

Higon Tower Defense Documentation

Created by

Rodchananat Khunakornophat 6031323521

Thuantham Chitbhakdee 6031025521

2110215 Programming Methodology

Semester 1 Year 2018

Chulalongkorn University

Higon Tower Defense Documentation

Introduction

Higon Defense is a Tower defense game. The objective of this game is defending your base by preventing enemy from reaching your base by placing towers which attack enemy. Each tower has different effect and can target different types of enemy (ground and air); some can target only ground enemy (tanks, trucks, soldiers), some can target only air enemy (planes), some can target both. There is also some type of supportive tower of which main ability isn't dealing damage but provide some useful effects. Strategy is also important in this game, placing tower effectively to maximize enemy walking path to reach your base will make them take more damage by your tower. There's various type of enemy. Some enemy will have special effect such as spawning smaller enemy when it dies, summoning more enemy while it's live. Enemy will spawn by wave, once you defend through 20 waves of enemy you will win the game. In later wave, there will be more enemy, and they will be tougher. That is when upgrade become important upgrading will make tower stronger, by increasing damage, range, or attack speed – making them more effective against stronger enemy.

User Manual

Loading Scene



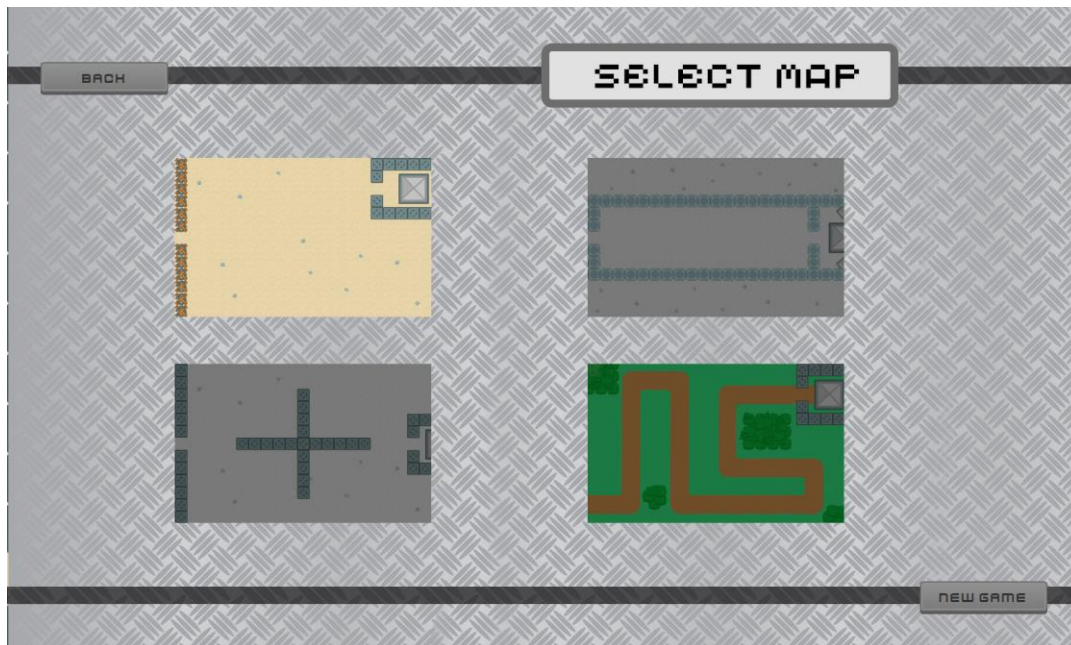
This scene represent loading progress of game resources (Image and sound), Once loading complete, you will be taken to Main Menu Scene.

Main Menu Scene



- Press New Game to start a new game. **Progress of last game will be lost.**
- Press Resume to resume your last game (when you go back from Game to Main Menu)

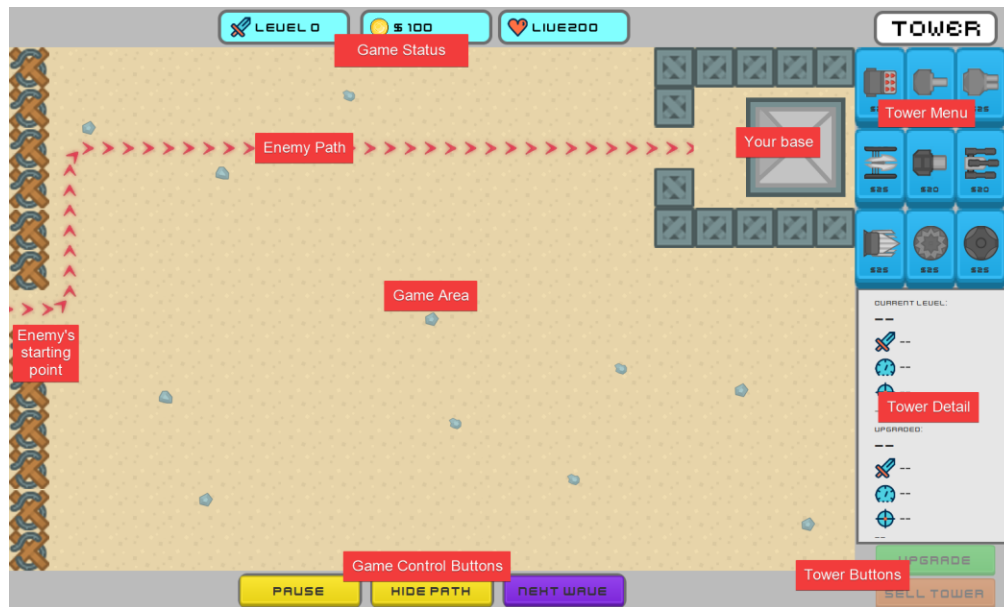
Map Selection Scene



In this scene, you will choose the map you want to play and press New Game. The map currently selected will be highlighted (in this picture: the top left map), each map has different layout, so the strategy in each map is different.

Game Scene

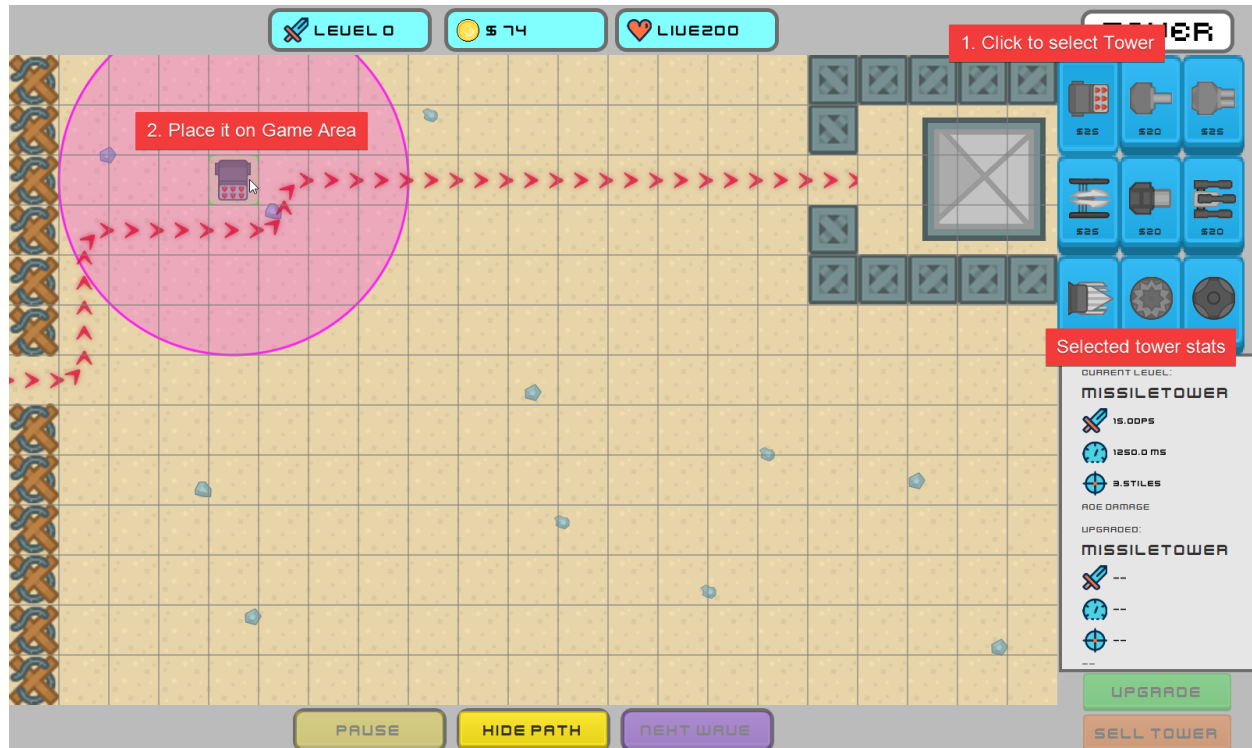
User Interface



Once you're in game you will see various elements of game.

- Game Status: This will display Level (numbers of wave), money (used to buy tower), lives (decrease when enemy reach your base)
- Game Area: This is Main game area where you will place towers.
- Enemy Path: This will display how enemy moves through Game Area
- Starting Point :Enemy will spawn here
- Your Base: Don't let enemy get here, you will lose you base health
- Tower Detail: Show details about selected tower and it's upgrade
- Tower Buttons: Buttons for upgrading and selling tower
- Game Control Buttons:
 - Pause: show pause menu
 - Hide path: Hide path if it's too obstructive to your view (it will still show when placing tower)
 - Next wave: press when you're ready for next wave

Gameplay

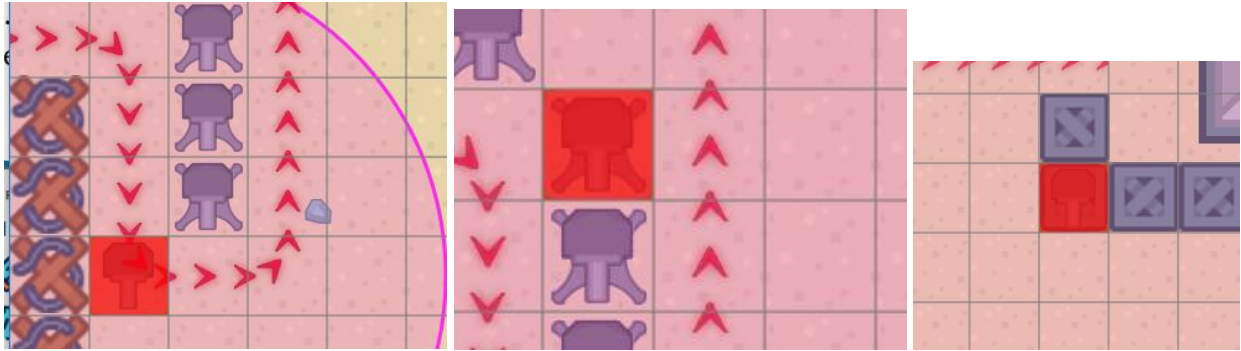


You should place some tower before spawning first wave. Each tower's cost is displayed below its image, selected tower's detail will be displayed on Tower detail panel, the radius of targeting of tower will also be displayed.

