Gem Defender

Made by

Mr.Satasuk Vipaksinlapin 5931062221

Mr.Supanat Wongwiwatchai 5931068021

2110215 Programming Methodology (2017/1)

Gem Defender

1 Problem statement : tower defense game

Gem defender is a tower defense game which players place towers to defence hoards of monsters and to protect the gem from the monsters. When the players kill the monsters, they get a score and some gold.If the monsters reach the gem, the gem's hitpoint will be decreased.If the gem's hitpoint is 0, the game is over.

The game has 2 scenes, a game scene(Figure 2) and a main menu scene(Figure 1).The main menu scene is the first one shown up when players start the program.

Figure 1 the main menu screen

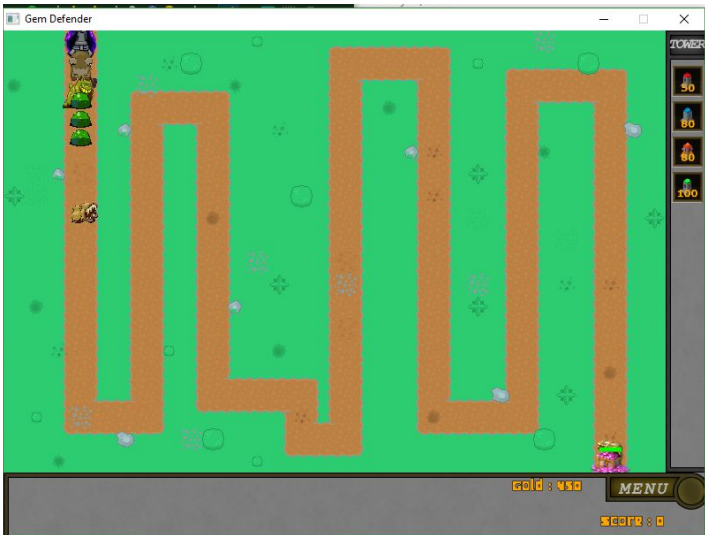


Figure 2 the game screen

2 Implementation Detail

2.1 Package : (default package)

2.1.1 Class Game

2.1.1.1 Field

+ Thread wait	True if any button was clicked
- GameScreen gameScreen	For drawing game component
- PaneForRenderImageViews pr	for add ImageView to game
- GameLogic logic	For update game object
+ boolean pause	True when paused
- Rectangle pauserec	To add event handler when game paused
- Rectangle endrec	To add event handler when game ended
- StackPane root2	Stackpane that contain pr,pausepane,endpane,gameScreen
- Pane pausepane	To display when game paused
- Pane endpane	To display when game ended

2.1.1.2 Constructor

+ Menu()	Initialize root,GridPane topbase,GridPane bottombase,TowerMenu towerMenu, BottomMenu bottomMenu,root2,logic,gameScreen,pr, pauserec. Run method mouseEventOnField from InputUtility,makePause(),makeEnd() makeBgmusic(),Fadingin() .play bgsound

2.1.1.3 Method

+ void run()	Initialize thread wait ,Thread sleep for one second then set wait in GameLogic to false,if game not ended and not paused paintImageView,update RenderableHolder , logicUpdate ,start wait.if game ended set pause to true play new background music Paint all text ,clear game screen ,endpane set visible to true .if game is paused set pauserec visible to false and set pausepane visible to true
+ void makeBgmusic()	Initialize bgsound
+ void makepause()	Initialize pausepane,ImageView pausemenu ,ImageView mubutton, ImageView reumebutton add all to pausepane , set event
+ void makeEnd()	Initialize endpane set opacity to 0 set on event , set visible to false

2.1.2 Abstract Class GameState

2.1.2.1 Field

+ static final int WIDTH	Set to 800
+ static final int HEIGH	Set to 600
# ImageView bg1	First background image
# ImageView bg2	Second bg image
+ GraphicsContext gc	Graphicscontext for each game state
+ Canvas canvas	Canvas for each game state
# MediaPlayer bgsound	Background music

# Scene scene	Scene in each gamestate
+ Static Pane root	Root in each game state
# FadeTransition ft	For fading effect

2.1.2.2 Method

+ Abstract void run()	
+ Abstract void makeBgmusic()	Set back ground music
+ void Fadingin()	For fading in effect
+ void Fadingout()	For fading out effect
+ Getter and Setter	

