

Tower defense

Created by

Punnawat Lohanuit 6430235721

Puun Pinyawat 6430239221

2110215 Programming Methodology

Samester 1 Year 2022

Chulalongkorn University

Tower Defense

Introduction

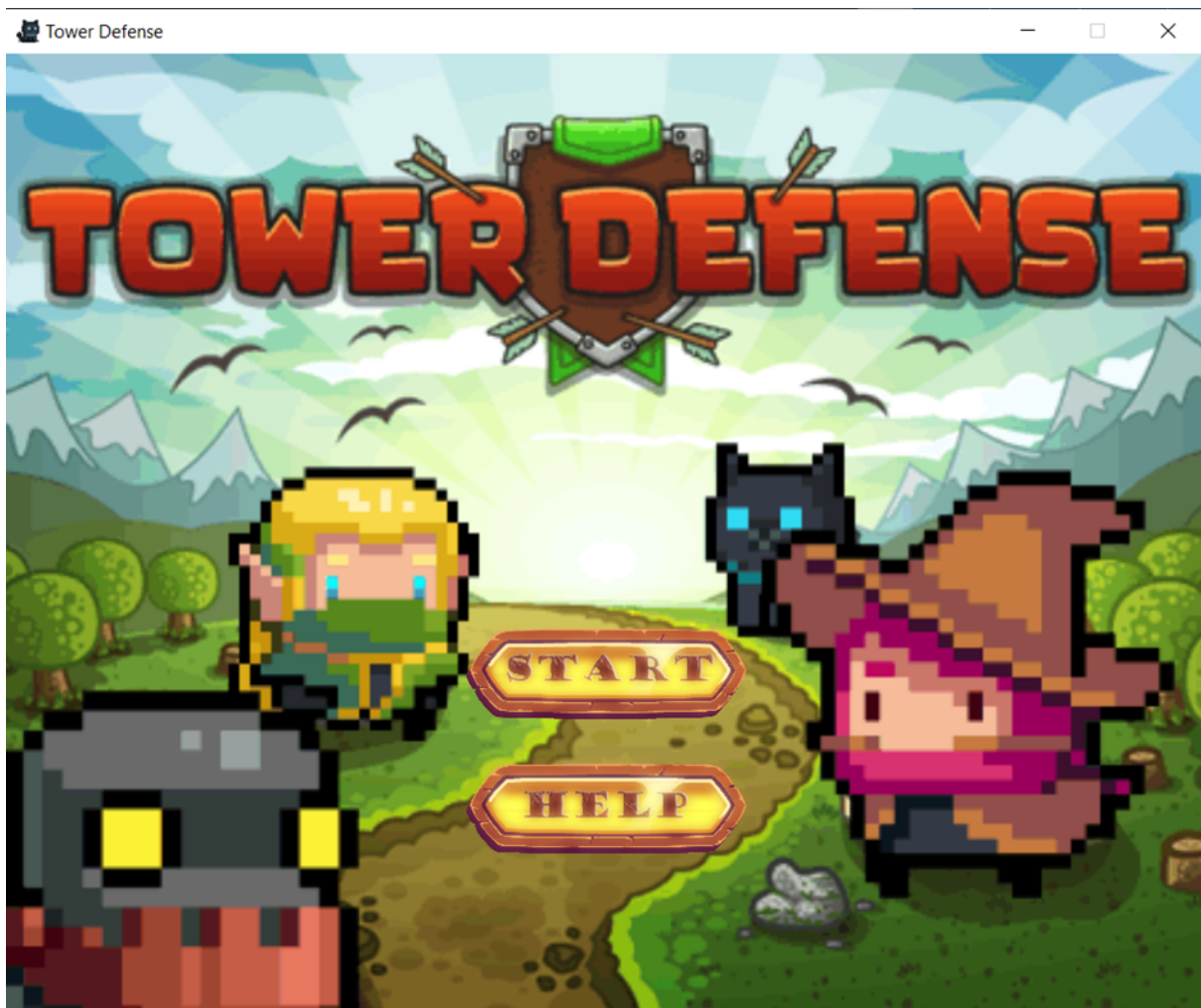
Tower Defense is inspired by Bloons TD game. Your goal is to be defense waves of monsters by your tower.