You can do so by clicking tower on menu to select and place it on game area (Towers will align to the grid). The path will change in real time to reflect what path would be if tower is placed there. You can cancel placing by right click or clicking it's button again. Enemy will take their shortest path to your base.

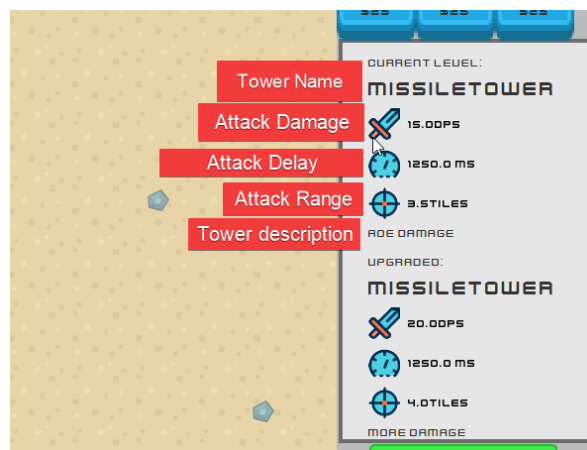
Shortcut Keys:

- Digit 1 – 9: select a Tower from menu.
- S: Sell current tower
- D: Upgrade current tower
- G: Spawn next wave
- Z: (Cheat) Give you \$1000 instantly



You can't place tower to block enemy path. Also, you can't place tower on top of another tower or obstacle. It will highlight in red to indicate that tower can't be placed there

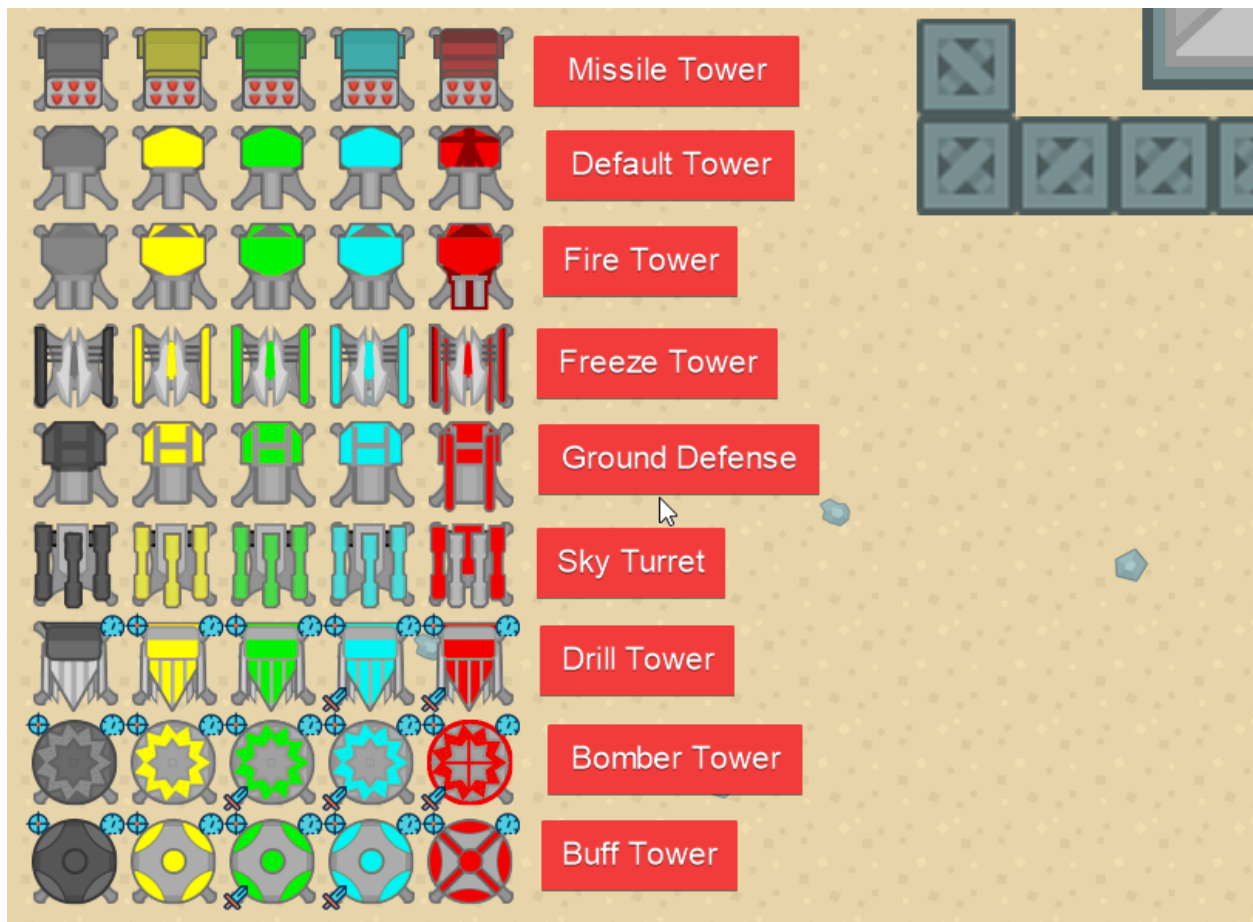
Tower detail panel



You can upgrade tower by selecting it (clicking on it, press shortcut key 1-9) and press the upgrade button (the green one). It will cost some money.

You can sell tower to gain half amount of money spent on it.

You can see tower's stat before it's upgraded; the stats below is upgraded tower's stats.



There are 9 different towers with various price and ability. Each can be upgraded up to level 5.

Tower Types

Name	Damage	Range	Attack Speed	Target	Extra
Missile Tower	High	Far	Slow	Both	Attack Multiple enemy in range, Missile explodes on impact More targets on each upgrade
Default Tower	Low	Far	Fast	Both	At level 5 it will shoot split shot.
Fire Tower	Medium	Medium	Slow	Ground	Deal continuous damage by shooting fire on ground
Freeze Tower	Very low	Medium	Very Fast	Both	Slow multiple enemy, more slowness with each upgrade. At level 5. Ice projectile will pierce through enemy
Ground Defense	High	Medium	Medium	Ground	At level 5 it will shoot split shot.
Sky Turret	High	Medium	Fast	Air	At level 5 it will shoot split shot.
Drill Tower	Very low	Medium	Fast	Both	Amplify damage of enemy. Effectiveness increase with level
Bomber Tower	High	Short	Slow	Ground	Deal damage to every enemy in range
Buff Tower	None	None	None	None	Increase adjacent tower's attack speed level 3: also increase range, level 5: also damage.






















Once you pressed next wave button, enemy will be spawned. And tower will shoot them. destroying them will give you money to place or upgrade more tower to defend more of them.



Don't let them reach your base! Or you will lose your base health.




Enemy Types

Image	Name		Description
	Normal Truck	Trucks spawn their 5 of respective type of soldier on their death. They're tougher and move faster than soldiers	
	Fast Truck		Fast variant of enemy moves faster but have less health and armor
	Armored Truck		Armored variant of enemy has more armor but move slower than normal ones
	Heavy Truck		Heavy variants are heavily armored and have a lot of health.
	Boss Truck	Boss Truck spawns 5 heavy trucks on death.	
	Normal Plane	Planes cannot be shot by ground towers. They have more health than trucks and moves a bit faster.	
	Fast Plane		Fast variant of enemy moves faster but have less health and armor
	Armored Plane		Armored variant of enemy has more armor but move slower than normal ones
	Heavy Plane		Heavy variants are heavily armored and have a lot of health.
	Normal Tank	Tanks have more armor than any other types of enemy, but they are also slowest.	
	Fast Tank		Fast variant of enemy moves faster but have less health and armor
	Armored Tank		Armored variant of enemy has more armor but move slower than normal ones
	Heavy Tank		Heavy variants are heavily armored and have a lot of health.
	Boss Tank	Boss Tank gives nearby enemy Armor Buff	

		which reduce damage taken	
	Normal Solider	Soldier are basic type of enemy in this game, they're also spawned when trucks are destroyed	
	Fast Soldier		Fast variant of enemy moves faster but have less health and armor
	Armored Soldier		Armored variant of enemy has more armor but move slower than normal ones
	Heavy Soldier		Heavy variants are heavily armored and have a lot of health.
	Boss soldier	Boss soldier spawn heavy sodiers periodically	

You will lose your lives (base health) if enemy reach your base, if life drops down to zero you will lose the game and you will have to start over the game.

Some special effect

Image	Effect	Description
	Drill Tower effect	Enemy Affected by Drill tower will take more damage, indicated by Yellow Health bar
	Freeze tower Effect	Enemy Affected by Freeze tower will have their movement speed reduced. Indicated by being tinted blue.
	Armor Buff Effect	Enemy affected by Tank Boss' Armor Buff will take less damage. Indicated by Shield.

Ending Game

The game ends either when you defeat all 20 waves or you lose all base health (live)



If you defend all 20 waves of enemy you will win the game.

