolder.Object[objectNum + spriteNum - 1]
+ void update()	updateSpriteNum

4.6 Class Hole extends entity.Object

This object is a trap that will decrease plyer HP

4.6.1 Fields

- double onHoleX	state the x position of hole which stepped on
- double onHoleY	state the y position of hole which stepped on
- int objectNum	number of the image of this object

4.6.2 Constructor

+ Hole(StageGame sg, int x, int y, int objectNum)	Constructor method Initializes with the following specifications: call super constructor set this objectNum to objectNum add this as NewObject setZ to RenderNum.holeZ set onHoleX and onHoleY to -1

4.6.3 Methods

+ void draw(GraphicsContext gc)	-assign image with RenderableHolder.Object[objectNum + spriteNum - 1] -draw image at (x,y)
+ void update()	This method update the hole by -updateSpriteNum -if not on Hole : check object is collide with player or not if yes : play audio set player onHole keep position that hole is decrease HP by 30 -if on hole and if this hole is the same hole that be stepped and player step out : set collision to false set player not on hole set onHoleX and onHoleY to -1

4.7 Class Star extends entity.Object

This is the effect of endpoint.

4.7.1 Fields

- int objectNum	number of the image of this object

4.7.2 Constructor

+ Star(StageGame sg, double x, double y, int objectNum)	Constructor method Initializes with the following specifications: call super constructor set this objectNum to objectNum add this as NewObject setZ to RenderNum.starZ draw this

4.7.3 Methods

+ void draw(GraphicsContext gc)	draw image with RenderableHolder.Object[objectNum + spriteNum - 1]
+ void update()	updateSpriteNum

5. Package scene

5.1 Class AudioController

Use for control the background music

5.1.1 Fields

+ final AudioController audioController	static field that is AudioController
+ AudioClip currentAudioClip	currently playing music background

5.2 Class ButtonUI

Use for create button

5.2.1 Fields

- AudioClip audioClip	sound effect when click button

5.2.2 Methods

+ Button createButton(String name)	create button for scene that’s not a stage game
+ void createButton(StageGame sg, String name, Button button)	create button for stage game scene

5.3 Class MainMenu extends VBox

This is the VBOX of main menu scene

5.3.1 Fields

- Scene scene	Scene of this pane
- static final MainMenu instance	static field which is MainMenu
- ArrayList<Button> Buttons	static ArrayList that contains buttons of this pane

5.3.2 Constructor

+ MainMenu()	Constructor method : -create new scene and set up -create VBOX that contains game name image -create VBOX that contains play, tutorial,quit button -add both VBOX to this pane

5.3.3 Methods

+ MainMenu getInstance()	return static field instance
+ void setupButton(String name)	add button to ArrayList Buttons

5.4 Class RenderNum

static final number using for rendering

5.4.1 Fields

+ final double scale	scale for FPS
+ final int FPS	FPS of gameStage thread
+ final int FPSUI	FPS of UI thread
+ final int priorBoom	priority of Boom thread
+ final int priorGame	priority of gameStage thread
+ final int priorUI	priority of UI thread
+ final int playerSpeed	speed of player
+ final int bulletSpeed	speed of the bullet
+ final int defaultSprite	default spriteTicker
+ final int playerSprite	spriteTicker for player

+ final int bulletSprite	spriteTicker for bullet
+ final int headTowerSprite	spriteTicker for headTower
+ final int holeSprite	spriteTicker for hole
+ final int diamondSprite	spriteTicker for diamond
+ final int boomSprite	spriteTicker for boom
+ final int starSprite	spriteTicker for star
+ final int defaultZ	default position in Z-Axis for
+ final int bulletZ	position in Z-Axis for bullet
+ final int boomZ	position in Z-Axis for boom
+ final int diamondZ	position in Z-Axis for diamond
+ final int endpointZ	position in Z-Axis for endpoint
+ final int headtowerZ	position in Z-Axis for headtower
+ final int holeZ	position in Z-Axis for hole
+ final int starZ	position in Z-Axis for star

5.5 Class SceneController

This class contains method that control stage switching.

