Module Interface Specification

SFWRENG 3XA3

Group 30
Alan Yin (yins1)
Huajie Zhu (zhuh5)
Junni Pan (panj10)

November 09, 2017

Contents

List of Tables

List of Figures

1 Major Revision History

Date	Revision
November 6, 2017	Rough draft of sections
November 7, 2017	Revised sections
November 9, 2017	Revision 0 complete
December 6, 2017	Revision 1 complete

Table 1: Major Revision History

2 Module Hierarchy

Level 1	Level 2
Hardware Hiding Module	
Behaviour Hiding Module	MainWindow AreaSelector SelectAreaCaptureScreen Data Output
Software Decision Module	FrameCapture Point2D

Table 2: Module Hierarchy

3 MIS of Game Module

3.1 Interface Syntax

3.1.1 Exported Access Programs

Name	In	Out	Exceptions
main	args	-	SlickException

3.2 Interface Semantics

3.2.1 State Variables

Not Applicable

3.2.2 Environmental Variables

external environment

3.2.3 Assumptions

None

3.2.4 Access Program Semantics

main():
Input: args
Transition: Start the application, set the resolution to 1000*700
Output: None
Exceptions: SlickException

4 MIS of GameController Module

4.1 Interface Syntax

4.1.1 Exported Access Programs

Name	In	Out	Exceptions
GameController	TDMap	-	-
setPanelAndButtonProperties	-	-	-
setInitialValues	-	-	-
setMainFrame	-	-	-
startNewWave	-	-	-
paintComponent	Graphics	-	-
stateChanged	-	-	-
setPlaybackSpeed	-	-	-
doPause	-	-	-
doReturnToMainMenu	-	-	-
doStartWave	-	-	-
doSelectTower	ActionEvent	-	-
doUpgrade	-	-	-
doSell	-	-	-
doDisplayCritterInfo	-	-	-
actionPerformed	ActionEvent	-	-
Draw	-	-	-
observerUpdate	-	-	-
endGame	-	-	-
disableAllGameButtons	-	-	-
resetPlayerWaveStats	-	-	-
spendMoney	int	-	-
getControlPanel	-	GameControlPanel	-
getPlayPanel	-	MapPanel	-
updateTowerInfoText	-	-	-
reactToLeftClick	-	-	-
buildTower	Tower	-	-
updateSelectedTowerInfoAndButtons	-	-	-
reactToMouseMove	Point	-	-
reactToRightClick	Point	-	-
itemStateChanged	-	ItemEvent	-

4.2 Interface Semantics

4.2.1 State Variables

controlPanel: GameControlPanel - the game control panel
mainFrame: JFrame - the main frame of this game

bPause: JButton - a button to pause the game
bStartWave: JButton - abutton to start the wave
bUpgrade: JButton - a button to upgrade the selected tower
bSell: JButton - a button to sell the selected tower
jsSpeed: JSlides - a slide to chage the game speed
cbStrategies: JComboBox<String> - a list of strategies
bCritterInfo: JButton - a button to show the critter information
timer: Timer - the timer
gamePlayer: Player - the player
waveStartMoney: int - the wave start money
waveStartLives: int - the wave start lives
waveNumber: int - the wave number
activeCritterIndex: int - the index of active critter
drawableEntities: ArrayList<DrawableEntity> - a list of drawable entities
tdMap: TDMap - the map to use
crittersInWave: ArrayList<Critter> - a list of critter on map
towersOnMap: ArrayList<Tower> - a list of tower on map
gamePaused: boolean - is the game paused
gameOver: boolean - is the game over
selectedTowerToBuild: String - name of a tower to build
towerBeingPreviewed: Tower - a tower which is being previewed
selectedTower: Tower - a tower which is selected
selectedTile: Maptile - a maptile which is selected
artist: Artist_Swing - a artist helper
clock: GameClock - a clock helper
helpers: ArrayList<Helper> - a list of helpers
subjects: ArrayList<Subject> - a list of subjects

4.2.2 Environmental Variables

None

4.2.3 Assumptions

Variables should be set before trying to access them

4.2.4 Access Program Semantics

GameController():
Input: none
Transition: This takes a TDMap object as the map on which to play the game.
Output: None
Exceptions: None

setPanelAndButtonProperties():
Input: none
Transition: set this to be our game panel, get all of our Swing objects, and add action listener
Output: None
Exception: none

setInitialValues():
Transition: sets the initial values of the variables for the game. Also initializes arrays and gets the instances of the singleton classes.

