

# Tower defense

Created by

Punnawat Lohanuit 6430235721

Puun Pinyawat 6430239221

2110215 Programming Methodology

Samester 1 Year 2022

Chulalongkorn University

# Tower Defense

## Introduction

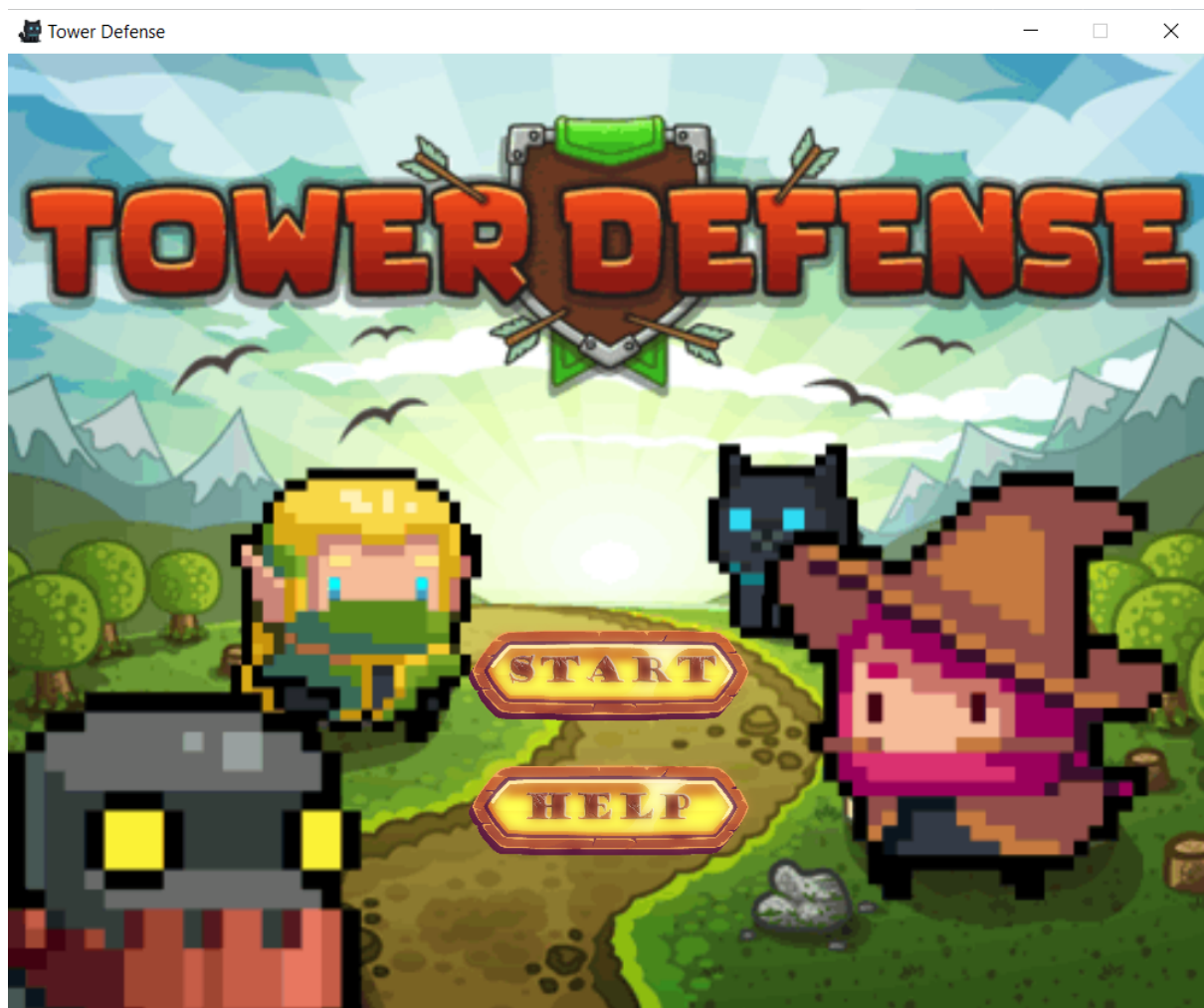
Tower Defense is inspired by Bloons TD game. Your goal is to be defense waves of monsters by your tower.

