

Project Report

We implemented the A star algorithm in our game as it calculates the shortest path from the source (in our scenario, the enemies) to the destination (player). We added the UI “god mode on” when the player took the power pellet. We allowed the enemies to start moving after the player's second move to give him an advantage over the enemies. We allowed the player to move diagonally and then disabled that feature as it made it so hard to beat the game. Some of the obstacles we faced while doing this algorithm were that we only sent the first position of the player, so the enemies' route did not change. Another thing was that we sent the position of only one of the enemies, so the other was not moving. We managed to overcome these errors after some trials and errors. Finally, we added the new levels, and the game window class reads the documents automatically, so this saved us time while adding the new levels.

This is a summary on who did what:

Hussein:

- Is valid function
- Is unblocked function
- Is destination function
- Calculated h value
- Trace path
- Added new levels

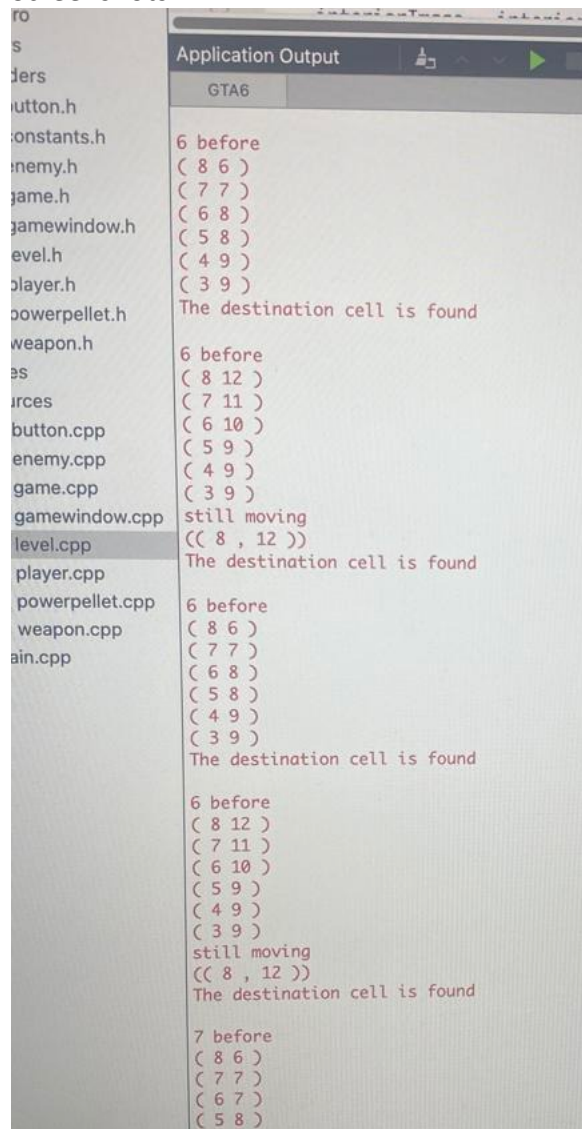
Farida:

- Cell Struct
- Trace path
- Applied a star algorithm function
- Applied a star algorithm to one enemy

Mazen:

- Changed God mode to be in a separate thread
- Changed the enemy's movement to be turn-based
- Applied a star algorithm to all enemies rather than only one
- Added God mode label on U