2.1.3 Class GemDefender

2.1.3.1 Field

+ static final int WIDTH	Set to 800
+ static final int HEIGH	Set to 600
+ GameState gs	Game state
+ static Stage window	Game stage

2.1.3.2 Method

+ void run()	Initialize gs ,AnimationTimer.set resizable to false set title to Gem Defender
+ static void main(String[] args)	
+ void stop()	System exit
+ Getter and Setter	

2.1.4 Class Menu

2.1.4.1 Field

- boolean buttonclick	True if any button was clicked

2.1.4.2 Constructor

+ Menu()	Initialize root ,background image ,canvas and then run method makeBgmusic() , creatBg() , draw().

2.1.4.3 Method

+ void creatBg()	Add background image and canvas to root pane
+ void draw()	Draw all text for the menu
+ void addHitbox(Pane root,GraphicContext gc)	Add Rectangle to cover all button and set mouse event
+ void run()	Run method bgrun()
+ void goGameScene()	Change GemDefender GameState gs to new Game
+ void bgrun()	Run background image as loop
+ makeBgmusic()	Initialize bgsound . Set on Media loop. Play bgsound

2.2 Package : core

2.2.1 Class GameCore

2.2.1.1 Field

- int gold	It’s a currency that is used in the game.
- int score	Game score
- Tower selectedTower	A tower that is selected.

- int buyStatus	Type of a tower that players want to buy
- int arrowTowerPrice,iceTowerPrice,sniperTowerPrice,fireTowerPrice	A price of a tower.
- boolean canPlace	True if the player can place the tower in the selected cell.

2.2.1.2 Method

+ void upgradeTower()	Upgrade the selected tower.
+ void sellTower()	Sell the selected tower.
+ void placeTower(int row,int column)	Place the tower to the specify cell.
+ boolean checkCanPlace(int row,int column)	True if the specify cell can place a tower.
+ Getters and Setters	

2.3 Package : drawing

2.3.1 Class GameScreen extends Canvas

2.3.1.1 Field

- GraphicsContext gc	GraphicsContext for method draw

2.3.1.2 Constructor

+ GameScreen(double width ,double height)	Initialize canvas with width and height, initialize gc, set visible to true

2.3.1.3 Method

+ void panitComponent()	Run method draw for every entity in entities from RenderableHolder
+ void clear()	Clear entities from RenderableHolder

2.3.2 Class PaneForRenderImageViews

2.3.2.1 Constructor

+ PaneForRenderImageViews(int width ,int height)	Set pane width and height

2.3.2.2 Method

+ void paintImageView()	Add ImageView to pane if it visible and not destroyed ,add event for every tower

2.4 Package : enemies

2.4.1 Abstract Class Enemy

2.4.1.1 Field

# int maxHp	Max hit points of the enemy.
# int hp	Current hit points of the enemy.
# double normalSpeed	Normal speed of the enemy.
# double speed	Current speed of the enemy.
# boolean freezed	True if the enemy is freezed.
# boolean goldAdded	True if the player got the gold from the enemy when it died.
# boolean up	True if the enemy is moving up.
# int freezedDuration	A freeze duration counter.
# int stage	Current move path stage.
# int price	The enemy bounty.
# int score	The score that the player gained when enemy is killed.

2.4.1.2 Methods

+ void move()	Move the enemy on the specify path.

+ void getHit(int damage)	Attack the enemy.Decrease it hp equals to the damage.If hp<=0,it'll die.
+ void freeze(int duration,int level)	Freeze the enemy for 'duration' and reduce it speed by 'level'
+ void update()	Decrease freezedDuration by time.if freezedDuration equals to 0,enemy won't be freezed anymore.
+ void abstract *draw*()	Draw the enemy sprite.

2.4.2 Class Cave extends Enemy

2.4.2.1 Constructor

+ Cave(int hp)	Set maxHp as hp, speed and normal speed to1.5, radius to 20, position to row 0 column 3, price to 5,animationFrame to 0 and score to 1.

2.4.1.2 Fields

- int animationFrame	Animation frame counter.

2.4.1.3 Methods

+ void draw()	Draw the enemy sprite.

2.4.3 Class Crap extends Enemy

As the class 'Cave' except it speed is 1.2.

2.4.4 Class Rock extends Enemy

As the class 'Cave' except it speed is 0.8 and it price is 7.

