1.

Object Class

doSomething: This function lets each object do something whenever it is called. It is pure virtual because each object in the program must have their own doSomething defined for them to take an action each tick.

getWorld: Function returns the StudentWorld private member of this Object to make it easier to call StudentWorld functions. It is in the object class so all objects return their StudentWorld private member

isAlive: Function returns the status of whether the object is alive (Boolean m\_alive private member). It is in the object class because all objects can be alive.

isetDead: Function makes an object dead (changes m\_alive to false). It is in the object class because all objects can be alive and they might need to be changed to dead so they can be removed from the game.

isProtester: Function returns false as most objects are not protesters. It is virtual because in protester classes, this function has to return true. It is in the object class because this function is called when we iterate through an object vector.

getState: Function returns the current state of object. It is in the object class because this function is called when we iterate through an object vector. It is virtual because not all objects have states and those will have empty state. However, those objects with states will have their own output.

loseHealth: Function allows objects to take damage. It is in the object class because this function is called when we iterate through an object vector. It is virtual because only Person class objects can take damage. Hence, if the other classes accidentally uses this function, nothing happens.

getBribed: Function allows object to be bribed. It is in the object class because this function is called when we iterate through an object vector. It is virtual because only Protester class objects can take damage. Hence, if the other classes accidentally uses this function, nothing happens.

Earth Class

doSomething: Function does nothing. Earth does not do anything. This function exists because object class’ doSomething is pure virtual. This is to prevent any errors from happening where and object class is created.

Squirt Class

doSomething:

Function checks if squirt annoys a protester. If yes, it will deal 2 damage to a protester and the squirt object will be set to dead and removed after this turn.

If distance travelled reaches 0 (from 4), it is also set to dead.

If it still has travel distance left (more than 0), it will continue to move in its current direction by 1, and this will reduce its travel distance left by 1. If it this the end of the oil field, or a boulder or earth, it will be set to dead.

This doSomething function is unique to the squirt class.

Boulder Class

doSomething:

Function does nothing and returns if boulder is not alive.

It checks the state of the boulder.

If state is ‘stable’ and there is no earth below it, it changes to ‘waiting’ state and the tick timer starts.

If state is ‘waiting’ and tick timer has not reach 30 ticks, tick timer increases by 1 tick.

If timer reaches 30 ticks, boulder state changes to ‘falling’ and play boulder falling sound.

If ‘falling’, boulder will continue to fall until it hits earth or another boulder. Object position at each location is updated as it falls.

If boulder reaches the bottom of oil field, or it hits another boulder or earth below, it is set to dead.

This doSomething function is unique to boulder class.

Person Class

setHP: Function sets m\_hp private member of Person class. This is the health of characters. This is in the Person class because all people-like classes (protesters and tunnelman) have health.

getHP: Function returns m\_hp value. This is in the Person class because all people-like classes (protesters and tunnelman) have health.

loseHealth: Function lets person take damage. If health is less or equal to zero, person gives up. This is in the Person class because all people-like classes have health but the tunnelman and protesters have different actions upon taking damage.

giveUp: Function is pure virtual and does nothing for person class objects. This is in the person class because all persons can give up. It is pure virtual because each person type has their own things to do when giving up.

Tunnelman Class

currentSquirts: Function returns current squirts (m\_water private member).

currentGold: Function returns current gold (m\_gold private member).

currentSonar: Function returns current sonar kits (m\_sonar private member).

addSquirts: Function adds to current squirts (m\_water private member).

addGold: Function adds to current gold (m\_gold private member).

addKit: Function adds to current sonar kits m\_sonar private member).

doSomething:

Function does nothing and returns immediately if not alive.

It will remove Earth objects if tunnelman object overlaps with them and plays the SOUND\_DIG sound.

It will receive key commands every tick.

If a direction is chosen, the tunnelman will turn to the correct direction and checks if the tunnelman is not at the end of the oil field and there is no boulder in its path. If true, the tunnelman will move to the next spot in the chosen direction. If false, it stays.

If ‘space’ is chosen, it checks if tunnelman has any water. Then it will check tunnelman’s current direction and whether they are not at the end of the oil field, if true, tunnelman will shoot the squirt in the same direction and m\_water is decreased by 1.

