BUBBLE SHOOTER

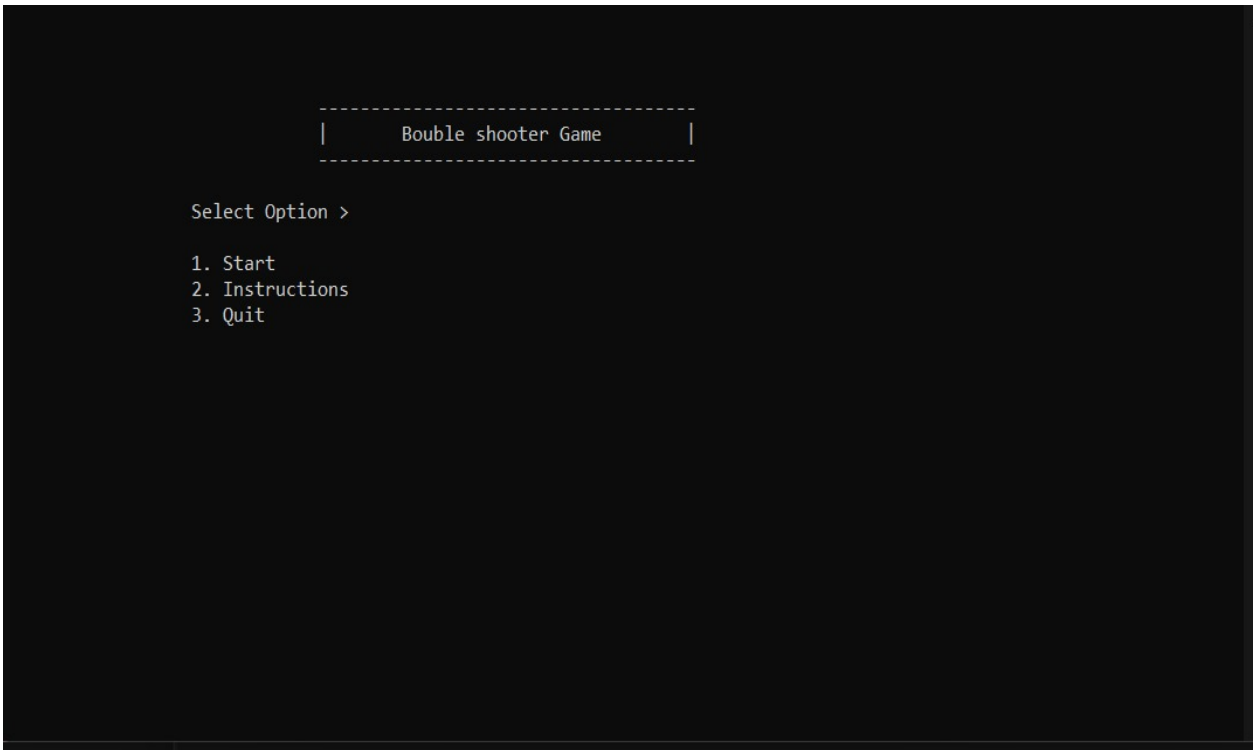
(CONSOLE GAME)

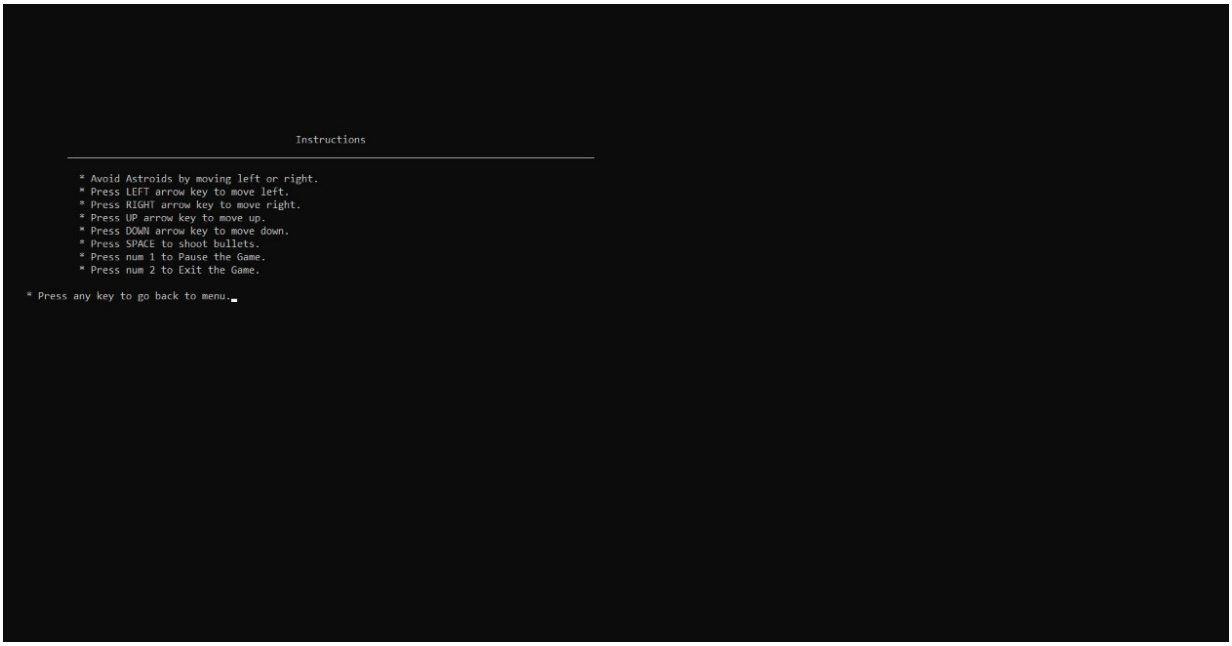
C++ PROGRAM

Team CodE4

Content:

1.Basic Documentation



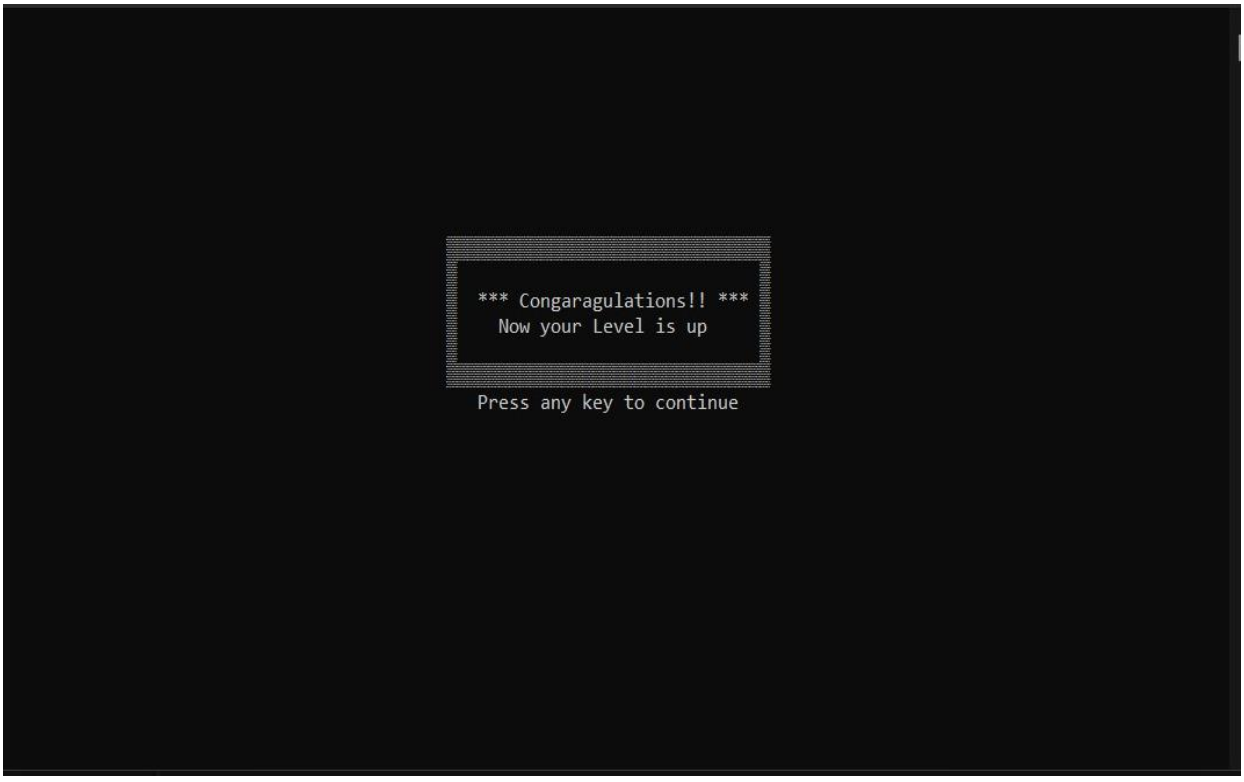



Press any key to play
Devoloped by Team Code4 PROJECT


HP:
ENERGY:
SCORE:0
Controlls
1) Press 1 to pause
2) Press 2 to exit


Select option
Press 0 to exit and any key to continue


Get ready..after 3 secs game will start.


*** Congaragulations!! ***
Now your Level is up
Press any key to continue

2.Instruction on how to play Bubble Shooter

This is a single player game which has 2 levels. Bubbles keep falling randomly, and you need to shoot the bubbles to score. For each bubble attack, you get 10 marks. After you score 100 marks, the 1st level is completed. Then you can move to level 2 as your preference. In the 2nd level, there is no time limit, so you can score marks until you lose all your lives. At the beginning of level 1, you are given 3 lives (represented as hearts ♥) and in level 2, 4 lives. For each life, you get 5 energy levels. When bubbles fall on to the shooter, you lose energy levels one by one. When all three lives are over, you lose the game.

In the first interface of the game, the player will be guided to press 1 to start the game, 2 to get instructions, and 3 to quit.

Instructions to play game -

Avoid Asteroids by moving left or right.

Press LEFT arrow key to move left.

Press RIGHT arrow key to move right

Press UP arrow key to move up.

Press DOWN arrow key to move down.

Press SPACE to shoot bullets.

Press num 1 to Pause the Game.

Press num 2 to Exit the Game.

3.Problems had with the report/ challenges

We had to face a lot of problems and challenges while building the code for the game.