Output: None
Exception: None

setMainFrame(mFrame):
Input: JFrame
Transition: sets the JFrame that the game is displayed on
Output: None
Exception: None

startNewWave():
Transition: start a new wave
Output: Return Value that was accessed (yReleased)
Exception: none

paintComponent(g):
Input: Graphics
Transition: update and draw all drawableEntities.
Output:
Exception: none

stateChanged(e):
Input: ChangeEvent
Transition: set the game speed
Output:
Exception: none

setPlaybackSpeed():
Transition: This relates to how fast or slow the wave apparently appears to the player.
Output: None
Exception: none

doPause():
Transition: pause the game
Output: None
Exception: none

doReturnToMainMenu():
Transition: return ti the main menu
Output: None
Exception: none

doStartWave():
Transition: unpause and start the wave
Output: None
Exception: none

doSelectTower():
Transition: select the tower
Output: None
Exception: none

doUpgrade():
Transition: upgrade the power

Output: None
Exception: none

doSell():
Transition: sell the tower
Output: None
Exception: none

doDisplayCritterInfo():
Transition: display the critter information
Output: None
Exception: none

doDisplayCritterInfo(arg0):
Input: ActionEvent
Transition: response to the acquired action
Output: None
Exception: none

Draw():
Transition: call the repaint method
Output: None
Exception: none

observerUpdate():
Transition: This will update the game whenever one of the subjects (Critters)of the Game Controller is changed. e.g. if a critter dies or a tower is upgraded.
Output: None
Exception: none

endGame():
Transition: Ends the game by disabling buttons, and pausing the clock.
Output: None
Exception: none

disableAllGameButtons():
Transition: disables all of the game buttons
Output: None
Exception: none

resetPlayerWaveStats():
Transition: resets the player's stats (so that a new game can be started with the same instance)
Output: None
Exception: none

spendMoney():
Transition: spends a certain amount of money of the Player
Output: None
Exception: none

getControlPanel():
Transition: return the controlPanel
Output: GameControlPanel

Exception: none

getPlayPanel():
Transition: return the gamePanel
Output: MapPanel
Exception: none

updateInfoLabelText():
Transition: updates the info text
Output: None
Exception: none

updateTowerInfoText():
Transition: updates the tower info text
Output: None
Exception: none

reactToLeftClick(point):
Input: Point
Transition: This method selects an existing tower to upgrade it, or puts a new tower on the selected tile, of the desired type. Hence, react to left click
Output: None
Exception: none

buildTower(t):
Input: Tower
Transition: builds a tower t and puts it in the drawable entities to be drawn.
Output: None
Exception: none

updateSelectedTowerInfoAndButtons():
Transition: enable upgrades if they have enough money and if the tower isn't at max level, and enable the sell button.
Output: None
Exception: none

reactToMouseMove(point):
Input: Point
Transition: react to the mouse move
Output: None
Exception: none

reactToRightClick(point):
Input: Point
Transition: a right click clears the current tower selection
Output: None
Exception: none

itemStateChanged(e):
Input: ItemEvent
Transition: if our strategies combobox changed, we want to change the strategy of the selected tower
Output: None

Exception: none

5 MIS of ArtistSwing Module

5.1 Interface Syntax

5.1.1 Exported Constants

PIXELWIDTH=1000
PIXELHEIGHT=700
GAMEPIXELHEIGHT = PIXELHEIGHT-100

5.1.2 Exported Access Programs

Name	In	Out	Exceptions
setGridWidth	int	-	-
setGridHeight	int	-	-
drawEmptyCircle	Graphics,color,int	-	-
drawFilledCircle	Graphics,color,int	-	-
drawFilledQuad	Graphics,color,int	-	-
drawEmptyQuad	Graphics,color,int	-	-
drawMap	TDMap, Graphics	-	-
drawCritter	Critter,Graphics	-	-
drawTower	Tower,Graphics	-	-
drawShot	Tower,Critter,Graphics	-	-

5.2 Interface Semantics

5.2.1 State Variables

width: int
height: int
g: graphics
c: color
x: int
y: int
radius: int
tdMap: TDMap
crit: Critter
tow: Tower

5.2.2 Access Program Semantics

setGridWidth(width):
Input: int
Transition: set gridwidth to width
Exception - None

setGridHeight(Height):

