

poon-au Documentation

Created by
Chayut Kritveeranant 6330088721
Napat Limsumpuncharoen 6330137821

2110215 Programming Methodology
Semester 1 Year 2021 Chulalongkorn
University

Tower Defense : The Legends of Tower

Introduction

Tower Defense : The Legends of Tower is inspired by normal Tower Defense games like "Plants VS Zombies" etc. The objective of this game is to protect your own Castle that you have built by yourself from the little knights that want to claim your castle. You have to hold the knight for 10 waves to win this game

Rules

On each wave, there will have a various number of knights that come in the lane. You will start with 50 golds and you will get more gold from killing the knights. All you have to do is place the Attack Tower and kill all the knights. There are 2 types of towers "Arrow Tower" and "Rock Tower" which have different damage and build costs. So you have to choose wisely which tower you have to put because after building you have no way back. If the knight finished path alive the castle's HP will reduce

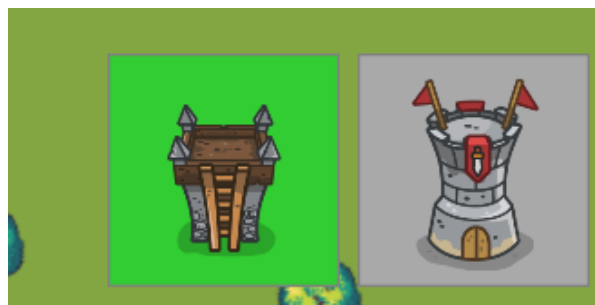
In case you have survived all the 10 waves or the HP bar has gone out. The game will end immediately but the ending will be different by the way you end the game.

Example

- When starting the game you will see your gold on the top left and the tower on the bottom right



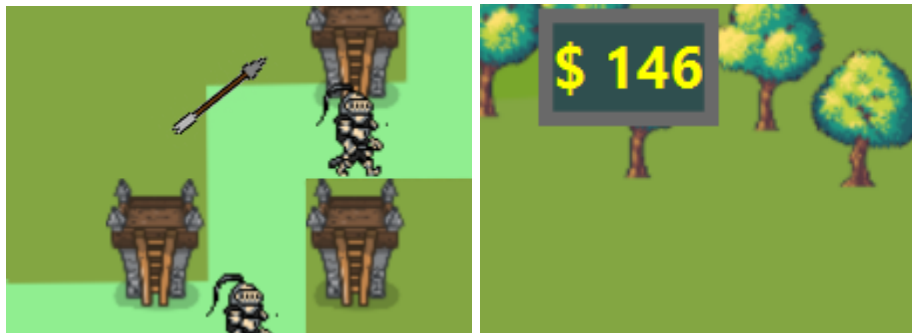
- When you click the tower button. The tower button will be highlighted and then you can click on the tile around the lane. The tower will be built and The gold will be reduced by its cost



- And if you don't have enough gold the button will be released automatically



- After building the Tower. The tower will attack knights immediately and you will get gold after killing knights.



- If knights reach the tower alive it will disappear then reduce the HP bar.



-

1. Package button
1.1 class InformationBar
1.1.1 Fields

# ProgressBar healthBar	ProgressBar that shows Player HP.
# Label moneyLabel	Label that shows Player money.
# WavePane wavePane	Pane that shows wave number and time remains on that wave.

1.1.2 Constructor

+ InformationBar()	Initialize all fields and setup size and alignment.
--------------------	---

1.1.3 Method

+ void InformationUpdate()	Update text of moneyLabel and wavePane.
----------------------------	---

1.2 class ItemButton extends Button
1.2.1 Field

- Tower tower	Tower that showed on this button.
---------------	-----------------------------------

1.2.2 Constructor

+ ItemButton(Tower tower)	- Initialize field - Set image to match the tower - Set tooltips.
---------------------------	---

1.2.3 Methods

+ void highlight()	Set background color to lime green.
+ void unhighlight()	Set background color to dark grey.
+ Tower getItem()	Return tower.
- void setTooltip()	Initialize Tooltip and set on mouse move and mouse exit.