Rules

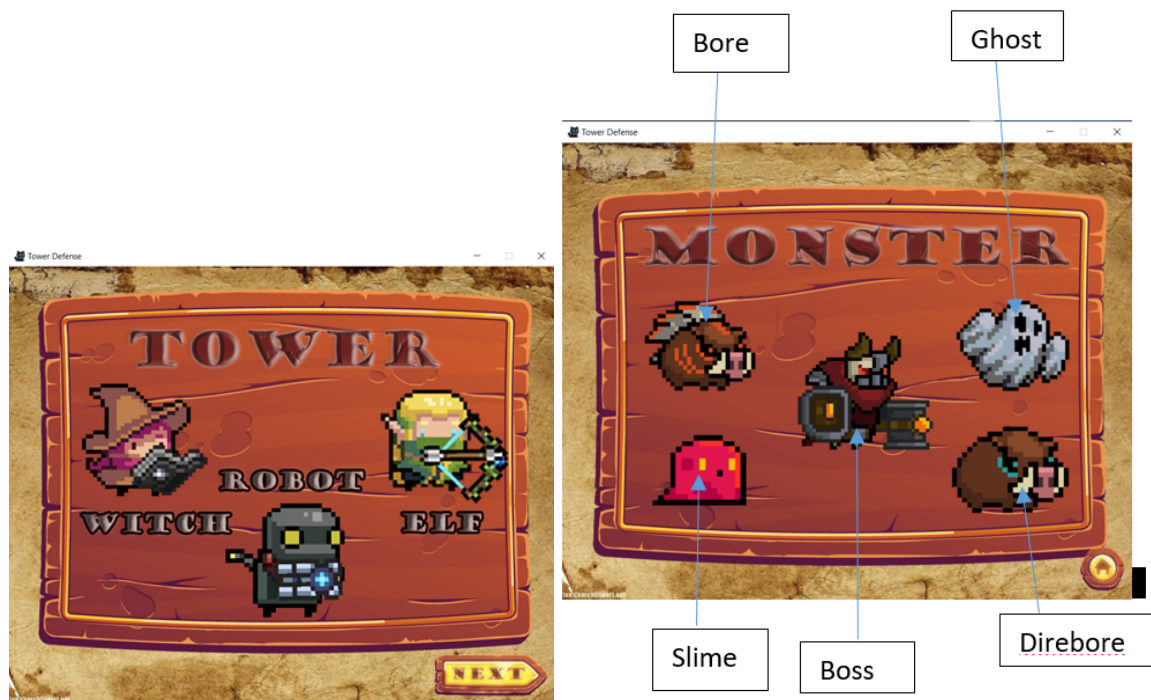
You have 20 health, if a monster can pass all of your tower, your health -1.

If you have health 0 you lose, but if you can survive until wave 40 you win.

Menu



Help Scene



Tower

	Witch	Elf	Robot
Cost	100	50	150
Damage	40 + slow a monster	50	140
Range	Long	Short	Medium
Cooldown	Slow	Fast	Very slow

Upgrade tower

	Witch	Elf	Robot
Cost	100	80	150
Damage	+5	+20	+50
Range	Increase much	Increase much	Increase medium
Cooldown	Decrease much	Decrease less	Decrease very much
Max tier	4	6	4