Input: int
Transition: set gridheight to height
Exception - none

drawEmptyCircle(g, c, x, y, radius):
Input: Graphics, Color, int, int, int
Transition - Sets the color, and draws a circle (oval with equal radii)
Exception - None

drawFilledCircle(g, c, x, y, radius):
Input: Graphics, Color, int, int, int
Transition - sets the color, and draws a filled circle (oval with radii equal)
Exception - None

drawFilledQuad(g, c, x, y, radius):
Input: Graphics, Color, int, int, int
Transition - sets the color and draws the rectangle
Exception - None

drawEmptyQuad(g, c, x, y, radius):
Input: Graphics, Color, int, int, int
Transition - sets the color and draws the empty rectangle
Exception - None

drawMap(tdMap,g):
Input: TDMap, Graphics
Transition - draws the map, gets the width and the height, finds the width and height of each block, sets the thickness of the line to 1, goes through the map in a nested for loop, if we have a path tile, draw it brown, if we have a scenery tile, draw it green.
Exception - None

drawCritter(crit, g):
Input: Critter, Graphics
Transition - gets the critter attribuets, drawing the space behind the critter, Drawing the actual critter, replace critter with image, replace critter with image, for Healthbar: we draw a green rectangle, and then draw a red rectangle of size depending on how damaged. we draw a green rectangle, and then draw a red rectangle of size depending on how damaged. supposing the critter is damaged , we draw the red part.
Exception - None

drawTower(tow,g):
Input: Tower, Graphics
Transition - sets our stroke to be size 1, gets the tile width and height of the gamemap, our outline tower is either black or blue (blue if selected), we draw the tower's rectangular part, and the outline and then we draw the tower's circular part, replace tower with image. for upgrades, we draw a circle (in white) around the main circle of the tower for each upgrade level 16 since max tower level is 4. (so the circle doesn't go out of bounds).
Exception - None

drawShot:(tow,crit,g):
Input: Tower, Critter, Graphics

Transition - gets the tile width and height, get tower color info, we set the stroke to be thicker than usually (2) and we draw the line.
Exception - None

6 MIS of CritterGenerator Module

6.1 Interface Syntax

6.1.1 Exported Access Programs

Name	In	Out	Exceptions
getGeneratedCritterWave	int, TDMap	ArrayList<Critter>	-

6.2 Interface Semantics

6.2.1 State Variables

BASECRITTERS: int - the base amount of critters
MAXWAVE: int - THE maximum wave number

6.2.2 Environmental Variables

None

6.2.3 Assumptions

None

6.2.4 Access Program Semantics

getGeneratedCritterWave(wavelevel, exampleMap):
Input: int, TDMap
Transition: generates a group of critters for a certain wave number. IF it is a multiple of 5, we do a boss round, with boss (infinity) and grouped (shuriken) critters.
Exception: None

7 MIS of Gameclock Module

7.1 Interface Syntax

7.1.1 Exported Access Programs

Name	In	Out	Exceptions
GameClock	-	-	-
getInstance	-	GameClock	-
deltaTime	-	int	-
setDeltaTime	int	-	-
pause	-	-	-
unPause	-	-	-

7.2 Interface Semantics

7.2.1 State Variables

dTime: int
dt: int
clock: GameClock

7.2.2 Access Program Semantics

GameClock():
    Input: None
    Transition: default tick is 1 second
    Exception: None

getInstance():
    Input: None
    Transition: none
    Output: return the instance (OF WHICH THERE IS ONLY 1) of the clock
    Exception:

deltaTime():
    Input: None
    Transition: None
    Output: return deltaTime
    Exception: none

setDeltaTime(dt):
    Input: int
    Transition: set deltaTime to dt
    Exception: none

pause():
    Input: none
    Transition: pause by setting deltaTime to 0
    Exception: none

unPause():
    Input: none
    Transition: unpause by setting deltaTime to 1
    Exception: none

8 MIS of MouseAndKeyboardHandler Module

8.1 Interface Syntax

8.1.1 Exported Access Programs

Name	In	Out	Exceptions
MouseAndKeyboardHandler	GameController	-	-
mouseClicked	MouseEvent	-	-
mousePressed	MouseEvent	-	-
mouseReleased	MouseEvent	-	-
mouseEntered	MouseEvent	-	-
mouseExited	MouseEvent	-	-
mouseMoved	MouseEvent	-	-
mouseDragged	MouseEvent	-	-

8.2 Interface Semantics

8.2.1 State Variables

gameController: GameController - the gameController that we are helping

8.2.2 Environmental Variables

external environment

8.2.3 Assumptions

None

8.2.4 Access Program Semantics

MouseAndKeyboardHandler(gameController):
Input: GameController
Transition: set the gameController to help
Exception: None

mouseClicked(event):
Input: MouseEvent
Transition: on mouse click, we alert the game controller
Exception: None

mousePressed(event):
Input: MouseEvent
Transition: on mouse prresed, we alert the game controller
Exception: None

mouseReleased(event):
Input: MouseEvent
Transition: on mouse released, we alert the game controller
Exception: None

mouseEntered(event):
Input: MouseEvent
Transition: on mouse entered, we alert the game controller