Example of creating long path



2. Implementation Details

2.1. Package buff

2.1.1. Class Buff implements IExpirable (Abstract)

2.1.1.1. Field

protected double duration	Duration time buff.
protected double age	Parameter for counting time.

protected double id	Buff's id
---------------------	-----------

2.1.1.2. Constructor

public Buff(int id, double duration)	Initialize all fields.
--------------------------------------	------------------------

2.1.1.3. Method

public void age()	Increase age.
public boolean isExpired()	Return status of expiration
public int getId()	Return id
public abstract void applyTo(Entity e)	Apply buff to e

2.1.2. Class AttackSpeedBuff extends Buff

2.1.2.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for increase attackSpeed

2.1.2.2. Constructor

public AttackSpeedBuff (double duration, double mutliplier)	Initialize all fields.
--	------------------------

2.1.2.3. Method

public void applyTo(Entity e)	If e is tower increase attackSpeed by mutliplier
-------------------------------	---

2.1.3. Class DamageBuff extends Buff

2.1.3.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for increase damage

2.1.3.2. Constructor

public DamageBuff (double duration, double multiplier)	Initialize all fields.
--	------------------------

2.1.3.3. Method

public void applyTo(Entity e)	If e is tower increase damage by mutliplier
-------------------------------	--

2.1.4. Class DamageTakenDebuff extends Buff

2.1.4.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for increase damageTaken

2.1.4.2. Constructor

public DamageTakenBuff (double duration, double multiplier)	Initialize all fields.
--	------------------------

2.1.4.3. Method

public void applyTo(Entity e)	If e is monster increase damageTeken by mutliplier
-------------------------------	---

2.1.5. Class MoveSpeedBuff extends Buff

2.1.5.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for increase moveSpeed

2.1.5.2. Constructor

public MoveSpeedBuff (double duration, double mutliplier	Initialize all fields. and Set age = 0.
---	---

2.1.5.3. Method

public void applyTo(Entity e)	If Entity is monster increase moveSpeed by mutliplier
-------------------------------	--

2.1.6. Class RangeBuff extends Buff

2.1.6.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for increase range

2.1.6.2. Constructor

public rangeBuff (double duration, double mutliplier)	Initialize all fields.
--	------------------------

2.1.6.3. Method

public void applyTo(Entity e)	If Entity is tower Increase range by mutliplier
-------------------------------	--

2.1.7. Class SlowDebuff extends Buff

2.1.7.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for decrease moveSpeed

2.1.7.2. Constructor

public SlowdeBuff (double duration, double mutliplier)	Initialize all fields. and Set age = 0.
---	---

2.1.7.3. Method

public void applyTo(Entity e)	If Entity is monster decrease moveSpeed by mutliplier
-------------------------------	--

2.1.8. Class ArmorDebuff extends Buff

2.1.8.1. Field

public static final int ID	Buff's id
private double mutliplier	Parameter for decrease Armor

2.1.8.2. Constructor

public ArmorDebuff (double duration, double mutliplier)	Initialize all fields.
--	------------------------

2.1.8.3. Method

public void applyTo(Entity e)	If Entity is monster decrease Armor by mutliplier
-------------------------------	--

2.2. Package constant

2.2.1. Class Images

2.2.1.1. Field

public static Image ["image's name"]	Image
public static final Image[] ["image's name"]	Array of image frame
public static final Image[][] ["image's name"]	Array of image
public static HashMap <String, Image[]> towerImages	Map of tower's image
private static DoubleProperty progress	Parameter for loading bar

2.2.1.2. Method

public static void loadResource()	Load all image
public static DoubleProperty getProgress()	return progress

2.2.2. Class Maps

2.2.2.1. Field

private static ArrayList<Map> maps	Array that contain Map
------------------------------------	------------------------

2.2.2.2. Method

public static void loadMap()	Add map(pixel array) to maps
------------------------------	------------------------------

public static Map getMap(int i)	Return “I”th map
public static int numOfAvaibleMaps()	Return maps’ size

2.2.3. Class Numbers

2.2.3.1. Field

public static final int WIN_WIDTH	Width of windows
public static final int WIN_HEIGHT	Height of windows
public static final int TILE_SIZE	Size of map pixel
public static final int COLUMNS	Number of columns in game pane
public static final int ROWS	Number of rows in game pane

2.2.4. Class Other

2.2.4.1. Field

public static int[][] dir	Array for circle loop (use for finding path)
public static Font normalButtonFont	Font fot bottoms.

2.2.4.2. Method

public static Font loadFontWithSize(int size)	Load Font and set size
---	------------------------

**there is a static initializer block that set normalButtonFont by loadFontWithSize

2.2.5. Class Sounds

2.2.5.1. Field

public static AudioClip ["name"]	AudioClip
----------------------------------	-----------

2.2.5.2. Method

private static AudioClip loadSound(String path)	Load AudioClip from "audio/" + path
public static void loadResources()	Load all AudioClip

2.2.6. Class TowerStats

2.2.6.1. Field

public static HashMap<String, HashMap<String, Object[]>> data	Map of tower's stat
---	---------------------

2.2.6.2. Method

public static void loadTower()	Load all of tower's stat to data
--------------------------------	----------------------------------

public static Object getData (String tower, String field, int level)	Return tower's stat
---	---------------------

2.3. Package controller

2.3.1. Class SuperManager

2.3.1.1. Field

private static SuperManager instance	
private BooleanProperty isInGameProp	inGame's status
private BooleanProperty isGamePausedProp	inGamePasused's status
private BooleanProperty nextWaveAvailableProp	nextWaveAvailable's status
private BooleanProperty canUpgradeProp	canUpgrade's status
private BooleanProperty canSellProp	canSell's status
private IntegerProperty towerChoiceProp	Number of towerChoice
private IntegerProperty gameStateProp	Number of gameState
private BooleanProperty shouldDisplayPathProp	shouldDisplay's status
private BooleanProperty shouldDisplayPath	shouldDisplay's status

2.3.1.2. Constructor

public SuperManager()	Set default parameter.
-----------------------	------------------------

2.3.1.3. Method

public void onGamePause()	Pause Game
public void onResumeGame()	Resume Game
public void onLeaveGame()	Leave Game and go to main menu
public void onReset()	Reset Game

****generate all Field's getter except shouldDisplayPath**

2.4. Package controller.game

2.4.1. Class GameManager

2.4.1.1. Field

private static final GameManager instance	Instance GameManager
ArrayList<Tower> towers	Array of tower
ArrayList<Monster> monsters	Array of monster
ArrayList<Projectile> projectiles	Array of projectiles
ArrayList<Particle> particles	Array of particles
TileStack[][] placedTiles	Array of placedTiles
cpp.pii[][] path	Array of path
cpp.pff mousePos	Position of mouse

cpp.pii tilePos	Position of tile
int money;	shouldDisplay's status
Tower selectedTower;	selectedTower
cpp.pii startTilePos;	Position of start tile
cpp.pii endTilePos;	Position of end tile
int lives;	Number of lives
21oolean isInitialized;	Initialized's status
Updater updater;	Use for render
TowerManager towerManager;	Use for manage tower
Handler handler;	Use for handler event
private BFSAlgo bfs	breadth first search Algo use for findingpath
private Image bgImage	Background image

2.4.1.2. Constructor

public GameManager()	Set default parameter.
----------------------	------------------------

2.4.1.3. Method

public void reset()	Clear map and other parameter in game
public boolean isInitialized()	Return Initialized's status
public void loadMap(int mapId)	Load 'l'th map to get ready for rendering
public Tower createTower(int towerChoice, int x, int y)	call createTower in tower manager class by choice.
Public void updateMousePos(MouseEvent e)	Call updateMousePos in handler
public void requestNextWave()	Call requestNextWave in updater
public void sellTower()	Call sellTower in tower manager
public void upgradeTower()	Call upgradeTower in tower manager
public void handleClick(MouseEvent e)	Call handleClick in handler
public void handleTileClick(int x, int y)	Call handleTileClick in handler
public boolean isPlaceable(int x, int y)	Return Placeable's status
public boolean isWalkable(int x, int y)	Return Walkable's status
public void handleKeyPress(KeyEvent e)	Call handleKeyPress in handler
public int getTowerChoice()	Return TowerChoice from supermanager
public int getSelectedPosition()	return getTilePos();
public void addMoney(int i)	Increase money by i

public boolean boundCheck(int x, int y)	Call boundCheck in tower manager
public void addMonsterDefault(Monster mon)	Add Monster to game pane by method in SharedObject
public void addMonster(Monster mon)	Add Monster to game pane by method in SharedObject
public void addProjectile(Projectile proj)	Add Projectile to game pane by method in SharedObject
public void removeMonster(Monster mon)	Remove Monster from game pane by method in SharedObject
public void removeProjectile(Projectile proj)	Remove Projectile from game pane by method in SharedObject
public void removeParticle(Particle part)	Remove Particle from game pane by method in SharedObject
public void updatePath()	Update path by BFSAlgo

**generate getter of mousepos, path, tower, monsters, lives, money, selectedTower, startTilePos, endTilePos, tilePos, placedTiles, bgimage
 **generate setter of selectedTower, tilePos, placedTiles

2.4.2. Class Handler

2.4.2.1. Field

private GameManager gm;	GameManger that this Handler is binding to
-------------------------	--

2.4.2.2. Constructor

public Handler(GameManager gm)	Initialize a field.
--------------------------------	---------------------

2.4.2.3. Method

public void updateMousePos(MouseEvent e)	Update mouse position
public boolean shouldHandle(MouseEvent e)	Return Mouse is out of game pane or not
public void handleClick(MouseEvent e)	Manage Click button event and call the right method
public void handleTileClick(int x, int y)	Manage Click tower in map event and and call the right method
public void handleKeyPress(KeyEvent e)	Manage pressing Hot key and and call the right method
public void setTowerChoice(int towerChoice)	call Set TowerChoice SuperManager

2.4.3. Class MonsterSpawner

2.4.3.1. Field

private static MonsterSpawner instance	
private static ArrayList<MonsterSpawningStage> stages	Array of Spawning stages
private int index;	Index of Spawning

2.4.3.2. Method

public void nextWave()	Spawn next wave monster
public void pauseWave()	pause wave monster
public void cancelWave()	Cancel wave monster
public void reset()	Cancel wave monster and set index = 0
public int getLevel()	return index
public boolean isReady()	Return thread MonsterSpawnerThread's status
public static MonsterSpawner getInstance()	return instance

2.4.4. Class MonsterSpawnerThread extends Thread

2.4.4.1. Field

private static MonsterSpawnerThread instance	monster wave spawning's thread
private static ConcurrentLinkedQueue<MonsterSpawningSequence> jobs	Queue of MonsterSpawningSequence
private static MonsterSpawningSequence currentJob	Present job
private static boolean shouldBreak	Status for interrupt thread

2.4.4.2. Constructor

public MonsterSpawnerThread()	Build the thread to spawn the monster wave
-------------------------------	--

2.4.4.3. Method

public static void addSequence (MonsterSpawningSequence ...seq)	Add seq to jobs
public static void onGamePause()	Call onGamePause in MonsterSpawningSequence
public static void onGameResume()	Call onGameResume in MonsterSpawningSequence
public static MonsterSpawnerThread getInstance()	return instance
public static boolean isIdle()	If jobs is empty and currentjob is null return true,otherwise return false
public void interrupt()	Set shouldBreak = true
public void onGameReset()	Interrupt current job and clear jobs

2.4.5. Class MonsterSpawningSequence extends Thread

2.4.5.1. Field

private static boolean isPaused	Status of Paused
private static boolean shouldStop	Status of Stop

2.4.5.2. Constructor

public MonsterSpawningSequence(int delay, int delay2, int repeat, Monster ...monstersList)	Build the thread to spawn the monster
public MonsterSpawningSequence(int delay)	empty constructor