Monster

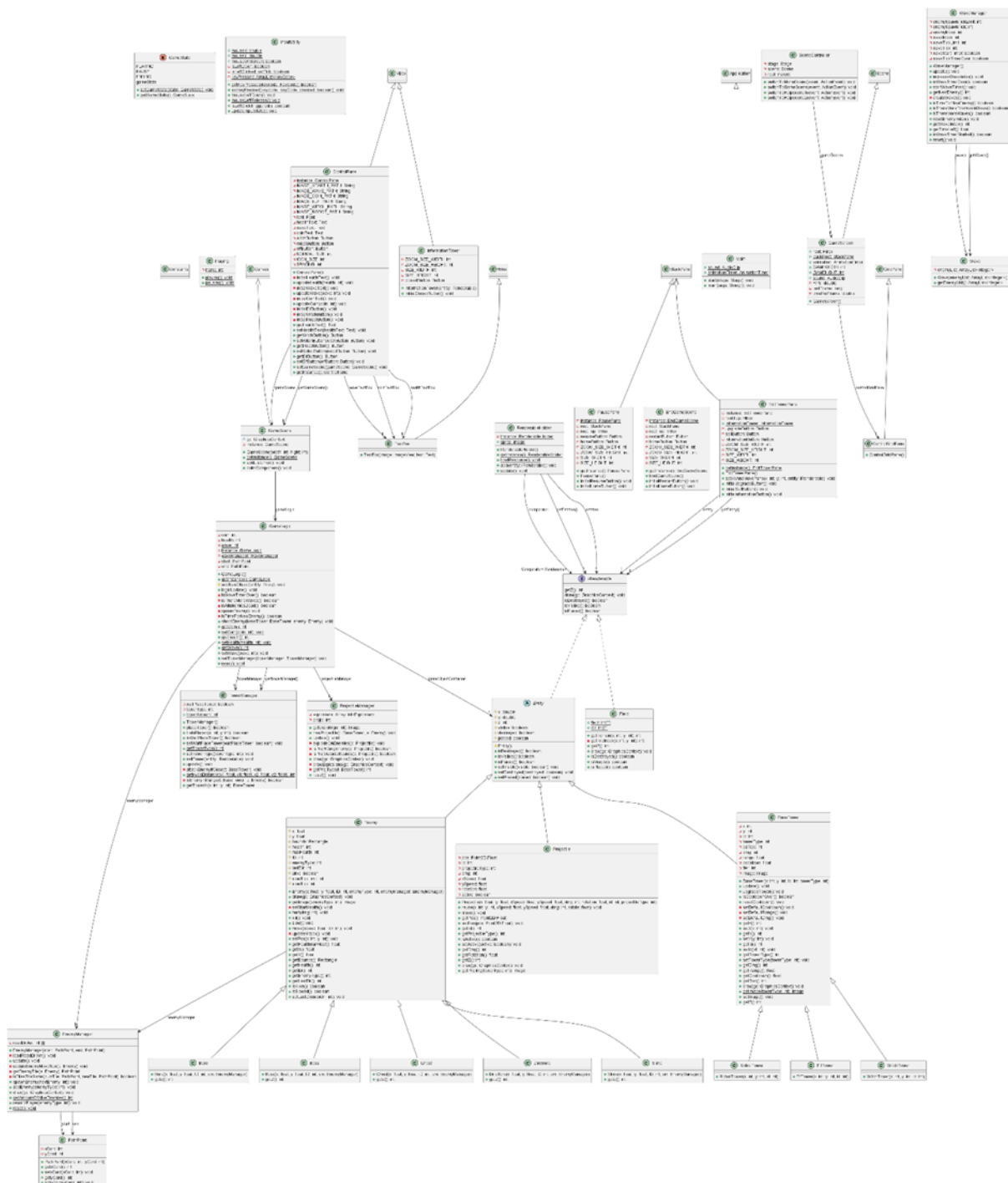
	Slime	Ghost	Bore	Direbore	Boss
Health	80	300	500	250	2000
Speed	Fast	Slow	Slow	Very fast	Very slow
Reward	5 coins	20 coins	25 coins	40 coins	300 coins

Game Scene



Click your tower at right and place it at field to kill monsters!

Class Diagram



1. package enemies

1.1 abstract class Enemy extends Entity

1.1.1 fields

-EnemyManager enemymanager	To manage enemies
-float x, y	Position on field
-int health	Current health
-int maxHealth	Max health
-int ID	ID
-int enemyType	Type of enemy
-int lastDir	Last direction that enemy moved
-boolean alive	This enemy alive
-int slowTickLimit	slowTickLimit
-int slowTick	slowTick
-Image image	Image of enemy

1.1.2 constructor

+Enemy(float x, float y, int ID, int enemyType, EnemyManager enemyManager)	Initialize the enemy withal parameters Set lastDir as -1
--	---

1.1.3 methods

+void draw(GraphicsContext gc)	Draw an enemy on gc
--------------------------------	---------------------

+void hurt(int dmg)	Decrease health by dmg
+void kill()	Set alive to false and destroy
+void slow()	To slow this enemy
+void move(float speed, int dir)	Move enemy
+void setPos(int x, int y)	Set position on field
+all getter/setter	