1.3 Class ItemPane extends HBox

1.3.1 Field

- ObservableList<ItemButton> itemButtonList	Contains all ItemButtons.
--	---------------------------

1.3.2 Constructor

+ ItemPane()	- Set alignment. - Create arrowtower and rocktower button set on action to call setSelectedButton and add to pane and list.
--------------	---

1.3.3 Methods

+ void setSelectedButton(ItemButton selecteditemButton)	-Set selectedButton in SimulationManager to match the argument and highlight it.
+ void resetButtonsBackGroundColor()	-Unhighlight all itemButton.

1.4 WavePane extends VBox

1.4.1 Fields

# Label waveLabel	Show wave number.
# Text leftTime	Show left time of current wave.
# int counter	Store left time of current wave.

1.4.2 Constructor

+ WavePane()	- Initialize fields, set properties of them and add to the pane.
--------------	---

1.4.3 Method

+ void upDate()	Update property of waveLabel and
-----------------	----------------------------------

	leftTime text to current value.
--	---------------------------------

2. Package data

2.1 class SpriteAnimation

2.1.1 Fields

- final ImageView imageView	ImageView of animation sprite.
- final int count	Number of frames to loop the animation.
- final int columns	Number of all columns.
- final int offsetX	Horizon offset distance to crop frame from imageView.
- final int offsetY	Vertical offset distance to crop frame from imageView.
- final int width	Width of each cropped frame.
- final int height	height of each cropped frame.
- int lastIndex	Using to help interpolate method.

2.1.2 Constructor

+ SpriteAnimation(ImageView imageView, Duration duration, int count, int columns, int offsetX, int offsetY, int width, int height)	- Set all fields to match parameter values. - Set the interpolator to linear.
--	--

2.1.3 Method

+ void interpolate(double k)	Crop image to be each frame in animation.
------------------------------	---

3 Package drawing

3.1 Class GameScreen extends Canvas

3.1.1 Constructor

+ GameScreen(double width , double height)	- create the Canvas with height and width. - set this visible to be true
--	---

3.1.2 Methods

+ void paintComponent(GraphicsContext gc)	- draw the entity(knight) and remove it if the entity is dead
--	---

3.2 Class HomeScene extends Gridpane

3.2.1 Fields

+ Button start	- the button that is used to start the game.
+ Button exit	- the button that is used to exit the game.

3.2.2 Constructor

+ HomeScene()	<ul style="list-style-type: none"> - use super to initiate the GridPane - set alignment to TOP_CENTER - set padding to 5 - set height and width gap to 2 - Initiate Label with the text "Tower Defense". Set color to dark red and set font to "Serif Bold Italic". Set its size to 60. Then add this label to the gridpane at 0,0 - Initiate Label with the text "The Legends of Tower". Set color to dark orange. Set font to "Serif Bold Italic" and size to 30. Then add this label to gridpane at 0,1 - initiate the start button with the text "Start". Set pref size to 120,40. fill background with color gold. Then add this button to 0,150. - initiate the exit button with the text "Exit". Set pref size to 120,40. Fill background with gold color. Then add to gridpane at 1,150. - Set both button fonts to "Serif
---------------	---

	Bold Italic".
--	---------------

3.3 Class LoseScene extends Gridpane

3.3.1 Fields

+ Button exit	- the button that is used to exit the game.
---------------	---

3.3.2 Constructor

+ LoseScene()	<ul style="list-style-type: none"> - use super to initiate the GridPane - set alignment to CENTER. - set padding to 25.- initiate exit button with Text "Exit" - set the font with "Serif Bold Italic". - set pref size with 120,40. - set visible to false. - add exit button to the pane with pos 0,0.
---------------	---

