Super Killing Wars

Name :

1. Pattara Teerapong	5831051221
2. Rachata Kampitak	5831059321

Introduction

In Super Killing Wars you’ll have to survive in an unknown world full of aliens who try to kill you. You’ll be gathering natural resources and building base to generate more resources while keeping the aliens off. The primary objective is to build a rocket to escape from this world and get back to your lovely home.

Objective of game

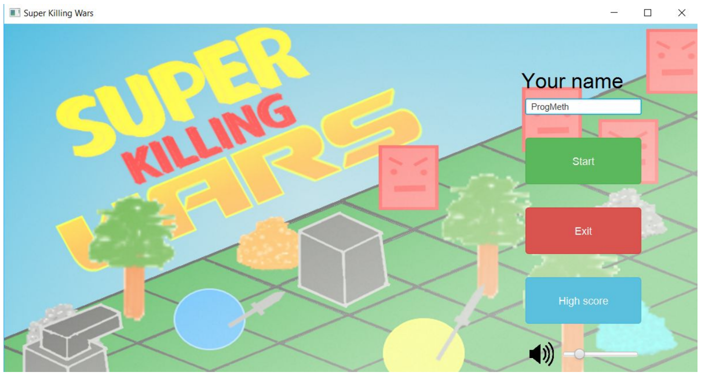
- Gather natural resources
- Build your base
- Survive from endless waves of enemies
- Build a rocket to end the game

Control

- WASD : Move the player
- Mouse left down : Shoot
- Mouse right down : Harvest item
- Mouse left click : Buying items
- Mouse right click : Exit buy mode
- SPACE : Pause game

How to play

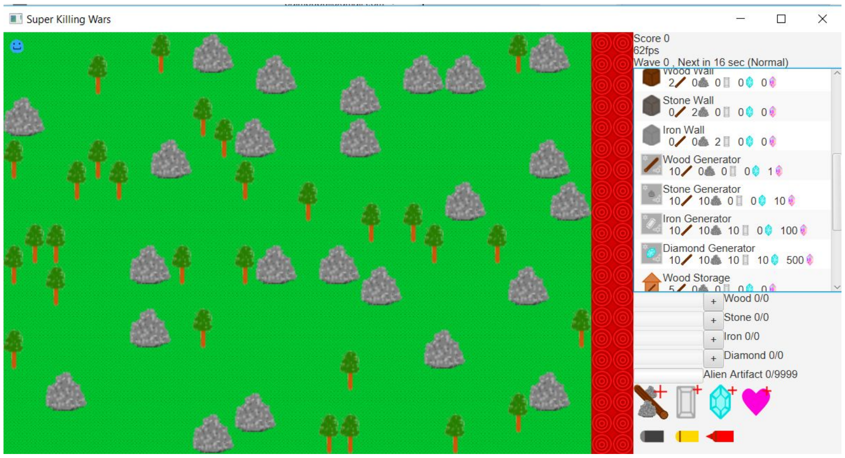
Basic



ProgMeth

Main menu

- Enter your name to text field.
- Slide bar to adjust volume
- Press “Start” button to start the game
- Press “Exit” button to close the game
- Press “High score” to show highscore



Score 0
62fps
Wave 0 , Next in 16 sec (Normal)

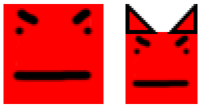
Wood Wall	2	0	0	0	0
Stone Wall	0	2	0	0	0
Iron Wall	0	0	2	0	0
Wood Generator	10	0	0	0	1
Stone Generator	10	10	0	0	10
Iron Generator	10	10	10	0	100
Diamond Generator	10	10	10	10	500
Wood Storage	5	0	0	0	0

+ Wood 0/0
+ Stone 0/0
+ Iron 0/0
+ Diamond 0/0
Alien Artifact 0/9999

Game Screen

Player

- Control player with WASD
- Left click to shoot at your cursor

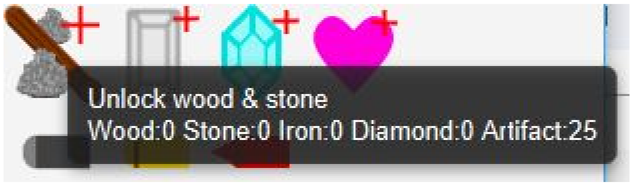


Enemy (Left : Basic, Right : Boss)

- Every 20 seconds, a wave of enemy will spawn. (Enemy will get stronger every wave)
- Enemy will follow and attack the player (They will even attack object on screen)
- If you get killed, the game is over
- You can shoot enemy to hurt them.
- Kill them to get some alien artifact.

Item building

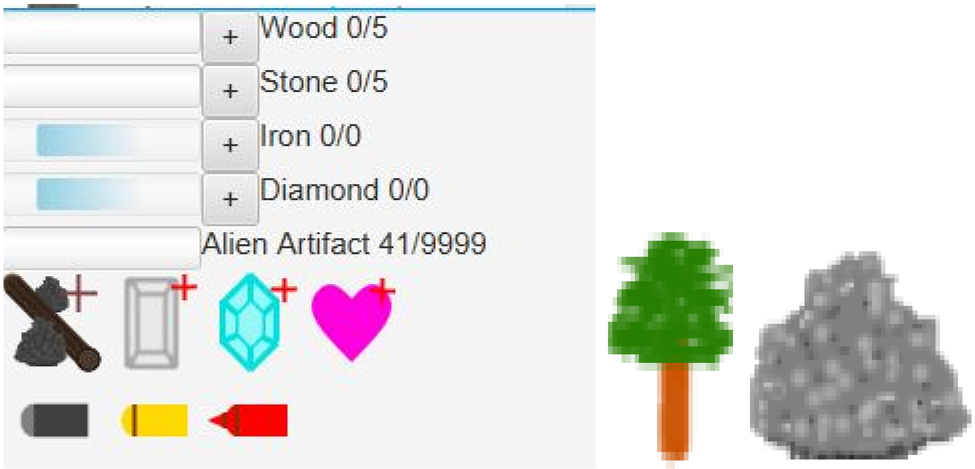
- You need to unlock item building from research (because at first your resource capacity is 0 so you can't buy anything)



Unlock wood & stone
Wood:0 Stone:0 Iron:0 Diamond:0 Artifact:25

"Unlock wood & stone" research (the first one)

- You need to collect 25 alien artifact to buy this research
- After unlock wood & stone, both your wood & stone capacity will be 5



Wood 0/5

Stone 0/5

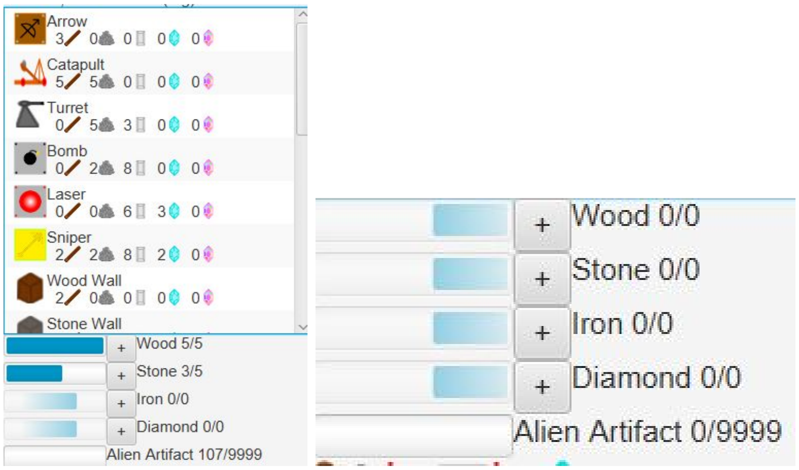
Iron 0/0

Diamond 0/0

Alien Artifact 41/9999

Now you can harvest resource (Wood & stone)

- There are stone and tree on the map. You can hold right mouse over them to harvest them.
- After you have enough resources, you can build your first item.
- You can also convert alien artifact to other resources.
  - 1 artifact -> 1 wood
  - 10 artifact -> 1 stone
  - 100 artifact -> 1 iron
  - 300 artifact -> 1 diamond



Arrow	3	0	0	0	0
Catapult	5	5	0	0	0
Turret	0	5	3	0	0
Bomb	0	2	8	0	0
Laser	0	0	6	3	0
Sniper	2	2	8	2	0
Wood Wall	2	0	0	0	0
Stone Wall					

Wood 5/5

Stone 3/5

Iron 0/0

Diamond 0/0

Alien Artifact 107/9999

Wood 0/0

Stone 0/0

Iron 0/0

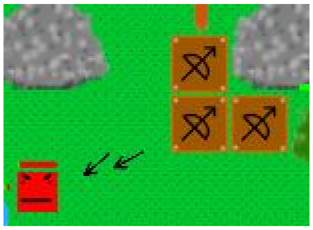
Diamond 0/0

Alien Artifact 0/9999

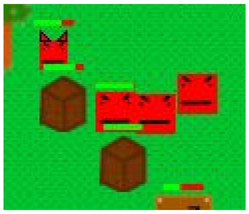
Get resource by harvesting or converting alien artifact (plus button)

There are 4 type of item

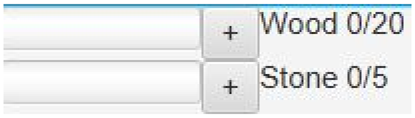
1. Tower
   Tower will shoot nearby enemy. Different tower shoot different projectiles.



2. Wall
   Wall is perfect for blocking enemy.

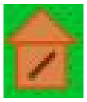


3. Storage
   Your resource capacity is very limited. Storage can extend your resource capacity

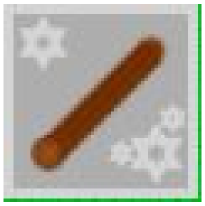


+ Wood 0/20

+ Stone 0/5



4. Generator
   Generate resource over the time



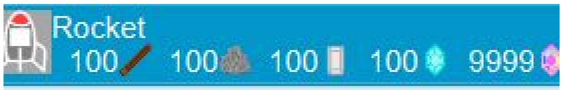
Resource generating

Generator cost 10 resource unit. But after research has unlocked you have only 5 capacity. So you need to build a storage first in order to get enough resource to build a generator.

To infinity and beyond

Now you can build and unlock everything. Iron can be researched from wood, stone and alien artifact. Diamond can be researched from iron. You can build many type of towers and walls to keep your base survived. But enemy will get stronger and stronger. So you can't stay here forever.

After you have enough resource, you can build rocket silo to escape from this world. Beware that when you try to escape, you'll face the worst nightmare you've ever seen.