One of them was that it was difficult to find the required inbuilt function. Another problem we faced was that we could not print some of the emojis and symbols we wanted using the functions of the c++ language, so we used the printf function of the c language for that. And this work was done online. So, there were some difficulties to work together as a team and it took a long time.

4.What we have done different

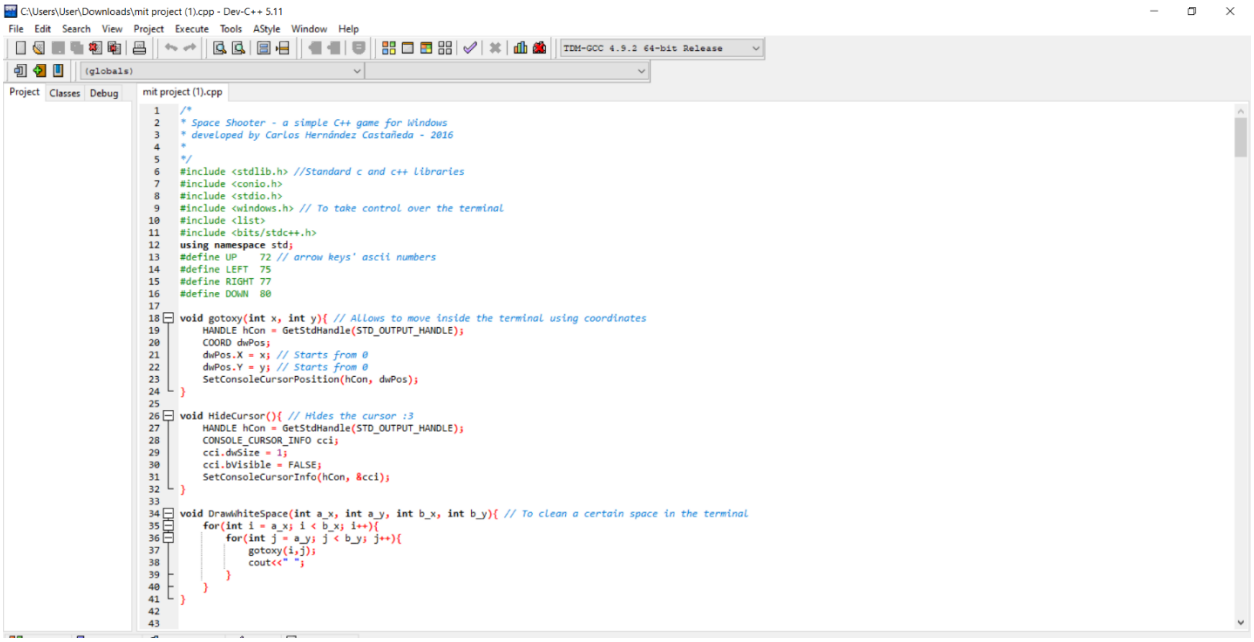
- We can add graphic effects.
- Add sound effects to the game.

  Those things increase the attractiveness of the game

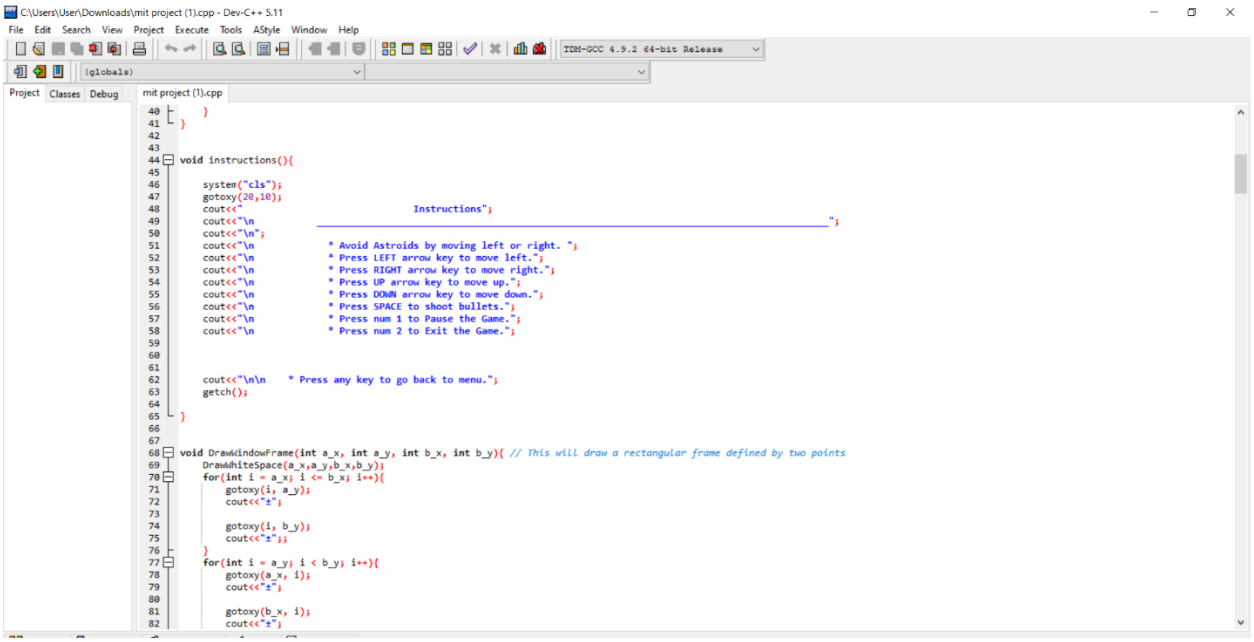
- We can add more levels and make the game more fun

  by giving things like gifts (or rewards) at those levels.