5.5.1 Fields

- SceneController	static field which is SceneController
- Stage primaryStage	static Stage which is tha main and only one Stage of this game

5.5.2 Constructor

+ SceneController()	Constructor method Initializes with the following specifications: Initialize primaryStage setup primaryStage

5.5.3 Methods

+ SceneController getSceneController()	return static field scenecontrol
+ void switchToStageSelect()	setup screen for stage select scene then show
+ void switchToMainMenu()	setup screen for main menu scene then show
+ void switchToTutorial()	setup screen for tutorial scene then show
+ void switchToStage(int stageNumber)	setup screen for stage game scene then show
+ void setupScreen(Parent root)	set scene for primaryStage
+ void closeGame()	close game
+ void playMusic(String path)	play current music from AudioController

5.6 StageSelect extends VBox

5.6.1 Fields

- final StageSelect instance	static field instance which is StageSelect
- ArrayList<StageButton> stageButtons	static ArrayList that contains Buuton for StageSelect scene
- GridPane stageGrid	Grid for contains buttons
- Button mainMenuButton	back to main menu button
- Scene scene	scene for this StageSelect pane

5.6.2 Constructor

+ StageSelect()	Constructor method : -create new scene and set up -create mainmenu button with “BACK” and setup -create stageGrid that contains all stage Buttons -add all every node to this pane

5.6.3 Methods

+ StageSelect getInstance()	return static field instance
+ void setupStageButton(int i)	create new stageButon then add to ArrayList stageButtons and setOnAction

5.6.4 Inner Class StageButton

This class is the button that use to choose stage game to play.

5.6.4.1 Fields

- int stageNumber	number of stage of this button

5.6.4.2 Constructor

+ StageButton(int number)	-call super constructor -set stageNumber

5.6.4.3 Methods

+ int getStageNumer()	return stageNumber

5.7 Class Tutorial extends AnchorPane

5.7.1 Fields

- final Tutorial instance	static field instance which is Tutorial
- Button mainMenuButton	click to switch to mainmenu scene
- Scene scene	scene for this Tutorial pane

5.7.2 Constructor

+ Tutorial()	Constructor method : -create new scene and set up -create mainmenu button with “BACK” and setup -create Imageview fire,trap,diamond -add all every node to this pane

5.7.3 Methods

+ Tutorial getInstance()	return static field instance

6. Package scene.StageGame

6.1 Class DieWindow extends StackPane implements UI

6.1.1 Fields

- StageGame sg	stage game that this window exists

6.1.2 Constructor

+ DieWindow(StageGame sg)	Constructor method : -assign this sg with sg -set up this pane -create rectangle for window and setup -create text “YOU DIE” and setup -create play again button and setup -create detail VBOX contains text and buttons then add this and window rectangle to this pane

6.1.3 Methods

+ void update()	update this window to show when game state is DIE

6.2 Class OptionWindow extends StackPane implements UI

6.2.1 Fields

- StageGame sg	stage game that this window exists

6.2.2 Constructor

+ OptionWindow(StageGame sg)	Constructor method : -assign this sg with sg -set up this pane -create rectangle for window and setup -create text “Game Pause” and setup -create continue button and mainmenu then setup -create detail VBOX contains text, continue button and mainmenu then add this and window rectangle to this pane

6.2.3 Methods

+ void update()	update this window to show when game state is PAUSE

6.3 Class PauseButton extends Button implements UI

6.3.1 Fields

- StageGame sg	stage game that this button exists

6.3.2 Constructor

+ PauseButton(StageGame sg)	Constructor method : -assign this sg with sg -create new pause button then setup

6.3.3 Methods

+ void update()	update this button to show when game state is not PAUSE and vice versa