1.2 class Bore extends Enemy

1.2.1 constructor

+Bore(float x, float y, int ID, EnemyManager em)	Initialize enemyType as Constants.Enemies.BORE
--	--

1.3 class Boss extends Enemy

1.3.1 constructor

+Boss(float x, float y, int ID, EnemyManager em)	Initialize enemyType as Constants.Enemies.BOSS
--	--

1.4 class DireBore extends Enemy

1.4.1 constructor

+DireBore(float x, float y, int ID, EnemyManager em)	Initialize enemyType as Constants.Enemies.DIREBORE
--	--

1.5 class Ghost extends Enemy

1.5.1 constructor

+Ghost(float x, float y, int ID, EnemyManager em)	Initialize enemyType as Constants.Enemies.GHOST
---	---

1.6 class Slime extends Enemy

1.6.1 constructor

+Slime(float x, float y, int ID, EnemyManager em)	Initialize enemyType as Constants.Enemies.SLIME
---	---

1.7 class Constants

1.7.1 inner public static class Direction

1.7.1.1 fields

+static final int LEFT	Set to 0
+static final int UP	Set to 1
+static final int RIGHT	Set to 2
+static final int DOWN	Set to 3

1.7.2 inner public static class Enemies

1.7.2.1 fields

+static final int SLIME	Set to 0
+static final int GHOST	Set to 1
+static final int BORE	Set to 2
+static final int DIREBORE	Set to 3
+static final int BOSS	Set to 4

1.7.2.2 methods

+static int GetReward(int enemyType)	Return reward when killed
+static int GetSpeed(int enemyType)	Return the enemy's speed
+static int GetStartHealth(int enemyType)	Return the enemy's start health

1.7.3 inner public static class Tiles

1.7.3.1 fields

+static final int GRASS_TILE	Set to 0
+static final int ROAD_TILE	Set to 1

1.8 class PathPoint

1.8.1 fields

-int xCord, yCord	Position in field
-------------------	-------------------

1.8.2 constructor

+PathPoint(int xCord, int yCord)	Initialize the position
----------------------------------	-------------------------

1.8.3 methods

+All getter/setter	
--------------------	--

1.9 class EnemyManager

1.9.1 fields

-PathPoint start, end	Start and end point
-int[][] roadDirArr	Array of road that enemy can move

1.9.2 constructor

+EnemyManager(PathPoint start, PathPoint end)	Initialize start and end point Create road by loadRoadDirArr()
---	---

1.9.3 methods

-void loadRoadDirArr()	Load the array of map
------------------------	-----------------------

+void update()	Update all enemies
-void updateEnemyMoveNew(Enemy e)	Update enemy move follow the road
-PathPoint getEnemyTile(Enemy e)	Get the PathPoint that enemy in
-boolean isTilesTheSame(PathPoint currTile, PathPoint newTile)	Check if 2 PathPoint the same
+void spawnEnemy(int nextEnemy)	To use addEnemy()
+void addEnemy(int enemyType)	Add enemy to start point
+void draw(GraphicsContext gc)	Draw enemy in gc
+static int getAmountOfAliveEnemies()	Count the alive enemies
+void rewardPlayer(int enemyType)	Give Reward to player follow the enemy type
+static void reset()	Remove all enemies

1.10 class Wave

1.10.1 fields

-ArrayList<Integer> enemyList	List of enemies
-------------------------------	-----------------

1.10.2 constructor

+Wave(ArrayList<Integer> enemyList)	Initialize enemyList
-------------------------------------	----------------------

1.10.3 methods

+ ArrayList<Integer> getEnemyList()	Get enemyList
-------------------------------------	---------------

1.11 class WaveManager

1.11.1 fields

-ArrayList<Wave> waves	ArrayList of waves
-int enemySpawnTickLimit	spawn tick limit
-int enemySpawnTick	spawn tick
-int enemyIndex, waveIndex	enemy index, wave index
-int waveTickLimit	wave tick limit
-int waveTick	wave tick
-boolean waveStartTimer, waveTickTimerOver	to start counting and reset ones it set reaches

1.11.2 constructor

+WaveManager()	Create wave by createWaves()
----------------	------------------------------