5.Code



```
/*
* Space Shooter - a simple C++ game for Windows
* developed by Carlos Hernández Castañeda - 2016
*
*/
#include <stdlib.h> //Standard c and c++ libraries
#include <conio.h>
#include <stdio.h>
#include <windows.h> // To take control over the terminal
#include <list>
#include <bits/stdc++.h>
using namespace std;
#define UP    72 // arrow keys' ascii numbers
#define LEFT  75
#define RIGHT 77
#define DOWN  80

void gotoxy(int x, int y){ // Allows to move inside the terminal using coordinates
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x; // Starts from 0
    dwPos.Y = y; // Starts from 0
    SetConsoleCursorPosition(hCon, dwPos);
}

void HideCursor(){ // Hides the cursor :3
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO cci;
    cci.dwSize = 1;
    cci.bVisible = FALSE;
    SetConsoleCursorInfo(hCon, &cci);
}

void DrawWhiteSpace(int a_x, int a_y, int b_x, int b_y){ // To clean a certain space in the terminal
    for(int i = a_x; i < b_x; i++){
        for(int j = a_y; j < b_y; j++){
            gotoxy(i,j);
            cout<<" ";
        }
    }
}
```



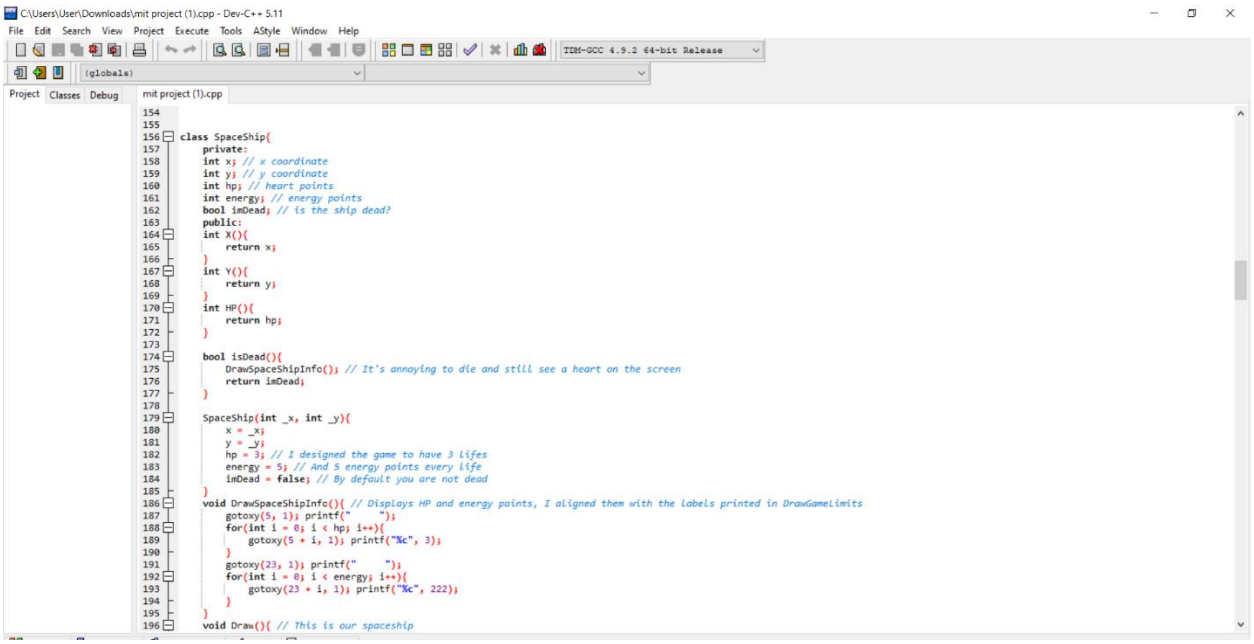
```
    }
}


void instructions(){

    system("cls");
    gotoxy(20,10);
    cout<<"                    Instructions";
    cout<<"\n                  ____________________________________________________________________";
    cout<<"\n";
    cout<<"\n            * Avoid Astroids by moving left or right. ";
    cout<<"\n            * Press LEFT arrow key to move left.";
    cout<<"\n            * Press RIGHT arrow key to move right.";
    cout<<"\n            * Press UP arrow key to move up.";
    cout<<"\n            * Press DOWN arrow key to move down.";
    cout<<"\n            * Press SPACE to shoot bullets.";
    cout<<"\n            * Press num 1 to Pause the Game.";
    cout<<"\n            * Press num 2 to Exit the Game.";


    cout<<"\n\n    * Press any key to go back to menu.";
    getch();
}


void DrawWindowFrame(int a_x, int a_y, int b_x, int b_y){ // This will draw a rectangular frame defined by two points
    DrawWhiteSpace(a_x,a_y,b_x,b_y);
    for(int i = a_x; i <= b_x; i++){
        gotoxy(i, a_y);
        cout<<"±";

        gotoxy(i, b_y);
        cout<<"±";;
    }
    for(int i = a_y; i < b_y; i++){
        gotoxy(a_x, i);
        cout<<"±";

        gotoxy(b_x, i);
        cout<<"±";
```