6.4 Class StageGame extends Canvas implements Runnable

6.4.1 Fields

+ StackPane root	pane that is root of this scene
+ Scene scene	scene for this StageGame pane
+ StageGameTile stageGameTile	stageGameTile for this Stage Game
+ StageGameUI stageGameUI	stageGameUI for this Stage Game
+ final static int originalTileSize	self explanation
+ final static int scale	self explanation
+ final static int tileSize	self explanation
+ final static int maxScreenCol	self explanation
+ final static int maxScreenRow	self explanation
+ final static int screenWidth	self explanation
+ final static int screenHeight	self explanation
+ int countdraw	number of render times
+ logic.KeyHandler keyH	key handler for this game
+ int stageNumber	stage number of this stage game
- Thread gameThread	thread for this logic game
+ GameLogic logic	logic for this stage game
+ CollisionChecker collisionCheck	use for check about collision
+ GraphicsContext gc	GraphicsContext for draw this stage game

6.4.2 Constructor

+ StageGame(int stagenumber)	Constructor method : -call super constructor -initialize stageNumber -set visible -call addListener() -initialize logic -add new obstacles using logic -initialize stageGameTile -initiallize stageGameUI -add stageGameTile,this StageGame,stageGameLogic to root orderly -initialize scene

6.4.3 Methods

+ void addListener()	setOnKeyPressed setOnKeyReleased and overide their handle
+ void startGameThread()	-reset game to default -create this game thread and setup priority then start
+ void run()	-override run method for game thread -do this until this stage game end : -clear canvas -update logic -paintComponent -update and draw hpBar

6.5 Class StageGameInstance

6.5.1 Fields

+ static final StageGameInstance instanceAllStage	static field which is Tutorial
- static List<StageGame> allStage	List that contains all StageGame

6.5.2 Constructor

+ StageGameInstance(int numStage)	create all stage game and add to List allStage

6.5.3 Methods

+ static StageGameInstance getInstance()	return this static field
+ StageGame getStage(int stageNumber)	return StageGame upto the number of stage

6.6 Class StageGameTile extends Canvas

6.6.1 Fields

+ GameLogic logic	logic of stage game that this StageGameTile Belongs to

6.6.2 Constructor

+ StageGameTile(StageGame stageGame)	Constructor method : -call super constructor -initialize logic -call paintTile()

6.6.3 Methods

+ void paintTile()	-initialize GraphicsContext gc as GraphicsContext of this canvas -draw all tiles form reHolder.getTiles()

6.7 Class StageGameUI extends AnchorPane implements Runnable

6.7.1 Fields

- StageGame sg	Stage game that this belongs to
- Thread gameThread	thread to run this StageGameUI
+ UIHolder uiHolder	UIHolder for this StageGameUI

6.7.2 Constructor

+ StageGameUI(StageGame sg)	Constructor method : -initialize sg -set style of this pane -initialize uiHolder -call setup()

6.7.3 Methods

+ void setUp()	-setUp uiHolder -add all UI to pane
+ void startGameThread()	-reset state game -create new thread for UI and setup then run
+ void run()	-override run method for UI thread -do this until this stage game end : -update every UI in uiHolder

6.8 Class SuccessWindow extends StackPane implements UI

6.8.1 Fields

- StageGame sg	Stage game that this belongs to

6.8.2 Constructor

+ SuccessWindow(StageGame sg)	Constructor method : -initialize sg -set style for this pane -create new Rectangle to be window and setup -create new text with “SUCCESS” and setup -create playagainButton with “PLAY AGAIN” -create detail VBOX that contains text and button then setup style -add window and that detail in to pane -set not visible and set disable

6.8.3 Methods

+ void update()	update this window to show when gamestate is SUCCESS

6.9 Interface UI

6.9.1 Methods

+ void update()	This method will call when the UI update more information in the other class that implements this interface