1.11.3 methods

+void update()	update and check if enemy spawn tick more than or equal to enemy spawn
----------------	--

	tick limit, update and check if wave tick more than or equal to wave tick limit
+void increaseWaveIndex()	wave index +1
+void startWaveTimer()	start the wave timer
+int getNextEnemy()	return next enemy in current wave
-void createWaves()	create all waves
+void reset()	reset all waves
+all getter/setter	

2. Package tower

2.1 Class BaseTower

2.1.1 fields

-int x, y	Point on field
-int id	id
-int towerType	Type of tower
-int cdTick	To count time
-int dmg	Tower's damage
-float range	Tower's attack range
-float cooldown	Tower's attack cooldown

-int tier	Tower's level
-Image image	Image of tower

2.1.2 constructor

+BaseTower(int x, int y, int id, int towerType)	Initialize x, y, id, towerType Set tier = 1 Set image by setImage() Set dmg by setDefaultDmg() Set range by setDefaultRange() Set cooldown by setDefaultCooldown()
---	---

2.1.3 methods

+void upgradeTower()	To upgrade the tower switch by type
+void update()	To increase cdTick by 1
+Boolean isCooldownOver()	To check if cdTick >= cooldown
+void resetCooldown()	To reset cdTick as 0
-void setDefaultCooldown()	To set cooldown follow to this towerType
-void setDefaultRange()	To set range follow to this towerType
-void setDefaultDmg()	To set damage follow to this towerType

+void draw(GraphicsContext gc)	To draw tower on gc
+all getter/setter	

2.2 class TowerManager

2.2.1 fields

-boolean waitPlaceTower	is placing now
-static int towerType	Type of tower
+static int towerAmount	Amount of tower

2.2.2 constructor

+TowerManager()	Set waitPlaceTower as false
-----------------	-----------------------------

2.2.3 methods

+boolean placeTower()	Place tower by their x, y and decrease coin by their cost and return true If can't return false
+Boolean tileIsPlaced(int x, int y)	Check if tile already have tower

+void selltower(IRenderable entity)	sell the tower, coin +30
+void update()	Update and attack all towers
-void attackEnemyIfClose(BaseTower t)	Attack alive enemy that is in range
+static int getHypoDistance(float x1, float y1, float x2, float y2)	return distance
-boolean isEnemyInRange(BaseTower t, Enemy e)	Check if enemy in tower's range
+BaseTower getTowerAt(int x, int y)	Return BaseTower at x, y

2.3 class constants

2.3.1 inner public static class Projectile

2.3.1.1 fields

+static final int ARROW	Set as 0
+static final int SNOWBALL	Set as 1
+static final int LASER	Set as 2

2.3.1.2 methods

+static float GetSpeed(int Type)	Return speed
----------------------------------	--------------

2.3.2 inner public static class Towers

2.3.2.1 fields

+static final int ELF	Set as 0
+static final int WITCH	Set as 1
+static final int ROBOT	Set as 2

2.3.2.2 methods

+static int GetTowerCost(int towerType)	Return tower's cost
+static String GetName(int towerType)	Return tower's name
+static int GetStartDmg(int towerType)	Return tower's damage
+static float GetDefaultRange(int towerType)	Return tower's attack range
+static float GetDefaultCooldown(int towerType)	Return tower's attack cooldown

2.3.3 inner public static class Tiles

+static final int GRASS_TILE	Set as 0
+static final int ROAD_TILE	Set as 1

2.4 class ElfTower extends BaseTower

2.4.1 constructor

+ElfTower(int x, int y, int id)	Constructor by superclass, towerType = 0
---------------------------------	--

2.5 class WitchTower extends BaseTower

2.5.1 constructor

+WitchTower(int x, int y, int id)	Constructor by superclass, towerType = 1
-----------------------------------	--

2.6 class RobotTower extends BaseTower

2.6.1 constructor

+RobotTower(int x, int y, int id)	Constructor by superclass, towerType = 2
-----------------------------------	--

3. Package scenes

3.1 class Playing

3.1.1 fields

-static int frame	Frame of scene
-------------------	----------------

3.1.2 constructor

+static void Playing()	Increase 1 coin per 60 frame Update GameLogic, RenderableHolder, InputUtility Paint all components
------------------------	--

