Wai Yan (Will) Htun

104-941-153

CS 32, Nachenberg

Project 3 (Nachenblaster) Report

**High Level Description of Member Functions**

1. StudentWorld – inherited from GameWorld
   1. ~StudentWorld

Calls the cleanUp() function to deallocate all the memories taken up by actor pointers.

* 1. virtual init()

Called when the level is first initiated. Creates a NachenBlaster object and 30 star objects. Calculates how many ships are needed to be destroyed to advance to next level.

* 1. virtual move()

Calls doSomething() function from all the actors present on screen, including the NachenBlaster. If NachenBlaster is not alive, decrease live by 1, delete the NachenBlaster object and informs the framework that the player has died. If number of ships destroyed is more than or equal to target Ships, informs the framework that the player has finished the level. If there can be more alien ships on the screen, introduces a new type of alien ships using the algorithm given in the specs, and creates a new object of that alien class. Calls functions to remove dead objects and updates the display text at the top of screen.

* 1. virtual cleanUp()

Delete all the actors on screen and clear the vector holding the actor pointers.

* 1. removeDeadGameObjects()

Delete all the dead game objects on screen. There is 1 in 16 chance that a new start is created on the right side of the screen.

* 1. updateDisplayText()

Displays the statistics of the game on top of the screen.

* 1. shootCabbage()

Creates cabbage object 12 pixels right of the NachenBlaster.

* 1. shootTurnip(double x, double y)

Creates a Turnip object at the given x,y coordinates.

* 1. shootTorpedo(double x, double y, bool fromNB)

Creates a Torpedo object at the given x,y coordinates. If NB is true, it is a friendly torpedo, else it is an enemy torpedo.

* 1. dropExtraLifeGoodie(double x, double y)

Creates an extra life goodie at the given x,y coordinates.

* 1. dropRepairGoodie(double x, double y)

Creates a repair goodie at the given x,y coordinates.

* 1. dropTorpedoGoodie(double x, double y)

Creates a torpedo goodie at the given x,y coordinates.

* 1. increaseShipsDestroyed()

Increase the number of ships destroyed by one.

* 1. getNachenBlaster()

Return the pointer pointing to the NachenBlaster

* 1. getActors()

Return the vector of pointers holding all the actors on screen besides the NachenBlaster.

1. Actor – inherited from GraphObject
   1. virtual doSomething() = 0

To be overridden in derived classes. This function is abstract because Actor is an abstract class and this version of the function will never be called.

* 1. virtual getFirePower()

To be overridden in NachenBlaster class. Not virtual because Actor pointer will be used to hold NachenBlaster object, and this function will be called from that pointer.

* 1. virtual isAlienShip()

To be overridden in derived clases. Returns false everywhere except in the derived class Alien and its derived classes. Used to check in StudentWorld if the object the pointer points to is an enemy ship or not.

* 1. virtual getHealth()

To be overridden in NachenBlaster and Alien class.

* 1. virtual setHealth(double hp)

To be overridden in NachenBlaster and Alien class.

* 1. virtual setTorpedo(int t)

To be overridden in NachenBlaster class.

* 1. virtual getTorpedo()

To be overridden in NachenBlaster class.

* 1. virtual getPoints()

To be overridden in Alien class.

* 1. getWorld()

Returns the pointer to the StudentWorld object the game is currently taking place in. This pointer is then used to control the game mode.

* 1. setCurrentWorld(StudentWorld\* sw)

Called when creating a new object in the StudentWorld object. When called, the private member \*currentWorld is set to the address of the StudentWorld object the game is currently taking place in.

* 1. isAlive()

Returns true of the current object is alive.

* 1. setIsAlive(bool status)

Set the state of the current object’s status (alive/dead).

1. Star – inherited from Actor
   1. virtual doSomething()

If x position of the star is more than -1, move the star 1 pixel left. Else, set the alive status to false.

1. Explosion – inherited from Actor
   1. virtual doSomething()

Increases the size by 50%, and decrease life time by one. If lifetime is 0, set the alive status to false.

1. NachenBlaster – inherited from Actor
   1. virtual doSomething()

If NachenBlaster is not alive, return. If health is 0 or less than 0, set alive status to false and return. Else, if it is alive, check to see if the player has pressed any of the following buttons: WASD, arrow keys, space, tab, and esc. If WASD or arrow keys are pressed, move the NachenBlaster in the corresponding direction. If space is pressed and cabbage point is more than 0%, shoot a cabbage. If tab is pressed and torpedo count is more than 0, shoot a torpedo.

* 1. virtual getHealth()

Returns the health of the NachenBlaster at the current tick.

* 1. virtual getFirePower()

Returns the cabbage fire power of the NachenBlaster.

* 1. virtual setHealth(double hp)

Set the current health of the NachenBlaster to the value of hp.