3.4 Class WinScene extends Gridpane

3.4.1 Fields

+ Button exit	- the button that is used to exit the game.
---------------	---

3.4.2 Constructor

+ WinScene()	<ul style="list-style-type: none"> - use super to initiate the GridPane - set alignment to CENTER. - set padding to 25. - initiate exit button with Text "Exit" - set the font with "Serif Bold Italic". - set pref size with 120,40. - set visible to false. - add exit button to the pane with pos 10,5000.
--------------	---

3.5 Class Main extend Application

3.5.1 Field

- static final String image_path	- url of game background
- static final Image IMAGE	- image from image_path
+ static final int RESOLUTION_X	- resolution of x. Set it to 800
+ static final int RESOLUTION_Y	- resolution of y. set it to 560
+ static AudioClip buyingTowerSound	- sfx of buying tower.
+ static AudioClip backGroundSound	- background music.
+ static AudioClip menuSound	- main menu background music
+ Pane pane	- main pane

3.5.2 Method

+ void start(Stage stage)	<ul style="list-style-type: none"> - initiate root with StackPane; - initiate scene with root and resolution x,y - initiate canvas with resolution x,y. -initiate GraphicsContext with canvas - add canvas to pane. -initiate homepage url "HomeScene.png". -draw image graphiccontext with homepage url . - initiate HomeScene - add HomeScene to root. -set stage scene with scene - set title with "Tower defense : The Legends of Tower" - set resizable to false. - show stage - set menuSound volume at 60% - play menuSound - setOnMouseClicked on start button in homeScene to initiate LoseScene,WinScene -setOnMouseClicked on exit button of LoseScene,WinScene to close stage - initiate
---------------------------	--

	<p>ItemPane,InformationBar,GameLogic,GameScreen,GraphicContext,TileMap</p> <ul style="list-style-type: none"> - remove homescene from root and add winPage, losePage, and others. -add listener with root,TileMap and GameLogic -play game bgm. - initiate animationTimer override handle method with <ul style="list-style-type: none"> - If the player loses, remove gamescene and other components that are part of the game. Draw background with lose background. Set up the text to show what wave you reach and the text "You Lose".both is Serif Bold Italic.Size 40,80 by order. Set exit button of lose scene to be visible. stop game bgm - if the player wins, remove gamescene and other components. draw win background. set win scene exit button to be visible and stop game bgm -else run the game. -set on mouse click homescene to stop bgm and close stage
+ static void addAnimationToPane(Node node)	- add node to pane
+ static void removeAnimationFromPane(Node node)	- remove node from pane
+ static void drawBackground(GraphicContext gc)	- draw background with gc
+ static void removeFromPane(ImageView imageView)	- remove imageView from pane

+ static void addToPane(ImageView imageView)	- add imageView to pane
+ static void relocate(double x , double y , ImageView im)	- relocate the im with x,y
+ void addListener(Pane root, TileMap map,GameLogic logic)	-set on mouse click on root to place the tower and reduce the money
+ void stop() throws Exception	- stop bgm
+ static void main	- launch the arguments

4. Package entity

4.1 Abstract class Entity implements IRenderable

4.1.1 Fields

#int pixelX	Horizon location of entity in pixel.
# int pixelY	Vertical location of entity in pixel.
# int idxX	Horizon location of entity in index of array.
# int idxY	Vertical location of entity in index of array.
# int healthPoints	Health point of entity.
# boolean isDestroyed	Signal of entity removal.
- static int Z	Use to determine the order to draw each object.
- int speed	Speed of enemy to complete path.
- int reward	Reward to receive when killing an enemy.
- boolean isDead	Signal of enemy removal.
- boolean isPathFinished	Signal the enemy finished the path alive.