Exception: None

mouseExited(event):
Input: MouseEvent
Transition: on mouse exited, we alert the game controller
Exception: None

mouseMoved(event):
Input: MouseEvent
Transition: let the game controller know if the mouse is moved
Exception: None

mouseDragged(event):
Input: MouseEvent
Transition: let the game controller know if the mouse is dragged
Exception: None

9 MIS of Critter Module

9.1 Interface Syntax

9.1.1 Exported Constants

MAXCRITTERLEVEL = 50
MAXSPEED = 15
CRITTERMESSAGE = "Below is a description of each of the colored critters." + "Yellow:˜Critter. Very hard to kill" + "White:˜but weak" + "Red:˜below average" + "Pink:˜but slow" + "Orange:˜resistant to fire and slow" + "Cyan:˜Critter";

9.1.2 Exported Access Programs

9.2 Interface Semantics

9.2.1 State Variables

currHitPoints: double
maxHitPoints: double
speed: double
size: int
regen: int
resistance: double
cColor: Color
reward: int
level: int
name: string
slowFactor: double
slowTime: int
image: Image
beenSlowedFor: int
damageOverTimeVal: double
dotTime: int
beenDOTFor: int
burning: boolean

Name	In	Out	Exceptions
critter	int,TDMap	-	-
setInitialValues	-	-	-
calculateLevelMultiplier	-	-	-
getIndexInPixelPath	-	double	-
getListPixelPath	-	ArrayList<Point>	-
setSlowFactor	double	-	-
getColor	-	Color	-
getPixelPosition	-	point	-
hasReachedEnd	-	boolean	-
isAlive	-	boolean	-
isBurning	-	boolean	-
getSize	-	int	-
getLoot	-	int	-
getImage	-	image	-
setHitboxRadius	int	-	-
getHitPoints	-	double	-
getMaxHitPoints	double	-	-
getRawSpeed	-	double	-
setRawSpeed	int	-	-
getLevel	-	int	-
setLevel	int	-	-
isActive	-	boolean	-
setActive	boolean	-	-
getSpeed	-	double	-
updateAndDraw	graphics	-	-
updateHealth	-	-	-
updatePositionAndDraq	graphics	-	-
moveAndDrawCritter	int,graphics	-	-
drawCritter	graphics	-	-
damage	double	-	-
slowCritter	int,double	-	-
damageOverTimeCritter	int,double	-	-

pixelPosition: Point
active: boolean
alive: boolean
reachedEnd: boolean
pixelPathToFollow: ArrayList<Point>
indexInPixelPath: double
intIndexInPixelPath: int
g: graphics

9.2.2 Access Program Semantics

critter(level, m):
Input: int, TDMap
Transition: set the level from input, sets the size to scale with the grid size (bigger blocks = bigger critters), sets the initial values of the critter attributes.
Exception - None

setInitialValues():
Input: none
Transition: sets the initial values of the critter attributes.
Exception - none

calculateLevelMultiplier():
Input: none
Transition - calculates the current level multiplier of the critter,This will be called by extending critters, usually
Exception - None

getIndexInPixelPath():
Input: none
Transition: none
Output: return indexInPixelPath
Exception - None

getListPixelPath():
Input: none
Transition: none
Output: return pixelPathToFollow
Exception - None

setSlowFactor(slowFactor):
Input: double
Transition - sets slow factor
Exception - None

setDOTAmount(dot):
Input: double
Transition: set up amount of dot
Exception - None

getColor():
Input: none

Transition: none
Output: return cColor
Exception - None

getPixelPosition():
Input: none
Transition: none
Output: return pixelPosition
Exception - None

hasReachedEnd():
Input: none
Transition: none
Output: return reachedEnd
Exception - None

isAlive():
Input: none
Transition: none
Output: return alive
Exception - None

isBurning():
Input: none
Transition: none
Output: return reachedEnd
Exception - None

getSize():
Input: none
Transition: none
Output: return size
Exception - None

getLoot():
Input: none
Transition: none
Output: return reward
Exception - None

getImage():
Input: none
Transition: none
Output: return image
Exception - None

setHitboxRadius(size):
Input:int
Transition: set size
Exception - None

getHitPoints():
Input: none
Transition: none

Output: return currHitPoints
Exception - None

getMaxHitPoints():
Input: none
Transition: none
Output: return maxHitPoints
Exception - None

getRawSpeed():
Input: none
Transition: none
Output: return speed
Exception - None

setRawSpeed(speed):
Input:int
Transition: set speed
Exception - None

getLevel():
Input: none
Transition: none
Output: return level
Exception - None

setLevel(level):
Input:int
Transition: set level
Exception - None

isActive():
Input: none
Transition: none
Output: return active
Exception - None

setActive(act):
Input:int
Transition: set active
Exception - None

getSpeed():
Input: none
Transition: none
Output: return speed
Exception - None

updateAndDraw(g):
Input: graphics
Transition: we only want to do something if the critter is active. See if we are being slowed, if so, tick the total amount of time we have been slowed for. See if we are being damaged over time, if so, tick the time we have been

