SE 3XA3: Module Interface Specification
Dragon Age

Group 8: Team Eight
Stanley Liu (MacID: liuz23)
Toni Miharja (MacID: miharjat)
Zhi Zhang (MacID: zhangz1)

November 10 2017

Contents

1 Revision History

Date	Developer	Change	Revision
November 10, 2017	Zhi	Part 3, 5, 7, 8	1.0
November 10, 2017	Stanley	Part 2, 4, 6	1.1
November 10, 2017	Toni	Part 9, 10, 11, 12, 13, 14	1.2
Dec 6, 2017	Zhi	Improve	2.0

Table 1: Revision History: Module Interface Specification

2 Module Hierarchy

Level 1	Level 2
Hardware Hiding Module	
Behaviour Hiding Module	Dragon Tower Module Time Bullet Module Time Enemy Module Time Hover Module Time Fired Module Draw Module Game Manager Module Gragon Age Module
Software Decision Hiding Module	Dragon Module Enemy Module Bullet Module Path Module Game Date Module

Table 2: Revision History: Module Hierarchy

3 MIS of Dragon Tower Module

3.1 Interface Syntax

3.1.1 Exported Access Programs

Name	In	Out	Exceptions
setDragons	-	-	-
isInRangeEquation	-	float	-
isInRange	-	boolean	-
drawTower	-	-	-
drawRadius	-	-	Insufficient building space
canUpgrade	-	boolean	No tower is been built
upgradeTower	-	-	Highest level reached

3.2 Interface Semantics

3.2.1 State Variables

Not Applicable

3.2.2 Environmental Variables

Not Applicable

3.2.3 Assumptions

Game started.

3.2.4 Access Program Semantics

setDragons():

- Output : set three types of dragons
- Exceptions: None

isInRangeEquation(x,y):

- Input: x, y
- Transition: get inRange value from x, y
- Output: inRange

isInRange(bounds):

- Transition: x0, x1, y0, y1 := bounds
- Output: return the boolean value whether or not the enemy is in range of tower

drawTower(canvas):

- Exceptions: insufficient building space

drawRadius(canvas):

- Output: draw radis of the enemy

canUpgrade(canvas):

- Output: whether or not the tower can still evolve

- Exceptions: No tower is been built

upgradeTower():

- Transition: dragon tower evolve to its next level
- Exception: Highest level reached

4 MIS of Timer Bullet Module

4.1 Interface Syntax

4.1.1 Exported Access Programs

Name	In	Out	Exceptions
moveAllBullets	-	-	Bullet out of bound
removeBullets	-	-	-
setTarget	-	-	-
shootEnemies	-	-	-
bulletEffect	canvas	-	-
setDamage	float	float	-
setBullets	-	-	-
allBulletsRemoved	-	boolean	-

4.2 Interface Semantics

4.2.1 State Variables

Not Applicable

4.2.2 Environmental Variables

Not Applicable

4.2.3 Assumptions

Assume dragon tower is placed onto board

4.2.4 Access Program Semantics

moveAllBullets():

- Transition: move bullets, if bullet goes out of bounds, remove bullets
- Exception: bullet out of bound

removeBullets():

- Input: x, y
- Transition: check whether bullets are removed for every frame and replace bullet list

setTarget():

- Transition: set target for each tower

shootEnemies():

- Transition: check if bullet hits enemy, if hit, set damage done to enemy or if enemy dies, enemy exit board and player get coins

bulletEffect():

- Input: object enemy, bullet
- Transition: reduce enemy speed if hit by bullet

setDamage():

- Output: the damage done to enemy

setBullets():

- Transition: set bullets for tower if tower has a target

allBulletRemoved():

- Transition: check if all bullets are removed from the board
- Output: return the value of bullet.remove

5 MIS of Timer Enemy Module

5.1 Interface Syntax

5.1.1 Exported Access Programs

Name	In	Out	Exceptions
moveAllEnemies	-	boolean	-
moveAllEnemies2	-	boolean	-
removeAllEnemies	-	boolean	-
roundOver	-	-	-

5.2 Interface Semantics

5.2.1 State Variables

None

5.2.2 Environmental Variables

Not Applicable

5.2.3 Assumptions

Enemy are running on the board.