- int damage	Damage the enemy dealt to you when it finished the path.
--------------	--

4.1.2 Constructor

+ Entity(int healthPoints,int speed, int reward, int damage)	- Initialize fields. - Set location to starting point in map.
--	--

4.1.3 Methods

+ boolean isDestroyed() + boolean isPathFinished() + boolean isDead() + int getPixelX() + int getPixelY() + int getIdxX() + int getIdxY() + int getZ() + int getReward() + int getHealthPoints() + abstract ImageView getImageView()	Getter of each field.
+ void takeDamage(int damage)	-Reduce healthPoints with damage. -Change isDead and isDestroy to true and isPathFinished to false if healthPoints <= 0.
+ void walk(int[][] map)	Make the enemy's move follow the path.
- void moveX()	Move the enemy horizontally.
- void moveY(int factor)	Move the enemy vertically using factor to determine to move up or down.
+ void setHealthPoints(int healthPoints)	Setter of healthPoints.

4.2 Class KingKnight extends Entity

4.2.1 Fields

# ImageView imageView	ImageView sprite of KingKnight
# Animation animation	Walking animation of KingKnight
- static final int HEALTHPOINTS	HEALTHPOINTS = 400
- static final int SPEED	SPEED = 1
- static final int REWARD	REWARD = 200
- static final int DAMAGE	DAMAGE = 100
- static final String image_path	Image path from ClassLoader
- static final Image IMAGES	Image from image_path
- static final int COLUMNS	COLUMNS = 6
- static final int COUNT	COUNT = 6
- static final int OFFSET_X	OFFSET_X = 0
- static final int OFFSET_Y	OFFSET_Y = 0
- static final int WIDTH	WIDTH = 67
- static final int HEIGHT	HEIGHT = 49

4.2.2 Constructor

+ KingKnight()	-Initialize field of entity by call super(HEALTHPOINTS, SPEED, REWARD, DAMAGE) -Call createFirstSprite() and createSprite().
----------------	---

4.2.3 Methods

+ void draw(GraphicsContext gc)	Make a move in the user screen.
+ ImageView getImageView()	Getter of imageView
# void createFirstSprite()	Initialize imageView
# void createtSprite()	Set each animation frame and play it

	infinitely
--	------------

4.3 Class Knight extends Entity

4.3.1 Fields

# ImageView imageView	ImageView sprite of Knight
# Animation animation	Walking animation of Knight
- static final int HEALTHPOINTS	HEALTHPOINTS = 15
- static final int SPEED	SPEED = 2
- static final int REWARD	REWARD = 4
- static final int DAMAGE	DAMAGE = 10
- static final String image_path	Image path from ClassLoader
- static final Image IMAGES	Image from image_path
- static final int COLUMNS	COLUMNS = 4
- static final int COUNT	COUNT = 4
- static final int OFFSET_X	OFFSET_X = 0
- static final int OFFSET_Y	OFFSET_Y = 0
- static final int WIDTH	WIDTH = 29
- static final int HEIGHT	HEIGHT = 40

4.3.2 Constructor

+ Knight()	-Initialize field of entity by call super(HEALTHPOINTS, SPEED, REWARD, DAMAGE) -Call createFirstSprite() and createSprite().
------------	---

4.3.3 Methods

+ void draw(GraphicsContext gc)	Make a move in the user screen.
+ ImageView getImageView()	Getter of imageView

# void createFirstSprite()	Initialize imageView
# void createtSprite()	Set each animation frame and play it infinitely

4.4 Class ShovelKnight extends Entity

4.4.1 Fields

# ImageView imageView	ImageView sprite of KingKnight
# Animation animation	Walking animation of KingKnight
- static final int HEALTHPOINTS	HEALTHPOINTS = 250
- static final int SPEED	SPEED = 1
- static final int REWARD	REWARD = 100
- static final int DAMAGE	DAMAGE = 30
- static final String image_path	Image path from ClassLoader
- static final Image IMAGES	Image from image_path
- static final int COLUMNS	COLUMNS = 6
- static final int COUNT	COUNT = 6
- static final int OFFSET_X	OFFSET_X = 0
- static final int OFFSET_Y	OFFSET_Y = 0
- static final int WIDTH	WIDTH = 42
- static final int HEIGHT	HEIGHT = 34