Your way home and your worst nightmare

Class detail

1. application

1.1 Main

Main class of the program

Fields

+ int SCREEN_WIDTH = 1200	Screen width (1200px)
+ int SCREEN_HEIGHT = 600	Screen height (600px)
+ Scene scene	Current scene (Game or MainMenu)

Methods

+ void main(String[] args)	Launch the JavaFX application
+ void start(Stage primaryStage) throws Exception	Call when application starts
+ void stop()	Call when application stops
+ void changeSceneToGame()	Change current scene to game
+ void changeSceneToMain()	Change current scene to main

2. exception

2.1 HighscoreException

This exception will be thrown when highscore can't be fetched.

Fields

- long serialVersionUID	6882787462646581816L

Methods

+ HighscoreException()	Initialize error message

2.2 InvalidNameException

This exception will be thrown when player's name is invalid (contains whitespace).

Fields

- long serialVersionUID	4658717663341535289L

Methods

+ InvalidNameException()	Initialize error message

3. Logic

3.1 BuyManager

Manage items buying.

Fields

+ BuyManager instance	Singleton object
+ boolean isBuyMode	Is buying
+ Image currentObjectImage	Buying object image
+ Class currentObjectClass	Buying object class

Methods

+ boolean canBuy()	Check if resources are enough for buying object

3.2 CollisionUtility

Methods for calculating collisions.

Methods

+ boolean isCollide(ICollidable o1, ICollidable o2)	Check if two ICollidable objects are intersect
+ boolean isBlock(ICollidable object)	Check if ICollidable object collides with other blocking objects
+ void checkCollision()	Update object when collision happen

+ double findDistance(ICollidable ic1, ICollidable ic2)	Find distance between two objects

3.3 EnemyManager

Managing enemy spawning (spawn in waves).

Fields

+ EnemyManager	Singleton object
- int timer	Ticks since last wave
- int wave	Current wave number
- int WAVE_DELAY	Delay between each wave (1200 ticks = 20 seconds)
- int ROCKET_WAVE_DELAY	Wave Delay when rocket is present (120 ticks = 2 seconds)

Methods

+ int getWaveNumber()	Getter for wave
+ void update()	Update timer and spawn new wave if delay is reached
+ int getRemainingTime()	Get remaining time until next wave (in seconds)
+ String getNextWaveName()	Get next wave name (“Rocket”, ”Big”, ”Boss” or “Normal”)
- boolean isBigWave(int wave)	Big wave on wave number 4, 9, 14, ...
- boolean isBossWave(int wave)	Boss wave on wave number 5, 10, 15, ...
- void spawn()	Spawn new enemy wave
- void spawnEnemiesOnSpawner(int basicCount, int bossCount, int level)	Spawn enemies on spawner tile

3.4 GameManager

Managing game states and update every entities.

Fields

+ GameManager instance	Singleton object
+ Random globalRNG	Global random number generator for this game
- ThreadGameManager thread	Thread for update game state
- int score	Game score
- int fps	Frame per second
- int rocketCount	Number of rocket
- boolean isRocketLaunched	Is rocket launched (the game will ended if true)
- boolean isGamePause	Is game pause
- Player player	Player

Methods

+ GameManager()	Initialize singleton and add player to the game

+ void addEntity(IRenderable entity)	Add entity to RenderableHolder
+ void update()	Update game state - Update buying overlay - Update entities - Check collisions and remove destroyed entities - Check ending condition
- boolean isPause()	Update pause status and return it
- boolean isGameEnd()	Game when when - Player is destroyed or - Rocket is launched
- void onGameEnded()	Call when game ended - Show ending dialog - Switch to main menu
- void updateEntity()	Call update on every entity
- void updateOverlay()	Update state when in buying mode
- void removeDestroyEntity()	Remove destroyed entities
+ void increaseScore(int amount)	Increase score by amount
+ void startUpdateThread()	Create and start update thread
+ void stopUpdateThread()	Stop update thread (by interrupt)
+ Getter for rocketCount, isGamePause, player, score, fps	
+ Setter for rocketLaunched, rocketCount, fps	

3.5 HighscoreManager

Managing HTTP request to get highscore from the server.

Methods

- String req(String path) throws HighscoreException	Make request to server and return response (throw HighscoreException when http request failed)
+ String getScore() throws HighscoreException	Wrapper function for req to fetch score from server
+ void postScore(String name,int score) throws HighscoreException	Wrapper function for req to post name and score to server

3.6 IBlockable

Interface for blockable object (object that can't be collide with other blockable object)

Methods

+ void undoMove()	Call when objects collided to undo move to valid position

3.7 ICollidable

Interface for collidable object (Can check collision with other Collidable Object)

Methods

+ void onCollision(ICollidable entity)	Call when object collide with other collidable object
+ double getX()	Get object’s x position
+ double getY()	Get object’s y position
+ double getWidth()	Get object’s width
+ double getHeight()	Get object’s height

3.8 IMovable

Object that can be moved (ex. MovingEntity)

Methods

+ void move()	Move the object

3.9 InputUtility

Utility for managing input events.

Fields

+ InputUtility instance	Singleton object
- ArrayList<KeyCode> keyPressed	Contains currently pressed key code
- ArrayList<KeyCode> keyTriggered	Contains currently triggered key code
- double mouseX,mouseY	Current mouse’s coordinate
- boolean isMouseLeftTriggered	Is mouse left triggered
- boolean isMouseLeftDown	Is mouse left down
- boolean isMouseRightTriggered	Is mouse right triggered
- boolean isMouseRightDown	Is mouse right down
- boolean isMouseOnScreen	Is mouse on screen

Methods

+ InputUtility()	Initialize singleton and variables.
+ void setEventHandler(Scene scene)	Add event handlers to a scene. (keyPressed, keyReleased, mousePressed, mouseReleased, mouseEntered, mouseExited, mouseDragged, mouseMoved)
+ void reset()	Reset triggered status
+ boolean isKeyDown(KeyCode a)	Check if key a is currently down (check in keyPressed)
+ boolean isKeyTriggered(KeyCode a)	Check if key a is currently triggered (check in keyTriggered)
+ Getter for isMouseLeftTriggered, isMouseRightTriggered,	

isMouseLeftDown, isMouseRightDown, isMouseX,isMouseY, isMouseOnScreen	

3.10 IRenderable

Interface for renderable objects (can be drawn on canvas)

Methods

+ int getZ()	Get object's Z
+ void draw(GraphicsContext gc)	Draw object on canvas
+ boolean isDestroy()	Is object destroyed

3.11 ResourceManager

Managing resources (wood / stone / iron / diamond / alien artifact)

Fields

+ ResourceManager instance	Singleton object
+ int WOOD = 0	Index for wood in resources array
+ int STONE = 1	Index for stone in resources array
+ int IRON = 2	Index for iron in resources array
+ int DIAMOND = 3	Index for diamond in resources array
+ int ARTIFACT	Index for alien artifact in resources array
+ String[] NAME	Name for each resources {"Wood", "Stone", "Iron", "Diamond", "Alien Artifact"}
+ int[] EXCHANGE_RATE	Exchange rate from x artifacts to 1 of target resources {1,10,100,250}
- int[] capacity	Capacity for each resource (initial : {0,0,0,0,9999})
- int[] resource	Resources count for each resource (initial : {0,0,0,0,0})

Methods

+ Getter for capacity, resource	
+ void addResource(int index,int amount)	Add resource by some amount
+ void addCapacity(int index,int amount)	Add capacity to a resource by some amount
+ void normalize(int index)	Check and correct resource overflow

3.12 SoundManager

Managing sounds (Background music and sound effects)

Fields

- AudioClip bgm, gunshot	AudioClip for background music and gunshot

Methods

- AudioClip getRes(String path)	Load AudioClip for given path using Classloader
static	Load sound and set default volume to 0.2
+ void start()	Play background music
+ void stop()	Stop background music
+ Getter for bgm, gunshot	
+ double getVolume()	Get volume from bgm
+ setVolume(double value)	Set volume bgm and gunshot

3.13 TileManager

Managing tiles

Fields

+ TileManager instance	Singleton object
+ int TILE_COUNT_X = 30	Number of column
+ int TILE_COUNT_Y = 20	Number of row
+ double TILE_SIZE = 30	Tile size
+ List<Tile> tileList	List of tile
+ Tile[][] tileArray	Direct access to tile at given index

Methods

+ TileManager()	Initialize tileList and tileArray
- ICollidable createCollidableFromTile(Tile tile, int sizeX, int sizeY)	Create temporary ICollidable object from tile and size to check if the object is placeable
+ boolean canPlace(Tile tile, int sizeX, int sizeY)	Check if object with size of sizeX, sizeY can be placed on top of a tile
+ void generateMap(int seed)	Generate tiles using seed to feed the RNG. - Create tile ground / spawner / void - Create tile object void at outside of the screen - Randomly create tree / stone
+ int getMouseTileX()	Get tile's X from tile that mouse is over
+ int getMouseTileY()	Get tile's Y from tile that mouse is over
+ boolean isOutOfBound(int x,int y)	Check if x, y is out of bound

4. model

4.1 BlockingEntity

MovableEntity that is also blockable

Fields

# double lastX, lastY	Last valid x,y

Methods

+ BlockingEntity(double x,double y, double width, double height, double speed, int hp)	Call super's constructor with (x, y, width, height, speed, hp)
+ void move()	Move with blocking behavior

4.2 Entity

All object in game

Fields

# double x,y	Object's x,y
# double width,height	Object's width,height
# boolean isDestroyed	Is object destoryed
# int hp	Object's health point
# int maxHp	Object's max health point
- double HEALTHBAR_WIDTH = 20	Healthbar's width (20px)
- double HEALTHBAR_HEIGHT = 3	Healthbar's height (3px)

Methods

+ Entity(double x,double y,double width,double height, int hp)	Initialize fields with parameters, also initialize isDestroyed to false and maxHp to hp
+ void destroy()	Destroy this entity (set isDestroyed to true and call onDestroy())
# void onDestroy()	Call when destroyed (default is doing nothing)
+ void update()	Call onDestroy() when object's hp reaches 0
+ void onCollision(ICollidable collider)	Call when this entity collides with another object (default is doing nothing)
+ Getter for isDestroy, hp, x, y, width,height	
+ Setter for hp, x, y, width, height	
+ void drawHealthbar(GraphicsContext gc)	Draw healthbar over this entity

# void draw(GraphicsContext gc,Image img)	default draw function for all entity
+ void reduceHP(int damage)	Reduce this entity hp by damage
+ double getCenterX()	Get X of center of object
+ double getCenterY()	Get Y of center of object

4.3 MovingEntity

Entity that can move

Fields

# double speed	Object's speed
# double velX,velY	Object's velocity

Methods

+ MovingEntity(double x, double y, double width, double height, double speed, int hp)	Call super's constructor with (x, y, width, height, hp) and initialize object's speed. Also initialize velX, velY to 0
+ void update()	Call super.update() and call move() if this object is not destroyed
# void move()	Move (change x, y according to velocity and speed)
+ Getter for speed, velX, velY	
+ Settter for speed, velX, velY	