3.1.3 methods

+static void pausing()	Do nothing
------------------------	------------

4. Package input

4.1 class InputUtility

4.1.1 fields

+static double mouseX, mouseY	Position of mouse
+static boolean mouseOnScreen	Set as true
-static boolean isLeftDown	Set as false
-static boolean isLeftClickedLastTick	Set as false
-static ArrayList<KeyCode> keyPressed	All key

4.1.2 methods

+static boolean getKeyPressed(KeyCode keycode)	Check if key pressed
---	----------------------

+static void setKeyPressed(KeyCode keycode, Boolean pressed)	Set key pressed
+static void MouseLeftDown()	Set isLeftDown as true Set isLeftClickedLastTick as true
+static void MouseLeftRelease()	Set isLeftDown as false
+static boolean isLeftClickTriggered()	Return isLeftClickedLastTick
+static void updateInputState()	Set isLeftClickedLastTick as false

5. Package sharedObject

5.1 class RenderableHolder

5.1.1 fields

-static RenderableHolder instance	RenderableHolder
-ConcurrentSkipListSet<IRenderable> entities	all entities
-Comparator<IRenderable> comparator	comparator
+static Image sprite	resource imge

5.1.2 constructor

+RenderableHolder()	Initialize RenderableHolder
---------------------	-----------------------------

5.1.3 methods

+static RenderableHolder getInstance()	Make sure RenderableHolder != null
+static void loadResource()	Load resource by ClassLoader
+void add(IRenderable entity)	Add entity
+void update()	check is destroyed and update all entity
+ ConcurrentSkipListSet<IRenderable> getEntities()	return entities

5.2 interface IRenderable

5.2.1 methods

+int getZ()
+void draw(GraphicContext gc)
+boolean isDestroyed()
+boolean isVisible()
+boolean isPlaced()

6. Package drawing

6.1 class ControlGridPane extends GridPane

6.1.1 constructor

+ ControlGridPane()	Create ControlGridPane and fill color background
---------------------	--

6.2 class ControlPane extends VBox

6.2.1 fields

-final String IMAGE_HEARTH_PATH	Image of hearth
-final String IMAGE_WAVE_PATH	Image of wave
-final String IMAGE_COIN_PATH	Image of coin
-final String IMAGE_ELF_PATH	Image of elf
-final String IMAGE_WITCH_PATH	Image of witch
-final String IMAGE_ROBOT_PATH	Image of robot
-final String Font font	To set font
-Text healthText	Player's current health text
-Text waveText	Player's current wave text
-Text coinText	Player's current coin text
-TextBox healthTextBox	Player's current health text box
-TextBox waveTextBox	Player's current wave text box
-TextBox coinTextBox	Player's current coin text box

-Button witchButton	Witch Button
-Button robotButton	Robot Button
-Button elfButton	Elf Button
-GameScene gamescene	Game scene at left
-static ControlPane instance	controlPane
-final int NORMAL_SIZE	Set as 120
-final int ICON_SIZE	Set as 40
-final int SPACING	Set as 10

6.2.2 constructor

+ControlPane()	Initialize the control pane
----------------	-----------------------------

6.2.3 methods

-void initialHealthText()	Initialize - Health text - Health Image
+void updateHealth(int Health)	Update Health text follow to Health

-void initialWaveText()	Initialize - Wave text - Wave Image
+void updateWave(int Wave)	Update Wave text follow to Wave
-void initialCoinText()	Initialize - Coin text - Coin Image
+void updateCoin(int Coin)	Update Coin text follow to Coin
-void initialElfButton()	Initialize elf button
-void initialWitchButton()	Initialize witch button
-void initialRobotButton()	Initialize robot button
+all getter/setter	

6.3 class GameScene extends Canvas

6.3.1 fields

GraphicContext gc	GraphicContext
GameLogic gameLogic	gameLogic
-static GameScene instance	Game scene

6.3.2 constructor

GameScene()	Initialize game scene at left
-------------	-------------------------------

6.3.3 methods

+void addListener()	
+void paintComponent()	Set color background and draw all visacle and not destroyed entity