DOT for. update the health of the critter,update the position of the critter and draw it.
Exception - None

updateHealth():
Input: none
Transition - updates the health of the critter (called on every "tick" of the clock)
Exception - None

updatePositionAndDraw(g):
Input: Graphics
Transition - updates the position (and draws it), called on every tick of clock
Exception - None

moveAndDrawCritter(index,g):
Input: int, Graphics
Transition - Moves the critter to a given position and draws it as it moves.
Exception - None

moveAndDrawCritter(g):
Input: Graphics
Transition - draws the critter using the artist class
Exception - None

damage(dam):
Input: double
Transition - Damages the critter for a certain amount
Exception - None

slowCritters(Factor, sTime):
Input: double, int
Transition - set the slow factor and slow time
Exception - None

damageOverTimeCritter(Factor, sTime):
Input: double, int
Transition - set the damage over time factor and time
Exception - None

10 MIS of Tower Module

10.1 Interface Syntax

10.1.1 Exported Access Programs

Name	In	Out	Exceptions
Tower	String, Point, ArrayList<Critter>	-	-
getSellPrice	-	int	-
getUpPrice	-	int	-
setStrategy	IStrategy	-	-
getPosX	-	int	-
getPosY	-	int	-
getRange	-	int	-
getName	-	String	-
getImage	-	Image	-
getEnabled	-	boolean	-
setEnabled	boolean	-	-
getColor	-	Color	-
isSelected	-	boolean	-
getStrategy	-	IStrategy	-
setSelected	boolean	-	-
getDefaultStrategy	-	String	-
setColor	Color	-	-
shootTarget	Criiter, Graphics	-	-
upgradeTower	-	-	-

10.2 Interface Semantics

10.2.1 State Variables

MAXTOWERLEVEL: int - the max level of a tower(4)
DEFAULTSTRATEGY: String - the default strategy to use("Closest")
position: Point - the position of the tower
damage: double - the damage of the tower
rateOfFire: int - the fire rate of a tower
range: int - the fire range of the tower
sellPrice: int - the sell price of the tower
upCost: int - the upgrade cost of the tower
name: String - name of the strategy
level: int - level of the tower
tColor: Color - color if the tower
shotColor: Color - shoot color of the tower
image: Image - model image of the tower
icon: ImageIcon - the icon of the image
strategy: IStrategy - the strategy to use
enabled: boolean - if it is enabled
selected: boolean - if it is selected

10.2.2 Environmental Variables

external environment

10.2.3 Assumptions

None

10.2.4 Access Program Semantics

Tower(n, p, crittersOnMap):
Input: String, Point, ArrayList<Critter>
Transition: constructor to construct a tower
Exception: None

getSellPrice():
Transition: return the sell price of the tower
Output: int
Exception: None

getUpPrice(event):
Transition: return the upgrade price of the tower
Output: int
Exception: None

setStrategy(strategy):
Input: IStrategy
Transition: set the strategy of the tower
Exception: None

getPosX():
Transition: return the x position of the tower
Output: int
Exception: None

getPosY():
Transition: return the y position of the tower
Output: int
Exception: None

getRange():
Transition: return the attack range of the tower
Output: int
Exception: None

getName():
Transition: return the name of the tower
Output: String
Exception: None

getImage():
Transition: return the image of the tower
Output: Image
Exception: None

getEnabled():
Transition: return if it is enabled
Output: boolean

Exception: None

setEnabled(state):
Input: boolean
Transition: set enabled or dis-enabled
Exception: None

getColor():
Transition: return the color of the tower
Output: Color
Exception: None

isSelected():
Transition: return if it is selected
Output: boolean
Exception: None

getStrategy():
Transition: return the strategy of the tower
Output: IStrategy
Exception: None

setSelected(s):
Input: boolean
Transition: set it is been selected or not
Exception: None

getDefaultStrategy():
Transition: return the name of default strategy of the tower
Output: String
Exception: None

setColor(newColor):
Input: Color
Transition: set the color of the tower
Exception: None

shootTarget(target, g):
Input: Criiter, Graphics
Transition: deals damage to the criiter based on amount of damage of the tower
Exception: None

upgradeTower():
Transition: upgrades the tower based on properties
Exception: None