C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11

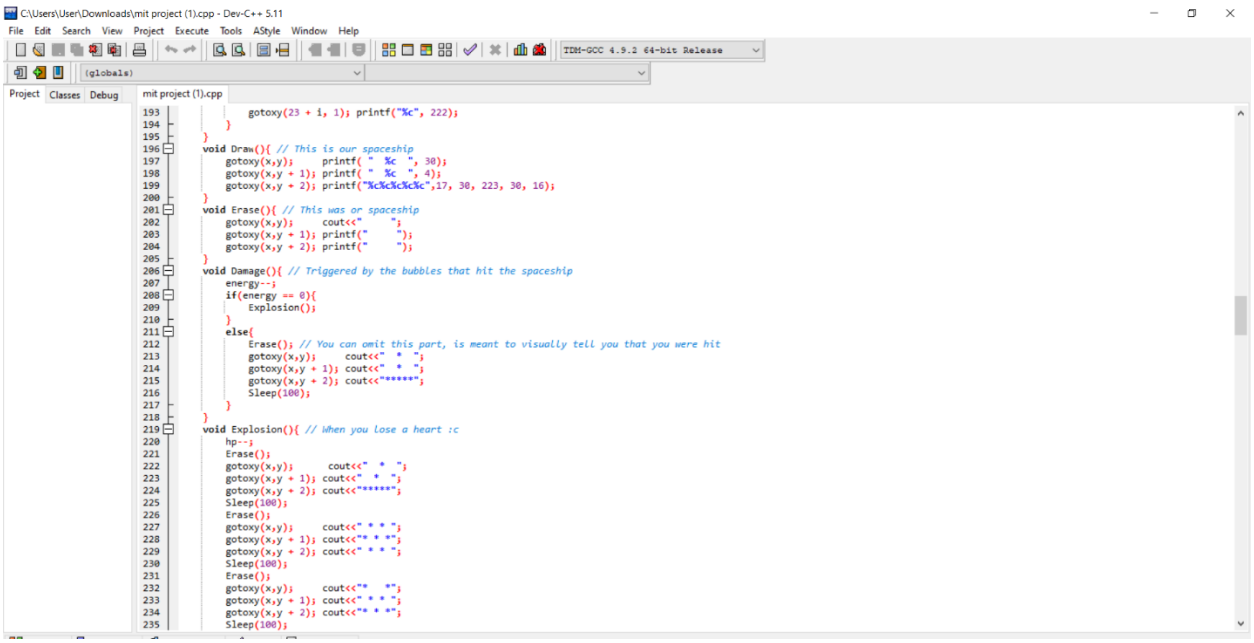
```cpp
            cout<<"±";

            gotoxy(b_x, i);
            cout<<"±";
        }
}
void DrawGameLimits(){ // Draws the game limits, and information that doesn't have to be printed repeatedly.
    DrawWindowFrame(0,2,79,23); // The default size of the Windows terminal is 25 rows x 80 columns
    gotoxy( 2,1); cout << "HP:";
    gotoxy(16,1); cout << "ENERGY:";
    gotoxy(50,1); cout << "SCORE:";
    gotoxy(82,15); cout<<"        Controlls ";
    gotoxy(82,16); cout<<" -------------------- ";
    gotoxy(82,18); cout<<" 1) Press 1 to pause ";
    gotoxy(82,19); cout<<" 2) Press 2 to exit ";

}
void WelcomeMessage(){ // The main title, I tried to center it as best as I could
    int x = 13;
    int y = 6;
    gotoxy(x, y  ); cout <<" ____        _   _   _          ____  _                 _                 ";
    gotoxy(x, y+1); cout <<"| __ )_   _| |_| |_| |__       / ___|| |__   ___   ___ | |_ ___ _ __ ";
    gotoxy(x, y+2); cout <<"|  _ \ | | | '_ \ '_ \ |/ _ \  \___ \| '_ \ / _ \ / _ \| __/ _ \ '__|";
    gotoxy(x, y+3); cout <<"| |_) | |_| | |_) | |_) | |  __/  ___) | | | | (_) | (_) | ||  __/ |   ";
    gotoxy(x, y+4); cout <<"|____/ \__,_|_.__/|_.__/|_|\___| |____/|_| |_|\___/ \___/ \__\___|_|   ";

    gotoxy(x, y+6); cout <<"                          Press any key to play";
    gotoxy(x, y+7); cout <<"                   Devoloped by Team Code4 PROJECT";
}
void GameOverDefeatMessage(){ // When you lose the game you see this in screen
    int a_x = 45;
    int a_y = 9;
    int b_x = a_x + 30;
    int b_y = a_y + 5;
    DrawWindowFrame(a_x,a_y,b_x,b_y);
    gotoxy(a_x+1,a_y+2); cout << "          YOU LOSE !!!";
}
void GameOverVictoryMessage(){ // When you win the game you see this in screen
    int a_x = 45;
    int a_y = 9;
    int b_x = a_x + 30;
    int b_y = a_y + 5;
    DrawWindowFrame(a_x,a_y,b_x,b_y);
```