## Rules

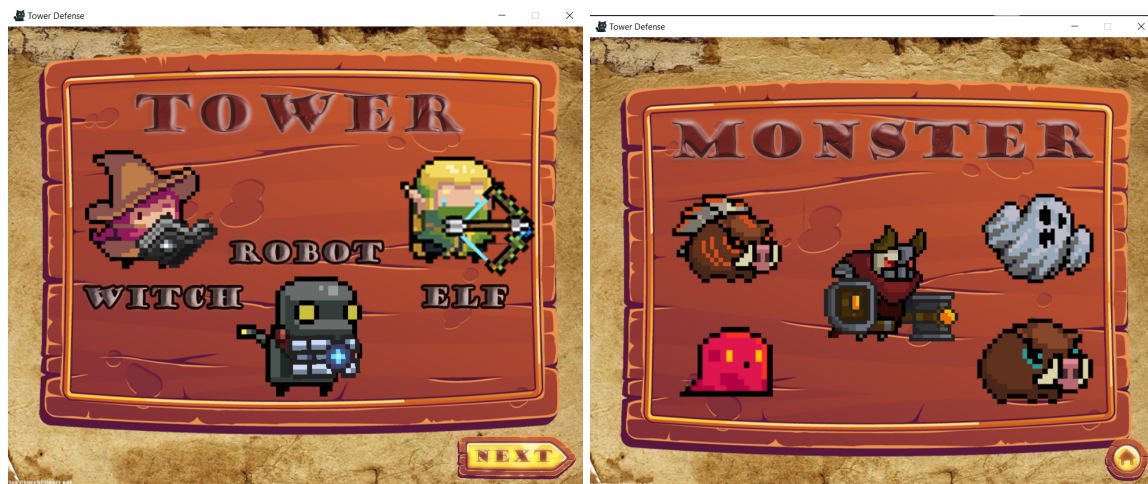
You have 20 health, if a monster can pass all of your tower, your health -1.

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

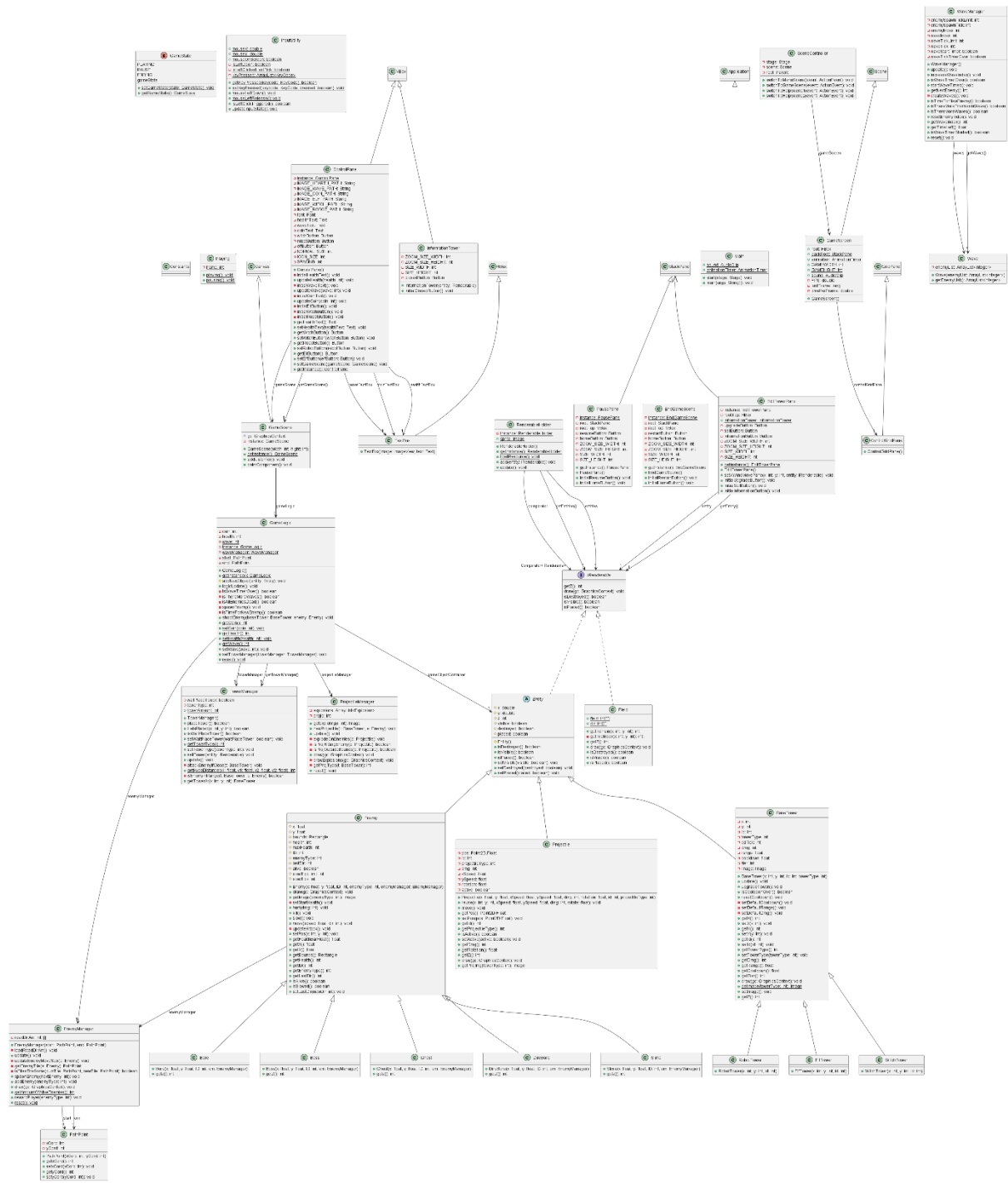
	Slime	Ghost	Bore	Direbore	Boss
Health	80	300	500	250	2000
Speed	Fast	Slow	Slow	Very fast	Very slow
Reward	5 coins	20 coins	25 coins	40 coins	300 coins