11 MIS of TDMap Module

11.1 Interface Syntax

11.1.1 Exported Access Programs

Name	In	Out	Exceptions
TDMap	-	-	-
TDMap	String	-	-
initializeGrid	-	-	
getPIXELWIDTH	-	int	-
getPIXELHEIGHT	-	int	-
getGridWidth	-	int	-
getGridHeight	-	int	-
getPointsOfShortestPath	-	ArrayList<Point>	-
updateAndDraw	Graphics	-	-
print	-	-	-

11.2 Interface Semantics

11.2.1 State Variables

PIXELWIDTH: int - the pixelwidth
PIXELHEIGHT: int - the pixelheight
gridWidth: int - the gridthwidth
gridHeight: int - the gridthheight
shortestPath: LinkedList<Integer> - the shortest path of the map ;

11.2.2 Environmental Variables

None

11.2.3 Assumptions

None

11.2.4 Access Program Semantics

TDMap():
Transition: set route and wall image, the grid width and height as default
Exception: None

TDMap(add):
Input: String
Transition: load grass and wall of the map
Exception: None

initializeGrid():
Transition: initializes the gridTile array to be all new MapTile objects
Exception: None

getPIXELWIDTH():
Transition: return the PIXELWIDTH of the map
Output: int

Exception: None

getPIXELHEIGHT():
Transition: return the PIXELHEIGHT of the map
Output: int
Exception: None

getGridWidth():
Transition: return the GridWidth of the map
Output: int
Exception: None

getGridHeight():
Transition: return the GridHeight of the map
Output: int
Exception: None

getPointsOfShortestPath():
Transition: return the shortest path of the map
Output: ArrayList<Point>
Exception: None

updateAndDraw():
Transition: uses the artist to draw the map
Exception: None

print():
Transition: This method provides an easy way to print out the grid to display the map. It also prints out the shortest path the critters will take to move from the Start cell to the End Cell.
Exception: None

12 MIS of Point Module

12.1 Interface Syntax

12.1.1 Exported Access Programs

Name	In	Out	Exceptions
Point	integer, integer	-	-
getX	-	int	-
getY	-	int	-
setX	int	-	-
setY	int	-	-
setPoint	int, int	-	-
equals	Point	boolean	-

12.2 Interface Semantics

12.2.1 State Variables

X: int - x value of point
Y: int - y value of point

12.2.2 Environmental Variables

Not Applicable

12.2.3 Assumptions

Variables should be set before trying to access them

12.2.4 Access Program Semantics

setX(x):
Input: Integer of X value
Transition: updates the X value of the point
Output: none
Exception: none

setY(y): Input: Integer of Y value
Transition: updates the Y value of the point
Output: none Exception: none

getX():
Input: none
Transition: accesses the X value
Output: Returns the X value of the Point
Exception: none

getY():
Input: none
Transition: accesses the Y value
Output: Returns the Y value of the point
Exception: none

setPoint(x, y):
Input: int, int
Transition: set both coords of a point at once
Output: none
Exception: none

equals(p):
Input: Point
Transition: check if one point equals another
Output: boolean
Exception: none

13 MIS of Player Module

13.1 Interface Syntax

13.1.1 Exported Access Programs

Name	In	Out	Exceptions
Player	-	-	-
getInstance	-	Player	-
getLives	-	int	-
getMoney	-	int	-
setLives	int	-	-
setMoney	int	-	-
addToMoney	int	-	-
takeAwayALife	-	-	-
getStartingLives	-	int	-
getStartingMoney	-	int	-
resetStats	-	-	-

13.2 Interface Semantics

13.2.1 State Variables

STARTINGLIVES: int - the default starting live STARTINGMONEY: int - the default starting money lives: int - the current lives money: int - the current money playerInstance: Player - the player

13.2.2 Environmental Variables

external environment

13.2.3 Assumptions

Variables should be set before trying to access them

13.2.4 Access Program Semantics

Player():
Transition: The constructor
Output: none
Exception: none

getInstance():
Transition: return the playerInstance
Output: Player
Exception: none

getLives():
Transition: return the current lives
Output: int
Exception: none

getMoney():
Transition: return the current money
Output: int

Exception: none

setLives(l):
Input: int
Transition: set the value of lives
Exception: none

setMoney(m):
Input: int
Transition: set the value of money
Exception: none

addToMoney(moneyToAdd):
Input: int
Transition: add money to the player
Exception: none

takeAwayALife():
Transition: reduce the player's lives by one
Exception: none

getStartingLives():
Transition: return the default starting lives
Output: int
Exception: none

getStartingMoney():
Transition: return the default starting money
Output: Player
Exception: none

resetStats():
Transition: reset the stats of the player
Output: Player
Exception: none