C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11

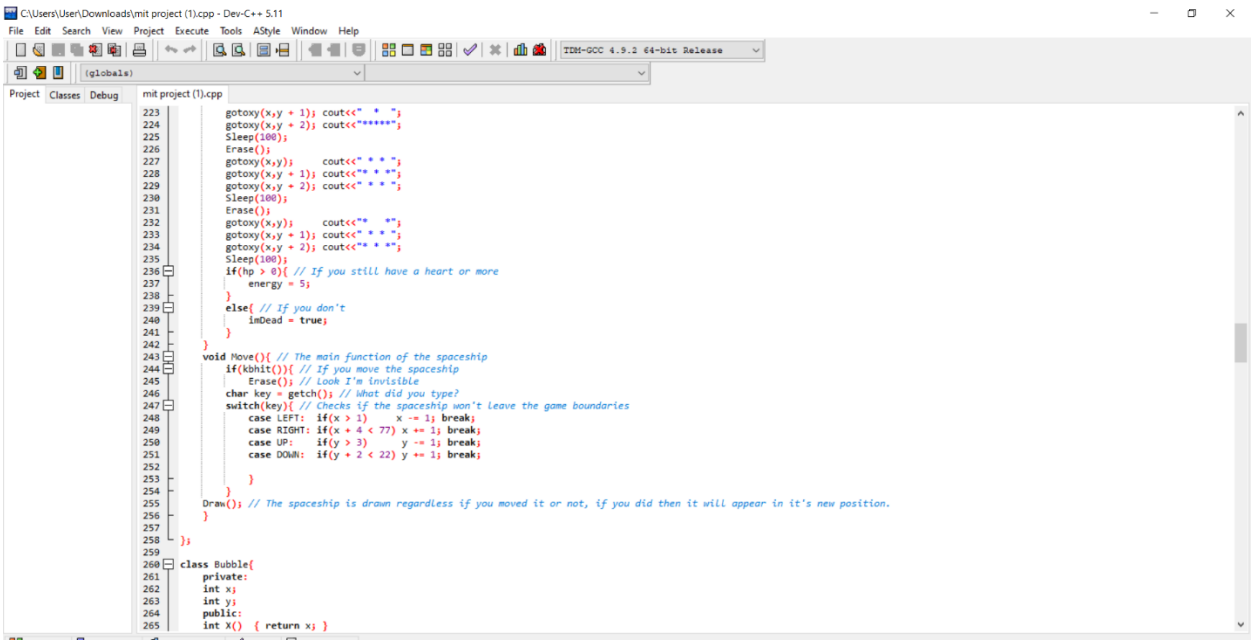
```cpp
    int a_y = 9;
    int b_x = a_x + 30;
    int b_y = a_y + 5;
    DrawWindowFrame(a_x,a_y,b_x,b_y);
    gotoxy(a_x+1,a_y+2); cout << "    YOU WIN.";
}
void LevelUpMessage(){ // When you win the game you see this in screen
    int a_x = 45; //30
    int a_y = 9;//11
    int b_x = a_x + 30; //23
    int b_y = a_y + 5; //4
    DrawWindowFrame(a_x,a_y,b_x,b_y);
    gotoxy(a_x+1,a_y+2); cout << "   *** Congaragulations!! *** ";
    gotoxy(a_x+1,a_y+3); cout << "     Now your Level is up";

    gotoxy(a_x+1,a_y+4);cout<<"\n\n                                             Press any key to continue\n"<<endl;
}
void SpecialMessage(){ // A special message for your special needs
    DrawWhiteSpace(1,1,80,24);
    gotoxy(30,11);cout<< "                _________________________________  ";
    gotoxy(30,12);cout<<"               |                                 |" ;
    gotoxy(30,13);cout<<"               |         Thanks for playing !!!  |  ";
    gotoxy(30,14);cout<<"               |_________________________________|  ";
    //gotoxy(30,11); printf("Thanks for playing!! ");
}
void GameclosedMassege(){
    int a_x = 45;
    int a_y = 9;
    int b_x = a_x + 30;
    int b_y = a_y + 5;
    DrawWindowFrame(a_x,a_y,b_x,b_y);
    gotoxy(a_x+1,a_y+2); cout<<"        Game closed!! ";
    Sleep(5000);
    SpecialMessage();

}


class SpaceShip{
    private:
    int x; // x coordinate
    int y; // y coordinate
    int hp; // heart points
```

