Project 3 report.docx

1.

Well.h

void display(Screen& screen, int x, int y): Displays the screen

Piece.h

class Coord(Represents a x and y coord of a position on the screen);

int getX() const: returns the x coord

int getY() const; returns the y coord

int setX(int m\_x): sets the x coord to be m\_x

int setY(int m\_y); sets the y coord to be m\_y

class Piece:

Piece();

~Piece() {}

PieceType accessType() const; returns the Piecetype of the Piece

int returnOrientation() const; //Returns the current orientation of the piece

char mSquares[12][19]; Stores the location of the piece regarding its position of the well

Coord coord[4]; Stores the coordinates of the screen of each square. Each square in a piece has a coordinate.

void rotate(); Rotates the piece

Game.h

class Game:

Game(int width, int height);

~Game();

void play();

bool playOneLevel();

void displayPrompt(std::string s);

void displayStatus();

void print(); //Prints out the piece on the screen every time it moves down

void eraseChar(int col, int row): Clears the screen at the position indicated

void moveCharDown(int col, int row); //Moves the pos of the char at pos col, row down one row

char getChar(int col, int row) const: Returns the char at the pos indicated

double findMaximum(double x, double y) const: Finds max of 2 doubles

void updateScreen(char c): When a piece comes to rest, this function keeps track of where it lies to rest

void updateScreenVapor(char c); //Serves the same function as updateScreen but for PIECE\_VAPOR

void updateScreenFoam(char c); //Serves the same function as updateScreen but for PIECE\_FOAM

void display(char c); //Prints at the char at the current coordinates of the current piece

void displayVapor(char c); //Same function as display but for PIECE\_VAPOR

void displayFoam(char c); //Same function as display but for PIECE\_FOAM

void move(string direction); //Moves the piece down, left, or right

bool overlap() const: Checks to see if a piece overlaps with an existing block in the grid

void translateCommands(char ch): Translate keyboard commands using a switch statement to either move the current piece or end the game

void shiftLeft(): Moves the piece left

void shiftRight(): Moves the piece right

void shiftDown(): Moves the piece down

void spaceDown(); Moves the piece down as far as it can go

void rotateShape(): rotates the piece

void changeBool(): Checks to see if the top row is filled. If so it sets playOneLevel to false and ends the game

bool rowisFilled(int row) const; Checks to see if a row is filled

void rowisFilledHelper(int row); if a row is filled. This function is called to erase the row and drop any blocks in rows above it down one row

void displaynextPiece(); Displays the next piece that is about to be dropped on the screen

void clearForNextPiece(); Clears the display that displaynextPiece() presented

2. Due to time constraints, I did not use inheritance in any of my classes. In addition, I was not able to implement the special features of the Foam or the Vapor pieces, so in the program they were treated as regular pieces. Also, whenever I moved the piece left or right, it would cause the dollar signs that the piece collided with to disappear, although the dollar signs were still considered to be there by the program, meaning that other blocks couldn’t pass through the positions that the empty blocks were in. In addition, the space button did not function properly, being unable to move the block all the way down without being stuck in an infinite loop, forcing me to comment out its implementation so that the game could run smoothly. Also, the display prompt did not work properly, being unable to display the score and number of rows left properly after they were updated.

3. For the classes, in order to display and update the position of the pieces on the screen. I was unsure how to implement it. Since I displayed and updated the position of the pieces based on how many squares they had. I eventually decided to use separate functions to display the PIECE\_FOAM and PIECE\_VAPOR functions, since they only had 1 and 2 squares, respectively.