4.4 Player

Player entity

Fields

- double SPEED = 5	Player's speed
- double WIDTH = 20, HEIGHT = 20	Player's width,height
- int START_HP = 750	Player's start health point
- int healthRegenerationTimer = 0	Health regeneration timer
- int HEALTH_REGEN_DELAY = 10	Delay between regeneration
- int healthRegenerationRate = 0	Amount of regenerate hp
- int shootingTimer = 0	Shooting timer
- int SHOOTING_DELAY = 15	Delay before player can shoot again
- int HARVEST_POWER = 1	Player's harvest power

Methods

+ Player(double x,double y)	Call super's constructor with (x, y, width, height, speed, startHp)
+ void move()	Call super.move() and set velocity to 0 after moving
+ void undoMove()	Return to lastX, lastY position when blocked

+ int getZ()	Return Player’s Z (0)
+ void draw(GraphicsContext gc)	Draw player’s image
- updateVelocity()	Update velocity according to input
- updateShoot()	Shoot if mouse left is down
- updateHealthRegeneration()	Regenerate health
- updateHarvest()	Harvest object under mouse if mouse right is down
+ void update()	Update everything and call super.update()
+ Getter for speed, velX, velY	
+ Settter for speed, velX, velY, healthRegenerationRate	

4.5 RenderableHolder

Holder for IRenderable objects

Fields

+ RenderableHolder instance	Singleton object
- List<IRenderable> entities	List that hold Renderable
- Comparator<IRenderable> comparator	Comparator for IRenderable based on Z value
+ *_img	Image of everything including : - Tile : ground, spawner - TileObject : tree, stone, rocket - Resources images - Projectile : arrow, laser, rock, bullet, bomb - Enemy : basic, boss - Entity : player - Tower : arrow, laser, catapult, turret, sniper, bomb - Wall : wood, stone, iron - Generator : wood, stone, iron, diamond - Storage : wood, stone, iron, diamond - Research : building, iron, diamond, gun, supergun, regen, smallgun - UI : Volume

Methods

+ RenderableHolder()	Initialize variables
- loadResource()	Load every images
- getRes(String path)	Load image from given path using ClassLoader
+ void add(IRenderable entity)	Add entity to entities list
+ void remove(int index)	Remove entity from index
+ Getter for entities	

4.6 ScoreRecord

A score record for high score.

Fields

- int score	Score of this record
- String name	Player’s name

Methods

+ ScoreRecord(int score, String name)	Initialize variables to parameters.
+ int compareTo(ScoreRecord o)	Compare function (higher score come first)
+ String toString()	Convert record to string in format “[name] : [score]”

4.7 Tile

A tile (background)

Fields

- TileObject tileObject	Tile Object that place over this tile
# double x, y	X, Y position of this tile
# int tileX, tileY	Column and row of this tile

Methods

+ Tile(int tileX, int tileY)	Initialize variables to parameters and calculate X, Y from tileX, tileY
+ int getZ()	Return tile’s Z (-10 because it’s drawn on background)
+ boolean isDestroy()	Return false (tile is never destroyed)
+ void draw(Graphics gc, Image img)	Draw this tile image (using x, y and TileManager.tileSize)
+ Getter for tileX, tileY, x, y, tileObject	
+ Setter for x, y, tileObject	Convert record to string in format “[name] : [score]”

4.8 TileGround

Tile ground (grass)

Methods

+ TileGround(int tileX, int tileY)	Call super(tileX, tileY)
+ void draw(GraphicsContext gc)	Call super.draw with tile ground image

4.9 TileSpawner

Tile spawner (enemy spawn on this tile)

Methods

+ TileSpawner(int tileX, int tileY)	Call super(tileX, tileY)
+ void draw(GraphicsContext gc)	Call super.draw with tile spawner image

4.10 TileVoid

Tile void (outside of game screen for placing tile object void)

Methods

+ TileVoid(int tileX, int tileY)	Call super(tileX, tileY)
+ void draw(GraphicsContext gc)	Do nothing (don't need to draw anything)

5. model.enemy

5.1 Enemy

Base class for enemy

Fields

- double WIDTH = 20	Default width of enemy (20px)
- double HEIGHT = 20	Default height of enemy (20px)
- int damage	Damage of this object
- int attackTimer	Attack timer for this object
- int reward	Reward (Alien artifact) received when killed
- int ATTACK_DELAY = 20	Attack delay for enemy (20 ticks)
- double ATTACK_RANGE = 5	Attack range for enemy (5px around it)
- Entity target	Target of this enemy (Enemy will move to it)

Methods

+ Enemy(double x, double y, double speed, int startHp, int damage, int reward)	Call super constructor with (x, y, WIDTH, HEIGHT, speed, startHp) and initialize damage, reward
- Entity findTarget(Class targetClass)	Find closest Entity that is class/subclass of targetClass
+ void update()	- Call super.update() - Assign new target if the old one is destroyed - Count attack timer and attack if delay is reached
- void attack()	Attack entities around this enemy that are in range
+ void undoMove()	Return to lastX, lastY position when blocked
+ int getZ()	Return enemy's Z (15)
+ void onDestroy()	Call super.onDestroy() and add alien artifact to resources
+ Setter for target	

5.2 EnemyBasic

Basic enemy (weak)

Fields

- double SPEED = 2	Default speed of basic enemy

Methods

+ EnemyBasic(double x, double y, int level)	Call super constructor with (x, y, SPEED, HP, DAMAGE, REWARD) where HP, DAMAGE, REWARD scales with level - HP = 20 + 20 * level - DAMAGE = Floor(3 + 0.7 * level) - REWARD = 3 * level
- void draw(GraphicsContext gc)	Call super.draw with enemy basic image

5.3 EnemyBoss

Enemy boss

Fields

- double SPEED = 3	Default speed of basic enemy

Methods

+ EnemyBoss(double x, double y, int level)	Call super constructor with (x, y, SPEED, HP, DAMAGE, REWARD) where HP, DAMAGE, REWARD scales with level - HP = 50 + 40 * level - DAMAGE = Floor(4 + 0.8 * level) - REWARD = 10 * level
- void draw(GraphicsContext gc)	Call super.draw with enemy boss image

6. model.projectile

6.1 Projectile

Base class for projectile

Fields

# int damage	Damage of this projectile
- double angle	Angle of this projectile (direction to enemy)
- double originalWidth,originalHeight	Original width and height of this projectile
- Entity target	Target of this projectile

Methods

+ Projectile(double x, double y, double width, double height, double speed, int damage, double targetX, double targetY)	Call super constructor with (X, Y, width, height, speed, 100) where X = x - width/2, Y = y - height /2 (because x,y = center) and initialize damage, originalWidth, originalHeight and calculate angle to (targetX, targetY)
+ Projectile(double x, double y, double	The same as above constructor and initialize target

width, double height, double speed, int damage, Entity target)	
- void setTarget(double tx, double ty)	Calculate angle and velocity to the target and calculate width/height to reflect the hitbox
+ void update()	Call super.update() and if this projectile has a target then recalculate trajectory to the target
+ int getZ()	Return projectile’s Z (20)
+ void onCollision(ICollidable collider)	- If this object is destroyed : do nothing - If hit another projectile : do nothing - If hit TileObjectVoid : destroy this (out of screen) - If hit enemy : destroy this and damage enemy
+ void draw(GraphicsContext gc, Image img)	Draw rotated image
+ void reduceHP(int damage)	Override reduceHP function to do nothing (projectile can’t be damaged)

6.2 ProjectileArrow

Arrow (From arrow tower)

Fields

- double WIDTH = 20	Width of arrow
- double HEIGHT = 8	Height of arrow
- double SPEED = 8	Speed of arrow
- double DAMAGE = 5	Damage of arrow

Methods

+ ProjectileArrow(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectileArrow(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void draw(GraphicsContext gc)	Call super.draw with projectile arrow image

6.3 ProjectileBomb

Bomb (From bomb tower)

Fields

- double WIDTH = 20	Width of bomb
- double HEIGHT = 8	Height of bomb
- double SPEED = 8	Speed of bomb
- double DAMAGE = 5	Damage of bomb
- double EXPLOSIVE_RANGE = 120	Range of explosion of this bomb
- double EXPLOSIVE_DAMAGE = 60	Damage of explosion of this bomb

Methods

+ ProjectileBomb(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectileBomb(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void onDestroy()	Call super.onDestroy and explode (damage enemy around this)
+ void draw(GraphicsContext gc)	Call super.draw with projectile bomb image

6.4 ProjectileBullet

Bullet (from Turret tower)

Fields

- double WIDTH = 20	Width of bullet
- double HEIGHT = 8	Height of bullet
- double SPEED = 12	Speed of bullet
- double DAMAGE = 40	Damage of bullet

Methods

+ ProjectileBullet(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectileBullet(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void draw(GraphicsContext gc)	Call super.draw with projectile bullet image

6.5 ProjectileBulletSniper

Sniper bullet (from Sniper tower)

Fields

- double WIDTH = 40	Width of sniper bullet
- double HEIGHT = 8	Height of sniper bullet
- double SPEED = 20	Speed of sniper bullet
- double DAMAGE = 200	Damage of sniper bullet

Methods

+ ProjectileBulletSniper(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, targetX, targetY) and set width, height, speed, damage
+ ProjectileBulletSniper(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)

6.6 ProjectileLaser

Laser (from laser tower)

Fields

- double WIDTH = 60	Width of laser
- double HEIGHT = 7	Height of laser
- double SPEED = 20	Speed of laser
- double DAMAGE = 6	Damage of laser

Methods

+ ProjectileLaser(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectileLaser(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void draw(GraphicsContext gc)	Call super.draw with projectile laser image

6.7 ProjectilePlayerBullet

Player’s bullet (Damage can be modified from research)

Fields

- double WIDTH = 20	Width of bullet
- double HEIGHT = 8	Height of bullet
- double SPEED = 12	Speed of bullet
- int damage = 15	Damage of bullet

Methods

+ ProjectilePlayerBullet(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectilePlayerBullet(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void draw(GraphicsContext gc)	Call super.draw with projectile bullet image
+ void addDamage(int d)	Add bullet’s damage by amount
+ Getter for damage	

6.8 ProjectileRock

Rock (From catapult)

Fields

- double WIDTH = 30	Width of rock
- double HEIGHT = 30	Height of rock

- double SPEED = 5	Speed of rock
- double DAMAGE = 30	Damage of rock

Methods

+ ProjectileRock(double x, double y, double targetX, double targetY)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, targetX, targetY)
+ ProjectileRock(double x, double y, Entity target)	Call super constructor with (x, y, WIDTH, HEIGHT, SPEED, DAMAGE, target)
+ void draw(GraphicsContext gc)	Call super.draw with projectile rock image