If ‘Z’ or ‘z’ is chosen, it checks if tunnelman has any sonar kits. If yes, sonar kit will reduce by 1 and it will show hidden objects in the area.

If ‘tab’ is chosen, it checks if tunnelman has any gold nuggets. If yes, gold will reduce by 1 and a gold nugget will be dropped into the game world.

If ‘esc’ is chosen, tunnelman is set to dead.

giveUp: Function sets tunnelman to dead and plays SOUND\_PLAYER\_GIVE\_UP.

Protester Class

getState: Function returns state(m\_state private member) of protester.

setLeaveState: Function changes state(m\_state private member) to ‘leave-the-oil-field’ and sets resting ticks to 0

isProtester: Function returns true so that the program knows that this objects is in the protester class

doNothing: Protester increases its ticks since last shout, ticks since last turn and decreases its resting ticks left.

getTicksLeft: Function returns resting ticks left.

getTicksSinceShout: Function returns ticks passed since previous shout.

resetShoutTicks: Function resets ticks passed since previous shout to 0.

increaseShoutTicks: Function increases ticks passed since previous shout by 1.

resetRestTicks: Function resets current resting ticks left before protester can move again to the preset number of resting ticks when protester was first created (saved under m\_ticksToWaitBetweenMoves private member)

findPath:

Function initializes a 2D array which tracks the steps, with all the values in each cell being 9999 at the start. Function initializes a queue (exitPath) of points (structs which contain x and y values, representing location coordinates).

Output variable starts off as false.

The endpoint is pushed into exitPath and stepMatrix at the exit point is set to 0.

While the queue is not empty

currX is initialized as x value of the point at the front of the queue and currY is initialized as y value of the point at the front of the queue. Front point of queue is popped off.

If the point is already the initial point, output variable is changed to true.

Variable currSteps is assigned value of stepsMatrix at current point and 1 is added to it.

If protester is not at the left end of the oil field and there is no boulder or earth on its left and the steps matrix of left spot is not 9999, the stepsmatrix of left spot is updated with currSteps value (representing step count need to reach there and that it is visited already) and point of the left spot is pushed onto queue.

If protester is not at right end of the oil field and there is no boulder or earth on its right and the steps matrix of right spot is not 9999, the stepsmatrix of right spot is updated with currSteps value (representing step count need to reach there and that it is visited already) and point of the right spot is pushed onto queue.

If protester is not at upper end of the oil field and there is no boulder or earth on its upper spot and the steps matrix of upper spot is not 9999, the stepsmatrix of upper spot is updated with currSteps value (representing step count need to reach there and that it is visited already) and point of the upper spot is pushed onto queue.

If protester is not at lower end of the oil field and there is no boulder or earth on its lower spot and the steps matrix of lower spot is not 9999, the stepsmatrix of lower spot is updated with currSteps value (representing step count need to reach there and that it is visited already) and point of the lower spot is pushed onto queue.

Function will determine the next best step to take depending on minimum steps from around the initial spot.

Steps variable is updated to minimum steps and output(Boolean value) is returned.

Shout: Function plays SOUND\_PROTESTER\_YELL and causes tunnelman within distance of 4 and in the direction protester is facing to lose 2 health.

resetNumSquaresToMoveInCurrDir: Function sets m\_numSquaresToMoveInCurrDir to 0.

decreaseNumSquaresToMoveInCurrDir: Function decreases m\_numSquaresToMoveInCurrDir by 1.

getNumSquaresToMoveInCurrDir: : Function returns m\_numSquaresToMoveInCurrDir .

setNumSquaresToMoveInCurrDir: : Function randomly rolls new number of steps (between 8 to 60 inclusive) to move in current direction, updating m\_numSquaresToMoveInCurrDir

getTicksSinceTurn: Function returns m\_ticksSinceTurn

resetTurnTicks: Function sets m\_ticksSinceTurn to 0, usually called after protester makes a turn

increaseTurnTicks: Function increases m\_ticksSinceTurn by 1

loseHealth:

Function calls loseHealth function under the person class.

If protester health is more than 0, function plays SOUND\_PROTESTER\_ANNOYED and protester gets stunned (getStunned function is called).

If protester health is 0 or less

If damage is 2

If protester is hardcore, score increases by 250.

Else, score increases by 100.