4.4.2 Constructor

+ ShovelKnight()	-Initialize field of entity by call super(HEALTHPOINTS, SPEED, REWARD, DAMAGE) -Call createFirstSprite() and createSprite().
------------------	---

4.4.3 Methods

+ void draw(GraphicsContext gc)	Make a move in the user screen.
+ ImageView getImageView()	Getter of imageView
# void createFirstSprite()	Initialize imageView
# void createtSprite()	Set each animation frame and play it infinitely

5. Package logic

5.1 Class GameLogic

5.1.1 Field

+ static List<IRenderable> gameObjectContainer	Make changes in this list to avoid errors when rendering.
--	---

5.1.2 Methods

+ void addNewObject(IRenderable iRenderable)	Add parameter to gameObjectContainer and irenders list from RenderableHolder class.
+ void logicUpdate()	-Remove all IRenderable that is destroyed from gameObjectContainer -Consider giving a reward when it is killed by a player.
+ void updateLocation(int[][] map)	-Call walk method from all entities that are not destroyed in gameObjectContainer.

5.2 Class SimulationManager

5.2.1 Fields

+ static ItemButton selectedItemButton	Store the pressed ItemButton.
+ static ProgressBar healthBar	Health bar implemented by Progressbar.
- static final double MAXHP	MAXHP = 100
- static double health	Current HP starts with 100.
- static int money	Current money starts with 50.
- static int wave	Current wave starts with 0.
- static int damageOnThisFrame	Store damage that the player receives each frame.

- static boolean isGameOver	Be true if the player loses the game.
- static boolean isGameEnd	Be true if the player wins or loses the game.
- static boolean isWin	Be true if the player wins the game used in Main.
- static boolean isLose	Be true if the player loses the game used in Main.
- static boolean hasItem	Be true if selectedItemButton is not null.
- static long lastTimeTriggered	Latest time the animation timer was triggered.
+ static int timer	Down counter of current wave time.
+ static int waveTime	Time used in that wave.
# static AudioClip arrowSound	Arrow shooting SFX
# static AudioClip lavaRockSound	Lava rock shooting SFX
# static AudioClip gameOverSound	Song that is played when the player loses the game.
# static AudioClip winningSound	Song that is played when the player wins the game.

5.2.2 Methods

static	Set the volume of the arrow and lavaRock sound.
+ static void setSelectedItemButton(ItemButton selectedItemButton)	-Set selectedItemButton -hasItem = true
+ static void receiveReward(int reward)	money += reward
+ static void reduceMoneyBuyTower()	-Reduce player's money equal to item cost. -Remove the highlight of selectedButton if the player can't

	afford that item anymore set selectedItemButton to null.
+ static void attacked(int damage)	damageOnThisFrame += damage
+ static void updateHealth()	-Call reduceHealth(damageOnThisFrame). -Update healthBar to match current HP. -Set damageOnThisFrame to 0.
- static void reduceHealth(int damage)	-Reduce health by damage -If health < 0, set it to 0. -isGameOver = true if health is 0.
+ static void createProjectiles()	Control the showing of canon in UI.
+ static void waveManage(GameLogic logic, long now)	-Change wave number considering time from animation timer. -Check the lose and win condition.
+ static void generateMonster(GameLogic logic)	Generate enemies relate to wave number.
- static boolean isGameClear()	Check if the number of existing enemies is 0.
+ static ItemButton getSelectedItemButton() + static int getWave() + static int getMoney() + static boolean hasItem() + static boolean isGameOver() + static boolean isGameEnd() + static boolean isWin() + static boolean isLose()	Getter method

5.3 Class TileMap

5.3.1 Fields

- int[][] map	Two dimensional array represents the position of each element in the game map.
---------------	--