6.4 class GameScreen extend Scene

6.4.1 fields

+HBox root	Root
+static StackPane backRoot	Back root
-ControlGridPane controlGridPane	controlGridPane
+static AnimationTimer animation	AnimationTimer
+static final int GAMEWIDTH	Set as 800
+static final int GAMEHIGHT	Set as 640
+static AudioClip sound	Game sound
-final double FPS	Set as 60.0
-long lastFrame	Last frame
-double timePerFrame	Time per frame

6.4.2 constructor

GameScreen()	Initialize game screen using animationTimer
--------------	---

6.5 class TextBox extends HBox

6.5.1 constructor

TextBox(ImageView image, Text text)	Initialize text box
-------------------------------------	---------------------

7. Package logic

7.1 abstract class Entity implements IRenderable

7.1.1 fields

Protected double x, y	Location
Protected int z	Layer
Protected boolean visible	Visible
Protected boolean destroyed	Destroyed
Protected boolean placed	Placed

7.1.2 constructor

Entity()	Initialize entity
----------	-------------------

7.1.3 methods

+all getter/setter	
--------------------	--

7.2 class Field implements IRenderable

7.2.1 fields

+static int[][] field	Field created by 2D array
+static int[][] dir	Direction can move created by 2D array

7.2.2 methods

getTerrain(int x, int y)	Check if x, y out of range return -3
-int[] getTileIndex(int x, int y)	Get tile index for crop image
+int getZ()	Return -9999 to backest layer
+void draw(GraphicContext gc)	Draw field
+boolean isDestroyed()	Return false
+boolean isVisible()	Return true
+boolean isPlaced()	Return false

7.3 class GameLogic

7.3.1 fields

-List<Entity> gameObjectContainer	All of entities
-static int coin	Set as 120
-static int health	Set as 20
-static int wave	Set as 1
-static GameLogic instance	GameLogic
-TowerManager towerManager	towerManager
-EnemyManager enemyManager	enemyManager
-static WaveManager waveManager	waveManager
-ProjectileManager projectileManager	projectileManager
-static PathPoint start	Start point
-static PathPoint end	End point

7.3.2 constructor

GameLogic()	Initialize gamelogic
-------------	----------------------

7.3.3 methods

+static GameLogic getInstance()	Make sure instance != null
---------------------------------	----------------------------

Protected void addNewObject(Entity entity)	Add entity
+void logicUpdate()	Check all and update
-boolean isWaveTimerOver()	Check if wave time over
-boolean isThereMoreWaves()	Check if there is next wave
-boolean isAllEnemiesDead()	Check if all enemies dead
-void spawnEnemy()	Spawn enemy
-boolean isTimeForNewEnemy()	Check let enemy go in wave
+void shootEnemy(BaseTower baseTower, Enemy enemy)	Shoot enemy
+static void reset()	Set game
+all getter/setter	

7.4 class Projectile extends Entity

7.4.1 fields

-Point2D.Float pos	Position
-int id	Id
-int projectileType	Type of projectile
-int dmg	Damage

-float xSpeed, ySpeed	Speed in x axis, y axis
-float rotation	Rotation of projectile
-boolean active	Active

7.4.2 constructor

+Projectile(float x, float y float xSpeed, float ySpeed, int dmg, float rotation, int id, int projectileType)	Initialize projectile
---	-----------------------

7.4.3 methods

+void reuse(int x, int y, float xSpeed, float ySpeed, int dmg, float rotate)	Reuse the projectile
+void move()	Move the projectile
+int getZ()	Return 0
+void draw(GraphicsContext gc)	Draw the projectile on gc
+Image getProjImg(int towerType)	Return image of projectile
+all getter/setter	

7.5 class TextBox

7.5.1 constructor

TextBox(ImageView image, Text text)	Initialize text box
-------------------------------------	---------------------

8. Package main

8.1 class EditTowerPane

8.1.1 fields

-static EditTowerPane instance	This implementation confirms that we have only one EditTowerPane
- HBox rootTop	A field that represents the rootTop
+ static InformationTower informationTower	A field that represents the informationTower
- Button upgradeButton	A field that represents the upgradeButton
- Button sellButton	A field that represents the sellButton
- Button informationButton	A field that represents the informationButton
- final int ZOOM_SIZE_WIDTH	A field that represents the ZOOM_SIZE_WIDTH
- final int ZOOM_SIZE_HEIGHT	A field that represents the ZOOM_SIZE_HEIGHT
- final int SIZE_WIDTH	A field that represents the SIZE_WIDTH