2.4.5 Class Slime extends Enemy

As the class 'Cave' except it speed is 1, and it radius is 12.

2.4.6 Class Worm extends Enemy

As the class 'Cave' except it speed is 1.5 and it price is 2..

2.5 Package : input

2.5.1 Class InputUtility

2.5.1.1 Fields

+ double mouseX,mouseY	Cursor position.
+ boolean mouseOnScreen	True if a cursor is on the game screen.
+ int hover	Type of the towers that cursor is pointing to.
+ void addImageViewEvents(Tower tower)	Add events to a sprite of a tower.
+ void setSellButton(ImageView btn)	Add events to the sell button.

2.6 Package : logic

2.6.1 Class Arrow extends Bullet

2.6.1.1 Constructor

+ Arrow(double x , double y ,Enemy lockedEnemy,int damage)	Initilize damage ,x,y,lockedEnemy,set radius to 10 ,set speedx and speedy to 0.

2.6.1.2 Method

+ void draw(GraphicsContext gc)	Set gc color to khaki and draw oval at x,y
+ void misile(Enemy enemy)	Calculate speedx and speedy ,if enemy is within radius enemy get hit then destroy this.
+ void update()	Run misile method

2.6.2 Class ArrowTower extends Tower

2.6.2.1 Constructor

+ ArrowTower(int row,int column)	Set price to 50, damage to 12, splashAttack to false, range to 60, fireRate to 20, fireCount to 0 and the others field as ‘IceTower’

2.6.2.2 Methods

- void attack()	Attack the nearest enemies that is in the attack range by creating ‘Arrow’.
Other methods are like ‘IceTower’	

2.6.3 Abstract Class Bullet extends CollidableEntity

2.6.3.1 Fields

+ int damage	Damage of the bullet
+ double x	X position of the bullet
+ double y	Y position of the bullet
+ double speedx	Velocity of the bullet in x direction
+ double speedy	Velocity of the bullet in y direction
+ Enemy lockedEnemy	Enemy that get locked from tower

2.6.3.2 Methods

+ public abstract void draw(GraphicsContext gc)	Draw bullet
+ public abstract void update()	Update bullet

2.6.4 Abstract Class CollidableEntity extends Entity

2.6.4.1 Fields

# int radius	Radius of this entity

2.6.4.2 Methods

# boolean collideWith(CollidableEntity other)	Return true if these are collide

2.6.5 Abstract Class Entity IRenderable

2.6.5.1 Fields

# double x	X location of this entity
# double y	Y location of this entity
# int z	Depth of the entity on the screen
# boolean visible	Visible status
# boolean destroyed	Destroyed status

2.6.5.2 Constructor

# Entity()	Set visible to true and destroyed to false

2.6.5.3 Methods

# boolean isDestroyed	Return destroyed
+ boolean IsVisible()	Return visible
+ int getZ()	Return z

2.6.6 Class Field implements IRenderable

2.6.6.1 Fields

+ static int[][] field	Set to int[30][20]

2.6.6.3 Methods

+ static double getPositionX(int column)	Return clumn*25+12.5
+ static double getPositionY(int row)	Return row*25+12.5

+ int getTerrain(int row , int column)	If row is between 1 to 75,column is between 1 to 50 return field[row][column]
+ boolean isDestroyed()	Return false
+ boolean isVisible()	Return true
+ int getZ()	Return -999
+ void draw(GraphicsContext gc)	Initialize and draw field image, if getWait() is true draw text that show current wave ,if tower button was clicked and tower hasn't been placed yet draw tower image that follow mouse cursor

2.6.7 Class FireBullet extends Bullet

As 'IceBullet' but no freeze attack and slower.

Change the circle color to orange and stroke it with red.

2.6.8 Class FireTower extends Tower

2.6.8.1 Constructor

+ FireTower(int row,int column)	Set price to 80, damage to 3, splashAttack to true, range to 65, fireRate to 15, fireCount to 0 and the others field as 'IceTower'

2.6.8.2 Methods

- void Attack()	Attack 5 enemies in a row that are in the attack range by creating fireBullet.
Other methods are like 'IceTower'	

2.6.9 Class GameLogic

2.6.9.1 Fields

+ static List<Entity> getObjectContainer	All entities that need to be updated.
- int Spawner spawner	Spawner

- Gem gem	Gem
+ static boolean wait	Wait status
+ static boolean end	End status
+ static int wave	Current wave