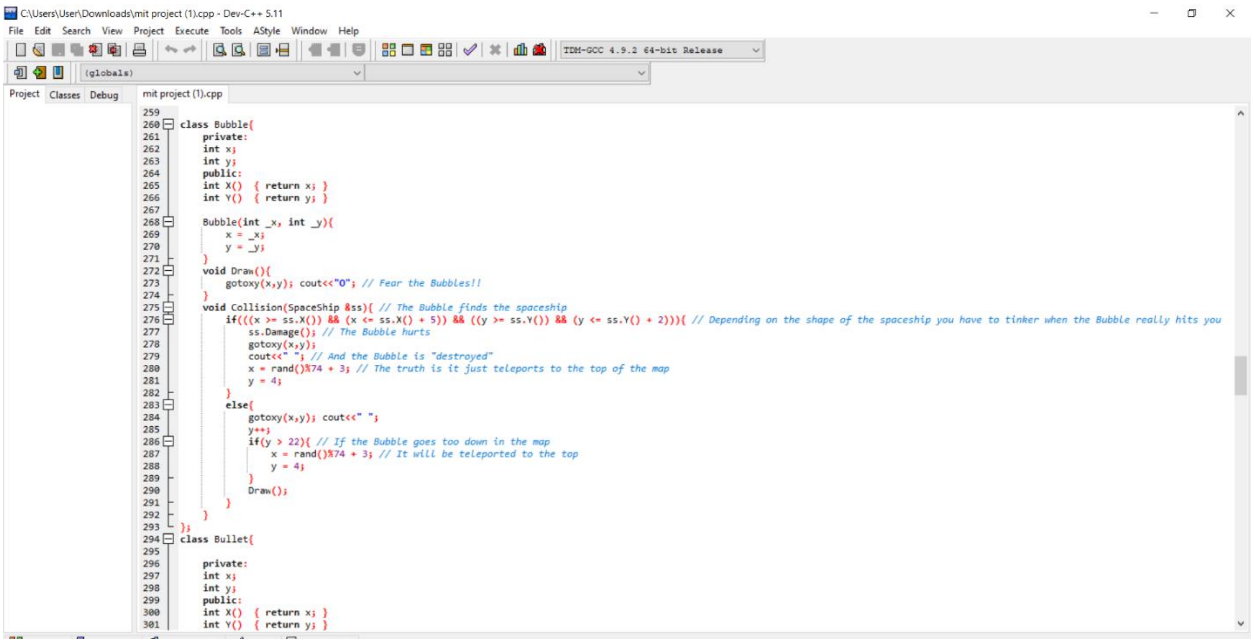


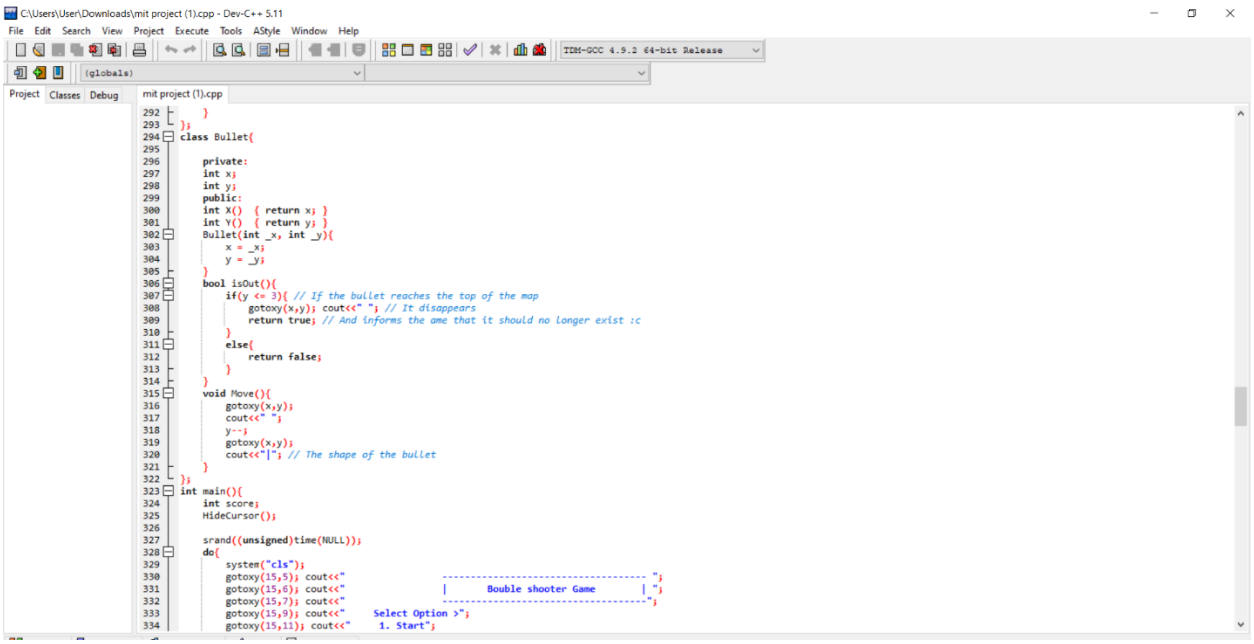
```
154
155
156 class SpaceShip{
157     private:
158     int x; // x coordinate
159     int y; // y coordinate
160     int hp; // heart points
161     int energy; // energy points
162     bool imDead; // is the ship dead?
163     public:
164     int X(){
165         return x;
166     }
167     int Y(){
168         return y;
169     }
170     int HP(){
171         return hp;
172     }
173
174     bool isDead(){
175         DrawSpaceShipInfo(); // It's annoying to die and still see a heart on the screen
176         return imDead;
177     }
178
179     SpaceShip(int _x, int _y){
180         x = _x;
181         y = _y;
182         hp = 3; // I designed the game to have 3 lifes
183         energy = 5; // And 5 energy points every life
184         imDead = false; // By default you are not dead
185     }
186     void DrawSpaceShipInfo(){ // Displays HP and energy points, I aligned them with the labels printed in DrawGamelimits
187         gotoxy(5, 1); printf("     ");
188         for(int i = 0; i < hp; i++){
189             gotoxy(5 + i, 1); printf("%c", 3);
190         }
191         gotoxy(23, 1); printf("     ");
192         for(int i = 0; i < energy; i++){
193             gotoxy(23 + i, 1); printf("%c", 222);
194         }
195     }
196     void Draw(){ // This is our spaceship
```