7. model.tileObject

7.1 TileObject

An object that is fix to a tile

Fields

# List<tile> tile	List of tile that this object place on
+ int sizeX,sizeY	Dimension of tile object

Methods

+ TileObject(Tile tile, int sizeX, int sizeY, int hp)	Call super constructor with (x, y, width, height, hp) , calculate x, y, width, height from tile and sizeX, sizeY
# void onDestroy()	Call super then remove back link on tiles
+ int getZ()	Return tile object’s Z (10)
+ void undoMove()	Do nothing, tile object aren't supposed to move anyway
+ void place(Tile tile)	Place on tile(s) - Foreach tile , set back link to this object and add to tile list - Add this object to game

7.2 TileObjectRocket

Rocket silo (When placed, it will countdown from 30 sec to 0 sec and then win)

Fields

- double START_HP = 1500	Start HP of rocket
+ double SIZE_X = 3	SizeX of rocket
+ double SIZE_Y = 3	SizeY of rocket
- double LAUNCH_DELAY = 1800	Launch delay (1800 ticks = 30 sec)
- int launchTimer	Launch timer (count up from 0 to 1800)

Methods

+ TileObjectRocket(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP) and add 1 to rocket count
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# void onDestroy()	Call super.onDestroy and reduce rocket count
+ void update()	Call super.update and count timer, after timer ends, set rocket launched to true to end the game
+ void draw(GraphicsContext gc)	- Call super draw with rocket image - Draw timer
+ int[] getResourceNeeded()	Get resources needed to build this object (Wood = 100, Stone = 100, Iron = 100, Diamond = 100, Alien Artifact = 9999)

7.3 TileObjectStone

Stone (Give 3 stones when destroy)

Fields

- double START_HP = 250	Start HP of stone
+ double SIZE_X = 2	SizeX of stone
+ double SIZE_Y = 2	SizeY of stone

Methods

+ TileObjectStone(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# void onDestroy()	Call super.onDestroy and add 3 stones to resource
+ void draw(GraphicsContext gc)	Call super draw with stone image

7.4 TileObjectTree

Tree (Give 2 woods when destroyed)

Fields

- double START_HP = 100	Start HP of tree
+ double SIZE_X = 1	SizeX of tree
+ double SIZE_Y = 1	SizeY of tree

Methods

+ TileObjectTree(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# void onDestroy()	Call super.onDestroy and add 2 wood to resource
+ void draw(GraphicsContext gc)	Call super draw with tree image

7.5 TileObjectVoid

TileObjectVoid is place around the outside of the screen so entity won’t go out (It will be blocked with this)

Fields

- double START_HP = -1	TileObjectVoid don’t die
+ double SIZE_X = 1	SizeX of TileObjectVoid
+ double SIZE_Y = 1	SizeY of TileObjectVoid

Methods

+ TileObjectVoid(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ void draw(GraphicsContext gc)	Do nothing (Don’t draw this object)
+ void drawHealthBar(GraphicsContext gc)	Do nothing (Don’t draw object health bar)
+ void update()	Do nothing (Don’t update this object)
+ void place(Tile tile)	Override place function to place on TileVoid

7.6 TileObjectWallIron

Iron wall

Fields

- double START_HP = 1000	Start HP of iron wall
+ double SIZE_X = 1	SizeX of iron wall
+ double SIZE_Y = 1	SizeY of iron wall

Methods

+ TileObjectWallIron(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with iron wall image
+ int[] getResourceNeeded()	Return resource needed to buy iron wall (Iron = 2)

7.7 TileObjectWallStone

Stone wall

Fields

- double START_HP = 700	Start HP of stone wall
+ double SIZE_X = 1	SizeX of stone wall
+ double SIZE_Y = 1	SizeY of stone wall

Methods

+ TileObjectWallStone(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with stone wall image
+ int[] getResourceNeeded()	Return resource needed to buy stone wall (Stone = 2)

7.8 TileObjectWallWood

Wood wall

Fields

- double START_HP = 300	Start HP of wood wall
+ double SIZE_X = 1	SizeX of wood wall
+ double SIZE_Y = 1	SizeY of wood wall

Methods

+ TileObjectWallWood(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with wood wall image
+ int[] getResourceNeeded()	Return resource needed to buy wood wall (Wood = 2)

8 model.tileObject.generator

8.1 Generator

Base class for generator

Fields

- int resource	Resource index
- int amount	Amount of resource to add
- int resourceGenerateTimer	Timer
- int resourceGenerateDelay	Delay between resource generation

Methods

+ Generator(Tile tile, int sizeX, int sizeY, int hp, int resource, int delay, int amount)	Call super constructor with (tile, sizeX, sizeY, hp) and initialize resource, resourceGenerateDelay, amount
+ void update()	Call super.update() and count timer, if timer reach delay then generate resource

8.2 GeneratorDiamond

Diamond generator (1 diamond every 3 seconds)

Fields

+ double SIZE_X = 2	SizeX of diamond generator
+ double SIZE_Y = 2	SizeY of diamond generator
- double START_HP = 1000	Start HP of diamond generator
- int RESOURCE = DIAMOND	Generate diamond
- int DELAY = 180	Delay every 3 sec
- int AMOUNT = 1	Generate 1 every delay

Methods

+ GeneratorDiamond(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, delay, amount)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with diamond generator image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 10, Stone = 10, Iron = 10, Diamond = 10, Alien Artifact = 500)

8.3 GeneratorIron

Iron generator (1 every 2 seconds)

Fields

+ double SIZE_X = 2	SizeX of iron generator
+ double SIZE_Y = 2	SizeY of iron generator
- double START_HP = 750	Start HP of iron generator
- int RESOURCE = IRON	Generate iron
- int DELAY = 120	Delay every 2 sec
- int AMOUNT = 1	Generate 1 every delay

Methods

+ GeneratorIron(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, delay, amount)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with iron generator image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 10, Stone = 10, Iron = 10, Diamond = 0, Alien Artifact = 100)

8.4 GeneratorStone

Stone generator (1 every 1 second)

Fields

+	double SIZE_X = 2	SizeX of stone generator
+	double SIZE_Y = 2	SizeY of stone generator
-	double START_HP = 500	Start HP of stone generator
-	int RESOURCE = STONE	Generate stone
-	int DELAY = 60	Delay every 1 sec
-	int AMOUNT = 1	Generate 1 every delay

Methods

+	GeneratorStone(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, delay, amount)
+	boolean canPlace(Tile tile)	Can we place this object at a tile
+	void draw(GraphicsContext gc)	Call super draw with stone generator image
+	int[] getResourceNeeded()	Resources needed to buy this (Wood = 10, Stone = 10, Iron = 0, Diamond = 0, Alien Artifact = 10)

8.5 GeneratorWood

Generate wood (1 every 1 second)

Fields

+	double SIZE_X = 2	SizeX of wood generator
+	double SIZE_Y = 2	SizeY of wood generator
-	double START_HP = 250	Start HP of wood generator
-	int RESOURCE = WOOD	Generate wood
-	int DELAY = 60	Delay every 1 sec
-	int AMOUNT =	Generate 1 every delay

Methods

+	GeneratorWood(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, delay, amount)
+	boolean canPlace(Tile tile)	Can we place this object at a tile
+	void draw(GraphicsContext gc)	Call super draw with wood generator image
+	int[] getResourceNeeded()	Resources needed to buy this (Wood = 10, Stone = 0, Iron = 0, Diamond = 0, Alien Artifact = 1)

9 model.tileObject.storage

9.1 Storage

Base class for storage

Fields

-	int resource	Resource index
-	int amount	Capacity of this storage

Methods

+	Storage(Tile tile, int sizeX, int sizeY, int hp, int resource, int amount)	Call super constructor with (tile, sizeX, sizeY, hp) and initialize variables and add amount to resource's storage
+	void onDestroy()	Call super.onDestroy and decrease resource total capacity

9.2 StorageDiamond

Storage for diamond (add 15 to diamond's capacity)

Fields

+	double SIZE_X = 1	SizeX of diamond storage
+	double SIZE_Y = 1	SizeY of diamond storage
-	double START_HP = 500	Start HP of diamond storage
-	int RESOURCE = DIAMOND	Store diamond
-	int AMOUNT = 15	Store 15 diamond

Methods

+	StorageDiamond(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, amount)
+	boolean canPlace(Tile tile)	Can we place this object at a tile
+	void draw(GraphicsContext gc)	Call super draw with diamond storage image
+	int[] getResourceNeeded()	Resources needed to buy this (Wood = 5, Stone = 0, Iron = 0, Diamond = 5, Alien Artifact = 0)

9.3 StorageIron

Storage for iron (add 15 to iron's capacity)

Fields

+	double SIZE_X = 1	SizeX of iron storage
+	double SIZE_Y = 1	SizeY of iron storage
-	double START_HP = 500	Start HP of iron storage

- int RESOURCE = IRON	Store iron
- int AMOUNT = 15	Store 15 iron

Methods

+ StorageIron(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, amount)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with iron storage image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 5, Stone = 0, Iron = 5, Diamond = 0, Alien Artifact = 0)

9.4 StorageStone

Storage for stone (add 15 to stone's capacity)

Fields

+ double SIZE_X = 1	SizeX of stone storage
+ double SIZE_Y = 1	SizeY of stone storage
- double START_HP = 200	Start HP of stone storage
- int RESOURCE = STONE	Store stone
- int AMOUNT = 15	Store 15 stone

Methods

+ StorageStone(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, amount)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with stone storage image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 5, Stone = 5, Iron = 0, Diamond = 0, Alien Artifact = 0)

9.5 StorageWood

Storage for wood (add 15 to wood's capacity)

Fields

+ double SIZE_X = 1	SizeX of wood storage
+ double SIZE_Y = 1	SizeY of wood storage
- double START_HP = 100	Start HP of wood storage
- int RESOURCE = WOOD	Store wood
- int AMOUNT = 15	Store 15 wood

Methods

+ StorageWood(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, resource, amount)

+ boolean canPlace(Tile tile)	Can we place this object at a tile
+ void draw(GraphicsContext gc)	Call super draw with wood storage image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 5, Stone = 0, Iron = 0, Diamond = 0, Alien Artifact = 0)

10 model.tileObject.tower

10.1 Tower

Base class for tower

Fields

- int shootingDelay	Delay between each shot
- int shootingTimer	Timer for shooting
- double shootingRange	Maximum distance that tower can shoot