2.4.5.3. Method

public static void onGamePause()	Set isPaused = true
public static void onGameResume()	Set isPaused = false
public void onGameReset()	Set shouldstop = true

2.4.6. Class MonsterSpawningStage

2.4.6.1. Field

private ArrayList<MonsterSpawningSequence> seqs	ArrayList of MonsterSpawning Sequence in this Stage
---	---

2.4.6.2. Constructor

public MonsterSpawningStage(int delay, MonsterSpawningSequence ...seqs)	Add Sequence and delay in seqs
---	--------------------------------

2.4.6.3. Method

public void play()	Add Sequence to MonsterSpawnerThread
Public void addSequence (MonsterSpawningSequence ...Sequence)	Add sequences in to this stage's ArrayList

2.4.7. Class TowerManager

2.4.7.1. Field

private GameManager gm	Array of MonsterSpawningSequence
private BFSAlgo bfs	this is BreadthFirstSearch instance for tower manager (to calculate preview path)

2.4.7.2. Constructor

public TowerManager(GameManager game)	Set gm = current gameManager
---------------------------------------	------------------------------

2.4.7.3. Method

public void sellTower()	increase money by half price and remove tower
public void removeTower(int x, int y)	Remove tower from game pane
public boolean canUpgrade()	Return ture if selected the tower already ,the tower can upgrade ,and have enough money
public boolean canSell()	Return ture if selected the tower already

void upgradeTower()	Upgrade Tower And set some implicated parameter If cannot place have to alert the error
public Tower createTower(int towerChoice, int x, int y)	createTower by choices in position x y
public void placeAt(int x, int y)	Place the tower to position x y And set some implicated parameter If cannot place have to alert the error
public boolean boundCheck(int x, int y)	x and y have to be in game pane
public boolean isPlaceable(int x, int y)	Return true if can place the tower
public boolean isWalkable(int x, int y)	Return true if monster can walk

2.4.8. Class Updater

2.4.8.1. Field

private GameManager gm	
private Timeline updateLoop	a Timeline for update game pane

2.4.8.2. Constructor

public Updater(GameManager gm)	Update GameManager
public void update()	Update Game pane
public void requestNextWave()	If current wave end, Call next wave
public boolean shouldSpawnNextWave()	If MonsterSpawner is ready and there is not monster in game pane return true

2.4.8.3. Method

public void play()	Add Sequence to MonsterSpawnerThread
--------------------	--------------------------------------

2.5. Package exception (*all the class extends Exception)

2.5.1. Class FullyUpgradedException

2.5.1.1. Field

private static final long serialVersionUID	-826516752779885631L
--	----------------------

2.5.2. Class NotEnoughMoneyException

2.5.2.1. Field

private static final long serialVersionUID	1783017797965554864L
--	----------------------

2.5.3. Class PathBlockedException

2.5.3.1. Field

private static final long serialVersionUID	-6925175092474437275L
--	-----------------------

2.5.4. Class UnplaceableException

2.5.4.1. Field

private static final long serialVersionUID	8854169360717212347L
--	----------------------

2.6. Package main

2.6.1. Class Main extends Application

2.6.1.1. Field

private static Stage stage	windows
private static MainMenuScene mainMenu;	Main menu scene
private static GameScene gameScene;	Game scene (after select map)

private static LoadingScene loadingScene;	Intro before main menu
---	------------------------

2.6.1.2. Method

public void start(Stage primaryStage)	Set stage component
public static void main(String[] args)	An entry point of the application.
public void stop() throws Exception	Reset game and interrupt MonsterSpawnerThread
public void loadAll()	Load component such as image Maps Sound Towerstat
public static void setScene(Scene scene)	Set scene to the stage
public static MainMenuScene getMainMenu()	return mainMenu
public static GameScene getGameScene()	return gameScene

2.7 Package model

2.7.1 Class Entity

2.7.1.1 Field

protected Image image;	entity's Image
protected double w;	entity's image width, used to calculate bounding box when rendering
protected double h;	entity's image height
protected double x;	entity's x (horizontal) position. This is based on center of entity

protected double y;	entity's y (vertical) position. This is based on center of entity
protected double zIndex;	entity's zIndex
protected double size;	entity's hitbox size (used for collision detection)
protected double rotation = 0;	entity's rotation (0-360) degree
getter & setters for fields zIndex, rotation, x, y	
getter for size, image	

2.7.1.2 Constructor

public Entity(Image image, double x, double y, double zIndex, double size)	Construct entity, with w, h inferred from image if image is null w, h will be 0, 0
--	--

2.7.1.3 Method

public cpp.pff getPosition()	return (x,y) pair of position
public double distanceTo(Entity e)	return distance to Entity e
public double distanceTo(double rx, double ry)	return distance to position (rx, ry)
public boolean isCollideWith(Entity e)	return whether this collides with Entity e (check like two circle touching)
public double getRenderX()	return x of topLeft position of image when this Entity is rendered
public double getRenderY()	return y of topLeft position of image when this Entity is rendered
public void render(GraphicsContext gc)	draw Entity image in to gc , rotate if needed
public void rotateTo(Entity e)	convenience method for rotating to face Entity e
public String toString()	Return string representation of this entity

2.7.2 Class Map

2.7.2.1 Field

private int[][] tiles;	array of integer representing whether each grid(tile) are - placeable : if $x\%4 \ \& \ 1 == 1$ - walkable : if $x\%4 \ \& \ 2 == 2$
private HashMap<Integer, Image> tileMap;	represent map from integer to tile Image. Only used when we want to modify visual of base map.

	- if tile has value x => image of that tile will be tileMap.get(x/4);
private cpp.pii start, end;	pair<int, int> of where enemy will spawn and where enemy will go to
private Image bgImage;	background image of map which is visible when in game
private Image previewImage;	preview image of map in MapSelectionMenu
private int[][] tiles;	array of integer representing whether each grid(tile) are - placeable: if $x\%4 \& 1 == 1$ - walkable: if $x\%4 \& 2 == 2$
private HashMap<Integer, Image> tileMap;	represent map from integer to tile Image. Only used when we want to modify visual of base map. - if tile has value x => image of that tile will be tileMap.get(x/4);
private cpp.pii start, end;	pair<int, int> of where enemy will spawn and where enemy will go to
private Image bgImage;	background image of map which is visible when in game

2.7.2.2 Constructor

public Map(int[][] tiles, HashMap<Integer, Image> tileMap, pii start, pii end, Image previewImage, Image bgImage)	Construct a map using specified details
---	---

2.7.2.3 Method

getters for tiles, tileMap, start, end, previewImage, bgImage	
---	--

2.7.3 Abstract class Monster extends Entity implements Cloneable

2.7.3.1 Field

protected double maxHealth;	maximum hp
protected double health;	current hp
protected double armor;	armor (which reduce damage taken)
protected double moveSpeed;	moveSpeed (scalar)
protected double vx;	x-velocity
protected double vy;	y-velocity

protected ArrayList<Buff> buffs = new ArrayList<>();	buffs affecting this monster
protected WritableImage coldImage;	image when monster is affected by freeze tower
protected int targetFlag;	targetFlag 1 for GroundMonster 2 for FlyingMonster
protected double moveSpeedMultiplier;	multiplier for calculating movespeed after buff
protected double damageTakenMultiplier;	multiplier for calculating damage after buff
protected int money	Reward for killing this monster

2.7.3.2 Constructor

public Monster(Image image, double x, double y, double size, double maxHealth, double armor, double moveSpeed, int money)	construct monster with specified details, and generate coldImage
---	--

2.7.3.3 Method

public void render(GraphicsContext gc)	call super.render() and render health bar and buff
public void onTick()	is called eachUpdate, will call methods preUpdate, updateBuff, updateVelocity, move,
protected void preUpdate()	action before monster is update
public void onDeath()	action when monster is dead, called by Updater
protected abstract void updateVelocity()	movement logic of monster
public void updateBuff()	aply and "age" all buffs
public boolean takeDamage(double damage)	take damage logic (reduced by armor)
public void takePureDamage(double damage)	take damage logic (NOT reduced by armor)
public void forceKill()	kill this monster
public boolean isDead()	return rue when monster is dead (hp <= 0)
public void addMoveSpeedMultiplier(double moveSpeedMultiplier)	increase moveSpeedMultiplier by that amount (can be negative)
public void addDamageTakenMultiplier(double damageTakenMultiplier)	increase damageTakenMultiplier by that amount (can be negative)
public void addBuff(Buff b)	add new buff, remove buff of same type before update
public boolean hasBuff(int id)	return true if monster has buff with that ID
public Monster clone()	clone this monster (used in MonsterSpawner)
public String toString()	return string with monster name + hp + armor + moveSpeed

getters for targetFlag, money, buffs

2.7.4 Interface IExpirable

2.7.4.1 Method

public abstract boolean isExpired()	Return true when this Entity should be removed
-------------------------------------	--

2.7.5 Class Projectile extends Entity implements IExpirable

2.7.5.1 Field

protected double vx, vy;	x, y velocity
protected double maxDistance = 3;	maximum travel distance
protected double maxAge = 5000;	maximum age
protected int targetFlag;	targetFlag x & 2 == 2 if can target Air Monster x & 1 == 1 if can target Ground Monster
protected double distance = 0;	current travel distance
protected double age = 0;	current age
protected boolean isExpired = false;	boolean for forcing expire
protected double damage;	damage on collide

2.7.5.2 Constructor

public Projectile(Image image, double x, double y, double vx, double vy, double maxDistance, double damage)	construct projecitle with specified detail, target flag default to 3
---	--

2.7.5.3 Method

public boolean shouldCollide(Monster m)	return whether should collide (do some action) with Monster m
public abstract boolean collideWith(Monster m)	action on collide with Monster m

public void onTick()	called every update
public void move()	move projectile and rotate accord to velocity angle, update distance
public void preUpdate()	custom action to be called before moving
private void age()	increase projectile age
public boolean isExpired()	return whether projectile should be removed
public boolean forceExpire()	called to mark for removal (such as when hit monster)

2.7.6 Class Particle extends Entity implements IExpirable

2.7.6.1 Field

protected double maxAge;	maximum age
protected double vx, vy;	x, y velocity
protected Image[] frames;	image array of frame
protected double age = 0;	current age
private int frameCount = 0;	frameCount, used in animation
private int nFrames;	number of total frames in animation
private int tickPerFrame = 4;	frame duration, in Tick
public boolean isExpired()	return whether projectile should be removed

2.7.6.2 Constructor

public Particle(Image[] images, double x, double y, double vx, double vy, double maxAge)	real constructor for animated particle, all particle has zIndex = 1.5 by default
public Particle(Image[] images, double x, double y, double vx, double vy, int tickPerFrame)	constructor for animated particle, specify tickPerFrame
public Particle(Image image, double x, double y, double vx, double vy, double maxAge)	constructor for static particle, specify maxAge