If damage is 100, score increases by 500.

getStunned:

Function updates the resting ticks of protester using a formula of max of 50 or 100 – level time 10 and changes protester state to rest. Protesters cannot do anything while he is stunned.

isHardcore: Functions returns false. It is virtual because function will return true if it is a hardcore protester. Function is in protester class because it is called in another function which is also in the protester class.

giveUp: Function set state of protesters to ‘leave-the-oil-field’ and plays SOUND\_PROTESTER\_GIVE\_UP.

RegularProtester Class

doSomething:

Function does nothing and returns if protester is dead

If resting ticks left is more than 0, function calls doNothing function and returns.

If protester is in ‘leave-the-oil-field’ state,

If protester is already at exit point, it is set dead and returns

It finds a path to the exit and moves accordingly.

If it can shout at tunnelman,

If it has been 15 ticks since last shout,

It will shout at the tunnelman, reset shout ticks and calls endTurn function and returns

Else it will just call endTurn function and return

If it is in the same row or column as the tunnel man(canFace), it will face the tunnelman and move towards it.

If it is not facing tunnelman, it will decrease number of squares to move in current direction

If number of squares to move in current direction drops to 0 or less, it will generate new direction and sets new number of squares to move in new direction

Else if ticks since last turn is more than 200, it can make a perpendicular turn at the appropriate spot (no boulder or earth and not at the end of the oil field). Turn tick is then reset to 0.

Protester will continue to move in the current direction by 1 step, if it is not obstructed by anything or not at the end of oil field.

Function calls endTurn function.

getBribed: Plays SOUND\_PROTESTER\_FOUND\_GOLD, increases score by 25 and changes protester state to ‘leave-the-oil-field’ state

HardcoreProtester Class

doSomething:

Function does nothing and returns if protester is dead

If resting ticks left is more than 0, function calls doNothing function and returns.

If protester is in ‘leave-the-oil-field’ state,

If protester is already at exit point, it is set dead and returns

It finds a path to the exit and moves accordingly.

If it can shout at tunnelman,

If it has been 15 ticks since last shout,

It will shout at the tunnelman, reset shout ticks and calls endTurn function and returns

Else it will just call endTurn function and return

If it is near tunnelman(nearTunnelman function),

It will initialize variable M = 16 + level times 2

It will calculate minimum steps to tunnelman using findPath function.

If minimum steps smaller than or equal to M, protester will change direction to move in the shortest path to tunnelman.

endTurn function is called and return.

If it is in the same row or column as the tunnel man(canFace), it will face the tunnelman and move towards it.

If it is not facing tunnelman, it will decrease number of squares to move in current direction

If number of squares to move in current direction drops to 0 or less, it will generate new direction and sets new number of squares to move in new direction

Else if ticks since last turn is more than 200, it can make a perpendicular turn at the appropriate spot (no boulder or earth and not at the end of the oil field). Turn tick is then reset to 0.

Protester will continue to move in the current direction by 1 step, if it is not obstructed by anything or not at the end of oil field.

Function calls endTurn function.

getBribed: Plays SOUND\_PROTESTER\_FOUND\_GOLD, increases score by 50 and stuns protester

Goodies Class

setVisible: Function changes object to be visible by calling GraphObject’s setVisible function and updates m\_visible private member

is\_visible: Function returns m\_visible private member

setPickup: Function changes whether goodie can be picked up by tunnelman or protesters (using Boolean)

can\_pickup: Function returns m\_tunnelpickup private member

setTicks: Function sets m\_restTicksLeft (how many ticks a goodie has left to be alive)

getTicks: Function returns m\_restTicksLeft

reduceTicks: Function decrease m\_restTicksLeft by 1

Sonar Class

doSomething:

Function does nothing and returns if sonar is dead.

If it is near tunnelman(nearTunnelman function), sonar will die, SOUND\_GOT\_GOODIE is played, score increase by 75 and tunnelman obtains a sonar kit.

Else if sonar runs out of ticks, it will be set to dead

Else, it will decrease ticks.

Water Class

doSomething:

Function does nothing and returns if water is dead.

If it is near tunnelman(nearTunnelman function), water will die, SOUND\_GOT\_GOODIE is played, score increase by 100 and tunnelman obtains a water for squirts.

Else if water runs out of ticks, it will be set to dead

Else, it will decrease ticks.

Oil Class

doSomething:

Function does nothing and returns if oil is dead.