Methods

+ Tower(Tile tile, int sizeX, int sizeY,int hp, int shootingDelay, double shootingRange)	Call super constructor with (tile, SIZE_X, SIZE_Y, hp) and initialize variables
+ void update()	Call super.update and shoot when shootingTimer reach shooting delay
# Projectile createProjectile(double x, double y, Entity target)	Create Projectile object to shoot
- void shoot()	Shoot closest enemy

10.2 TowerArrow

Arrow tower (Shoot ProjectileArrow)

Fields

+ double SIZE_X = 1	SizeX of arrow tower
+ double SIZE_Y = 1	SizeY of arrow tower
- double START_HP = 200	Start HP of arrow tower
- int SHOOTING_DELAY = 20	Delay 20 tick
+ int SHOOTING_RANGE = 150	Range 150 px

Methods

+ TowerArrow(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x,	Create new ProjectileArrow object

double y, Entity target)	
+ void draw(GraphicsContext gc)	Draw arrow tower image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 3, Stone = 0, Iron = 0, Diamond = 0, Alien Artifact = 0)

10.3 TowerBomb

Bomb tower (Shoot ProjectileBomb)

Fields

+ double SIZE_X = 2	SizeX of bomb tower
+ double SIZE_Y = 2	SizeY of bomb tower
- double START_HP = 800	Start HP of bomb tower
- int SHOOTING_DELAY = 60	Delay 60 tick
+ int SHOOTING_RANGE = 300	Range 300 px

Methods

+ TowerBomb(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x, double y, Entity target)	Create new ProjectileBomb object
+ void draw(GraphicsContext gc)	Draw bomb tower image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 0, Stone = 2, Iron = 8, Diamond = 0, Alien Artifact = 0)

10.4 TowerCatapult

Catapult (Shoot ProjectileRock)

Fields

+ double SIZE_X = 2	SizeX of catapult
+ double SIZE_Y = 1	SizeY of catapult
- double START_HP = 250	Start HP of catapult
- int SHOOTING_DELAY = 60	Delay 60 tick
+ int SHOOTING_RANGE = 400	Range 400 px

Methods

+ TowerCatapult(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x, double y, Entity target)	Create new ProjectileRock object
+ void draw(GraphicsContext gc)	Draw catapult image

+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 5, Stone = 5, Iron = 0, Diamond = 0, Alien Artifact = 0)

10.5 TowerLaser

Laser tower (Shoot ProjectileLaser)

Fields

+ double SIZE_X = 2	SizeX of laser tower
+ double SIZE_Y = 2	SizeY of laser tower
- double START_HP = 400	Start HP of laser tower
- int SHOOTING_DELAY = 3	Delay 3 tick
+ int SHOOTING_RANGE = 300	Range 300 px

Methods

+ TowerLaser(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x, double y, Entity target)	Create new ProjectileLaser object
+ void draw(GraphicsContext gc)	Draw laser tower image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 0, Stone = 0, Iron = 6, Diamond = 3, Alien Artifact = 0)

10.6 TowerSniper

Sniper tower (Shoot ProjectileBulletSniper)

Fields

+ double SIZE_X = 2	SizeX of sniper tower
+ double SIZE_Y = 2	SizeY of sniper tower
- double START_HP = 500	Start HP of sniper tower
- int SHOOTING_DELAY = 180	Delay 180 tick
+ int SHOOTING_RANGE = 700	Range 700 px

Methods

+ TowerSniper(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x, double y, Entity target)	Create new ProjectileBulletSniper object
+ void draw(GraphicsContext gc)	Draw sniper tower image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 2, Stone = 2, Iron = 8, Diamond = 2, Alien Artifact = 0)

10.7 TowerTurret

Turret (Shoot ProjectileBullet)

Fields

+ double SIZE_X = 1	SizeX of arrow tower
+ double SIZE_Y = 2	SizeY of arrow tower
- double START_HP = 500	Start HP of arrow tower
- int SHOOTING_DELAY = 30	Delay 30 tick
+ int SHOOTING_RANGE = 250	Range 250 px

Methods

+ TowerTurret(Tile tile)	Call super constructor with (tile, SIZE_X, SIZE_Y, START_HP, SHOOTING_DELAY, SHOOTING_RANGE)
+ boolean canPlace(Tile tile)	Can we place this object at a tile
# Projectile createProjectile(double x, double y, Entity target)	Create new ProjectileBullet object
+ void draw(GraphicsContext gc)	Draw turret tower image
+ int[] getResourceNeeded()	Resources needed to buy this (Wood = 0, Stone = 5, Iron = 3, Diamond = 0, Alien Artifact = 0)

11 thread

11.1 ThreadGameManager

While the game is running, this thread will sleep for 16 ms and call GameManager.update() to update the game state.

11.2 ThreadNewScore

Post new score and player name to the server or show an alert if there’s an error

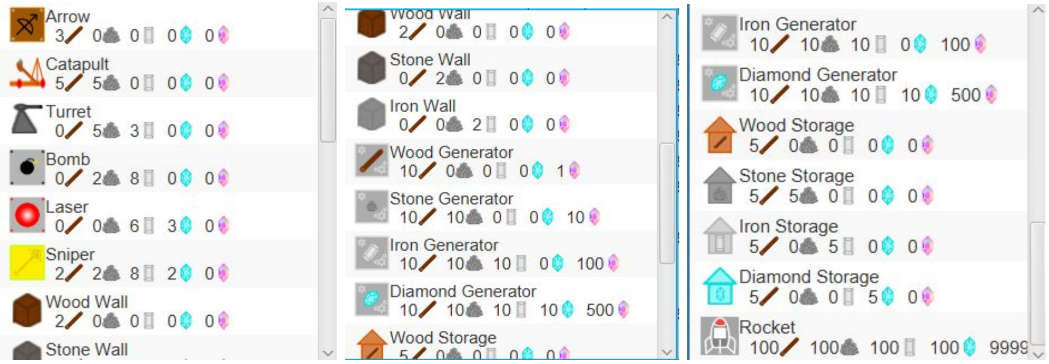
11.3 ThreadShowHighscore

- Fetch high score from server
- Sort score record
- display top 10 in a dialog or show an alert if there’s an error.

12 ui

12.1 BuyBox

Scrollable list view that hold buyItem



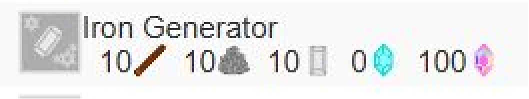
Arrow
3 0 0 0 0
Catapult
5 5 0 0 0
Turret
0 5 3 0 0
Bomb
0 2 8 0 0
Laser
0 0 6 3 0
Sniper
2 2 8 2 0
Wood Wall
2 0 0 0 0
Stone Wall

Wood Wall
2 0 0 0 0
Stone Wall
0 2 0 0 0
Iron Wall
0 0 2 0 0
Wood Generator
10 0 0 0 1
Stone Generator
10 10 0 0 10
Iron Generator
10 10 10 0 100
Diamond Generator
10 10 10 10 500
Wood Storage
5 0 0 0 0

Iron Generator
10 10 10 0 100
Diamond Generator
10 10 10 10 500
Wood Storage
5 0 0 0 0
Stone Storage
5 5 0 0 0
Iron Storage
5 0 5 0 0
Diamond Storage
5 0 0 5 0
Rocket
100 100 100 100 9999

12.2 BuyItem

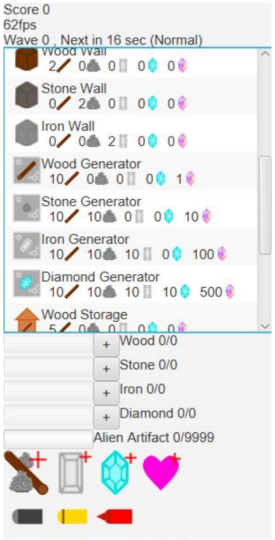
Click on a buy item place on the map. Show item image , name, resource needed to buy



Iron Generator
10 10 10 0 100

12.3 GameMenu

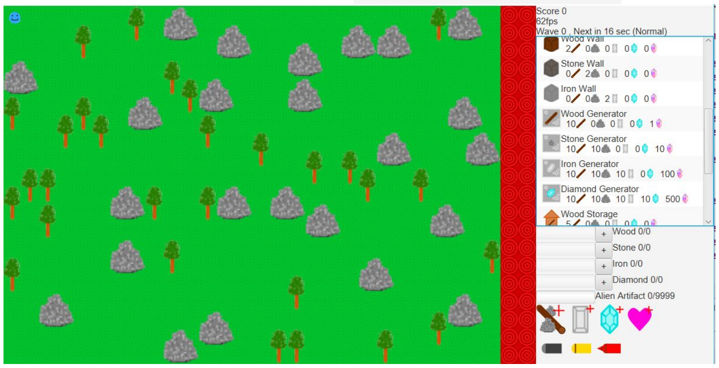
VBox holding StatsBox, BuyBox, ResourceBox, ResearchBox



Score 0
62fps
Wave 0 , Next in 16 sec (Normal)
Wood Wall
2 0 0 0 0
Stone Wall
0 2 0 0 0
Iron Wall
0 0 2 0 0
Wood Generator
10 0 0 0 1
Stone Generator
10 10 0 0 10
Iron Generator
10 10 10 0 100
Diamond Generator
10 10 10 10 500
Wood Storage
5 0 0 0 0
+ Wood 0/0
+ Stone 0/0
+ Iron 0/0
+ Diamond 0/0
Alien Artifact 0/9999

12.4 GamePane

- HBox holding GameScreen and GameMenu.
- When the game start :
    - Create GameManager instance



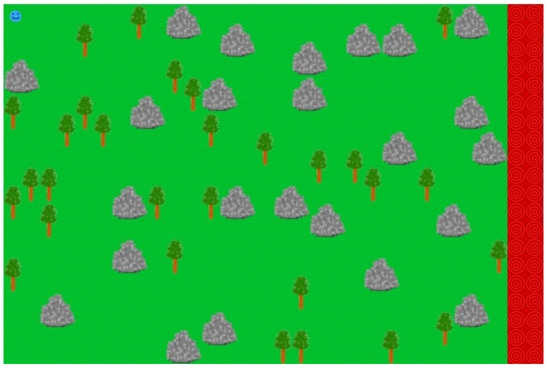
Score 0
62fps
Wave 0 , Next in 16 sec (Normal)
Wood Wall
2 0 0 0 0
Stone Wall
0 2 0 0 0
Iron Wall
0 0 2 0 0
Wood Generator
10 0 0 0 1
Stone Generator
10 10 0 0 10
Iron Generator
10 10 10 0 100
Diamond Generator
10 10 10 10 500
Wood Storage
5 0 0 0 0
+ Wood 0/0
+ Stone 0/0
+ Iron 0/0
+ Diamond 0/0
Alien Artifact 0/9999

  - Create animationTimer to draw game
  - Start GameManager update thread
- When the game stop :
  - Stop GameManager update thread
  - Stop animationTimer