2.6.9.2 Constructor

+ GameLogic	Initialize gameObjectContainer ,field Gem ,spawner , then add them to instance,add gem with spawner to gameObjectContainer

2.6.9.3 Method

+ void removeDestroyed()	Remove every game object that is destroyed
+ void addNewObject(Entity entity)	Add entity to gameObjectContainer and instance
+ static List<Enemy> getEnemy()	Return enemyList that contain every current enemy
+ static List<Tower> getTower()	Return towerList that contain every current tower
+ List<Bullet> getBullet()	Return bulletList that contain every current bullet
+ static boolean getWait()	Return wait
+ static int getWave()	Return wave
+ void logicUpdate()	Remove all destroyed items from the list. Call every items that need to be updated.Check if the gem hp is 0, the game is over.

2.6.10 Class Gem extends CollidableEntity

2.6.10.1 Fields

- int hp	Current hitpoint of the gem.

- int maxHp	Maximum hitpoint of the gem.

2.6.10.2 Constructor

+ Gem(int row,int column)	Set the gem to the specify location. Set radius to 12 Set hp and maxHp to 20 Set z to -100

2.6.10.3 Method

+ void update()	If the enemy collides with the gem, destroy the enemy and decrease it hp by 1.
+ void draw(GraphicsContext gc)	Draw the hp bar for the gem and draw the gem sprite.
+ int getHp()	Return hp of the gem.

2.6.11 Class IceBullet extends Bullet

2.6.11.1 Fields

- int duration	Duration of freezing.
- int level	Level of freezing.

2.6.11.2 Constructor

+ IceBullet(double x , double y ,Enemy lockedEnemy,int damage,int duration, int level)	Initialize damage ,duration,level,x,y,lockedEnemy,set radius to 15,set speedx and speedy to 0.

2.6.11.3 Method

+ void draw(GraphicsContext gc)	Draw the light blue circle and stroke it with royal blue.
+ void misile(Enemy enemy)	Calculate speedx and speedy ,if the enemy is within the radius,the enemy'll get hit then destroy this

	and set freeze to the enemy.
+ void update()	Run misile method

2.6.12 Class IceTower extends Tower

2.6.12.1 Constructor

+ IceTower(int row,int column)	Set price to 80, damage to 1, splashAttack to true, range to 55, fireRate to 15, fireCount to 0, radius to 10, row to row, column to column,x,y, angle to 0, isDrew to false,eventAdded to false, level to 0, isSelected to false.

2.6.12.2 Fields

+ int duration	Freeze duration.

2.6.12.3 Methods

- void attack()	Attack all the enemies that are in the attack range by creating IceBullet.
+ void update()	If fireCount is more than fireRate call the attack function and then reset fireCount to 0. Increase fireCount by 1 every frame.
+ void draw(GraphicsContext gc)	If the tower is selected, draw the tower's attack range.
+ ImageView drawImageView()	Initialize 'tower' at the specify location. Set isDrew to true.
+ void changeSprite()	Change the tower sprite by it level.
+ void move()	Relocate and rotate the tower.
+ void destroy()	Set the 'tower' image to null. Set 'destroyed' to true.
+ ImageView getImageView()	Return 'tower'

2.6.13 Class SniperBullet extends Bullet

As ‘IceBullet’ but no freeze attack and faster.

Change the circle color to royalblue and stroke it with lightblue

2.6.14 Class SniperTower extends Tower

2.6.14.1 Constructor

+ SniperTower(int row,int column)	Set price to 100, damage to 50, splashAttack to false, range to 100, fireRate to 45, fireCount to 0 and the others field as ‘IceTower’

2.6.14.2 Methods

- void Attack()	Attack the nearest enemies that is in the attack range by creating SniperBullet.
Other methods are like ‘IceTower’	

2.6.15 Class Spawner extends Entity

2.6.15.1 Fields

- int radius	Radius of the spawner
- int wave	Current wave number
- int time	Time to wait until spawn next enemy

2.6.15.2 Constructor

+ Spawner (int row, int column)	Set this radius to 12 ,wave to 1,time to 150 ,set x equal to getPositionX(column) and y equal to getPositionY(row)