- final int SIZE_HEIGHT	A field that represents the SIZE_HEIGHT
- IRenderable entity	A field that represents the entity

8.1.2 constructor

+ EditTowerPane()	Initialize EditTowerPane
-------------------	--------------------------

8.1.3 methods

+ static EditTowerPane getInstance()	Make sure instance != null
+ void setXYAndMovePane(int x, int y, IRenderable entity)	Set new position x, y
+ void initialUpgradeButton()	Initialize upgrade button
+ IRenderable getEntity()	return entity
+ void initialSellButton()	Initialize sell button
+ void initialInformationButton()	Initialize information button

8.2 class EndGameScene

8.2.1 fields

- static EndGameScene instance	This implementation confirms that we have only one EndGameScene
--------------------------------	---

- StackPane root	A field that represents the root
- VBox rootTop	A field that represents the rootTop
- Button restartButton	A field that represents the restartButton
- Button homeButton	A field that represents the homeButton
- final int ZOOM_SIZE_WIDTH	A field that represents the ZOOM_SIZE_WIDTH
- final int ZOOM_SIZE_HEIGHT	A field that represents the ZOOM_SIZE_HEIGHT
- final int SIZE_WIDTH	A field that represents the SIZE_WIDTH
- final int SIZE_HEIGHT	A field that represents the SIZE_HEIGHT

8.2.2 constructor

+ EndGameScene()	Initialize EndGameScene
------------------	-------------------------

8.2.3 methods

+ static EndGameScene getInstance()	Make sure instance != null
+ void initialRestartButton()	Initialize restart button
+ void initialHomeButton()	Initialize home button

8.3 class enum GameState

PLAYING, PAUSE, ENDING;

8.4 class InformationTower

8.4.1 fields

- final int ZOOM_SIZE_WIDTH	A field that represents the ZOOM_SIZE_WIDTH
- final int ZOOM_SIZE_HEIGHT	A field that represents the ZOOM_SIZE_HEIGHT
- final int SIZE_WIDTH	A field that represents the SIZE_WIDTH
- final int SIZE_HEIGHT	A field that represents the SIZE_HEIGHT
- Button closedButton	A field that represents the closedButton

8.4.2 constructor

+ InformationTower(IRenderable entity)	Initialize InformationTower
--	-----------------------------

8.4.3 methods

+ void initialClosedButton()	Initialize close button
------------------------------	-------------------------

8.5 class Main

8.5.1 fields

- static AudioClip sound	Game soundtrack
- static AnimationTimer animationTimer	AnimationTimer

8.5.2 methods

+ void start(Stage stage)	Initialize icon, scene, AnimationTimer and soundtrack, start the program
+ static void main(String[] args)	Launch args

8.6 class PausePane

8.6.1 fields

- static PausePane instance	This implementation confirms that we have only one PausePane
- StackPane root	A field that represents the root
- VBox rootTop	A field that represents the rootTop
- Button resumeButton	A field that represents the resumeButton
- Button homeButton	A field that represents the homeButton
- final int ZOOM_SIZE_WIDTH	A field that represents the ZOOM_SIZE_WIDTH

- final int ZOOM_SIZE_HEIGHT	A field that represents the ZOOM_SIZE_HEIGHT
- final int SIZE_WIDTH	A field that represents the SIZE_WIDTH
- final int SIZE_HEIGHT	A field that represents the SIZE_HEIGHT

8.6.2 constructor

+ PausePane()	Initialize PausePane
---------------	----------------------

8.6.3 methods

+ void initialResumeButton()	Initialize resume button
+ void initialHomeButton()	Initialize home button

8.7 class Scene

8.7.1 fields

- Stage stage	A field that represents the stage
- Scene scene	A field that represents the scene
- Parent root	A field that represents the root
- GameScreen gameScreen	A field that represents the gameScreen

8.7.2 methods

+ void switchToMenuScene(ActionEvent event)	Switch scene to menu scene
+ void switchToGameScene(ActionEvent event)	Switch scene to game scene
+ void switchToHelpScene1(ActionEvent event)	Switch scene to help 1 scene
+ void switchToHelpScene2(ActionEvent event)	Switch scene to help 2 scene