12.5 GameScreen

Canvas for draw game

- paintComponents()
  - synchronized on Renderable.instance.getEntities()
  - drawBackground() and drawEntities()
  - If in buying mode then drawOverlay() and drawBuyingItem()
  - If currently pausing then drawPause()
- drawPause()
  - Draw black overlay and a “PAUSE” text at the center
- drawBuyingItem()
  - Draw a buying object under the mouse
  - If can’t buy, an object will be lightened
- drawOverlay()
  - Draw red overlay on every tile object
  - Otherwise draw green overlay
- drawBackground()
  - Draw black blackground
- drawEntities()
  - Draw all entities (by calling entity.draw())
  - Draw health bar of all entities



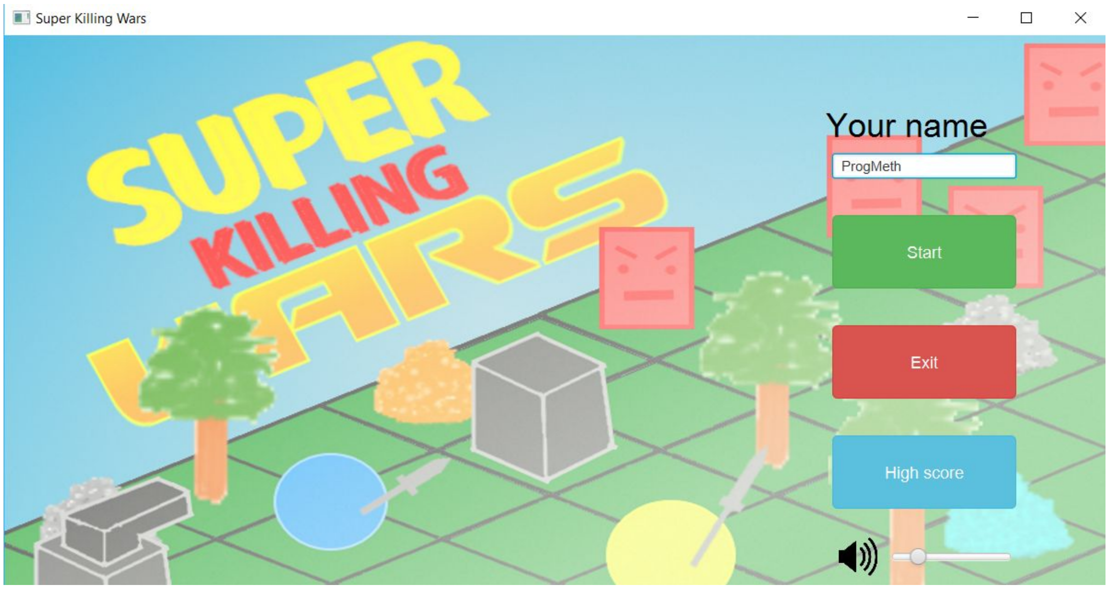
12.6 IStoppable

Stop unfinished job before pane get destroyed

12.7 MainPane

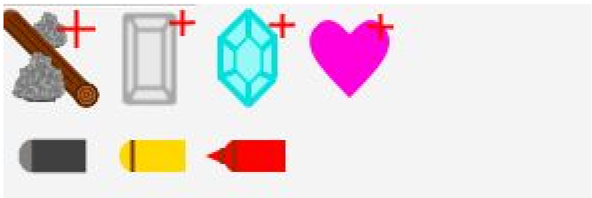
VBox on the main menu with image background, containing :

  - Label “Your Name”
  - Text Field to enter your name
  - Start button (Start the game)
  - Exit button (Exit the game)
  - Highscore button (Show high score)
  - VolumePane



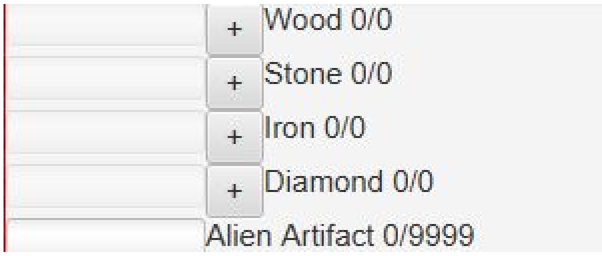
12.8 ResearchBox

GridPane that hold research item



12.9 ResourceBox

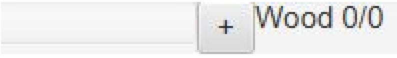
VBox hold resource items



Wood 0/0
Stone 0/0
Iron 0/0
Diamond 0/0
Alien Artifact 0/9999

12.10 ResourceItem

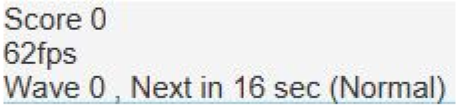
- Show current resource and capacity as a progressbar (with text at the back).
- Buy button to buy this resource from alien artifact



Wood 0/0

12.11 StatsBox

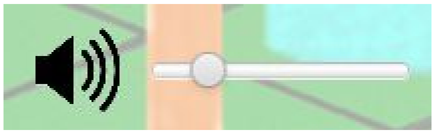
- Show game status



Score 0
62fps
Wave 0 , Next in 16 sec (Normal)

12.12 VolumePane

- Slider to adjust game’s volume



13 ui.research

13.1 BuildingResearch

onResearch() : Add 5 capacity to wood and stone (So you can start building items)
Cost : 25 alien artifacts

13.2 DiamondResearch

onResearch() : Add 5 capacity to diamond (So you can start building diamond items)
Cost : 50 wood, 50 stone, 30 iron, 2500 alien artifacts

13.3 GunResearch

onResearch() : Add 20 damage to ProjectilePlayerBullet
Cost : 1 iron

13.4 HealthRegenerationResearch

onResearch() : Increase health regeneration rate 10
Cost : 500 alien artifacts

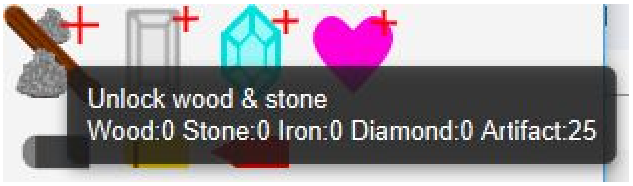
13.5 IronResearch

onResearch() : Add 5 capacity to iron (So you can start building iron items)
Cost : 10 wood, 10 stone, 1000 alien artifacts

13.6 ResearchItem

Base class for research item

- Display tooltip showing name and resource needed for this research
- After research complete call onResearch() and image is darkened