* 1. virtual setTorpedo(int t)

Set the current number of torpedoes held by the NachenBlaster to the value of t. These is no maximum number of torpedoes allowed to be held.

* 1. virtual int getTorpedo()

Returns the number of torpedoes the NachenBlaster is currently holding.

1. Projectile – inherited from Actor
   1. virtual doSomething()

If projectile is not alive, return. If it has flown of the screen, set its alive status to false. Else, if alive, call collisionCheck() to see if it has collided with anything. If the projectile is friendly (shot by NachenBlaster), move right by the speed set by the derived classes. Else, if the projectile is hostile (shot by Alien ships), move left by the speed set by the derived classes. If the projectile is supposed to rotate, rotate it by 20 degrees. Call collisionCheck() once again before the function finishes,

* 1. virtual collisionCheck() = 0

To be overridden in Cabbage, Turnip, Torpedo, and FriendlyTorpedo class. This function is abstract because Projectile is an abstract class and this version of the function will never be called.

* 1. getSpeed()

Returns the travelling speed of the projectile.

* 1. getDmgPoint()

Returns the amount of damange that the projectile can do to the NachenBlaster or the Alien ships, depending on the type of projectile.

1. Cabbage – inherited from Projectile
   1. virtual collisionCheck()

Overrides the collisionCheck() in Projectile class. Get all the actor pointers from StudentWorld object, and check if each of them if they are in contact with itself (Cabbage object) using the Euclidian distance formula. If they are in contact, and the object in contact is an Alien ship, this function will call setHealth() from the Alien ship in contact to reduce its health. Then, it will play SOUND\_BLAST and set to not alive so it can be deleted on the next tick.

1. Turnip – inherited from Projectile
   1. virtual collisionCheck()

Overrides the collisionCheck() in Projectile class. Check if this (Turnip object) is in contact with the NachenBlaster using the Euclidian distance formula. If they are in contact, this function will call setHealth() from the NachenBlaster to reduce its health. Then, it will play SOUND\_BLAST and set to not alive so it can be deleted on the next tick.

1. Torpedo – inherited from Projectile
   1. virtual collisionCheck()

Overrides the collisionCheck() in Projectile class. Check if this (Torpedo object) is in contact with the NachenBlaster using the Euclidian distance formula. If they are in contact, this function will call setHealth() from the NachenBlaster to reduce its health. Then, it will play SOUND\_BLAST and set to not alive so it can be deleted on the next tick.

1. FriendlyTorpedo – inherited from Projectile
   1. virtual collisionCheck()

Overrides the collisionCheck() in Projectile class. Get all the actor pointers from StudentWorld object, and check if each of them if they are in contact with itself (Cabbage object) using the Euclidian distance formula. If they are in contact, and the object in contact is an Alien ship, this function will call setHealth() from the Alien ship in contact to reduce its health. Then, it will play SOUND\_BLAST and set to not alive so it can be deleted on the next tick.

1. Alien – inherited from Actor
   1. virtual doSomething()

If Alien is not alive, return. If Alien has flown off screen, set its alive status to false. Else, if Alien is alive, call collisionCheck(), and changeTravelDirection(). If the NachenBlaster is +/-4 pixels up or down the Alien ship, there is a chance that the Alien ship will fire it. This chance is determined by current level. There is also the same amount of chance that the Alien ship will change flight plan. If travel direction is 0, move the ship up and left. If the travel direction is 1, move the ship left. If the travel direction is 2, move the ship down and left. Decrease flight plan by 1, and call collisionCheck() again before finishing the function.

* 1. virtual isAlienShip()

Returns true. To be used in StudentWorld object through Actor pointers.

* 1. virtual changeFlightPlan()

To be overridden in derived classes.

* 1. virtual changeTravelDirection()

If the ship reaches the top of the screen, set its travel direction to down and left. If the ship reaches the bottom of the screen, set its travel direction to up and left. If the ship runs out of flight plan, set its travel direction to one of the following by random: down and left, up and left, or left.

* 1. virtual fire() = 0

To be overridden in derived classes.

* 1. virtual getHealth()

Returns the health of the ship.

* 1. virtual setHealth(double hp)

Sets the health of the ship to hp.

* 1. getFlightPlan()

Returns the value of flight plan of the ship.

* 1. getTravelSpeed()

Returns the value of travel speed of the ship.

* 1. getTravelDirection()

Returns the value of travel direction of the ship.

* 1. getCollisionDmg()

Returns the amount of health to be reduced if collided with the NachenBlaster.

* 1. setFlightPlan(int fp)

Sets the value of flight plan to fp.

* 1. setTravelSpeed(double sp)

Sets the value of travel speed to sp.

* 1. setTravelDirection(int td)

Sets the index of travel direction to td, where td can be 0, 1, 2 indicating up and left, left, and down and left respectively.

