

21 August 2020

Professor Ambrosio

Project 4 Report

Public Member Functions

Actor (abstract base class):

- Constructor and virtual Destructor
- I chose to define a pure virtual version of the doSomething function because all Actors must have a doSomething function that tells them what actions to take during each tick, so each type of actor must define their own special version of it that has the object function as specified while it is alive.
- I defined virtual versions of the gotGold, getHitPoints, and getAnnoyed functions that do nothing as this function needs to be called on a Protester through an Actor pointer and will be implemented differently in the Protester and TunnelMan classes.
- I defined virtual versions of the blocksTunnelman, hardcore, and annoyedByBoulderOrSquirt functions that return false, as they will return different values in derived classes.
- I defined a non-virtual getWorld function that will just always return a pointer to the StudentWorld the actor is in, so it will never need to be modified by a derived class.
- I similarly defined non-virtual getState and setState functions that return and set an Actor's state, respectively, and will similarly function for all Actor classes.

- Finally, I defined a non-virtual move function that has the Actor object move one step in whatever direction it is facing. As this function will also behave the same way for all Actor classes, it is not virtual.

Goody (base class for items TunnelMan can pick up, derived class from Actor):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function because all Goodies must have a specialized doSomething function as each one plays a different sound when picked up and increases the game's score by a different amount.

WaterPool (derived class from Goody):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.

SonarKit (derived class from Goody):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.

OilBarrel (derived class from Goody):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.

GoldNugget (derived class from Goody):

- Constructor and virtual destructor

- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.
- I defined a non-virtual setLifeSpan function as the gold nugget is the only object that needs to have its life span redefined within a level.

Squirt (derived class from Actor):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.

Earth (derived class from Actor):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class. This function does nothing as the earth object does not take any actions during a tick.

Boulder (derived class from Actor):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.
- I defined a virtual function of the blocksTunnelman function for clarity, as this function was redefined to return true.

TunnelMan (derived class from Actor):

- Constructor and virtual destructor

- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.
- I defined non-virtual functions getWater, getSonarCharge, getGold, addSonar, addWater, and addGold, as the TunnelMan class is the only one that requires these functions. These are accessors and mutators of the member variables of TunnelMan.
- I defined virtual functions getAnnoyed and getHitPoints for clarity, as these functions were redefined from the base class to have the TunnelMan's specific functionality.

Protester (derived class from Actor):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.
- I defined virtual functions getAnnoyed, gotGold, annoyedByBoulderOrSquirt, and getHitPoints for clarity, as these functions were redefined from the base class to have the Protester's specific functionality.
- I defined non-virtual functions setTick, getTick, getShout, and getHitPoints, as the Protester class is the only one that requires these functions, and they do not need to be redefined anywhere. These are accessors and mutators of the member variables of Protester.

HardcoreProtester (derived class from Protester):

- Constructor and virtual destructor
- I chose to define a virtual version of the doSomething function for clarity as I was redefining a virtual function from the base class.

- I defined virtual functions `gotGold` and `hardcore`, as these functions were redefined from the base class to have the `HardcoreProtester`'s specific functionality.

Known Bugs

- My hardcore protester does not yet have the cell phone signal feature and does not act differently if it is within a certain distance of the `TunnelMan`; all other functionalities of the hardcore protester work well, however, such as getting stunned by Gold Nuggets and Squirts and incrementing the score as specified
- Protester sometimes has trouble sensing Earth objects right underneath it and walks onto them into a nonvalid location

Design Decisions and Assumptions

It was a bit ambiguous what to do when the `TunnelMan` approached a boulder, as the spec and various Piazza posts confirmed that the `TunnelMan` cannot go within a radius of 3.0 within a boulder. However, after implementing this in my program, it behaved a little differently from the sample, which seemed to not let the `TunnelMan` enter the entire area of 4 squares the boulder was occupying. I decided to keep the program checking for the radius of 3 from the boulder, as that is what most sources seemed to confirm.

I was also unsure if in the spec, the value `M`, which was the `probabilityOfHardcore` in the `StudentWorld` move function, meant that the new protester should have a 1 in `M` probability of being hardcore, or if it should have an `M%` probability of being hardcore. After studying the sample, it seemed that `M%` would be the correct answer as it produces a higher number of hardcore protesters, so this is what I decided to implement in my program as well.

I was not sure if a Water Pool was allowed to appear in the same space as a Boulder, but as the spec only said that the Water Pool must not appear where there are Earth objects, I allowed Water Pools to share a space with Boulders.

Testing

I tested my StudentWorld by building the game world multiple times, making every object visible, and making sure that the proper number of each object, such as Boulders, Gold Nuggets, Earth, Oil Barrels, and TunnelMan, appeared in their respective positions. Additionally, a Protester was initialized at the beginning of every level, and the status text line appeared at the top of the screen and had constantly updated values. Pressing the escape key caused the game to quit, and a completely annoyed TunnelMan resulted in a lost life. When the TunnelMan collected all the Oil Barrels in the game world, the game moved on to the next level without losing a life. At various intervals throughout the gameplay of a level, Water Pools, Sonar Kits, and more Protesters were created and added to the game world.