+ static final int TILE_LENGTH_X	Length of tiles TILE_LENGTH_X = 40
+ static final int TILE_LENGTH_Y	Length of tiles TILE_LENGTH_Y = 40
- final int NUMER_OF_TILE_X	Number of tiles of each row of map.
- final int NUMER_OF_TILE_Y	Number of tiles of each column of map.
+ static final int BEGIN_X_IDX	Spawning location of enemies in x dimension .
+ static final int BEGIN_Y_IDX	Spawning location of enemies in y dimension.
+ static final int SIMPLE_GRASS + static final int TURRET_PLACE + static final int UP_VERTICAL_PATH + static final int DOWN_VERTICAL_PATH + static final int HORIZON_PATH + static final int EAST_TO_NORTH_PATH + static final int NORTH_TO_EAST_PATH + static final int SOUTH_TO_EAST_PATH + static final int EAST_TO_SOUTH_PATH + static final int TOWER_TILE + static final int GOAL_TILE	Constants containing 0 - 9 respectively used when checking type of tile in map make code easier to understand.

5.3.2 Constructor

+ TileMap(int mapWidth , int mapHeight , GraphicsContext gc)	-NUMER_OF_TILE_X = (int) (mapWidth / TILE_LENGTH_X); -NUMER_OF_TILE_Y = (int) (mapHeight / TILE_LENGTH_Y); -Initialize map by calling generateMapArray(). -Call pathPaint(gc);
--	---

5.3.3 Methods

+ void pathPaint(GraphicsContext gc)	Paint the enemies path considering the value contained in each position of the map array.
- int[][] generateMapArray()	-Initialize map and return it.
+ void setMapNode(int xCord , int yCord , int updatedValue)	-Set the value of the given index map to updatedValue.
+ int[][] getMap()	Return map

6. Package sharedObject

6.1 Interface IRenderable

6.1.1 Methods

+ void draw(GraphicsContext gc)	Show itself on UI.
+ boolean isDestroyed()	Return true if that object should not appear on UI anymore.
+ int getZ()	Return priority to be drawn on UI.

6.2 Class RenderableHolder

6.2.1 Fields

- static final RenderableHolder instance	Itself in static terms to be called by other classes.
- List<IRenderable> irenders	List contains game objects that can be drawn.
- Comparator<IRenderable> comparator	Comparator use to sort the list.

6.2.2 Constructor

+ RenderableHolder()	Initialize irenders and comparator.
----------------------	-------------------------------------

6.2.3 Methods

+ static RenderableHolder getInstance()	Return instance
+ void add(IRenderable entity)	Add entity to irenders and sort.

+ List<IRenderable> getEntities()	Return renders
-----------------------------------	----------------

7 Package tower

7.1 Class Arrow extends Projectile

7.1.1 Field

- Entity target	- the target to attack
- final int startX	- location of tower x
- final int startY	- location of tower y
- int endX	- location of target x
- int endY	- location of target y
- static final String arrow_image_path	- url of arrow image.
- Image newArrow	- image of arrow

7.1.2 Constructor

+ Arrow(Entity target , int towerX , int towerY)	- initiate by super(target, towerX, towerY, 50) - set every field to it's value - initiate deltaX, deltaY by start - end - initiate angle by deltaX,deltaY - set fill with newArrow - rotate to the right angle
---	--

7.1.3 Method

+ getter()	- getter of every Field
------------	-------------------------

7.2 Class LavaRock extends Projectile

7.2.1 Field

- Entity target	- the target to attack
- final int startX	- location of tower x
- final int startY	- location of tower y

- static final String lava_image_path	- url of lava rock image.
- Image newLava	- image of lava rock

7.2.2 Constructor

+ LavaRock(Entity target , int towerX , int towerY)	- initiate by super(target , towerX ,towerY , 9) - set every field to it's value - set fill with newLava
---	--

7.2.3 Method

+ getter()	- getter of every Field
------------	-------------------------

7.3 Class Projectile extends Circle

7.3.1 Field

- Entity target	- the target to attack
- final int startX	- location of tower x
- final int startY	- location of tower y

7.3.2 Constructor

+ Projectile(Entity target , int towerX , int towerY,int size)	- initiate by super(towerX,towerY,size) - set every field to it's value
--	--