If it is near tunnelman and was not visible before, it will turn visible to tunnelman and returns.

If tunnelman gets even closer, it is picked up by tunnelman and plays SOUND\_FOUND\_OIL, increases score by 1000 and changes object position at that spot to empty. Barrel left decreases by 1.

Gold Class

doSomething:

Function does nothing and returns immediately if dead.

If it is near tunnelman and was not visible before, it will turn visible to tunnelman and returns.

If tunnelman gets even closer and it can be picked up by tunnelman(Boolean m\_tunnelpickup is true), it is picked up, SOUND\_GOT\_GOODIE is played, score increases by 10 and tunnelman calls addGold function

If it can only be picked up by protester, it will find the nearest protester (using findProtester function).

If there are no protesters nearby,

If rest ticks left is 0 or less, it will be set to dead and return;

Else, rest ticks reduce by 1.

Else, it will be picked up and bribe the protester.

StudentWorld Class

Init:

Functions first initializes earth array where each position in array is a nullptr(if no earth object exists in the spot in the game) or a new Earth object. objectMatrix is also initialized with ‘E’ value to represent earth.

objectMatrix is updated at spots without any earth (the initial tunnel down)

Number of boulders are initialized.

For each boulder, x and y coordinate are randomly generated

If there are no overlaps,

new Boulder object is pushed into objectV(object vector)

any earth in that spot is removed

objectMatrix at that spot is updated with ‘B’

Else, this loop repeats

Number of gold is initialized.

For each gold, x and y coordinate are randomly generated

If there are no overlaps,

new Gold object is pushed into objectV(object vector)

objectMatrix at that spot is updated with ‘G’

Else, this loop repeats

Number of oil is initialized.

For each oil, x and y coordinate are randomly generated

If there are no overlaps,

new Oil object is pushed into objectV(object vector)

objectMatrix at that spot is updated with ‘L’

Else, this loop repeats

ticksSinceLastProtester is initiazlied

ticksForNextProtester = ticksSinceLastProtester;

nProtesters is 0;

return GWSTATUS\_CONTINUE\_GAME;

digEarth:

output is initialized as false;

for each member in earthArray

if it is not nullptr

if earth object overlaps with current spot

earthArray member is deleted

earthArray member is nullptr

output changes to true

return output

move:

Function first updates display text

Initializes hardcoreProb and G

If new protesters can be added,

It randomly generates value from 1 to 100

If value is hardcoreProb or less,

New HardcoreProtester object is pushed to object V

Else New RegularProtester object is pushed to object V

Then, it determines if new goodies are to be added, and adds either sonar or water in probability ration of 1:4.

If player is still alive, player calls doSomething()

For each object in object,

If object is alive, it calls doSomething()

If player dies, return GWSTATUS\_PLAYER\_DIED

If player finishes level, return GWSTATUS\_FINISHED\_LEVEL

Call removeDeadGameObjects() to remove dead objects

If player dies, return GWSTATUS\_PLAYER\_DIED

If player finishes level, return GWSTATUS\_FINISHED\_LEVEL

Return GWSTATUS\_CONTINUE\_GAME

cleanUp: Function deletes all the earth objects, then tunnelman and then all the other objects found in object vector.

setDisplayText: Function returns string which will show the status of the game such as health, score, gold found, etc.

outputFormatter: Helper function to format output of setDisplayText to follow guidelines provided in specs.

finishedLevel: If there are no barrels left, returns true. Else, false.

shootSquirt: Function creates a new squirt object in the object vector and plays SOUND\_PLAYER\_SQUIRT.

isEarth: Function checks if there is earth on that spot given in parameters

isBoulder: Function checks if there is boulder on that spot given in parameters

setObjectPosition: Function sets which object is on that particular location. Characters are E for Earth, T for Empty, B for Boulder, G for Gold, L for Oil

getObjectPosition: Function returns which object is on that particular location

overlap: Function checks if two objects overlap.

canShout: Function checks direction that protester is facing and determines if it is facing tunnelman and it is close enough to shout.

getTunnelman: Function returns pointer to tunnelman object in world.

facingTunnelman: Function checks if current spot means that this object is on the same row or column to tunnelman.

nearTunnelman: Function checks if current spot means that this object is near to the tunnelman where near is defined by distance provided.

newDirRandom: Function generates a new random direction for protester.