14 MIS of Closest Module

14.1 Interface Syntax

14.1.1 Exported Access Programs

Name	In	Out	Exceptions
Critter findTarget	Tower, ArrayList<Critter>	Critter	-
toString	-	string	-

14.2 Interface Semantics

14.2.1 State Variables

tower: Tower
g1: ArrayList<Critter>

14.2.2 Environmental Variables

None

14.2.3 Assumptions

None

14.2.4 Access Program Semantics

findTarget(tower, g1):
Input: Tower, ArrayList<Critter>
Transition: finds the target based on who is closest, set arbitrary large number that will never be reached
Output: return closest enemy
Exception: None

toString():
Input: none
Transition: none
Output: return closest
Exception: None

15 MIS of Farthest Module

15.1 Interface Syntax

15.1.1 Exported Access Programs

Name	In	Out	Exceptions
Critter findTarget	Tower, ArrayList<Critter>	Critter	-
toString	-	string	-

15.2 Interface Semantics

15.2.1 State Variables

tower: Tower
g1: ArrayList<Critter>

15.2.2 Environmental Variables

None

15.2.3 Assumptions

None

15.2.4 Access Program Semantics

findTarget(tower, g1):
  Input: Tower, ArrayList<Critter>
  Transition: finds the Critter that is farthest along the path
  Output: return farthest enemy
  Exception: None

toString():
  Input: none
  Transition: none
  Output: return Farthest
  Exception: None

16 MIS of Fastest Module

16.1 Interface Syntax

16.1.1 Exported Access Programs

Name	In	Out	Exceptions
Critter findTarget	Tower, ArrayList<Critter>	Critter	-
toString	-	string	-

16.2 Interface Semantics

16.2.1 State Variables

tower: Tower
g1: ArrayList<Critter>

16.2.2 Environmental Variables

None

16.2.3 Assumptions

None

16.2.4 Access Program Semantics

findTarget(tower, g1):
  Input: Tower, ArrayList<Critter>
  Transition: finds target that is fastest
  Output: return fastest enemy
  Exception: None

toString():
  Input: none
  Transition: none
  Output: return Fastest

Exception: None

17 MIS of Strongest Module

17.1 Interface Syntax

17.1.1 Exported Access Programs

Name	In	Out	Exceptions
Critter findTarget	Tower, ArrayList<Critter>	Critter	-
toString	-	string	-

17.2 Interface Semantics

17.2.1 State Variables

tower: Tower
g1: ArrayList<Critter>

17.2.2 Environmental Variables

external environment

17.2.3 Assumptions

None

17.2.4 Access Program Semantics

findTarget(tower, g1):
Input: Tower, ArrayList<Critter>
Transition: finds the strongest enemy
Output: return Strongest enemy
Exception: None

toString():
Input: none
Transition: none
Output: return Strongest
Exception: None

18 MIS of Weakest Module

18.1 Interface Syntax

18.1.1 Exported Access Programs

Name	In	Out	Exceptions
Critter findTarget	Tower, ArrayList<Critter>	Critter	-
toString	-	string	-

18.2 Interface Semantics

18.2.1 State Variables

tower: Tower
g1: ArrayList<Critter>

18.2.2 Environmental Variables

None

18.2.3 Assumptions

None

18.2.4 Access Program Semantics

findTarget(tower, g1):
  Input: Tower, ArrayList<Critter>
  Transition: finds the Weakest enemy
  Output: return Weakest enemy
  Exception: None

toString():
  Input: none
  Transition: none
  Output: return Weakest
  Exception: None

19 MIS of GameApplicationFrame Module

19.1 Interface Syntax

19.1.1 Exported Constants

PIXELWIDTH=ArtistSwing.PIXELWIDTH
PIXELHEIGHT=ArtistSwing.PIXELHEIGHT
APPNAME = "Group 30 Tower Defense"
TIMEOUT = 30

19.1.2 Exported Access Programs

Name	In	Out	Exceptions
GameApplicationFrame	TDMap	-	-
init	-	-	-

19.2 Interface Semantics

19.2.1 State Variables

controlPanel: GameControlPanel
mapPanel: MapPanel

gameController: GameController
tdMap: TDMap

19.2.2 Environmental Variables

None

19.2.3 Assumptions

None

19.2.4 Access Program Semantics

GameApplicationFrame(tdMap):
Input: TDMap
Transition: set the tdmap
Exception: None

init():
Input: none
Transition: set the Frame properties, get the control and map panels, add them to the frame, set the x button as the default close operation
Exception: None