2.7.6.3 Method

public void onTick()	called every update
public void move()	Move particle
protected void updateFrame()	update particle frame to be draw
private void age()	increase projectile age
public boolean isExpired()	return whether particle should be removed
public boolean forceExpire()	called to mark for removal

2.7.7 Class Tile extends Entity

2.7.7.1 Field

<code>private boolean isWalkable, isPlaceable;</code>	boolean representing ability to place tower on, and monster to walk on
---	--

2.7.7.2 Constructor

<code>public Tile(Image image, double x, double y, boolean isWalkable, boolean isPlaceable)</code>	construct a tile with specified detail, tile has zIndex = 1 by default.
--	---

2.7.7.3 Method

<code>public boolean isSelectable()</code>	return false ** will be overridden by other
--	---

2.7.8 Class TileStack

2.7.8.1 Field

<code>private Vector<Tile> layers;</code>	vector of tiles in stack
---	--------------------------

2.7.8.2 Constructor

<code>public TileStack()</code>	construct empty Tile Stack
<code>public TileStack(Tile baseTile)</code>	construct Tile Stack and add a tile

2.7.8.3 Method

public Tile select()	return top of stack if I's selectable, else null
public void push(Tile t) throws UnplaceableException	add Tile t in to stack, throw UnplaceableException when top of stack is unplaceable (like when place tower on obstacle)
public void pop()	remove top of stack
public void clear()	clear stack
public boolean isPlaceable()	return true when all of tiles are placeable
public boolean isWalkable()	return true when all of tiles are walkable

2.7.9 Class Tower extends Tile

2.7.9.1 Field

protected double attackCooldown;	base attack cooldown
protected double baseAttack;	attack value before buffed
protected double baseRange;	range value before buffed
protected int price;	value of tower = baseprice plus all upgrade
protected int level;	tower level
protected int targetFlag;	targetFlag (1 for ground, 2 for air, 3 for both)
protected String typeName;	tower type, used to get data from TowerStats class
protected double attack;	attack after buffed
protected double range;	range after buffed
protected double currentCooldown;	time until can attack again
protected double attackSpeedMultiplier;	cooldown reduce faster by this multiplier
protected double rangeMultiplier;	range multiplier increase range
protected double attackMultiplier;	attack multiplier increase attack
protected Monster currentTarget;	target monster (tower targets nearest monster)
protected ArrayList<Buff> buffs = new ArrayList<>();	buffs applied to this tower
protected double minDist;	minimum distance -- distance to target monster

2.7.9.2 Constructor

public Tower(String typeName, double cellX, double cellY)	construct tower with specified type; use data from TowerStats class, tower has zIndex = 2 by default
---	--

2.7.9.3 Method

public void render(GraphicsContext gc)	render tower with base and buffs; call super.render() to draw rotated part
public void render(GraphicsContext gc, boolean showRadius)	used to force render tower with radius
public void renderBuff(GraphicsContext gc)	render buffs applied to this tower (called by render)
public void renderExtraElements(GraphicsContext gc)	render tower radius
public boolean canUpgrade()	return true when level < 5
public boolean upgrade() throws FullyUpgradedException	upgrade tower, throw Exception when already fully upgraded (level 5)
public boolean isSelectable()	override Tile's isSelectable
public void onTick()	logic to be called on each loop
public void updateBuff()	part of logic: apply and age buffs
private void updateCooldown()	part of logic: reduce tower cooldown
private boolean shouldAcquireTarget()	part of logic: condition for targeting monster on this tick (default: when ready to fire)
private void acquireTarget()	part of logic: loop through monster to find nearest target
public boolean isInRange(Monster m)	part of logic: return true when monster isInRange, and targetFlag is valid
public void tryTarget(Monster m)	part of logic: precondition before selecting monster as target (not dead etc)
public void fire()	part of logic: rotate to monster shoot bullet and set cooldown
public void clearTarget()	part of logic: reset target to null
public void addBuff(Buff b)	add new buff, remove buff of same type before update
public boolean hasBuff(int id)	return true if monster has buff with that ID
public void addAttackSpeedMultiplier(double attackSpeedMultiplier)	adds attackSpeedMultiplier
public void addRangeMultiplier(double rangeMultiplier)	adds arangeMultiplier
public void addAttackMultiplier(double attackMultiplier)	adds attackMultiplier
public String getDescription()	return this tower description from class TowerStats
public String getUpgradedDescription()	return description of next level upgrade from TowerStats
public double getUpgradedAttackCooldown()	return attackCooldown of next level upgrade from TowerStats
public double getUpgradedAttack()	return attack of next level upgrade from TowerStats

public double getUpgradedRange()	return range of next level upgrade from TowerStats
public int getUpgradePrice()	return price of next level upgrade from TowerStats
getters for attackCooldown, attack, range, price, buffs setters for price	

2.8 Package model.monster

2.8.1 Class GroundMonster extends Monster

2.8.1.1 Field

(no new fields)

2.8.1.2 Constructor

public GroundMonster(Image image, double x, double y, double size, double health, double armor, double moveSpeed, int money)	construct a monster with targetFlag = 1
--	---

2.8.1.3 Method

public void updateVelocity()	movement logic using Breadth First Search
------------------------------	---

2.8.2 FlyingMonster extends Monster

2.8.2.1 Field

(no new fields)

2.8.2.2 Constructor

public FlyingMonster(String name, Image image, double x, double y, double size, double health, double armor, double moveSpeed, int money)	construct a monster with targetFlag = 2
---	---

2.8.2.3 Method

protected void updateVelocity()	movement logic: displacement vector to endTile
public String toString()	return super.toString() + "Flying"

2.8.3 SplittingMonster extends Monster

2.8.3.1 Field

protected boolean wasForceKilled;	whether this die by forceKill()
protected Class<? extends Monster> childClass;	class for monster that's spawned when this dies

2.8.3.2 Constructor

public FlyingMonster(String name, Image image, double x, double y, double size, double health, double armor, double moveSpeed, int money)	construct a monster with targetFlag = 2
---	---

2.8.3.3 Method

protected void updateVelocity()	movement logic: displacement vector to endTile
public String toString()	return super.toString() + "Flying"

2.8.4 Car extends SplittingMonster

2.8.4.1 Field

public static double DEFAULT_SIZE;	default hitbox size for Car types
protected double level = 0;	Level of this monster

2.8.4.2 Constructor

public FlyingMonster(String name, Image image, double x, double y, double size, double health, double armor, double moveSpeed, int money, Class<? extends Monster> childClass)	calls SplittingMonster constructor, set level to 0
--	--

2.8.4.3 Method

public Car(Image image, double x, double y, double health, double armor, double moveSpeed,	calls SplittingMonster constructor, set level to 0
public void onDeath()	logic on death : create monster on death by calling childClass' constructor. Also create explosion onDeath, and play sound
public void takeDamage(double damage)	spawn spark when take damage
protected void updateVelocity()	movement logic: displacement vector to endTile

2.8.5 CarArmored extends Car

2.8.5.1 Field

private static final Image DEFAULT_IMAGE = Images.armoredCar;	default image
private static final double DEFAULT_HEALTH = 150;	default health
private static final double DEFAULT_ARMOR = 5;	default armor
private static final double DEFAULT_MS = 1.0;	default ms
private static final int DEFAULT_MONEY = 12;	default money

2.8.5.2 Constructor

public CarArmored(double x, double y, double health, double armor, double moveSpeed, int money)	construct car with specified stats and calculate level based on health, which spawn armored soldiers
public CarArmored(double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.5.3 Method

(no new methods)

2.8.6 CarFast extends Car

2.8.6.1 Field

private static final Image DEFAULT_IMAGE = Images.fastCar;	default image
private static final double DEFAULT_HEALTH = 80;	default health
private static final double DEFAULT_ARMOR = 0;	default armor
private static final double DEFAULT_MS = 1.7;	default ms
private static final int DEFAULT_MONEY = 10;	default money

2.8.6.2 Constructor

public CarFast(double x, double y, double health, double armor, double moveSpeed, int money)	construct car with specified stats and calculate level based on health which spawn fast soldiers
public CarFast(double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.6.3 Method

(no new methods)

2.8.7 CarHeavy extends Car

2.8.7.1 Field

private static final Image DEFAULT_IMAGE = Images.heavyCar;	default image
private static final double DEFAULT_HEALTH = 500;	default health
private static final double DEFAULT_ARMOR = 20;	default armor
private static final double DEFAULT_MS = 0.5;	default ms
private static final int DEFAULT_MONEY = 15;	default money

2.8.7.2 Constructor

public CarHeavy (double x, double y, double health, double armor, double moveSpeed, int money)	construct car with specified stats and calculate level based on health which heavy armored soldiers
public CarHeavy (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.7.3 Method

(no new methods)

2.8.8 CarNormal extends Car

2.8.8.1 Field

private static final Image DEFAULT_IMAGE = Images.normalCar;	default image
private static final double DEFAULT_HEALTH = 100;	default health
private static final double DEFAULT_ARMOR = 0;	default armor
private static final double DEFAULT_MS = 1.3;	default ms

private static final int DEFAULT_MONEY = 10;	default money
--	---------------

2.8.8.2 Constructor

public CarNormal (double x, double y, double health, double armor, double moveSpeed, int money)	construct car with specified stats and calculate level based on health which spawn normal soldiers
public CarNormal (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.8.3 Method

(no new methods)

2.8.9 CarBoss extends Car

2.8.9.1 Field

private static final Image DEFAULT_IMAGE = Images.heavyCar;	default image
private static final double DEFAULT_HEALTH = 800;	default health
private static final double DEFAULT_ARMOR = 45;	default armor
private static final double DEFAULT_MS = 1.3;	default ms
private static final int DEFAULT_MONEY = 100;	default money

2.8.9.2 Constructor

public CarBoss (double x, double y, double health, double armor, double moveSpeed, int money)	construct car with specified stats and calculate level based on health which spawn heavy cars
public CarBoss (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.9.3 Method

public void render(GraphicsContext gc)	Render aura then call super.render();
--	---------------------------------------

2.8.10 Plane extends FlyingMonster

2.8.10.1 Field

private static final double DEFAULT_SIZE = 0.5;	default size
protected double level = 0;	default level
private static double x0 = 13, y0 = 5, scale = 1.07;	Some value related to calculating shadow

2.8.10.2 Constructor

public Plane(Image image, double x, double y, double health, double armor, double moveSpeed, int money)	Calls FlyingMonster constructor
---	---------------------------------