Unlock wood & stone
Wood:0 Stone:0 Iron:0 Diamond:0 Artifact:25

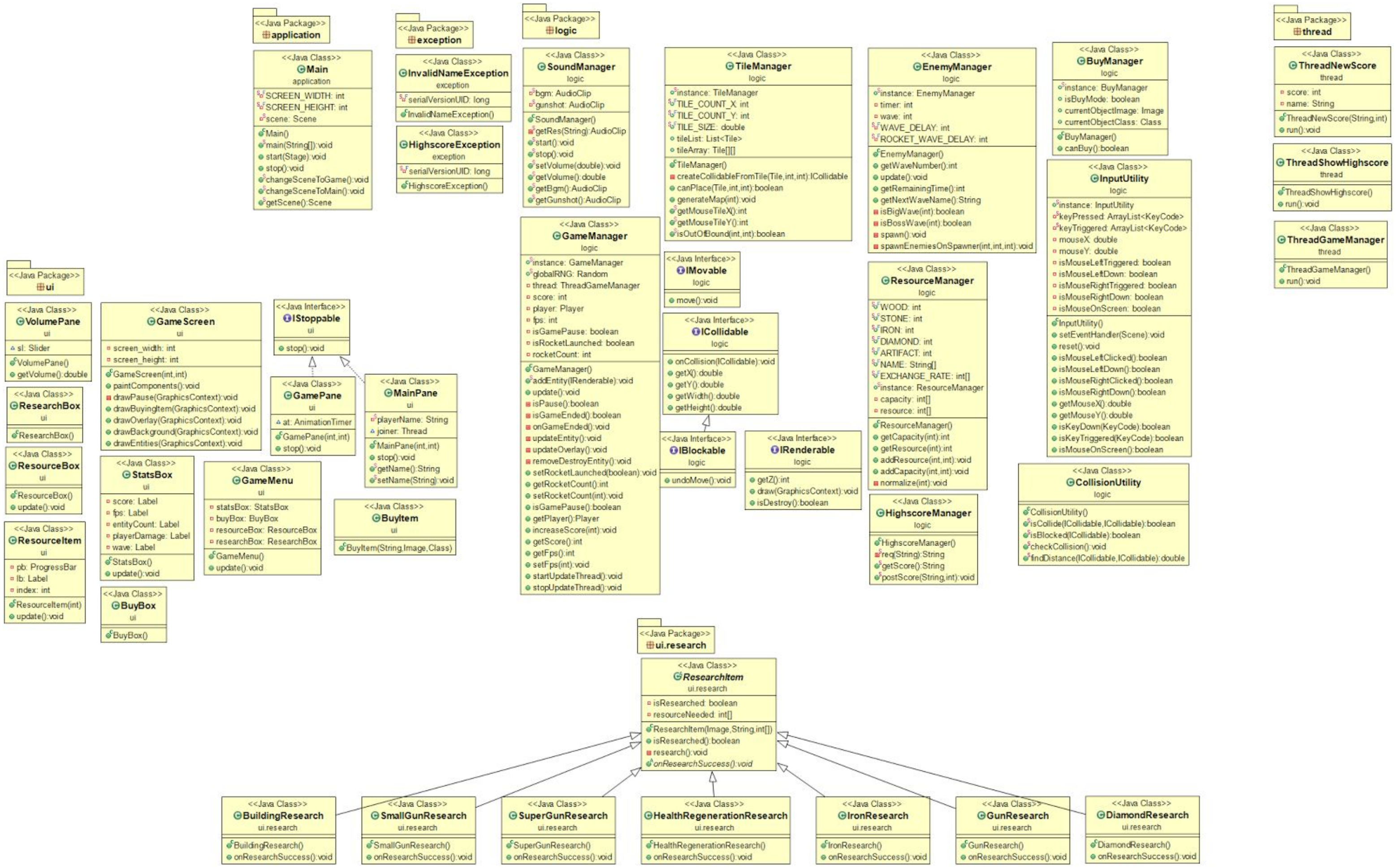
13.7 SmallGunResearch

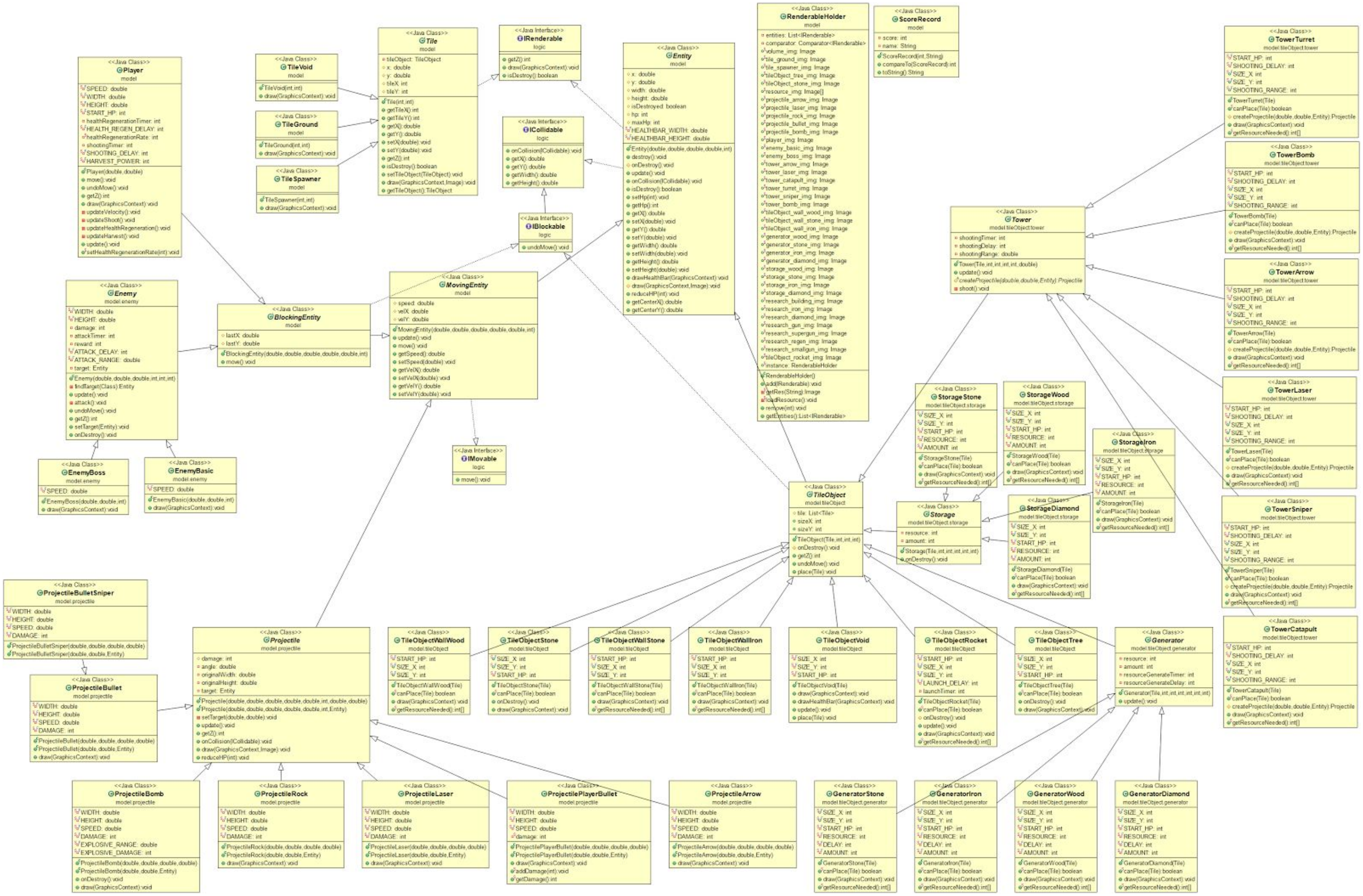
onResearch() : Add 10 damage to ProjectilePlayerBullet
Cost : 2 wood, 2 stone

13.8 SuperGunResearch

onResearch() : Add 40 damage to ProjectilePlayerBullet
Cost : 1 diamond

UML


<<Java Package>>
application
<<Java Class>>
Main
application
SCREEN_WIDTH: int
SCREEN_HEIGHT: int
scene: Scene
Main()
main(String[]):void
start(Stage):void
stop():void
changeSceneToGame():void
changeSceneToMain():void
getScene():Scene
<<Java Package>>
exception
<<Java Class>>
InvalidNameException
exception
serialVersionUID: long
InvalidNameException()
<<Java Class>>
HighscoreException
exception
serialVersionUID: long
HighscoreException()
<<Java Package>>
logic
<<Java Class>>
SoundManager
logic
bgm: AudioClip
gunshot: AudioClip
SoundManager()
getRes(String):AudioClip
start():void
stop():void
setVolume(double):void
getVolume():double
getBgm():AudioClip
getGunshot():AudioClip
<<Java Class>>
TileManager
logic
instance: TileManager
TILE_COUNT_X: int
TILE_COUNT_Y: int
TILE_SIZE: double
tileList: List<Tile>
tileArray: Tile[][]
TileManager()
createCollidableFromTile(Tile,int,int):ICollidable
canPlace(Tile,int,int):boolean
generateMap(int):void
getMouseTileX():int
getMouseTileY():int
isOutOfBound(int,int):boolean
<<Java Class>>
EnemyManager
logic
instance: EnemyManager
timer: int
wave: int
WAVE_DELAY: int
ROCKET_WAVE_DELAY: int
EnemyManager()
getWaveNumber():int
update():void
getRemainingTime():int
getNextWaveName():String
isBigWave(int):boolean
isBossWave(int):boolean
spawn():void
spawnEnemiesOnSpawner(int,int,int):void
<<Java Class>>
BuyManager
logic
instance: BuyManager
isBuyMode: boolean
currentObjectImage: Image
currentObjectClass: Class
BuyManager()
canBuy():boolean
<<Java Class>>
InputUtility
logic
instance: InputUtility
keyPressed: ArrayList<KeyCode>
keyTriggered: ArrayList<KeyCode>
mouseX: double
mouseY: double
isMouseLeftTriggered: boolean
isMouseLeftDown: boolean
isMouseRightTriggered: boolean
isMouseRightDown: boolean
isMouseOnScreen: boolean
InputUtility()
setEventHandler(Scene):void
reset():void
isMouseLeftClicked():boolean
isMouseLeftDown():boolean
isMouseRightClicked():boolean
isMouseRightDown():boolean
getMouseX():double
getMouseY():double
isKeyDown(KeyCode):boolean
isKeyTriggered(KeyCode):boolean
isMouseOnScreen():boolean
<<Java Class>>
GameManager
logic
instance: GameManager
globalRNG: Random
thread: ThreadGameManager
score: int
player: Player
fps: int
isGamePause: boolean
isRocketLaunched: boolean
rocketCount: int
GameManager()
addEntity(IRenderable):void
update():void
isPause():boolean
isGameEnded():boolean
onGameEnded():void
updateEntity():void
updateOverlay():void
removeDestroyEntity():void
setRocketLaunched(boolean):void
getRocketCount():int
setRocketCount(int):void
isGamePause():boolean
getPlayer():Player
increaseScore(int):void
getScore():int
getFps():int
setFps(int):void
startUpdateThread():void
stopUpdateThread():void
<<Java Interface>>
IMovable
logic
move():void
<<Java Interface>>
ICollidable
logic
onCollision(ICollidable):void
getX():double
getY():double
getWidth():double
getHeight():double
<<Java Interface>>
IBlockable
logic
undoMove():void
<<Java Interface>>
IRenderable
logic
getZ():int
draw(GraphicsContext):void
isDestroy():boolean
<<Java Class>>
ResourceManager
logic
WOOD: int
STONE: int
IRON: int
DIAMOND: int
ARTIFACT: int
NAME: String[]
EXCHANGE_RATE: int[]
instance: ResourceManager
capacity: int[]
resource: int[]
ResourceManager()
getCapacity(int):int
getResource(int):int
addResource(int,int):void
addCapacity(int,int):void
normalize(int):void
<<Java Class>>
HighscoreManager
logic
HighscoreManager()
req(String):String
getScore():String
postScore(String,int):void
<<Java Class>>
CollisionUtility
logic
CollisionUtility()
isCollide(ICollidable,ICollidable):boolean
isBlocked(ICollidable):boolean
checkCollision():void
findDistance(ICollidable,ICollidable):double
<<Java Package>>
thread
<<Java Class>>
ThreadNewScore
thread
score: int
name: String
ThreadNewScore(String,int)
run():void
<<Java Class>>
ThreadShowHighscore
thread
ThreadShowHighscore()
run():void
<<Java Class>>
ThreadGameManager
thread
ThreadGameManager()
run():void
<<Java Package>>
ui
<<Java Class>>
VolumePane
ui
sl: Slider
VolumePane()
getVolume():double
<<Java Class>>
ResearchBox
ui
ResearchBox()
<<Java Class>>
ResourceBox
ui
ResourceBox()
update():void
<<Java Class>>
ResourceItem
ui
pb: ProgressBar
lb: Label
index: int
ResourceItem(int)
update():void
<<Java Class>>
GameScreen
ui
screen_width: int
screen_height: int
GameScreen(int,int)
paintComponents():void
drawPause(GraphicsContext):void
drawBuyingItem(GraphicsContext):void
drawOverlay(GraphicsContext):void
drawBackground(GraphicsContext):void
drawEntities(GraphicsContext):void
<<Java Class>>
StatsBox
ui
score: Label
fps: Label
entityCount: Label
playerDamage: Label
wave: Label
StatsBox()
update():void
<<Java Class>>
BuyBox
ui
BuyBox()
<<Java Interface>>
IStoppable
ui
stop():void
<<Java Class>>
GamePane
ui
at: AnimationTimer
GamePane(int,int)
stop():void
<<Java Class>>
MainPane
ui
playerName: String
joiner: Thread
MainPane(int,int)
stop():void
getName():String
setName(String):void
<<Java Class>>
GameMenu
ui
statsBox: StatsBox
buyBox: BuyBox
resourceBox: ResourceBox
researchBox: ResearchBox
GameMenu()
update():void
<<Java Class>>
BuyItem
ui
BuyItem(String,Image,Class)
<<Java Package>>
ui.research
<<Java Class>>
ResearchItem
ui.research
isResearched: boolean
resourceNeeded: int[]
ResearchItem(Image,String,int[])
isResearched():boolean
research():void
onResearchSuccess():void
<<Java Class>>
BuildingResearch
ui.research
BuildingResearch()
onResearchSuccess():void
<<Java Class>>
SmallGunResearch
ui.research
SmallGunResearch()
onResearchSuccess():void
<<Java Class>>
SuperGunResearch
ui.research
SuperGunResearch()
onResearchSuccess():void
<<Java Class>>
HealthRegenerationResearch
ui.research
HealthRegenerationResearch()
onResearchSuccess():void
<<Java Class>>
IronResearch
ui.research
IronResearch()
onResearchSuccess():void
<<Java Class>>
GunResearch
ui.research
GunResearch()
onResearchSuccess():void
<<Java Class>>
DiamondResearch
ui.research
DiamondResearch()
onResearchSuccess():void