2.6.15.3 Method

+ void draw(GraphicsContext gc)	Draw the light blue circle and stroke it with royal blue.
+ int getTime()	Calculate speedx and speedy ,if the enemy is within the radius,the enemy'll get hit then destroy this and set freeze to the enemy.
+ nextWave()	Go to next wave
+ void setTime(int time)	Run the missile method
+ int getWave()	Return current wave
+ boolean update()	If there is no enemy run nextWave and return true ,else return false
+ List<Pair<Enemy,Integer>> WaveList()	Set time how fast enemy will spawn And add pair that first equal to enemy type and second equal to amount of the enemy to Wavelist then return Wavelist
+ class Pair<A,B>	Class pair to contain type of the enemy and amount of the enemy

2.6.16 Abstract Class Tower extends Entity

2.6.16.1 Fields

# int row,column	Location of the tower in the cell.
# ImageView tower	Sprite of the tower.
# int level	Level of the tower.
# int price	Price of the tower.
# int radius	Radius of the sprite.
# int range	Tower's attack range.
# int damage	Tower's damage.
# int fireRate	The higher fireRate the slower the tower attack.

# int fireCounter	Time counter for attack.
# boolean eventAdded	True if the event is added to the tower.
# boolean splashAttack	True if the tower can attack multiple enemies at a time.
# boolean isDrew	True if the tower's ImageView is drawn on the screen.
# boolean isSelected	True if the tower is selected by the player.
# double angle	Tower's rotation.

2.6.16.2 Methods

# int inRange(Enemy enemy)	Return the distance between the enemy and the tower.
# boolean isInRange(Enemy enemy)	True if the enemy is in the tower's attack range.
+ int getFireRate()	Return by Math.round(((1/(double)fireRate)) * 1000.0) / 100.0;
+ int getUpgradePrice()	Return the upgrade price by the tower level. If the tower's level is 0 return price*1.5. If the tower's level is 1 return price*2.5.
+ int upgrade()	Upgrade the tower then decrease the player gold. Return 0 if the gold is not enough.
+ void deSelect()	Deselect all the towers in the list and set effect to null.
+ void selected()	Deselect all the towers. Set this tower to be selected. And set the effect to the ImageView.
+ abstract void update()	Update the tower logic.

+ abstract ImageView drawImageView()	Return the tower's ImageView.
+ abstract void move()	Move or rotate the tower.
+ abstract void destroy()	Destroy the tower.
+ abstract ImageView getImageView()	Return the tower's ImageView.
+ abstract void changeSprite()	Change the tower's sprite.
+ Getters and Setters	

2.7 Package : menu

2.7.1 Class BottomMenu extends StackPane

2.7.1.1 Fields

- GraphicsContext gcStatus	For drawing status text
- Thread animationThread	Current wave number
- boolean buttonAdded	True if the button in the button menu is added.
- ImageView upgradeButton	Sprite of the upgrade button
- ImageView sellButton	Sprite of the sell button

2.7.1.2 Constructor

+ BottomMenu(double width,double height)	Create a new thread for status menu update it canvas.

2.7.1.3 Method

+ void drawStatus()	Draw the light blue circle and stroke it with royal blue.
+ void printStatus(Tower selected)	Draw status text of the selected tower

2.7.2 Class TowerMenu extends Pane

2.7.1.1 Constructor

+ TowerMenu(int width , int height)	Add all buy tower buttons and add events for them

2.8 Package : sharedObject

2.8.1 Interface Class IRenderable

2.8.1.1 Method

+ ubt getZ();	Return z
+ void draw(GraphicsContext gc)	To draw and update it animation.
+ boolean isDestroyed()	True if the entity was destroyed.
+ boolean isVisible()	True if the entity is visible.

2.8.2 Class RenderableHolder

2.8.2.1 Fields

- static final RenderableHolder instance	Instance of the RenderableHolder.
- List<IRenderable> entities	All entities that need to be drawn on the screen.
- Comparator<IRenderable> comparator	Entity's comparator
+ static Image mapSprite	Image of the map
+ static Image bottomMenuSprite	Bottom menu image
+ static Image towerMenuSprite	Tower menu image
+ static Image arrowButton	Arrow tower button image
+ static Image sniperButton	Sniper tower button image
+ static Image iceButton	Ice tower button image
+ static Image fireButton	Fire tower button image