2.8.10.3 Method

public void onDeath()	Explosion when death, and play sound
public void render(GraphicsContext gc)	Render calculated shadow using PerspectiveTransform

2.8.11 PlaneArmored extends Plane

2.8.11.1 Field

private static final Image DEFAULT_IMAGE = Images.armoredPlane;	default image
private static final double DEFAULT_HEALTH = 170;	default health
private static final double DEFAULT_ARMOR = 10;	default armor
private static final double DEFAULT_MS = 0.8;	default ms
private static final int DEFAULT_MONEY = 8;	default money

2.8.11.2 Constructor

public PlaneArmored (double x, double y, double health, double armor, double moveSpeed, int money)	construct plane with specified stats and calculate level based on health,
public PlaneArmored (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.11.3 Method

(no new methods)

2.8.12 PlaneFast extends Plane

2.8.12.1 Field

private static final Image DEFAULT_IMAGE = Images.fastPlane;	default image
private static final double DEFAULT_HEALTH = 100;	default health
private static final double DEFAULT_ARMOR = 2;	default armor
private static final double DEFAULT_MS = 2;	default ms
private static final int DEFAULT_MONEY = 8;	default money

2.8.12.2 Constructor

public PlaneFast (double x, double y, double health, double armor, double moveSpeed, int money)	construct plane with specified stats and calculate level based on health
---	--

public PlaneFast (double x, double y, double modifier)	lazy constructor with stats based on default stats
--	--

2.8.12.3 Method

(no new methods)

2.8.13 PlaneHeavy extends Plane

2.8.13.1 Field

private static final Image DEFAULT_IMAGE = Images.heavyPlane;	default image
private static final double DEFAULT_HEALTH = 250;	default health
private static final double DEFAULT_ARMOR = 20;	default armor
private static final double DEFAULT_MS = 0.4;	default ms
private static final int DEFAULT_MONEY = 13;	default money

2.8.13.2 Constructor

public PlaneHeavy (double x, double y, double health, double armor, double moveSpeed, int money)	construct plane with specified stats and calculate level based on health
public PlaneHeavy (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.13.3 Method

(no new methods)

2.8.14 PlaneNormal extends Plane

2.8.14.1 Field

private static final Image DEFAULT_IMAGE = Images.normalPlane;	default image
private static final double DEFAULT_HEALTH = 100;	default health
private static final double DEFAULT_ARMOR = 3;	default armor
private static final double DEFAULT_MS = 1.2;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.14.2 Constructor

public PlaneNormal (double x, double y, double health, double armor, double moveSpeed, int money)	construct plane with specified stats and calculate level based on health
---	--

public PlaneNormal (double x, double y, double modifier)	lazy constructor with stats based on default stats
--	--

2.8.14.3 Method

(no new methods)

2.8.15 Soldier extends GroundMonster

2.8.15.1 Field

private static final double DEFAULT_SIZE = 0.2;	default size
protected double level = 0;	default level

2.8.15.2 Constructor

public Soldier(Image image, double x, double y, double health, double armor, double moveSpeed, int money)	Calls GroundMonster constructor
---	---------------------------------

2.8.15.3 Method

public void onDeath()	Spawn blood on death
-----------------------	----------------------

2.8.16 SoldierArmored extends Soldier

2.8.16.1 Field

private static final Image DEFAULT_IMAGE = Images.armoredSoldier;	default image
private static final double DEFAULT_HEALTH = 30;	default health
private static final double DEFAULT_ARMOR = 2;	default armor
private static final double DEFAULT_MS = 0.75;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.16.2 Constructor

public SoldierArmored (double x, double y, double health, double armor, double moveSpeed, int money)	construct Soldier with specified stats and calculate level based on health
public SoldierArmored (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.16.3 Method

(no new methods)

2.8.17 SoldierFast extends Soldier

2.8.17.1 Field

private static final Image DEFAULT_IMAGE = Images.fastSoldier;	default image
private static final double DEFAULT_HEALTH = 15;	default health
private static final double DEFAULT_ARMOR = 0;	default armor
private static final double DEFAULT_MS = 2;	default ms
private static final int DEFAULT_MONEY = 3;	default money

2.8.17.2 Constructor

public SoldierFast (double x, double y, double health, double armor, double moveSpeed, int money)	construct Soldier with specified stats and calculate level based on health
public SoldierFast (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.17.3 Method

(no new methods)

2.8.18 SoldierHeavy extends Soldier

2.8.18.1 Field

private static final Image DEFAULT_IMAGE = Images.heavySoldier;	default image
private static final double DEFAULT_HEALTH = 50;	default health
private static final double DEFAULT_ARMOR = 4;	default armor
private static final double DEFAULT_MS = 0.6;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.18.2 Constructor

public SoldierHeavy (double x, double y, double health, double armor, double moveSpeed, int money)	construct Soldier with specified stats and calculate level based on health
public SoldierHeavy (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.18.3 Method

(no new methods)

2.8.19 SoldierNormal extends Soldier

2.8.19.1 Field

private static final Image DEFAULT_IMAGE = Images.normalSoldier;	default image
private static final double DEFAULT_HEALTH = 15;	default health
private static final double DEFAULT_ARMOR = 0;	default armor
private static final double DEFAULT_MS = 1.3;	default ms
private static final int DEFAULT_MONEY = 3;	default money

2.8.19.2 Constructor

public SoldierNormal (double x, double y, double health, double armor, double moveSpeed, int money)	construct Soldier with specified stats and calculate level based on health
public SoldierNormal (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.19.3 Method

(no new methods)

2.8.20 SoldierBoss extends Soldier

2.8.20.1 Field

private static final Image DEFAULT_IMAGE = Images.heavySoldier;	default image
private static final double DEFAULT_HEALTH = 50;	default health
private static final double DEFAULT_ARMOR = 4;	default armor
private static final double DEFAULT_MS = 0.6;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.20.2 Constructor

public SoldierBoss (double x, double y, double health, double armor, double moveSpeed, int money)	construct Soldier with specified stats and calculate level based on health
---	--

public SoldierBoss (double x, double y, double modifier)	lazy constructor with stats based on default stats
--	--

2.8.20.3 Method

public void render(GraphicsContext gc)	Render aura to indicate this is a boss
protected void preUpdate()	Random chance to summon HeavySoldier

2.8.21 Tank extends GroundMonster

2.8.21.1 Field

private static final double DEFAULT_SIZE = 0.7;	default size
protected double level = 0;	default level

2.8.21.2 Constructor

public Tank(Image image, double x, double y, double health, double armor, double moveSpeed, int money)	Calls GroundMonster constructor, Sets level to 0
--	--

2.8.21.3 Method

public void onDeath()	Create explosion on death, play sound
-----------------------	---------------------------------------

2.8.22 TankArmored extends Tank

2.8.22.1 Field

private static final Image DEFAULT_IMAGE = Images.armoredTank;	default image
private static final double DEFAULT_HEALTH = 300;	default health
private static final double DEFAULT_ARMOR = 20;	default armor
private static final double DEFAULT_MS = 0.4;	default ms
private static final int DEFAULT_MONEY = 20;	default money

2.8.22.2 Constructor

public TankArmored (double x, double y, double health, double armor, double moveSpeed, int money)	construct Tank with specified stats and calculate level based on health
---	---

public TankArmored (double x, double y, double modifier)	lazy constructor with stats based on default stats
--	--

2.8.22.3 Method

(no new methods)

2.8.23 TankFast extends Tank

2.8.23.1 Field

private static final Image DEFAULT_IMAGE = Images.fastTank;	default image
private static final double DEFAULT_HEALTH = 150;	default health
private static final double DEFAULT_ARMOR = 10;	default armor
private static final double DEFAULT_MS = 1.3;	default ms
private static final int DEFAULT_MONEY = 3;	default money

2.8.23.2 Constructor

public TankFast (double x, double y, double health, double armor, double moveSpeed, int money)	construct Tank with specified stats and calculate level based on health
public TankFast (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.23.3 Method

(no new methods)

2.8.24 TankHeavy extends Tank

2.8.24.1 Field

private static final Image DEFAULT_IMAGE = Images.heavyTank;	default image
private static final double DEFAULT_HEALTH = 500;	default health
private static final double DEFAULT_ARMOR = 30;	default armor
private static final double DEFAULT_MS = 0.6;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.24.2 Constructor

public TankHeavy (double x, double y, double health, double armor, double moveSpeed, int money)	construct Tank with specified stats and calculate level based on health
public TankHeavy (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.24.3 Method

(no new methods)

2.8.25 TankNormal extends Tank

2.8.25.1 Field

private static final Image DEFAULT_IMAGE = Images.normalTank;	default image
private static final double DEFAULT_HEALTH = 250;	default health
private static final double DEFAULT_ARMOR = 0;	default armor
private static final double DEFAULT_MS = 1.3;	default ms
private static final int DEFAULT_MONEY = 3;	default money

2.8.25.2 Constructor

public TankNormal (double x, double y, double health, double armor, double moveSpeed, int money)	construct Tank with specified stats and calculate level based on health
public TankNormal (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.25.3 Method

(no new methods)

2.8.26 TankBoss extends Tank

2.8.26.1 Field

private static final Image DEFAULT_IMAGE = Images.heavyTank;	default image
private static final double DEFAULT_HEALTH = 500;	default health
private static final double DEFAULT_ARMOR = 30;	default armor
private static final double DEFAULT_MS = 0.6;	default ms
private static final int DEFAULT_MONEY = 5;	default money

2.8.26.2 Constructor

public TankBoss (double x, double y, double health, double armor, double moveSpeed, int money)	construct Tank with specified stats and calculate level based on health
public TankBoss (double x, double y, double modifier)	lazy constructor with stats based on default stats

2.8.26.3 Method

public void render(GraphicsContext gc)	Render aura to indicate this is a boss
protected void preUpdate()	Apply armor buffs to nearby monsters

2.9 Package model.particle

2.9.1 Class Blood extends Particle

2.9.1.1 Field

(no new fields)

2.9.1.2 Constructor

public Blood(double x, double y, double vx, double vy)	Create blood particle with tickPerFrame = 3
--	---

2.9.1.3 Method

(no new methods)

2.9.2 Class FadingParticle extends Particle

2.9.2.1 Field

(no new fields)

2.9.2.2 Constructor

public FadingParticle(Image image, double x, double y, double vx, double vy, double maxAge)	It just calls Particle's constructor
---	--------------------------------------

2.9.2.3 Method

public double transparency()	Calculate transparency (when fading)
public void render(GraphicsContext gc)	Render with transparency calculated

2.9.3 Class FireAoE extends Particle

2.9.3.1 Field

protected double damage;	Damage per second for fire AoE
--------------------------	--------------------------------