* 1. collisionCheck()

Check if this object collides with the NachenBlaster using Euclidian distance formula. If so, decrease NachenBlaster’s health by the value of collisionDmg (dependent on type of Alien ship, defined in derived classes). Then, this function calls the dropGoodie() function, increases number of ships destroyed and score, plays SOUND\_DEATH, creates an Explosion object at its x,y coordinate, and set its alive status to false.

* 1. getPoints()

Returns the amount of points that should be rewarded to the player when an Alien ships is destroyed. The value is determined by each derived classes.

* 1. dropGoodie()

To be overridden in derived classes.

1. Smallgon – inheritied from Alien
   1. virtual fire()

Play SOUND\_ALIEN\_SHOOT, and create a turnip object by calling shootTurnip from StudentWorld.

1. Smoregon – inherited from Alien
   1. virtual changeFlightPlan()

Sets travel direction to 1 (left), sets it flight plan to VIEW\_WIDTH, and increases speed to 5.

* 1. virtual fire()

Play SOUND\_ALIEN\_SHOOT, and create a turnip object by calling shootTurnip from StudentWorld.

* 1. virtual dropGoodie()

Use rand() to simulate 1 in 3 chance. If so, there is 50% chance of dropping a repair goodie and 50% chance of dropping a torpedo goodie. Calls drop function from StudentWorld according to which goodie to drop.

1. Snaggleon – inherited from Alien
   1. virtual changeFlightPlan()

If the ship reaches the top of the screen, sets travel direction to down and left. If the shup reaches the bottom of the screen, sets travel direction to up and left.

* 1. virtual fire()

Plays SOUND\_TORPEDO and creates a torpedo object.

* 1. virtual dropGoodie()

Use rand() to simulate 1 in 6 chances. If so, drop an extra life goodie at the current position of the ship.

1. Goodie – inherited from Actor
   1. virtual doSomething()

If Goodie is not alive, return. If the Goodie has flown off screen, set its alive status to false. Call collisionCheck() to see if it has collided with any other object. Move left and down. Call collisionCheck() again before ending the function.

* 1. virtual powerup()

To be overridden in derived classes.

* 1. collisionCheck()

Checks if this (Goodie object) is in contact with the NachenBlaster using the Euclidian distance formula. If they are in contact, this function will call the powerUp() function Then, it will play SOUND\_GOODIE and set to not alive so it can be deleted on the next tick.

1. ExtraLifeGoodie – inherited from Goodie
   1. virtual powerup()

Increases score by 100 and live by 1 through the StudentWorld object.

1. RepairGoodie – inherited from Goodie
   1. virtual powerup()

Increases score by 100 and the health of the NachenBlaster by 10. If NachenBlaster health is more than 40, set its health to 50 (max).

1. TorpedoGoodie – inherited from Goodie
   1. virtual powerup()

Increases score by 100 and the number of torpedoes held by NachenBlaster by 1 through the StudentWorld object.

**Failed Functionality**

All the basic functionalities required by the project specification are implemented. The issue of stars abruptly disappearing cannot be resolved, even after updating the three files provided on the class website.

**Assumptions**

I assumed that the stars disappearing abruptly was due to the framework, and not the four files that I edited (StudentWorld.cpp, StudentWorld.h, Actor.cpp, Actor.h).

**Classes Test Cast**

The behavior of each classes are given below. Since testing each class individually is difficult, the testing is done by playing the game as a whole and documenting the behavior of each class. The pause and advance tick function provided by the framework are thoroughly used in testing. The given descriptions below are observed when playing the game, and is considered working/passing.

1. StudentWorld – inherited from GameWorld

The StudentWorld is a game manager that keeps track of the progress of the game. Aside from a few functions that are defined in GameWorld, StudentWorld has several member functions that are used to create new objects.

1. Actor

Actor is an abstract class and is purely meant to be overridden by its derived classes.

1. Star – inherited from Actor

Upon creation, a star object steadily moves left at 1px/tick speed and disappears when it reaches the left border of the window. The size of the star is determined randomly between 0.05 and 0.5, with depth 3 and no rotation.

1. Explosion – inherited from Actor

Upon creation, an explosion starts off with size 1, but increases by 50% with every tick, up to 4 ticks. After 4 ticks, it is set to dead so it can be deleted in the next tick. Explosion objects do not rotate and has depth of 0.

1. NachenBlaster – inherited from Actor

The NachenBlaster class is tested by movements, actions, and reactions to collisions. The NachenBlaster object can be controlled by WASD/arrow keys to move around the screen, given that the movement will not cause NachenBlaster to move off the screen. Pressing space will fire a cabbage given that it has more than 5 cabbage points (fire power is more than 0%). Fire power accumulates overtime and decreases whenever a cabbage is fired. The NachenBlaster can also shot torpedoes if it has any. The NachenBlaster can also collide with any alien ship and will lose health and destroys the alien ship.