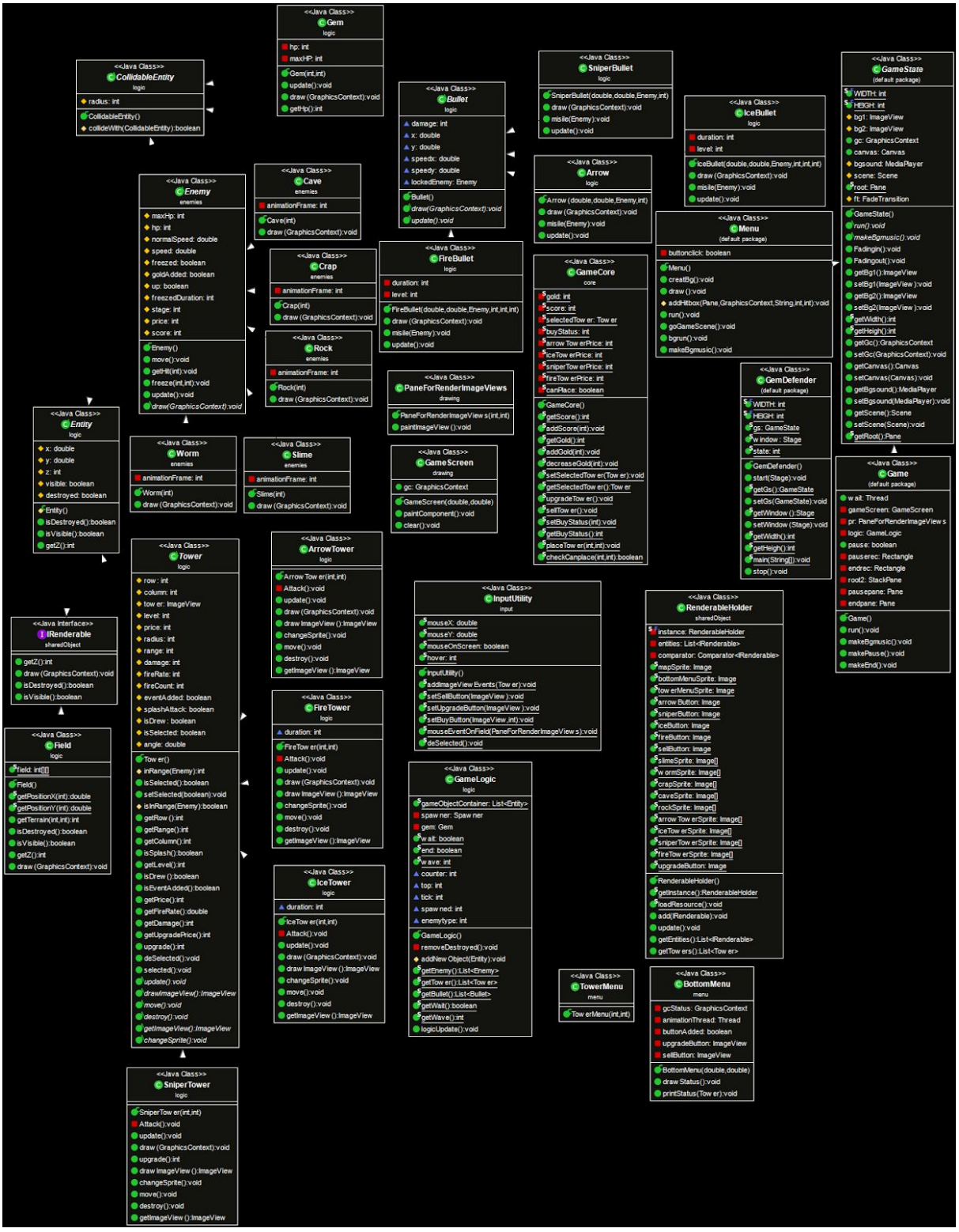
+ static Image sellButton	Sell button image
+ static Image[] slimeSprite	Slime image for each frame
+ static Image[] wormSprite	Worm image for each frame
+ static Image[] crapSprite	Crap image for each frame
+ static Image[] caveSprite	Cave image for each frame
+ static Image[] rockSprite	Rock image for each frame
+ static Image[] arrowTowerSprite	Arrow tower image for each level
+ static Image[] iceTowerSprite	Ice tower image for each level
+ static Image[] sniperTowerSprite	Sniper tower image for each level
+ static Image[] fireTowerSprite	Fire tower image for each level
+ static Image upgradeButton	Upgrade button image