2.9.3.2 Constructor

public FireAoE(Image[] images, double x, double y, double vx, double vy, double maxAge, double radius, double damage)	Construct a animated particle, set some radius as size and damage
---	---

2.9.3.3 Method

public void onTick()	Damage nearby monster by colliding
----------------------	------------------------------------

2.9.4 Class Corpse extends FadingParticle

2.9.4.1 Field

(no new fields)

2.9.4.2 Constructor

public Corpse(Image image, double x, double y, double vx, double vy)	Create fading particle with maxAge = 5000.0
--	---

2.9.4.3 Method

public double transparency()	Calculate transparency
------------------------------	------------------------

2.9.5 Class Explosion extends Particle

2.9.5.1 Field

(no new fields)

2.9.5.2 Constructor

public Explosion(Image[] image, double x, double y, double vx, double vy)	Crates particle with specified images, tickPerFrame = 3, and zIndex = 3 (to ensure visibility)
---	--

2.9.5.3 Method

public double transparency()	Calculate transparency
------------------------------	------------------------

2.9.6 Class Crater extends FadingParticle

2.9.6.1 Field

<code>private static Image DEFAULT_IMAGE = Images.crater;</code>	Default image for crater
--	--------------------------

2.9.6.2 Constructor

<code>public Crater(double x, double y, double maxAge)</code>	Create crater at specified position, with vx = vy = 0, and zIndex = 1.1
---	---

2.9.6.3 Method

<code>public double transparency()</code>	Calculate transparency
---	------------------------

2.10 Package model.projectile

2.10.1 Class NormalProjectile extends Projectile

2.10.1.1 Field

--	--

2.10.1.2 Constructor

<code>public NormalProjectile(Image image, double x, double y, double vx, double vy, double maxRange, double damage)</code>	create a projectile with target flag = 3
---	--

2.10.1.3 Method

<code>public boolean collideWith(Monster m)</code>	Deal damage to monster, and expire itself
--	---

2.10.2 Class SplittingProjectile extends NormalProjectile

2.10.2.1 Field

protected double splitDistance;	Distance where this projectile should split
---------------------------------	---

2.10.2.2 Constructor

public SplittingProjectile (Image image, double x, double y, double vx, double vy, double maxRange, double damage)	create a projectile with target flag = 3.
--	---

2.10.2.3 Method

public void preUpdate()	Check distance and split if distance > splitDistance
public boolean collideWith(Monster m)	When collide: split (never collide normally)

2.10.3 Class SplittingAirProjectile extends SplittingProjectile

2.10.3.1 Field

2.10.3.2 Constructor

public SplittingAirProjectile (Image image, double x, double y, double vx, double vy, double maxRange, double damage)	Same as splitting projectile but target flag is 2
---	---

2.10.3.3 Method

public void preUpdate()	Same as SplittingProjectile but spawn AirProjectile
-------------------------	---

2.10.4 Class PiercingProjectile extends NormalProjectile

2.10.4.1 Field

protected int slowTick = 0;	Number of tick this projectile should be slowed (it slows on hit)
-----------------------------	---

2.10.4.2 Constructor

public PiercingProjectile (Image image, double x, double y, double vx, double vy, double maxRange, double damagePerTick)	create a projectile with default image with target flag = 3.
--	--

2.10.4.3 Method

public void move()	Check for slowtick before move (and slow movement)
public void preUpdate()	Reduce slowTick each update

public boolean collideWith(Monster m)	Deal damage, slow itself but doesn't expire
---------------------------------------	---

2.10.5 Class Missile extends NormalProjectile

2.10.5.1 Field

protected double explosionRadius;	Missile Explosion radius
-----------------------------------	--------------------------

2.10.5.2 Constructor

public PiercingProjectile (Image image, double x, double y, double vx, double vy, double maxRange, double damagePerTick)	create a projectile with target flag = 3.
--	---

2.10.5.3 Method

public boolean collideWith(Monster m)	Create explosion, play sound, and deal damage in area
---------------------------------------	---

2.10.6 Class IceProjectile extends NormalProjectile

2.10.6.1 Field

private double slowness;	Movement speed reduction to be applied to monster
private double splashRadius;	Projectile splash radius
private double duration;	Slow debuff duration

2.10.6.2 Constructor

public IceProjectile(double x, double y, double vx, double vy, double maxRange, double damage, double slowness, double splashRadius, double duration)	create a projectile with targetFlag = 3, set slowness, duration, splashRadius
---	---

2.10.6.3 Method

public boolean collideWith(Monster m)	Apply slow debuff to all affected monster
public boolean shouldCollide(Monster m)	Override shouldCollide: don't collide with monster which already has slow debuff

2.10.7 Class HomingMissile extends NormalProjectile

2.10.7.1 Field

private double explosionRadius;	Missile Explosion radius
private Monster target;	Monster this missile should hit

2.10.7.2 Constructor

public HomingMissile(double x, double y, double vx, double vy, double maxRange, double damage, double radius, Monster target)	create a projectile with default image with targetFlag = 3, and set target
---	--

2.10.7.3 Method

public void preUpdate()	Rotate to target monster and set vx, vy correctly
public boolean collideWith(Monster m)	Same as Missile but with target checking

2.10.8 Class IcePiercingProjectile extends PiercingProjectile

2.10.8.1 Field

private double slowness;	Movement speed reduction to be applied to monster
--------------------------	---

2.10.8.2 Constructor

public IcePiercingProjectile(double x, double y, double vx, double vy, double maxRange, double slowness)	call PiercingProjectile constructor, with default image set slowness
--	--

2.10.8.3 Method

public boolean collideWith(Monster m)	Same as PiercingProjectile but slow enemy instead
---------------------------------------	---

2.10.9 Class GroundProjectile extends NormalProjectile

2.10.9.1 Field

private double slowness;	Movement speed reduction to be applied to monster
--------------------------	---

2.10.9.2 Constructor

public GroundProjectile(double x, double y, double vx, double vy, double maxRange, double damage)	Create normal projectile with default image then set targetFlag to 1
---	--

2.10.9.3 Method

public boolean collideWith(Monster m)	Same as PiercingProjectile but slow enemy instead
---------------------------------------	---

2.10.10 Class FireProjectile extends NormalProjectile

2.10.10.1 Field

protected double fireRadius;	Fire radius of fireAoE that spawns on impact
protected double fireDamage;	Fire damage of fireAoE that spawns on impact

2.10.10.2 Constructor

public GroundProjectile(double x, double y, double vx, double vy, double maxRange, double damage)	Create normal projectile with default image then set targetFlag to 1, And set this class' specific fields
---	--

2.10.10.3 Method

public boolean collideWith(Monster m)	Spawn fireAoE at monster location
---------------------------------------	-----------------------------------

2.10.11 Class DrillProjectile extends NormalProjectile

2.10.11.1 Field

private double multiplier;	Damage amplifier factor
private double duration;	Debuff duration

2.10.11.2 Constructor

public DrillProjectile(double x, double y, double vx, double vy, double maxRange, double damage, double multiplier, double duration)	Create normal projectile with default image, and set this class' specific field
--	---

2.10.11.3 Method

public boolean collideWith(Monster m)	Spawn fireAoE at monster location
public boolean shouldCollide(Monster m)	Override shouldCollide: don't collide with monster which already has damage amplify debuff

2.10.12 Class AirProjectile extends NormalProjectile

2.10.12.1 Field

(no new fields)

2.10.12.2 Constructor

public AirProjectile(double x, double y, AirProjectile	Create normal projectile with default image, set targetFlag to 2
public AirProjectile(Image image, double x, double y, double vx, double vy, double maxRange, double damage)	Create air projectile with custom image

2.10.12.3 Method

public boolean collideWith(Monster m)	Deal damage to monster and crate explosion
---------------------------------------	--

2.11 package model.tower

2.11.1 Class NormalTower extends Tower

2.11.1.1 Field

(no new fields)

2.11.1.2 Constructor

public NormalTower(double cellX, double cellY)	Create level 1 Normal Tower
--	-----------------------------

2.11.1.3 Method

public void fire()	Plays sound, Fire splitting projectile if level == 5 Else fire normal projectile
--------------------	--

public String toString()	Return tower name to be displayed on tower info panel
--------------------------	---

2.11.2 Class GroundTower extends Tower

2.11.2.1 Field

(no new fields)

2.11.2.2 Constructor

public GroundTower(double cellX, double cellY)	Create level 1 Ground Tower
--	-----------------------------

2.11.2.3 Method

public void fire()	Plays sound, fire Ground projectile
public String toString()	Return tower name to be displayed on tower info panel

2.11.3 Class BombTower extends Tower

2.11.3.1 Field

(no new fields)

2.11.3.2 Constructor

public BombTower(double cellX, double cellY)	Create level 1 Bom Tower=
--	---------------------------

2.11.3.3 Method

public void fire()	Plays sound, Deal damage to every monster nearby
public String toString()	Return tower name to be displayed on tower info panel

2.11.4 Class BuffTower extends Tower

2.11.4.1 Field

(no new fields)

2.11.4.2 Constructor

public BuffTower(double cellX, double cellY)	Create level 1 Buff Tower at cellX, cellY
--	---

2.11.4.3 Method

public void fire()	Apply buffs to nearby towers
--------------------	------------------------------

public String toString()	Return tower name to be displayed on tower info panel
--------------------------	---

2.11.5 Class DrillTower extends Tower

2.11.5.1 Field

private static final int DEBUFF_DURATION = 1250;	Damage amplify debuff duration
private double multiplier;	Current level's damage amplification

2.11.5.2 Constructor

public DrillTower (double cellX, double cellY)	Create level 1 Drill Tower
--	----------------------------

2.11.5.3 Method

public void fire()	Fires drill projectile
public String toString()	Return tower name to be displayed on tower info panel
public void tryTarget(Monster m)	Override : also check target for debuff before firing
public boolean upgrade() throws FullyUpgradedException	Override: also updates this tower specific stats

2.11.6 Class FireTower extends Tower

2.11.6.1 Field

private double fireRadius;	Fire AoE's radius
----------------------------	-------------------

2.11.6.2 Constructor

public FireTower (double cellX, double cellY)	Create level 1 Fire Tower
---	---------------------------

2.11.6.3 Method

public void fire()	Fires fire projectile
public String toString()	Return tower name to be displayed on tower info panel
public boolean upgrade() throws FullyUpgradedException	Override: also updates this tower specific stats

2.11.7 Class IceTower extends Tower

2.11.7.1 Field

private static final double SLOW_DURATION = 1000;	Slow debuff duration
private double slowness;	Movement speed reduction
private double splashRadius;	IceProjectile splash radius