20 MIS of GameState Module

20.1 Interface Syntax

20.1.1 Exported Access Programs

Name	In	Out	Exceptions
init	GameContainer, StateBasedGame	-	-
render	GameContainer, StateBasedGame, Graphics	-	-
update	GameContainer, StateBasedGame, init	-	-
getID	-	int	-

20.2 Interface Semantics

20.2.1 State Variables

mapToLoad: TDMap
arg0: GameContainer
arg1: StateBasedGame
g: Graphics
delta: init

20.2.2 Environmental Variables

external environment

20.2.3 Assumptions

None

20.2.4 Access Program Semantics

init(arg0, arg1):
  Input: GameContainer, StateBasedGame
  Transition: set main menu image, set background image and set background music
  Exception: None

render(arg0, arg1, g):
  Input: GameContainer, StateBasedGame, Graphics
  Transition: draw string at interface
  Exception: None

update(arg0, arg1, delta):
  Input: GameContainer, StateBasedGame, init
  Transition: set default map
  Exception: None

getID():
  Input: none
  Transition: none
  Output: return 1
  Exception: None

21 MIS of MainMenu Module

21.1 Interface Syntax

21.1.1 Exported Access Programs

Name	In	Out	Exceptions
init	GameContainer, StateBasedGame	-	-
render	GameContainer, StateBasedGame, Graphics	-	-
update	GameContainer, StateBasedGame, init	-	-
getID	-	int	-

21.2 Interface Semantics

21.2.1 State Variables

mapToLoad: TDMap
container: GameContainer
arg1: StateBasedGame
g: Graphics
delta: init
MainMenu: image
playNow: image
exitGame: image
mapToLoad: image

21.2.2 Environmental Variables

None

21.2.3 Assumptions

None

21.2.4 Access Program Semantics

init(container, arg1):
  Input: GameContainer, StateBasedGame
  Transition: set image to mainMenu, set image to playNow, set image to exitGame
  Exception: None

render(arg0, arg1, g):
  Input: GameContainer, StateBasedGame, Graphics
  Transition: draw string at interface, draw mainMenu image at interface, draw playNow image at interface, draw exitGame image at interface
  Exception: None

update(arg0, arg1, delta):
  Input: GameContainer, StateBasedGame, init
  Transition: set default map, set mapToLoad = new TDMap("res/Try1.TDMap"), set new GameApplicationFrame(mapToLoad).
  Exception: None

getID():
  Input: none
  Transition: none
  Output: return 0
  Exception: None

22 MIS of MenuApplicationFrame Module

22.1 Interface Syntax

22.1.1 Exported Constants

PIXELWIDTH=460
PIXELHEIGHT=200
APPNAME = "Main Menu"
TIMEOUT = 30
bPlay = Play a game
bCreateMap = Create a map
bQuit = Quit
bLoadMap = Load a map
bDefault = Default
lblMapToLoad = MAP: Default

22.1.2 Exported Access Programs

Name	In	Out	Exceptions
MenuApplicationFrame	-	-	-
actionPerformed	ActionEvent	-	-
init	-	-	-
setMapName	string	-	-

22.2 Interface Semantics

22.2.1 State Variables

mapToLoad: TDMap
fc: JFileChooser
mainPanel: JPanel
bPlay: JButton
bCreateMap: JButton
bQuit: JButton
bLoadMap: JButton
bDefault: JButton
lblMapToLoad: JLabel
e: ActionEvent

22.2.2 Environmental Variables

None

22.2.3 Assumptions

None

22.2.4 Access Program Semantics

MenuApplicationFrame():
Input: none
Transition: set default map, set all button application
Exception: None

actionPerformed(e):
Input: ActionEvent
Transition: set every button to correspond perfermed action
Exception: None

init():
Input: none
Transition: set panel properties, set mainPanel.setBackground(Color.BLACK), set mainPanel.add(bCreateMap), set the Frame properties, set the x button as the default close operation
Exception: None

setMapName(name):
Input: string
Transition: add name to map

Exception: None

23 MIS of SetupClass Module

23.1 Interface Syntax

23.1.1 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
etupClass	string	-	-
initStatesList	GameContainer	-	-

23.2 Interface Semantics

23.2.1 State Variables

title: string
container: GameContainer

23.2.2 Environmental Variables

None

23.2.3 Assumptions

None

23.2.4 Access Program Semantics

SetupClass():
Input: string
Transition: set up title
Exception: None

initStatesList(container):
Input: GameContainer
Transition: initial mainmenu and gamestate
Exception: None