showHidden: Function reveals objects which were not visible in the area

dropGold: Function creates new gold object and adds it to object vector

annoyProtester: Function iterates through object vector and causes any protester to give up if current position if within 3 squares of a protester

annoyTunnelman: : Function causes tunnelman to give up if current position if within 3 squares of tunnelman

decreaseBarrel: Function decreases total number of barrels left in game

findProtester: Function used to determine nearest protester to current point and will return pointer to protester if it is close enough

removeDeadGameObjects: Function checks alive status of each object in object vector and erases it if it is dead

canAddProtester: Function checks if ticks since last protester is greater than waiting time for next protester and that total number of protesters in game is still fewer than level generated max number, it will then increase protester and return true.

canAddWater: Function checks if any earth array is overlapping on current spot

playerDied: Function checks if player has died and decreases lives in game if player died

2.

a. I realized that after I lose all my lives, and I got my score, when I press enter, the game crashes seomtimes.

b. The game takes too long to calculate the exit path every tick once a regular protester is bribed.

3. Assumptions:

a. I treated all radiuses mentioned in the spec as in a circle, using Pythagoras to calculate distance

b. The order of some of the steps I assumed to make it make sense.

4. Test Cases

Tunnelman class:

I played the game to determine if tunnelman can move around and that all of his different actions work. He can shoot squirts, drop gold nuggets and activate sonar to reveal hidden objects, and pick up items which will update the display text. The escape button also allows him to lose a life. The tunnelman can only dig within the confines of the oil field. Proper sounds are made when digging, picking up various items, being annoyed by protesters and finishing a level.

Earth class:

When the game starts, I observe that the oil field was generated properly and that all non-boulder, non-person objects (like oil and gold) are not seen in some spots. When close to such objects, objects reveal themselves in front of the earth, just like the sample game. Digging sounds work. I also checked that tunnels get dug into when tunnelman walks through earth objects. Protesters only go through tunnels and not through the earth.

Squirt class:

When tunnelman shoots a squirt, it can move a distance of 4 before disappearing. It animates properly and stuns a protester accordingly. It also disappears if it hits a boulder or earth.

Boulder class:

Boulder falls until it hits earth or another boulder while there are not earth blocking its path below. Animation works. Sound is made at the correct time and it will damage the tunnelman and any protester if it is under the boulder while it falls.

Sonar class:

Sonar kits appear at the upper left-hand corner at random times and can be picked up by the tunnelman only. It disappears upon being picked up. Display text is properly updated (kit increases by 1) and the correct sound is played. When sonar kit is used, it is able to reveal hidden objects, same as the sample gameplay.

Water class:

Water appears randomly throughout the gameplay at spots without earth. It can be picked up by the tunnelman only. Display text is properly updated (water increases by 5) and the correct sound is played.

Oil class:

Oil barrels are hidden behind the earth and are not visible until the tunnelman is close by. It can be picked up by the tunnelman only, and display text is properly updated (barrels left – 1) when it is picked up. It is revealed upon sonar kit activation if it is in the area.

Gold class:

Gold nuggets are hidden behind the earth and are not visible until the tunnelman is close by. It can be picked up by the tunnelman, and display text is properly updated (gold increases by 1). It is revealed upon sonar kit activation if it is in the area.

Regular Protester class:

Protester follows tunnelman and moves around on its own if it is not close to tunnelman. Protester is able to turn at a junction. When in the same line (row or column) as tunnelman, it is able to change directionto face tunnelman and “chase” the tunnelman. Once tunnelman is not in the same line, it can change its action. It can pick up nugget and the score increases accordingly. However, I have issues with my shortest path algorithm being too long to solve and the game lags whenever a protester is in leave state. It can shout properly to damage the tunnelman when close.

Hard Protester class:

Protester follows tunnelman and moves around on its own if it is not close to tunnelman. Protester can turn at a junction. When in the same line (row or column) as tunnelman, it is able to change directionto face tunnelman and “chase” the tunnelman. When the tunnelman is close enough, it is also able to detect the tunnelman and find the path to chase it. It can shout properly to damage the tunnelman when close. Once tunnelman is not nearby, it can change its action. It can pick up nuggets, and the score increases accordingly. However, I have issues with my shortest path algorithm being too long to solve and the game lags whenever a protester is in leave state.