2.8.2.2 Constructor

+ RenderableHolder()	Initialize entities and comparator

2.8.2.3 Method

+ static RenderableHolder getInstance()	Return the instance
+ static void loadRecsource()	Initialize all images by using ClassLoader
+ void add(IRenderable entity)	Add entity to entities then sort the whole list
+ void update()	Remove entities that were destroyed
+ List<IRenderable> getEntities()	Return all entities
+ List<Tower> getTowers()	Initialize towerList ,add every tower in towerList then return towerList


Gem
hp: int
maxHP: int
Gem(int,int)
update():void
draw (GraphicsContext):void
getHp():int
CollidableEntity
radius: int
CollidableEntity()
collideWith(CollidableEntity):boolean
Bullet
damage: int
x: double
y: double
speedx: double
speedy: double
lockedEnemy: Enemy
Bullet()
draw(GraphicsContext):void
update():void
SniperBullet
SniperBullet(double,double,Enemy,int)
draw (GraphicsContext):void
misile(Enemy):void
update():void
IceBullet
duration: int
level: int
IceBullet(double,double,Enemy,int,int,int)
draw (GraphicsContext):void
misile(Enemy):void
update():void
GameState
(default package)
WIDTH: int
HEIGH: int
bg1: ImageView
bg2: ImageView
gc: GraphicsContext
canvas: Canvas
bgsound: MediaPlayer
scene: Scene
root: Pane
ft: FadeTransition
GameState()
run():void
makeBgmusic():void
Fadingin():void
Fadingout():void
getBg1():ImageView
setBg1(ImageView):void
getBg2():ImageView
setBg2(ImageView):void
getWidth():int
getHeigh():int
getGc():GraphicsContext
setGc(GraphicsContext):void
getCanvas():Canvas
setCanvas(Canvas):void
getBgsound():MediaPlayer
setBgsound(MediaPlayer):void
getScene():Scene
setScene(Scene):void
getRoot():Pane
Arrow
logic
Arrow (double,double,Enemy,int)
draw (GraphicsContext):void
misile(Enemy):void
update():void
Menu
(default package)
buttonclick: boolean
Menu()
creatBg():void
draw ():void
addHitbox(Pane,GraphicsContext,String,int,int):void
run():void
goGameScene():void
bgrun():void
makeBgmusic():void
Enemy
enemies
maxHp: int
hp: int
normalSpeed: double
speed: double
freezed: boolean
goldAdded: boolean
up: boolean
freezedDuration: int
stage: int
price: int
score: int
Enemy()
move():void
getHit(int):void
freeze(int,int):void
update():void
draw(GraphicsContext):void
Cave
enemies
animationFrame: int
Cave(int)
draw (GraphicsContext):void
Crap
enemies
animationFrame: int
Crap(int)
draw (GraphicsContext):void
Rock
enemies
animationFrame: int
Rock(int)
draw (GraphicsContext):void
FireBullet
logic
duration: int
level: int
FireBullet(double,double,Enemy,int,int,int)
draw (GraphicsContext):void
misile(Enemy):void
update():void
GameCore
core
gold: int
score: int
selectedTower: Tower
buyStatus: int
arrowTowerPrice: int
iceTowerPrice: int
sniperTowerPrice: int
fireTowerPrice: int
canPlace: boolean
GameCore()
getScore():int
addScore(int):void
getGold():int
addGold(int):void
decreaseGold(int):void
setSelectedTower(Tower):void
getSelectedTower():Tower
upgradeTower():void
sellTower():void
setBuyStatus(int):void
getBuyStatus():int
placeTower(int,int):void
checkCanplace(int,int):boolean
GemDefender
(default package)
WIDTH: int
HEIGH: int
gs: GameState
window : Stage
state: int
GemDefender()
start(Stage):void
getGs():GameState
setGs(GameState):void
getWindow ():Stage
setWindow (Stage):void
getWidth():int
getHeigh():int
main(String[]):void
stop():void
Game
(default package)
wait: Thread
gameScreen: GameScreen
pr: PaneForRenderImageViews
logic: GameLogic
pause: boolean
pauserec: Rectangle
endrec: Rectangle
root2: StackPane
pausepane: Pane
endpane: Pane
Game()
run():void
makeBgmusic():void
makePause():void
makeEnd():void
PaneForRenderImageViews
drawing
PaneForRenderImageViews(int,int)
paintImageView ():void
GameScreen
drawing
gc: GraphicsContext