1. Projectile – inherited from Actor

Projectile is an abstract class and is purely meant to be overridden by its derived classes.

1. Cabbage – inherited from Projectile

The cabbage object is created whenever the NachenBlaster shoots. A cabbage has size 0.5 with no rotation and depth of 1. When it hits an alien ship, it decreases its health by 2. It travels right at the speed of 8 pixel per tick, disappearing when it reaches the right border of the screen. It does not do damage to Goodies, Stars, Explosions, or the NachenBlaster itself since it only checks collision with Alien ships.

1. Turnip – inherited from Projectile

The turnip object is created whenever the Smallgon or Smoregon shoots. A turnip has size 0.5 with no rotation and depth of 1. When it hits NachenBlaster, it decreases its health by 2. It travels left at the speed of 6 pixel per tick, disappearing when it reaches the left border of the screen. It does not do damage to Goodies, Stars, Explosions, or the Alien ships since it only checks collision with NachenBlaster.

1. Torpedo – inherited from Projectile

The torpedo object is created whenever the Snagglegon shoots. A torpedo has size 0.5 with no rotation and depth of 1. When it hits NachenBlaster, it decreases its health by 8. It travels left at the speed of 8 pixel per tick, disappearing when it reaches the left border of the screen. It does not do damage to Goodies, Stars, Explosions, or the Alien ships since it only checks collision with NachenBlaster.

1. FriendlyTorpedo – inherited from Projectile

The friendly torpedo object is created whenever the NachenBlaster shoots with torpedo. A torpedo has size 0.5 with no rotation and depth of 1. When it hits an Alien ship, it decreases its health by 8. It travels right at the speed of 8 pixel per tick, disappearing when it reaches the right border of the screen. It does not do damage to Goodies, Stars, Explosions, or the NachenBlaster since it only checks collision with Alien ships.

1. Alien – inherited from Actor

Alien is an abstract class and is purely meant to be overridden by its derived classes.

1. Smallgon – inherited from Alien

A smallgon flies from right to left with the speed of 2.0, until it runs out of flight path. When flight path is 0, it will select a new flight path between 1 and 32. It also changes direction of travel during its travel time. If the NachenBlaster is +/-4 pixels above or below the smallgon, it has 1 in 3 changes of firing a turnip at the NachenBlaster. When a smallgon is destroyed, it rewards player with 250 points. It also plays SOUND\_DEATH, creates an explosion object at the current x,y coordinate and sets isAlive() to false.

1. Smoregon – inherited from Alien

A smoregon flies from right to left with the speed of 2.0, until it runs out of flight path. When flight path is 0, it will select a new flight path between 1 and 32. It also changes direction of travel during its travel time. If the NachenBlaster is +/-4 pixels above or below the smoregon, it has 1 in 3 changes of firing a turnip at the NachenBlaster. It also has a chance of slamming towards the NachenBlaster with speed of 5. This chance is decided by level. When a smoregon is destroyed, it rewards player with 250 points. It also plays SOUND\_DEATH, creates an explosion object at the current x,y coordinate and sets isAlive() to false. Upon its death, a smoregon has 1/3 chance of dropping a goodie – ½ chance of repair goodie and ½ chance of torpedo goodie.

1. Snagglegon – inherited from Alien

A snagglegon flies from right to left/down diagonally with the speed of 1.75, until it reaches the top or bottom of the screen. If it reaches the top, it travels left/down. If it reaches the bottom, it travels left/up. If the NachenBlaster is +/-4 pixels above or below the snagglegon, it has 1 in 3 changes of firing a torpedo at the NachenBlaster. When a snagglegon is destroyed, it rewards player with 1000 points. It also plays SOUND\_DEATH, creates an explosion object at the current x,y coordinate and sets isAlive() to false. Upon its death, a snagglegon has 1/6 chance of dropping an extra life goodie.

1. Goodie – inherited from Actor

A goodie calls collisionCheck() every tick, to see if it has collided with the NachenBlaster. If so, it will play SOUND\_GOODIE and call its powerUp(), whose functionality differs depending on the derived classes. If it flies off screen, it sets itself to dead so it can be deleted next tick. It always moves left 0.75 and down 0.75 every tick. It also does not collide with any other object besides the NachenBlaster.

1. ExtraLifeGoodie – inherited from Goodie

Extra life goodie is a type of goodie, that increases the score by 100, and life by 1 when eaten.

1. RepairGoodie – inherited from Goodie

Repair goodie is a type of goodie, that increases the score by 100, and the NachenBlaster health by 50 when eaten.

1. TorpedoGoodie – inherited from Goodie

Torpedo goodie is a type of goodie, that increases the score by 100, and torpedo count of the NachenBlaster by 1 when eaten.