<<Java Class>>
Player
model
SPEED: double
WIDTH: double
HEIGHT: double
START_HP: int
healthRegenerationTimer: int
HEALTH_REGEN_DELAY: int
healthRegenerationRate: int
shootingTimer: int
SHOOTING_DELAY: int
HARVEST_POWER: int
Player(double,double)
move():void
undoMove():void
getZ():int
draw(GraphicsContext):void
updateVelocity():void
updateShoot():void
updateHealthRegeneration():void
updateHarvest():void
update():void
setHealthRegenerationRate(int):void
<<Java Class>>
TileVoid
model
TileVoid(int,int)
draw(GraphicsContext):void
<<Java Class>>
TileGround
model
TileGround(int,int)
draw(GraphicsContext):void
<<Java Class>>
TileSpawner
model
TileSpawner(int,int)
draw(GraphicsContext):void
<<Java Class>>
Tile
model
tileObject: TileObject
x: double
y: double
tileX: int
tileY: int
Tile(int,int)
getTileX():int
getTileY():int
getX():double
getY():double
setX(double):void
setY(double):void
getZ():int
isDestroy():boolean
setTileObject(TileObject):void
draw(GraphicsContext,Image):void
getTileObject():TileObject
<<Java Interface>>
IRenderable
logic
getZ():int
draw(GraphicsContext):void
isDestroy():boolean
<<Java Interface>>
ICollidable
logic
onCollision(ICollidable):void
getX():double
getY():double
getWidth():double
getHeight():double
<<Java Interface>>
IBlockable
logic
undoMove():void
<<Java Class>>
Entity
model
x: double
y: double
width: double
height: double
isDestroyed: boolean
hp: int
maxHp: int
HEALTHBAR_WIDTH: double
HEALTHBAR_HEIGHT: double
Entity(double,double,double,double,int)
destroy():void
onDestroy():void
update():void
onCollision(ICollidable):void
isDestroy():boolean
setHp(int):void
getHp():int
getX():double
getY():double
setX(double):void
setY(double):void
getWidth():double
setWidth(double):void
getHeight():double
setHeight(double):void
drawHealthBar(GraphicsContext):void
draw(GraphicsContext,Image):void
reduceHP(int):void
getCenterX():double
getCenterY():double
<<Java Class>>
RenderableHolder
model
entities: List<IRenderable>
comparator: Comparator<IRenderable>
volume_img: Image
tile_ground_img: Image
tile_spawner_img: Image
tileObject_tree_img: Image
tileObject_stone_img: Image
resource_img: Image[]
projectile_arrow_img: Image
projectile_laser_img: Image
projectile_rock_img: Image
projectile_bullet_img: Image
projectile_bomb_img: Image
player_img: Image
enemy_basic_img: Image
enemy_boss_img: Image
tower_arrow_img: Image
tower_laser_img: Image
tower_catapult_img: Image
tower_turret_img: Image
tower_sniper_img: Image
tower_bomb_img: Image
tileObject_wall_wood_img: Image
tileObject_wall_stone_img: Image
tileObject_wall_iron_img: Image
generator_wood_img: Image
generator_stone_img: Image
generator_iron_img: Image
generator_diamond_img: Image
storage_wood_img: Image
storage_stone_img: Image
storage_iron_img: Image
storage_diamond_img: Image
research_building_img: Image
research_iron_img: Image
research_diamond_img: Image
research_gun_img: Image
research_supergun_img: Image
research_regen_img: Image
research_smallgun_img: Image
tileObject_rocket_img: Image
instance: RenderableHolder
RenderableHolder()
add(IRenderable):void
getRes(String):Image
loadResource():void
remove(int):void
getEntities():List<IRenderable>
<<Java Class>>
ScoreRecord
model
score: int
name: String
ScoreRecord(int,String)
compareTo(ScoreRecord):int
toString():String
<<Java Class>>
TowerTurret
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerTurret(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TowerBomb
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerBomb(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
Tower
model.tileObject.tower
shootingTimer: int
shootingDelay: int
shootingRange: double
Tower(Tile,int,int,int,int,double)
update():void
createProjectile(double,double,Entity):Projectile
shoot():void
<<Java Class>>
TowerArrow
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerArrow(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
Enemy
model.enemy
WIDTH: double
HEIGHT: double
damage: int
attackTimer: int
reward: int
ATTACK_DELAY: int
ATTACK_RANGE: double
target: Entity
Enemy(double,double,double,int,int,int)
findTarget(Class):Entity
update():void
attack():void
undoMove():void
getZ():int
setTarget(Entity):void
onDestroy():void
<<Java Class>>
BlockingEntity
model
lastX: double
lastY: double
BlockingEntity(double,double,double,double,double,int)
move():void
<<Java Class>>
MovingEntity
model
speed: double
velX: double
velY: double
MovingEntity(double,double,double,double,double,int)
update():void
move():void
getSpeed():double
setSpeed(double):void
getVelX():double
setVelX(double):void
getVelY():double
setVelY(double):void
<<Java Interface>>
IMovable
logic
move():void
<<Java Class>>
EnemyBoss
model.enemy
SPEED: double
EnemyBoss(double,double,int)
draw(GraphicsContext):void
<<Java Class>>
EnemyBasic
model.enemy
SPEED: double
EnemyBasic(double,double,int)
draw(GraphicsContext):void
<<Java Class>>
StorageStone
model.tileObject.storage
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
AMOUNT: int
StorageStone(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
StorageWood
model.tileObject.storage
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
AMOUNT: int
StorageWood(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
StorageIron
model.tileObject.storage
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
AMOUNT: int
StorageIron(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TowerLaser
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerLaser(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TileObject
model.tileObject
tile: List<Tile>
sizeX: int
sizeY: int
TileObject(Tile,int,int,int)
onDestroy():void
getZ():int
undoMove():void
place(Tile):void
<<Java Class>>
Storage
model.tileObject.storage
resource: int
amount: int
Storage(Tile,int,int,int,int,int)
onDestroy():void
<<Java Class>>
StorageDiamond
model.tileObject.storage
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
AMOUNT: int
StorageDiamond(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TowerSniper
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerSniper(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
ProjectileBulletSniper
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
ProjectileBulletSniper(double,double,double,double)
ProjectileBulletSniper(double,double,Entity)
<<Java Class>>
ProjectileBullet
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
ProjectileBullet(double,double,double,double)
ProjectileBullet(double,double,Entity)
draw(GraphicsContext):void
<<Java Class>>
Projectile
model.projectile
damage: int
angle: double
originalWidth: double
originalHeight: double
target: Entity
Projectile(double,double,double,double,double,int,double,double)
Projectile(double,double,double,double,double,int,Entity)
setTarget(double,double):void
update():void
getZ():int
onCollision(ICollidable):void
draw(GraphicsContext,Image):void
reduceHP(int):void
<<Java Class>>
TileObjectWallWood
model.tileObject
START_HP: int
SIZE_X: int
SIZE_Y: int
TileObjectWallWood(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TileObjectStone
model.tileObject
SIZE_X: int
SIZE_Y: int
START_HP: int
TileObjectStone(Tile)
canPlace(Tile):boolean
onDestroy():void
draw(GraphicsContext):void
<<Java Class>>
TileObjectWallStone
model.tileObject
START_HP: int
SIZE_X: int
SIZE_Y: int
TileObjectWallStone(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TileObjectWallIron
model.tileObject
START_HP: int
SIZE_X: int
SIZE_Y: int
TileObjectWallIron(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TileObjectVoid
model.tileObject
SIZE_X: int
SIZE_Y: int
START_HP: int
TileObjectVoid(Tile)
draw(GraphicsContext):void
drawHealthBar(GraphicsContext):void
update():void
place(Tile):void
<<Java Class>>
TileObjectRocket
model.tileObject
START_HP: int
SIZE_X: int
SIZE_Y: int
LAUNCH_DELAY: int
launchTimer: int
TileObjectRocket(Tile)
canPlace(Tile):boolean
onDestroy():void
update():void
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
TileObjectTree
model.tileObject
SIZE_X: int
SIZE_Y: int
START_HP: int
TileObjectTree(Tile)
canPlace(Tile):boolean
onDestroy():void
draw(GraphicsContext):void
<<Java Class>>
Generator
model.tileObject.generator
resource: int
amount: int
resourceGenerateTimer: int
resourceGenerateDelay: int
Generator(Tile,int,int,int,int,int,int)
update():void
<<Java Class>>
TowerCatapult
model.tileObject.tower
START_HP: int
SHOOTING_DELAY: int
SIZE_X: int
SIZE_Y: int
SHOOTING_RANGE: int
TowerCatapult(Tile)
canPlace(Tile):boolean
createProjectile(double,double,Entity):Projectile
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
ProjectileBomb
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
EXPLOSIVE_RANGE: double
EXPLOSIVE_DAMAGE: int
ProjectileBomb(double,double,double,double)
ProjectileBomb(double,double,Entity)
onDestroy():void
draw(GraphicsContext):void
<<Java Class>>
ProjectileRock
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
ProjectileRock(double,double,double,double)
ProjectileRock(double,double,Entity)
draw(GraphicsContext):void
<<Java Class>>
ProjectileLaser
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
ProjectileLaser(double,double,double,double)
ProjectileLaser(double,double,Entity)
draw(GraphicsContext):void
<<Java Class>>
ProjectilePlayerBullet
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
damage: int
ProjectilePlayerBullet(double,double,double,double)
ProjectilePlayerBullet(double,double,Entity)
draw(GraphicsContext):void
addDamage(int):void
getDamage():int
<<Java Class>>
ProjectileArrow
model.projectile
WIDTH: double
HEIGHT: double
SPEED: double
DAMAGE: int
ProjectileArrow(double,double,double,double)
ProjectileArrow(double,double,Entity)
draw(GraphicsContext):void
<<Java Class>>
GeneratorStone
model.tileObject.generator
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
DELAY: int
AMOUNT: int
GeneratorStone(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
GeneratorIron
model.tileObject.generator
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
DELAY: int
AMOUNT: int
GeneratorIron(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
GeneratorWood
model.tileObject.generator
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
DELAY: int
AMOUNT: int
GeneratorWood(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]
<<Java Class>>
GeneratorDiamond
model.tileObject.generator
SIZE_X: int
SIZE_Y: int
START_HP: int
RESOURCE: int
DELAY: int
AMOUNT: int
GeneratorDiamond(Tile)
canPlace(Tile):boolean
draw(GraphicsContext):void
getResourceNeeded():int[]