I tested my Water Pool class by paying close attention to where they were added to the game world during gameplay. They successfully always initialized in locations without Earth objects, and were able to be picked up by the TunnelMan when close enough. When picked up, the Water Pool then disappeared and the TunnelMan's amount of water increased by 5, allowing it to shoot more Squirt objects.

My Sonar Kit always successfully initialized in the top left corner of the game world, as specified and was able to be picked up by the TunnelMan when close enough. The Sonar Kit then disappeared after being picked up, and the TunnelMan gained one Sonar powerup to use.

I tested my Oil Barrel by making it visible initially and made sure that the correct number of these objects were created at the beginning of each level in valid locations. When the TunnelMan got close enough, these objects were made visible and then picked up when the TunnelMan got even closer. When picked up, the number of Oil Barrels left in the field decreased by one in the status bar, and when the last one was picked up, the level ended.

I tested my Gold Nugget by making it visible initially and made sure that the correct number of these objects were created at the beginning of each level in valid locations. When the TunnelMan got close enough, these objects were made visible and then picked up when the TunnelMan got even closer. When picked up, the amount of gold the TunnelMan had increased by one, and when dropped by the TunnelMan, the Gold Nugget appeared in the right location and was only able to be picked up by the Protesters, not the TunnelMan.

I tested my Earth class by first building the game world multiple times and making sure that all the Earth showed up in the right places, filling the entire screen except for the top portion, the column in the middle, and the areas where boulders were initialized. I also then had my TunnelMan move through various portions of the Earth, checking to make sure all the Earth blocks touched or passed through by the TunnelMan disappeared.

I tested my Boulder class by building the game world multiple times, making sure the correct number of Boulders were initialized for the respective levels, and that they were in a location other than the top portion of the screen without Earth and the middle column. I then changed the numbers in the StudenetWorld to have the Boulders all initialize in the same column. I then had the TunnelMan dig underneath the top boulders to make sure they

disappeared when colliding with the top of any other boulder. The Boulders also disappeared as specified when colliding with an Earth object or the bottom game world boundary.

I tested my Squirt class by doing repeated tests by pressing the spacebar with the TunnelMan facing various objects from different distances. I confirmed that when the TunnelMan was facing and very close to an Earth or Boulder object, as well as the boundaries of the game world, the Squirt object did not appear on the screen. Additionally, when the TunnelMan was facing an Earth or Boulder object or the boundary from a slightly further distance, the Squirt appeared and travelled straight until hitting one of these objects and disappearing. Similarly, the Squirt would behave similarly when encountering a Protester, freezing the Protester if it was in the Hardcore Protester class, and making a Regular Protester give up if hit too many times. The Squirt, when not blocked by any of these objects, travelled its proper distance of four squares before it disappeared.

I tested my TunnelMan through various gameplay trials. Pressing the arrow keys allowed the TunnelMan to move around the game world properly, not leaving the boundaries or walking through Boulders. The TunnelMan also changed directions properly when necessary, and was always facing the direction it was moving in. When close to Goody objects, the TunnelMan was able to pick these objects up and use their various features.

When the spacebar was pressed, if the TunnelMan had water, a Squirt was created and behaved properly, and when the tab key was pressed and the TunnelMan had gold, a Gold Nugget was dropped in the proper place and the TunnelMan was unable to pick it up. Similarly, when the Z or z keys were pressed and the TunnelMan had sonar, hidden objects within a certain radius of the TunnelMan were made visible. The TunnelMan was also able to dig through the

Earth that was in the way of its path. When the TunnelMan was very close to a Protester and got yelled at, the TunnelMan's health decreased, and if the health got to 0%, the level ended and a life was lost. Similarly, if the TunnelMan was bonked by a boulder, the level ended and a life was lost.

I tested my Protester by digging different paths in the Earth and seeing the Protester's movements. Whenever it had a clear straight path to the TunnelMan, the Protester would immediately change direction to face the TunnelMan and start moving towards it. For the most part, the Protesters only moved to valid locations, avoiding Earth and Boulder objects. The Protesters were able to pick up Gold Nuggets dropped by the TunnelMan, but no other objects, and stopped chasing the TunnelMan when sufficiently annoyed with the Squirts or Boulders.

The Protesters also moved slower than the TunnelMan, only performing actions every couple ticks, instead of every tick. The Protesters could also successfully find new perpendicular directions to turn to if clear, and move through these new paths. The Protesters also moved to the exit successfully and proceeded to disappear after being sufficiently annoyed.

I tested my Hardcore Protester by hitting it with Squirts. When annoyed, the Hardcore Protester just froze in place for a certain amount of ticks and then began its normal behavior once again. Similarly, this also happened when the Hardcore Protester was bribed with a Gold Nugget. Annoying the Hardcore Protester increased the score by a larger amount than annoying a Regular Protester.