2.11.7.2 Constructor

public IceTower (double cellX, double cellY)	Create level 1 Ice Tower
--	--------------------------

2.11.7.3 Method

public void fire()	Fires Ice projectile or Piercing Ice Projectile if level == 5
public String toString()	Return tower name to be displayed on tower info panel
public void tryTarget(Monster m)	Override : also check target for slow debuff before firing
public boolean upgrade() throws FullyUpgradedException	Override: also updates this tower specific stats

2.11.8 Class SkyTower extends Tower

2.11.8.1 Field

(no new fields)

2.11.8.2 Constructor

public SkyTower(double cellX, double cellY)	Create level 1 Sky Tower
---	--------------------------

2.11.8.3 Method

public void fire()	Plays sound, Fire Air projectile or splittingAirProjectile if level == 5
public String toString()	Return tower name to be displayed on tower info panel

2.11.9 Class MissileTower extends Tower

2.11.9.1 Field

private double splashRadius;	Missile Splash radius
private PriorityQueue<Monster> targets;	Priority Queue for storing nearest N targets
private int maxTargets = 2;	Maximum targets of Tower (increase with level)

2.11.9.2 Constructor

public MissileTower (double cellX, double cellY)	Create level 1 Missile Tower
--	------------------------------

2.11.9.3 Method

public void fire()	Fire Homing Missile to each target
public String toString()	Return tower name to be displayed on tower info panel
public void tryTarget(Monster m)	add targets to Priority queue
public boolean upgrade() throws FullyUpgradedException	Override: also updates this tower specific stats

2.12. Package sharedobject

2.12.1. Class SharedObject

2.12.1.1. Field

private static SharedObject instance	Declare instance variable
private ArrayList<Entity> renderables;	Array of entity for render
private ZIndexComparator comp;	Variable for compare the layer

2.12.1.2. Constructor

public SharedObject()	Empty Constructor
-----------------------	-------------------

2.12.1.3. Method

public void addRenderables(Entity ...entities)	Remove all the entities in renderables
public void removeRenderables(Entity ...entities)	Remove all the entities in renderables
public ArrayList<Entity> getRenderables()	return this.renderables
public void clear()	Clear all data in renderables
public static SharedObject getInstance()	return instance

2.13. Package ui

2.13.1. Class GameButton

2.13.1.1. Field

private Button nextButton	Next button
private Button resumeButton	Resume Button
private Button toMenuButton	Main menu Button

private Button pauseButton	Pause Button
private Button sellButton	Sell Button
private Button upgradeButton	Upgrade Button
private Button showPathButton	ShowPath Button
private ArrayList<ToggleButton> toggleButtons	Array of tower select buttons

2.13.1.2. Constructor

public GameButton()	Create and set action all button by bottom maker
---------------------	--

2.13.1.3. Method

public void addMenuButtons(Pane pane)	Add resumeButton, toMenuButton to pane
public void addTowerButtons(Pane pane)	Add toggleButtons to pane
public void addControlButton(Pane pane)	Add pauseButton, showPathButton, nextButton to pane
public void setUpgradeText(String text)	Set text of upgrade button
public void addUpgradeButton(Pane pane)	Add upgradeButton, sellButton to pane

2.13.2. Class GameScene extends Scene

2.13.2.1. Field

private GameButton buttonManager	
private StackPane root	Root pane

2.13.2.2. Constructor

public GameScene()	Add and set style status bar manu gameArea to the root pane
--------------------	---

2.13.2.3. Method

public GameButton getButtonManager()	return buttonManager
public void fadeIn()	Timeline for fade in animetion
public void fadeout()	Timeline for fade out animetion

2.13.3. Class LoadingScene extends Scene

2.13.3.1. Field

double currentProgress	Percent of processing bar
Timeline tl	Timeline for loading animetion

2.13.3.2. Constructor

public LoadingScene()	Create Scene for intro animetion
-----------------------	----------------------------------

2.13.4. Class MainMenuScene extends Scene

2.13.4.1. Field

private Timeline menuTick;	Timeline for display
private Pane root;	Root pane for add component
private MapSelectionMenu mapMenu;	Map manu there are 4 maps for select
private Timeline showMapMenu, hideMapMenu;	Timeline for animation of showing and hiding Mapmenu

2.13.4.2. Constructor

public MainMenuScene()	Create Scene for main menu
------------------------	----------------------------

2.13.4.3. Method

public void showMapSelect()	Pause menuTick and play showmapmenu
public void hideMapSelect()	Play menuTick and play hidemapmenu
public void tickle()	Set parameter for display Main menu

public void fadeIn()	Timeline for fade in animetion
public void fadeOut()	Timeline for fade out animetion

2.13.5. Class MapSelectionMenu extends Pane

2.13.5.1. Constructor

public MapSelectionMenu()	Create MapSelectionMenu pane with 4 maps ,back and next button
---------------------------	--

2.13.6. Class PauseMenu

2.13.6.1. Field

private static String[] TEXTS	Array of text for pause menu
private static boolean isShown	Status's for show Pause menu
private static GraphicsContext gc	Make text more beautiful

2.13.6.2. Method

public static void render()	Render the textbox
public static void setTargetGC(GraphicsContext gc)	Apply graphics to PauseMenu

public static void handleClick(MouseEvent e)	Add mouse event handle
--	------------------------

** there is a static initializer block to set up Pause menu

2.13.7. Class SnackBar

2.13.7.1. Field

private static boolean isShown	whether SnackBar (Toast message) is currently shown
private static String message	Message in SnackBar
private static DoubleProperty x	X position of SnackBar, to be animated by timeline
private static DoubleProperty y	Y position of SnackBar, to be animated by timeline
private static DoubleProperty opacity	Opacity of SnackBar, to be animated by timeline
private static double w h	Width and height of SnackBar
private static Timeline tl	Timeline that animates show SnackBar
private static Timeline rev	Timeline that animates hide SnackBar

2.13.7.2. Method

public static void render(GraphicsContext gc)	Render notification box
public static void play(String msg)	Play animation

** there is a static initializer block to set up notification box

2.14. Package ui.component

2.14.1. Class ButtonMaker

2.14.1.1. Constructor

public static Button make(Image imgNormal, Image imgPressed, Image imgHover, Image imgDisable, Font font, String text)	Consturctor for full feature button
public static Button make(Image imgNormal, Image imgPressed, Font font, String text)	Consturctor for button that not need to hover and disable.
public static ToggleButton makeTowerButton(double x, double y, Image imgNormal, Image imgPressed, Tower t,Font font, int value)	Consturstor for tower selector menu
public static RadioButton makeMapSelectButton(double x, double y, Image imgNormal, Image imgPressed, int value)	Consturstor map selector menu

2.14.2. Class IconText extends HBox

2.14.2.1. Field

private ImageView image	Timeline for display
private Text text	Root pane for add component

2.14.2.2. Constructor

public IconText(Image image, String text)	Constructor for build status panel
---	------------------------------------

public IconText(Image image, String text, Font font)	Constructor for build status panel and set font
--	---

2.14.2.3. Method

public void setImage(Image image)	Set panel's image
public void setText(String text)	Set panel's text

2.15. Package ui.game

2.15.1. Class GameUI

2.15.1.1. Field

private static GameUI instance	game UI in game scene
private TowerInfoPanel towerInfoPanel	Panel of current tower's info
private TowerInfoPanel upgradeInfoPanel	Panel of upgraded tower's info
private IconText levelPanel, moneyPanel, livePanel, debug;	Small Panel of user info
private cpp.pii lastPos	Pair of last position
private cpp.pii[][] path	Array contain position of path
private boolean isError	Status of error
private BFSAlgo bfs	A tool for findind path
private Tower[] previewTowers	Preview of all towers

2.15.1.2. Constructor

public GameUI()	Constructor for build gameUI such as levelPanel, moneyPanel, livePanel
-----------------	--

2.15.1.3. Method

<code>public void addinfo(Pane pane)</code>	Add levelPanel, moneyPanel, livePanel to pane
<code>public void mountPanel(Pane pane)</code>	Add towerInfoPanel, upgradeInfoPanel to pane
<code>public void render(GraphicsContext gc)</code>	Render the action in gameArea
<code>public void updateTowerInfo(Tile t, boolean showUpgrade)</code>	Update info of tower in towerinfo panel
<code>public void drawGrid(GraphicsContext gc)</code>	drawGrid for easier to place the tower
<code>public static GameUI getInstance()</code>	return instance

2.15.2. Class PathRenderer

2.15.2.1. Field

<code>private static final Bloom bloom</code>	Effect of path line
---	---------------------

2.15.2.2. Method

<code>public static void render(cpp.pii[][] path, cpp.pii start, cpp.pii end, GraphicsContext gc)</code>	Render current path on game area
--	----------------------------------

2.15.3. Class Renderer

2.15.3.1. Field

private static Renderer instance	Instance of Renderer object
private GraphicsContext gc	GraphicsContext this render will draw to
private Timeline renderLoop	Timeline that runs every 1/60 sec to render each frame
private int renderTick	Number of render time 1tick = 1/60 second

2.15.3.2. Constructor

public Renderer()	Create new renderer, will have to set GraphicsContext later
-------------------	---

2.15.3.3. Method

public void setGC(GraphicsContext gc)	Set GraphicsContext
public boolean isPlaceable(int x, int y)	Return true if can place the tower
public boolean isWalkable(int x, int y)	Return true if monster can walk
public void render()	Render game windows snackbar
public static Renderer getInstance()	return instance
public int getRenderTick()	return renderTick

2.15.4. Class TowerInfoPanel extends VBox

2.15.4.1. Field

private static final Font titleFont	load Font for title
private static final Font textFont	load Font for text
private Label superTitle, titleText, descText	All the label in info
private IconText attackInfo, cooldownInfo,rangeInfo	All the icon to make program more beautiful

2.15.4.2. Constructor

public TowerInfoPanel(String superTitleText, String towerName, String attack, String cooldown, String range, String desc)	Constructor for build Tower info panel
public TowerInfoPanel(String title)	Constructor for build Tower info panel with only title

2.15.4.3. Method

public void setTexts(String towerName, String attack, String cooldown, String range, String desc)	Set text by all the parameter
--	-------------------------------

2.16. Package util

This Package is very special.it contain special class
For using like a library

2.16.1. Class BFSAlgo

A Class that find the shortest path for monster
So we can allow user to make their own path
Because monsters can walk by the themselves

2.16.2. Class cpp

A Data structure like pair in c++ like a coordinate
Or use with comparator method

2.16.3. Class GameUtil

A Class use for compute the position of plane's shadow
so we can get plane's shadow that render naturally

2.16.4. Class Render

A Class use for render rotating of tower realtime