## 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	
-int enemyType	Type of enemy
-int lastDir	Last direction that enemy moved
-boolean alive	This enemy alive
-int slowTickLimit	
-int 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	
-int enemySpawnTick	
-int enemyIndex, waveIndex	
-int waveTickLimit	
-int waveTick	
-boolean waveStartTimer, waveTickTimerOver	

### 1.11.2 constructor

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

### 1.11.3 methods

+void update()	
+void increaseWaveIndex()	
+void startWaveTimer()	
+int getNextEnemy()	
-void createWaves()	
+void reset()	
+all getter/setter	

## 2. Package tower

### 2.1 Class BaseTower

#### 2.1.1 fields

-int x, y	Point on field
-int 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	
-static int towerType	Type of tower
+static int towerAmount	

### 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)	
+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)	
-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	
+static boolean mouseOnScreen	Set as true
-static boolean isLeftDown	Set as false
-static boolean isLeftClickedLastTick	Set as false

-static ArrayList<KeyCode> keyPressed	
---------------------------------------	--

#### 4.1.2 methods

+static boolean getKeyPressed(KeyCode keycode)	
+static void setKeyPressed(KeyCode keycode, Boolean 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	
-ConcurrentSkipListSet<IRenderable> entities	
-Comparator<IRenderable> comparator	
+static Image sprite	

##### 5.1.2 constructor

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

##### 5.1.3 methods

+static RenderableHolder getInstance()	
+static void loadResource()	
+void add(IRenderable entity)	
+void update()	
+ ConcurrentSkipListSet<IRenderable> getEntities()	

#### 5.2 interface IRenderable

##### 5.2.1 methods

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

## 6. Package drawing

### 6.1 class ControlGridPane

6.1.1 fields

6.1.2 constructor

6.1.3 methods

### 6.2 class ControlPane

6.2.1 fields

6.2.2 constructor

6.2.3 methods

### 6.3 class GameScene

6.3.1 fields

6.3.2 constructor

6.3.3 methods

### 6.4 class GameScreen

6.4.1 fields

6.4.2 constructor

6.4.3 methods

### 6.5 class TextBox

6.5.1 fields

6.5.2 constructor

6.5.3 methods

## 7. Package logic

### 7.1 abstract class Entity

7.1.1 fields

7.1.2 constructor

7.1.3 methods

### 7.2 class Field

7.2.1 fields

7.2.2 constructor

7.2.3 methods

### 7.3 class GameLogic

7.3.1 fields

7.3.2 constructor

7.3.3 methods

### 7.4 class Projectile

7.4.1 fields

7.4.2 constructor

7.4.3 methods

### 7.5 class TextBox

7.5.1 fields

7.5.2 constructor

7.5.3 methods

- 8. Package main
  - 8.1 class EditTowerPane
    - 8.1.1 fields
    - 8.1.2 constructor
    - 8.1.3 methods
  - 8.2 class EndGameScene
    - 8.2.1 fields
    - 8.2.2 constructor
    - 8.2.3 methods
  - 8.3 class GameState
    - 8.3.1 fields
    - 8.3.2 constructor
    - 8.3.3 methods
  - 8.4 class InformationTower
    - 8.4.1 fields
    - 8.4.2 constructor
    - 8.4.3 methods
  - 8.5 class Main
    - 8.5.1 fields
    - 8.5.2 constructor
    - 8.5.3 methods
  - 8.6 class PausePane
    - 8.6.1 fields
    - 8.6.2 constructor
    - 8.6.3 methods
  - 8.7 class Scene
    - 8.7.1 fields
    - 8.7.2 constructor
    - 8.7.3 methods
  - 8.8 Aplication.css
  - 8.9 Help1.fxml
  - 8.10 Help2.fxml
  - 8.11 Menu.fxml