GameScreen(double,double)
paintComponent():void
clear():void
Entity
logic
x: double
y: double
z: int
visible: boolean
destroyed: boolean
Entity()
isDestroyed():boolean
isVisible():boolean
getZ():int
Worm
enemies
animationFrame: int
Worm(int)
draw (GraphicsContext):void
Slime
enemies
animationFrame: int
Slime(int)
draw (GraphicsContext):void
Tower
logic
row : int
column: int
tower: ImageView
level: int
price: int
radius: int
range: int
damage: int
fireRate: int
fireCount: int
eventAdded: boolean
splashAttack: boolean
isDrew : boolean
isSelected: boolean
angle: double
Tower()
inRange(Enemy):int
isSelected():boolean
setSelected(boolean):void
isInRange(Enemy):boolean
getRow ():int
getRange():int
getColumn():int
isSplash():boolean
getLevel():int
isDrew ():boolean
isEventAdded():boolean
getPrice():int
getFireRate():double
getDamage():int
getUpgradePrice():int
upgrade():int
deSelected():void
selected():void
update():void
drawImageView():ImageView
move():void
destroy():void
getImageView():ImageView
changeSprite():void
ArrowTower
logic
ArrowTower(int,int)
Attack():void
update():void
draw (GraphicsContext):void
draw ImageView ():ImageView
changeSprite():void
move():void
destroy():void
getImageView ():ImageView
FireTower
logic
duration: int
FireTower(int,int)
Attack():void
update():void
draw (GraphicsContext):void
draw ImageView ():ImageView
changeSprite():void
move():void
destroy():void
getImageView ():ImageView
IceTower
logic
duration: int
IceTower(int,int)
Attack():void
update():void
draw (GraphicsContext):void
draw ImageView ():ImageView
changeSprite():void
move():void
destroy():void
getImageView ():ImageView
SniperTower
logic
SniperTower(int,int)
Attack():void
update():void
draw (GraphicsContext):void
upgrade():int
draw ImageView ():ImageView
changeSprite():void
move():void
destroy():void
getImageView ():ImageView
InputUtility
input
mouseX: double
mouseY: double
mouseOnScreen: boolean
hover: int
InputUtility()
addImageView Events(Tower):void
setSellButton(ImageView):void
setUpgradeButton(ImageView):void
setBuyButton(ImageView ,int):void
mouseEventOnField(PaneForRenderImageViews):void
deSelected():void
GameLogic
logic
gameObjectContainer: List<Entity>
spawner: Spawner
gem: Gem
wait: boolean
end: boolean
wave: int
counter: int
top: int
tick: int
spawned: int
enemytype: int
GameLogic()
removeDestroyed():void
addNew Object(Entity):void
getEnemy():List<Enemy>
getTower():List<Tower>
getBullet():List<Bullet>
getWall():boolean
getWave():int
logicUpdate():void
RenderableHolder
sharedObject
instance: RenderableHolder
entities: List<IRenderable>
comparator: Comparator<IRenderable>
mapSprite: Image
bottomMenuSprite: Image
towerMenuSprite: Image
arrowButton: Image
sniperButton: Image
iceButton: Image
fireButton: Image
sellButton: Image
slimeSprite: Image[]
wormSprite: Image[]
crapSprite: Image[]
caveSprite: Image[]
rockSprite: Image[]
arrowTowerSprite: Image[]
iceTowerSprite: Image[]
sniperTowerSprite: Image[]
fireTowerSprite: Image[]
upgradeButton: Image
RenderableHolder()
getInstance():RenderableHolder
loadResource():void
add(IRenderable):void
update():void
getEntities():List<IRenderable>
getTowers():List<Tower>
IRenderable
sharedObject
getZ():int
draw (GraphicsContext):void
isDestroyed():boolean
isVisible():boolean
Field
logic
field: int[][]
Field()
getPositionX(int):double
getPositionY(int):double
getTerrain(int,int):int
isDestroyed():boolean
isVisible():boolean
getZ():int
draw (GraphicsContext):void
TowerMenu
menu
TowerMenu(int,int)
BottomMenu
menu
gcStatus: GraphicsContext
animationThread: Thread
buttonAdded: boolean
upgradeButton: ImageView
sellButton: ImageView
BottomMenu(double,double)
drawStatus():void
printStatus(Tower):void