6.10 Class UIHolder

6.10.1 Fields

- List<UI> uiList	List that contains all UI of each UIHolder

6.10.2 Methods

+ void setUp(StageGame sg)	setUp UI for the StageGame by create and add PauseButton, OptionWindow, DieWindow, SuccessWindow to uiList
+ void addUI(UI ui)	add UI to uiList
+ void update()	update alls in uiList
+ List<UI> getUiList()	getter for UIList

7. Package sharedObject

7.1 Interface IRenderable

7.1.1 Methods

+ int getZ()	get Z value
+ void draw(GraphicsContext gc)	draw irenderable on gc
+ boolean isDestroyed()	self explanation
+ boolean isVisible()	self explanation
+ void update()	This method will call when the irenderable update more information in the other class that implements this interface

7.2 Class RenderableHolder

7.2.1 Fields

- List<IRenderable> entities	List of all entities in this game
- List<IRenderable> defaultEntities	default list of all entities in this game
- List<IRenderable> tiles	List of all tiles in this game
- Comparator<IRenderable> comparator	comparator for sort the list that contain irenderable
+ static Image[] character	Array of player image
+ static Tile[] tile	Array of tile image
+ static Image[] Object	Array of object image
+ static Map<String, AudioClip> audios	Map that the key is name of audio and the value is AudioClip
+ static Font brokenConsole	Font that names “brokenConsole”

7.2.2 Static Block

loadResource()	Static Block to load resource when open application

7.2.3 Constructor

+ RenderableHolder()	Constructor method : -initialize entities -initialize defaultEntities -initialize tiles -initialize comparator : compare between Irenderable using Z

7.2.4 Methods

+ static void loadResource()	load all resource that are Font, Image,Audio
- static void loadAudio()	load all Audio
- static void loadAudioFromURL(String path)	load Audio using URL as string
- static void loadObject()	load Image of all objects
+ static void loadTile()	load Image of all Tiles
+ static void loadTileFromURL(int index, String path, boolean collision)	loade one Image of Tiles using URL as string
+ static void loadTileFromBigImage(int number, int nCol, int nRow, int index, String path, boolean collision)	loade Image of Tiles by crop Big image
+ static void loadImageFromURL(Image[] array, int index, String path)	loade one Image using URL
+ static void loadImageFromBigImage(Image[] array, int number, int nCol, int nRow, int index, int size,String path)	loade Images by crop Big image which can set size of cropped image
+ static void loadImageFromBigImageResize(Image[] array, int number, int nCol, int nRow, int index,String path)	loade Images by crop Big image which resize to TileSize of this gane
+ void add(IRenderable entity)	add entity to List entities and defaultEntities and sort
+ void addTile(IRenderable tile)	add tile to List tiles and sort
+ void update()	update all of entities in List entities and check whether it has been destroyed or not, if yes remove it from List.
+ List<IRenderable> getEntities()	getter of List entities
+ List<IRenderable> getTiles()	getter of List tiles
+ void setDefault()	set entities to default state by assign the List entities with dafaultEntities

8. Package Tile

8.1 Class Tile

8.1.1 Fields

+ Image image	image for this tile
+ boolean collision	state that collide or not

8.2 Class TileMap implements IRenderable

8.2.1 Fields

+ int mapTileNum[][]	2D array of map tile
- int stageNumber	Number of stage
- int layer	layer of tiles

8.2.2 Constructor

+ TileMap(StageGame sg, int stageNumber, int layer)	Initialize all fields and call loadMap()

8.2.3 Methods

+ void loadMap()	load map to mapTileNum[][]
+ int getZ()	get Z value
+ void draw(GraphicsContext gc)	draw each tile to display map
+ boolean isDestroyed()	self explanation
+ boolean isVisible()	self explanation
+ void update()	do nothing

9. Package application

9.1 Class Main extends Application

9.1.1 Methods

+ void start(Stage primaryStage)	switch to Main menu pane
+ static void main(String[] args)	launch(args)