7.3.3 Method

+ getter()	- getter of every Field
------------	-------------------------

7.4 abstract Class Tower implements IRenderable

7.4.1 Field

# static int Z	- field that use to set dimension
# int pixelX	- position x in pixel scale
# int pixelY	- position y in pixel scale

# int idxX	- position x in index scale
# int idxY	- position y in index scale
# final int BUILDTIME	- time to build = 1
# int attackDamage	- damage of tower
# boolean destroyed	- check if this tower is destroyed
# int attackRange	- Range of tower
# int sellCost	- Cost of tower
# static ArrayList<Projectile> projectileList	- arraylist of Projectile
# TowerAttackService towerAttacker	- field that use to shoot Projectiles

7.4.2 Constructor

+ Tower(int x, int y)	-initiate all Field -idx and idy is pixelX/lengthX and pixelY/lengthY (length get from TileMap) -initiate towerAttacker and use towerAttacker.pollTower(BUILDTIME)
-----------------------	--

7.4.3 Methods

+ abstract void createProjectile(Entity target)	- this method is abstract
+ getter/setter	- getter/setter of all Field
+ String getToolText()	- return name, damage and cost
+ boolean isDestroyed()	- return destroyed
+ abstract String getUrl()/getName()	- this is abstract methods

7.5 ArrowTower extends Tower

7.5.1 Field

- static String url	- url of Arrow tower image.
- static String NAME	- name of this tower
- static String urlForDraw	- url of this tower. use to draw tower

- static final String image_path	- get url from urlForDraw
- static Image image	- image of this tower

7.5.2 Constructor

+ ArrowTower()	- set attackDamage equal to 5 - set attackRange to 0 - set sellCost to 20
+ ArrowTower(int x, int y)	- use super(x,y) - set attackDamage equal to 5 - set attackRange to 0 - set sellCost to 20

7.5.3 Methods

+ void draw(GraphicsContext gc)	- use gc to draw the tower
+ createProjectile(Entity target)	- initiate Arrow(target,pixelX,pixelY) and add to projectileList
+ String getUrl	- return url
+ String getName	- return NAME

7.6 RockTower extends Tower

7.6.1 Field

- static String url	- url of Rock tower image.
- static String NAME	- name of this tower
- static String urlForDraw	- url of this tower. use to draw tower
- static final String image_path	- get url from urlForDraw
- static Image image	- image of this tower

7.6.2 Constructor

+ ArrowTower()	- set attackDamage equal to 25 - set attackRange to 0 - set sellCost to 100
----------------	---

+ ArrowTower(int x, int y)	<ul style="list-style-type: none"> - use super(x,y) - set attackDamage equal to 25 - set attackRange to 0 - set sellCost to 100
----------------------------	---

7.6.3 Methods

+ void draw(GraphicsContext gc)	- use gc to draw the tower
+ createProjectile(Entity target)	- initiate LavaRock(target,pixelX,pixelY) and add to projectileList
+ String getUrl	- return url
+ String getName	- return NAME

7.7 Class TowerAttackService

7.7.1 Field

- final ScheduledExecutorService scheduler	- Scheduler for attacking. use Executors.newScheduledThreadPool(1)
- ScheduledFuture<?> pollHandler	- use to run the Schedule
- Tower tower	- tower of this schedule
- Entity target	- target in schedule

7.7.2 Constructor

+ TowerAttackService(Tower tower)	- set tower to the given object
-----------------------------------	---------------------------------

7.7.3 Methods

+ void pollTower(int delay)	<ul style="list-style-type: none"> - initiate Runnable - override run() in Runnable with - initiate min,max range in x,y by using tower position and tower attack range - then check if there are any Entity that are in range of the tower. If there are Entity that in range create projectile with tower.createProjectile(target) then
-----------------------------	---

	decrease the HP of target. After that use pollhandler to set scheduler's delay 1000 ms
+ void cancel()	- set the pollHandler.cancel to true.