5.2.4 Access Program Semantics

moveAllEnemies():

- Output : move walking enemies on the board ~~at different speed~~
- Exceptions: None

moveAllEnemies2():

- Output : move flying enemies on the board
- Exceptions: None

removeAllEnemies():

- Output : remove all enemies from enemy list
- Exceptions: None

roundOver(x,y):

- Output: whether or not the round is over
- Exception: None

6 MIS of Timer Hover Module

6.1 Interface Syntax

6.1.1 Exported Access Programs

Name	In	Out	Exceptions
hover	-	-	-
buildTowerHover	real, real	-	-
gameoverHover	-	-	-

6.2 Interface Semantics

6.2.1 State Variables

Not Applicable

6.2.2 Environmental Variables

None

6.2.3 Assumptions

The game is started.

6.2.4 Access Program Semantics

hover():

- Transition: x,y := pygame.mouse.get_pos()
- Output: put the tower on board

buildTowerHover(x,y):

- Input: x, y cordinates
- Transition: gameData.playerSelected.x, gameData.playerSelected.y= x,y
- Output: draw rectangle of size of dragon when building is legal

gameoverHover():

- Transition: remove all enemies when user's remaining life is 0

7 MIS of Timer Fired Module

7.1 Interface Syntax

7.1.1 Exported Access Programs

Name	In	Out	Exceptions
timeFired	-	-	-

7.2 Interface Semantics

7.2.1 State Variables

Not Applicable

7.2.2 Environmental Variables

None

7.2.3 Assumptions

None

7.2.4 Access Program Semantics

timeFired():

- Transition: runs all the time-based modules of the game

8 MIS of Draw Module

8.1 Interface Syntax

8.1.1 Exported Access Programs

Name	In	Out	Exceptions
~~drawIntro~~	-	-	-
drawEnemies	-	-	-
drawPlay	-	-	-
drawTowers	-	-	-
drawStatus	-	-	-
drawMessage	string	-	-
drawGameStats	-	-	-
drawParty	-	-	-
drawAllBullets	-	-	-
drawAll	-	-	-

8.2 Interface Semantics

8.2.1 State Variables

Not Applicable

8.2.2 Environmental Variables

None

8.2.3 Assumptions

The game is started.

8.2.4 Access Program Semantics

drawIntro():

- Transition: display the introduction page on the board

drawEnemies():

- Transition: display enemies on the board

drawPlay():

- Transition: display Play Button

drawTowers():

- Transition: draw all towers on board

drawStatus():

- Transition: display the status of a built tower selected by user

drawMessage():

- Transition: display the message

drawGameStats():

- Transition: display the money, level, remaining life of the user

drawParty():

- Transition: display options of dragon towers for game player

drawAllBullets():

- Transition: draw all bullets on board

drawAll():

- Transition: draw all items above on the board

9 MIS of Game Manager Module

9.1 Interface Syntax

9.1.1 Exported Access Programs

Name	In	Out	Exceptions
gameInit	-	-	-
runGame	-	-	-
mousePress	int, int	-	-

9.2 Interface Semantics

9.2.1 State Variables

Not Applicable

9.2.2 Environmental Variables

None

9.2.3 Assumptions

None

9.2.4 Access Program Semantics

gameInit():

- Transition: Initialise the game data

runGame():

- Transition: The functions that will be run continuously in the while loop of the main game

mousePress(x,y):

- Input: x and y coordinates
- Transition: Handle the mouse control of the game

10 MIS of Dragon Age Module

10.1 Interface Syntax

10.1.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	-	-
mouse	-	-	-
loadBackground	-	-	-
loadGameOverPage	-	-	-
loadBGM	-	-	-
loadIntro	-	-	-
game	-	-	-

10.2 Interface Semantics

10.2.1 State Variables

Not Applicable

10.2.2 Environmental Variables

None

10.2.3 Assumptions

None

10.2.4 Access Program Semantics

init():

- Transition: initialise pygame

mouse():

- Transition: handle the mouse control response of the game

loadBackground():

- Transition: load the game background

loadGameOverPage():

- Transition: display the gameover message

loadBGM():