Screenshots:



```
ro
s
ders
utton.h
onstants.h
nemy.h
game.h
gamewindow.h
evel.h
player.h
powerpellet.h
weapon.h
es
rces
button.cpp
enemy.cpp
game.cpp
gamewindow.cpp
level.cpp
player.cpp
powerpellet.cpp
weapon.cpp
ain.cpp

Application Output
GTA6

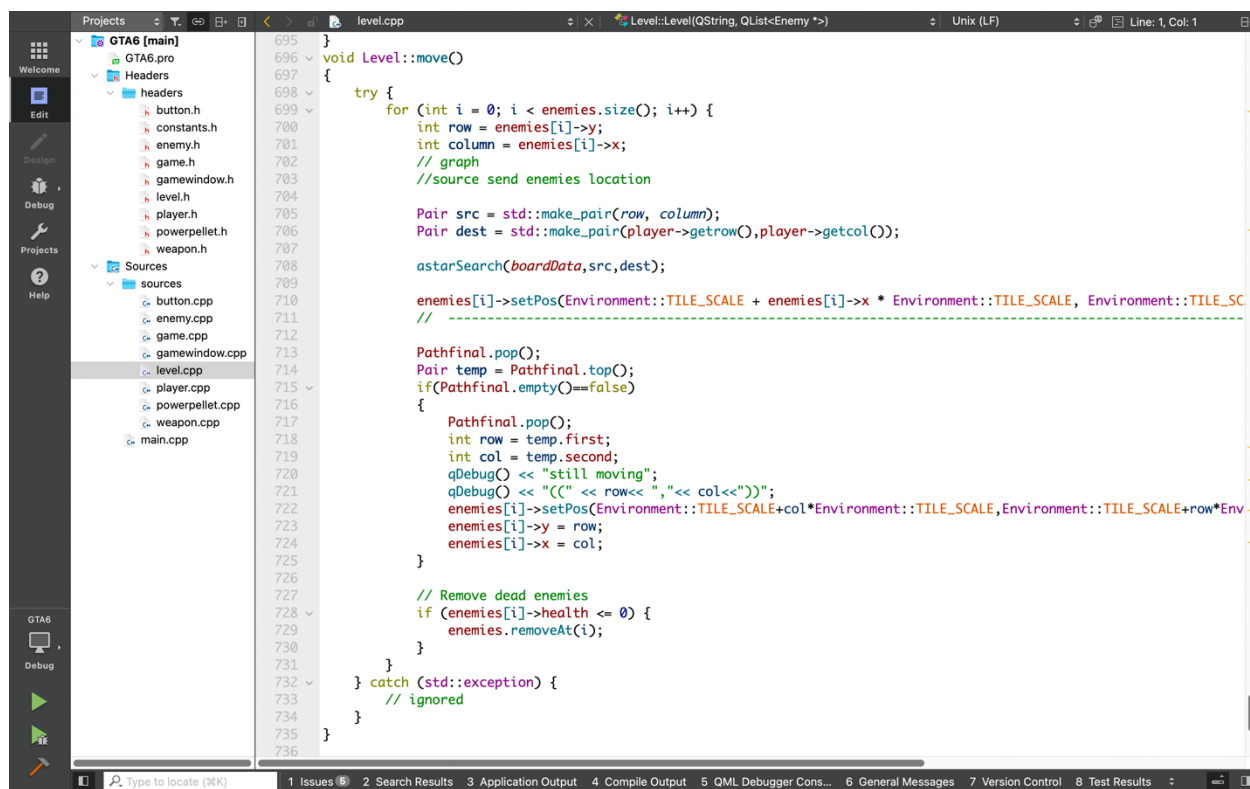
6 before
( 8 6 )
( 7 7 )
( 6 8 )
( 5 8 )
( 4 9 )
( 3 9 )
The destination cell is found

6 before
( 8 12 )
( 7 11 )
( 6 10 )
( 5 9 )
( 4 9 )
( 3 9 )
still moving
(( 8 , 12 ))
The destination cell is found

6 before
( 8 12 )
( 7 11 )
( 6 10 )
( 5 9 )
( 4 9 )
( 3 9 )
still moving
(( 8 , 12 ))
The destination cell is found

7 before
( 8 6 )
( 7 7 )
( 6 7 )
( 5 8 )
```

Debug the console to check if the algorithm works or not.



```
695 }
696 void Level::move()
697 {
698     try {
699         for (int i = 0; i < enemies.size(); i++) {
700             int row = enemies[i]->y;
701             int column = enemies[i]->x;
702             // graph
703             //source send enemies location
704
705             Pair src = std::make_pair(row, column);
706             Pair dest = std::make_pair(player->getrow(), player->getcol());
707
708             astarSearch(boardData, src, dest);
709
710             enemies[i]->setPos(Environment::TILE_SCALE + enemies[i]->x * Environment::TILE_SCALE, Environment::TILE_SCALE + enemies[i]->y * Environment::TILE_SCALE);
711             // -----
712
713             Pathfinal.pop();
714             Pair temp = Pathfinal.top();
715             if(Pathfinal.empty()==false)
716             {
717                 Pathfinal.pop();
718                 int row = temp.first;
719                 int col = temp.second;
720                 qDebug() << "still moving";
721                 qDebug() << "(" << row << ", " << col << ")";
722                 enemies[i]->setPos(Environment::TILE_SCALE+col*Environment::TILE_SCALE, Environment::TILE_SCALE+row*Environment::TILE_SCALE);
723                 enemies[i]->y = row;
724                 enemies[i]->x = col;
725             }
726
727             // Remove dead enemies
728             if (enemies[i]->health <= 0) {
729                 enemies.removeAt(i);
730             }
731         }
732     } catch (std::exception) {
733         // ignored
734     }
735 }
736 }
```

Move function