Name: Sungmin Cho

UCID: 205124669

1. classes
   1. Actor
      1. doSomething() is a pure virtual function for the subclasses to define. I made doSomthing() pure virtual because it would have to be implemented by all subclasses eventually.
      2. getBribed() is a virtual function that will be redefined by the protester classes. I made getBribed() just virtual because it would only be used by the protesters.
      3. getWorld() returns StudentWorld of the class.
      4. getAnnoyed() is a virtual function that will be redefined by subclasses that is capable of being annoyed.
      5. getAlive() returns a m\_alive, showing whether the class is alive or not.
      6. setAlive() changes m\_alive.
      7. getType() is used to return type, distinguishing each Actor subclasses by their type name.
      8. setType() sets the type for each subclass.
      9. move() move function receives direction, the current coordinate, and the distance variable will travel to successfully move but blocked by boulder.
   2. TunnelMan
      1. doSomething() receives the player’s input and applies the appropriate move. For the left, right, up, down, it moves to a proper location. Tab puts gold down while Z and z uses sonar. Spacebar squirts water and escape loses life.
      2. getAnnoyed() makes TunnelMan to lose 2 points of HP and when it is 0 or below, it sets its m\_alive to false(dead).
      3. getters and setters of this returns and changes the private variables.
   3. Boulder
      1. doSomething() checks if there are earth below and if there are not, turns its state to waiting, counts 30 ticks, then finally falls and declares its death upon hitting earth, boulder, or out of bound.
   4. Squirt
      1. doSomething() travels its m\_travel distance and for each move it decreases m\_travel by one. When m\_travel is 0 it no longer moves.
   5. Barrel
      1. doSomething() checks if TunnelMan is within 4 block radius and reveals itself if so. Within 3 block radius, it declares its death and increases score.
   6. GoldNugget
      1. doSomething() checks if TunnelMan is within 4 block radius and reveals itself if so. Within 3 block radius, it declares its death and increases score if its m\_pickup is true. When it is false, the same is occurred by Protester instead and calls their getBribed() function.
   7. SonarKit
      1. doSomething() checks if TunnelMan is within 3 block radius and declares its death and increases sonar if true. However, after counting its tick off, when tick reaches 0, it declares its death and disappears.
   8. Water
      1. doSomething() checks if TunnelMan is within 3 block radius and declares its death and increases water if true. However, after counting its tick off, when tick reaches 0, it declares its death and disappears.
   9. StudentWorld
      1. init() locates oil content in the right position. Earth and TunnelMan as indicated by the instruction and other content randomly as described.
      2. tooClose() checks if the random coordinates for init functions are too close to already spawned Actors.
      3. move() updates the text every tick, places goodie when necessary, calls doSomething of all actors, deletes dead actors. If player is dead, returns player dead status along with the sound and life decrease. If oil is all found, it returns finished level. Otherwise, continues game.
      4. setDisplayText() and displayText() work together to output the string text at the top of the game.
      5. deleteEarth() receives a coordinate input and deletes the earth in 4 block radius area.
      6. deleteDead() goes through the actors and deletes whatever actor that died.
      7. useSonar() uses sonar and reveals barrels or nuggets within 12 block radius.
      8. annoyProtesters() is used by squirt to annoy the protester it hits.
      9. fallProtester() is used to kill the protester that’s hit by falling boulder.
      10. protesterGold() is used by gold nugget class to call getBribed function of protesters.
      11. distCheck() checks if two coordinates are within a certain radius area.
      12. block() shows whether or not the coordinate is blocked by boulder object.
      13. earthBlock() shows whether or not the coordinate is blocked by the earth object.
2. The majority of protester class was not implemented. The exit algorithm was not implemented and so was the way to track ticks and the decisions made in earlier ticks. A few parts of getAnnoyed() function was implemented but it was not completed. Protester does not move unless seen in straight line. However, Protester does annoy the TunnelMan, decreases HP, and even kill him.
3. I assumed that the definition of distance in radius did not exactly mean radius but just the distance away in terms of coordinate difference. For instance, if the instruction said 4 block radius, I just made sure that objects were 4 blocks away in terms of x and y. I made this assumption after actually applying the distance equation that incorporates square root, which did not work out. Majority of the auxiliary functions were implemented inside the StudentWorld class because the game data was all stored in that object.
4. Class Test
   1. StudentWorld
      1. Each time I had an error, I tried the debug function of Visual Studio to check whether the memory leak issue came from. I was eventually able to figure out all that I need to find and was able to successfully create each level and clear it.
   2. TunnelMan
      1. I first implemented the move function to see if it was able to move in a free space not blocked by anything. Then I added the deleteEarth() function to see if it was able to detect the moved coordinate and delete earth accordingly.
   3. Earth
      1. I created a nested for loop in the init function of StudentWorld to check if the display was correct. Then, I checked to see if the earth deleted itself instantaneously as TunnelMan dug through it.
   4. Boulder
      1. I moved the TunnelMan object from all directions to see if it does not go through. Then, I checked to see if the boulder dropped after the earth below was gone by digging with TunnelMan object. Once checked, I went through multiple simulations to check if it died after hitting earth, after hitting another boulder, and after hitting the ground. Finally, I put the TunnelMan object under it to see if it died as intended.
   5. Squirt
      1. I first checked if the squirt object moved one by one to the appropriate destination by using TunnelMan. First checked the empty area, then boulder, then earth.
   6. Barrel
      1. I added a allVisible() function in StudentWorld to change back and forth between the visibilities of the Actor objects. I used it to find the location, moved TunnelMan near it. Turned the visibility off and then moved closer to it to see if Barrel became visible automatically. Then I moved into it to see if the score was added and the correct sound was produced.
   7. GoldNugget
      1. I did the same as Barrel for this class. Additionally, however, I used the tab ability of TunnelMan to drop and checked if TunnelMan could obtain it. Further, I waited the dropped gold nugget for a bit of a time to check if it disappeared after a certain time.
   8. SonarKit
      1. I added a line in move() in StudentWorld to add a SonarKit object in certain location. Then I moved my TunnelMan toward it and checked if the object disappeared, made sound, and increased sonar count.
   9. Water
      1. I added a line in move() in StudentWorld to add a water object in certain location. Then I moved my TunnelMan toward it and checked if the object disappeared, made sound, and increased the water supply.