1.

StudentWorld

virtual int init()

This function was used to initialize the objects in the game. It initialized all of the variables for the StudentWorld class as well as added all of the actors to the vector that I stored them in.

virtual int move();

This function was described in the spec and was used to attempt to move the TunnelMan in a variety of ways, whether it was through earth, into a boulder, into a barrel, a piece of gold, or any other goodie. I started by attempting to move as if there was nothing there, and then went along with each type of move and coded the specifics depending on what kind of goodie or protestor or whatever other actor it was.

bool removeEarth(int x,int y);

This function was used to remove pieces of the earth from the screen. I used a helper function in order to do this which is described later in the report.

virtual void cleanUp();

This is a helper function for my destructor. It goes through each actor in my vector and deletes each one. It uses an iterator to go through the vector and delete each actor.

TunnelMan\* getTM() {return m\_tm;}

This function simply returns a pointer to the Tunnel Man object.

std::vector<Base\*> getActs() {return m\_act;}

This function returns my vector of actors.

void addActs();

This function is a helper function for move. Before my player moves, it adds actors to the screen. It goes through the actor array and implements them into the program.

Earth\* getEarth(int a, int b);

This returns a pointer to a location on the map where there may be earth.

int getTick() {return m\_ticks;}

This returns the number of ticks that have passed.

void addTick() {++m\_ticks;}

This adds a tick.

void resetTick() {m\_ticks = 0;}

This resets the number of ticks.

int getTicksb4() {return m\_ticksb4Prot;}

This returns the number of ticks before a protester arrived.

void minusTicksb4() {m\_ticksb4Prot--;}

This decrements the previous variable.

void resetTicksB4();

This resets the previous described variable.

int getNumProt() {return m\_numProt;}

This returns the number of protesters.

void minusNumProt() {m\_numProt--;}

This decrements the number of protesters

void gatherBarrel() { m\_barrels--; }

This function decrements the number of barrels.

// helper functions

void createPath(GraphObject::Direction arr[][VIEW\_HEIGHT], int x, int y);

This function attempts to create a path from a protester to either the exit or the tunnel man. This does so by attempting each direction with a BFS.

bool closeToTM(int x, int y, double r);

This returns whether the protester is within the radius of tunnel man or not.

bool annoyProt(int x, int y, int a);

This functions annoys all protesters within the radius.

bool isMoveValid(int x, int y, GraphObject::Direction dir);

This returns whether or not a move is valid.

bool isCoordValid(int x, int y);

void createActor(Base\* actor) {m\_act.push\_back(actor);}

int howFarFromTM(int x, int y);

std::string formatWords(std::string &s, int lvl, int health, int lives, int gold, int water, int oil, int sonar, int scr);

bool removeHlp(int a, int b);

double dist(double x, double y, double x\_2, double y\_2);

void buryDead();

bool compareBldHlp(int a, int b);

void getPos(int &x, int &y);

void getWaterPos(int &x, int &y);

void getBoulderPos(int &x, int &y);

void setWords();

bool boulder(int x, int y);

bool earth(int x, int y);

bool closeToBld(int x, int y);

void lightUp(int x, int y);

Protester\* closestProt(int x, int y);

GraphObject::Direction exitDir(int x, int y) {return m\_exitpath[x][y];}

GraphObject::Direction tmDir(int x, int y) {return m\_tmpath[x][y];}

Test Cases:  
Earth:  
I tested the earth class by playing the game many times, attempting to go through the earth, simply waiting for goodies and protesters to spawn on top of earth to see if it would happen. I attempted to go out of bounds next to a piece of earth to see what would happen. I shot squirts at earth to see if they would stop moving.

TunnelMan:  
Most of my testing for the TunnelMan class involved simply moving around with the Tunnel Man to see if he reacted correctly to his environment. I ran into walls to see if I would be thrown out of bounds, I went through the earth to see if I interacted with it properly. I ran near and into protesters just to see what would happen. I essential tried to do everything I would try to do if I was playing the game for the first time.