- Transition: play background music of the game

loadIntro():

- Transition: display the menu which allow user to choose between play and quit

game():

- Transition: main loop of the game

11 MIS of Dragon Module

11.1 Interface Syntax

11.1.1 Exported Access Programs

Name	In	Out	Exceptions
setSize	-	-	-

11.2 Interface Semantics

11.2.1 State Variables

Not Applicable

11.2.2 Environmental Variables

None

11.2.3 Assumptions

None

11.2.4 Access Program Semantics

setSize():

- Transition: set the size of the dragon unit on the game board

12 MIS of Enemy Module

12.1 Interface Syntax

12.1.1 Exported Access Programs

Name	In	Out	Exceptions
setWave	-	-	-
setHP	-	float	-
setLevel	-	-	-
moveEnemy	-	-	-
drawEnemy	-	-	-

12.2 Interface Semantics

12.2.1 State Variables

Not Applicable

12.2.2 Environmental Variables

None

12.2.3 Assumptions

None

12.2.4 Access Program Semantics

setWave():

- Transition: spawn new enemy units for the current wave

setHP():

- Transition: set the individual hit point (HP) of the enemy
- Output: the hit point (HP)

setLevel():

- Transition: set the individual level of the enemy

moveEnemy():

- Transition: increase the coordinate of the enemy by the movement speed

drawEnemy():

- Transition: draw the enemy on the game board

Name	In	Out	Exceptions
setImage	-	-	-
getDirection	-	-	-
shotEnemy	-	boolean	-
moveBullet	-	-	-
drawBullet	-	-	-

13 MIS of Bullet Module

13.1 Interface Syntax

13.1.1 Exported Access Programs

13.2 Interface Semantics

13.2.1 State Variables

Not Applicable

13.2.2 Environmental Variables

None

13.2.3 Assumptions

None

13.2.4 Access Program Semantics

setImage():

- Transition: Transition: set the image of the bullet itself

getDirection():

- Transition: set the direction of the bullet

shotEnemy():

- Transition: determine if the enemy is within bound of the game
- Output: return âĂŸTrueâĂŹ if enemy is within bound of the game and hence can be damaged

moveBullet():

- Transition: move the bullet towards the enemy

drawBullet():

- Transition: draw the bullet on the screen

14 MIS of Path Module

14.1 Interface Syntax

14.1.1 Exported Access Programs

14.2 Interface Semantics

14.2.1 State Variables

Not Applicable

Name	In	Out	Exceptions
inPlay	int, int	boolean	-
onBoard	int, int	boolean	-
upgradeBound	int, int	boolean	-
inTowerBounds	-	boolean	-
canBuild	int, int	boolean	-
onRoute	-	boolean	-
onTopTower	-	boolean	-
inParty	int, int	boolean	-
createPath	-	-	-
verticalPath	-	-	-
horizontalPath	-	-	-

14.2.2 Environmental Variables

None

14.2.3 Assumptions

None

14.2.4 Access Program Semantics

inPlay():

- Input: the x and y coordinates
- Transition: determine if the object is inside the game screen
- Output: return true if the object is within the game screen

onBoard():

- Input: the x and y coordinates
- Transition: determine if the object is inside the game board area
- Output: return true if the object is within the game board area

upgradeBound():

- Input: the x and y coordinates
- Transition: determine if the coordinates is within the upgrade button
- Output: return true if the object is within the evolve button

inTowerBounds():

- Transition: determine if the tower is on top of another tower
- Output: return false if the tower is not on top of another tower

canBuild():

- Input: the x and y coordinates
- Transition: determines if the tower can be built in correct tower bound
- Output: return true if the tower can build in correct tower bound

onRoute():

- Transition: determines if the tower is on enemy path
- Output: returns false if the tower is not on enemy path

onTopTower():

- Transition: determines if the tower in on top of another tower
- Output: returns false if the tower is not on top of another tower

inParty():

- Input: the x and y coordinates
- Transition: determine if the object is inside the party menu
- Output: return true if the object is within the party menu

createPath():

- Transition: create the corners of the enemy path

verticalPath():

- Transition: create the vertical portion of the enemy path

horizontalPath():

- Transition: create the horizontal portion of the enemy path