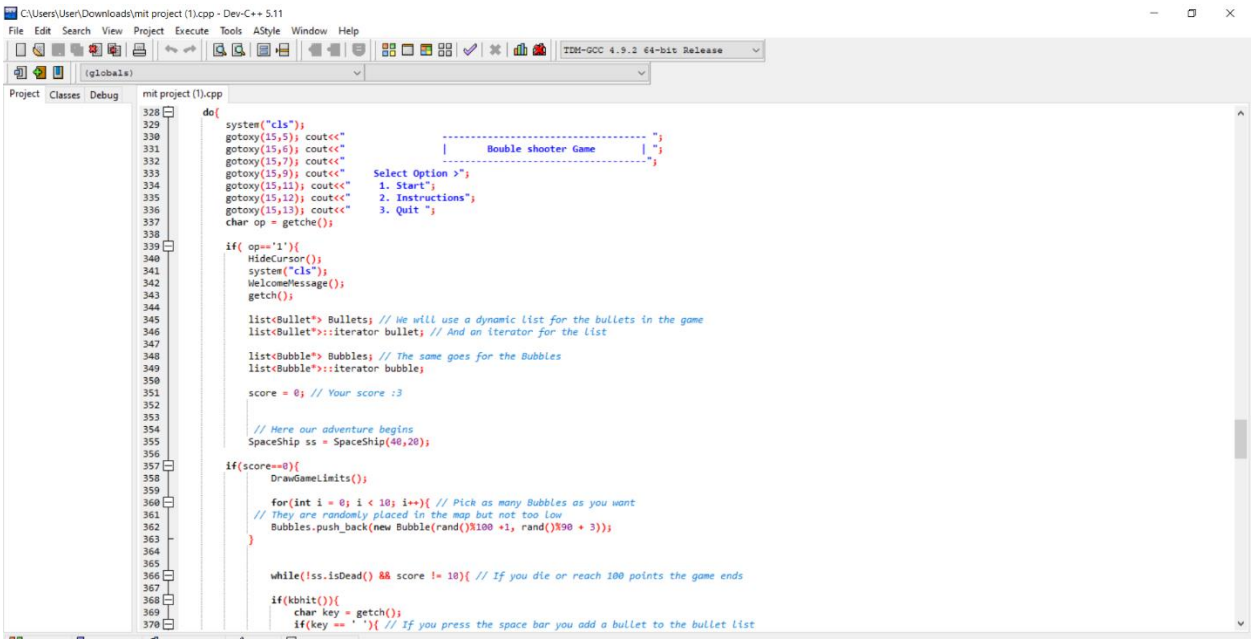
```
193             gotoxy(23 + i, 1); printf("%c", 222);
194         }
195     }
196     void Draw(){ // This is our spaceship
197         gotoxy(x,y);     printf("  %c  ", 30);
198         gotoxy(x,y + 1); printf(" %c%c ", 4);
199         gotoxy(x,y + 2); printf("%c%c%c%c%c",17, 30, 223, 30, 16);
200     }
201     void Erase(){ // This was or spaceship
202         gotoxy(x,y);     cout<<"     ";
203         gotoxy(x,y + 1); printf("     ");
204         gotoxy(x,y + 2); printf("     ");
205     }
206     void Damage(){ // Triggered by the bubbles that hit the spaceship
207         energy--;
208         if(energy == 0){
209             Explosion();
210         }
211         else{
212             Erase(); // You can omit this part, is meant to visually tell you that you were hit
213             gotoxy(x,y);     cout<<"  *  ";
214             gotoxy(x,y + 1); cout<<" * * ";
215             gotoxy(x,y + 2); cout<<"*****";
216             Sleep(100);
217         }
218     }
219     void Explosion(){ // When you lose a heart :c
220         hp--;
221         Erase();
222         gotoxy(x,y);     cout<<"  *  ";
223         gotoxy(x,y + 1); cout<<" * * ";
224         gotoxy(x,y + 2); cout<<"*****";
225         Sleep(100);
226         Erase();
227         gotoxy(x,y);     cout<<" * * ";
228         gotoxy(x,y + 1); cout<<"* * *";
229         gotoxy(x,y + 2); cout<<" * * ";
230         Sleep(100);
231         Erase();
232         gotoxy(x,y);     cout<<"*   *";
233         gotoxy(x,y + 1); cout<<" * * ";
234         gotoxy(x,y + 2); cout<<"* * *";
235         Sleep(100);
```


C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
mit project (1).cpp
gotoxy(x,y + 1); cout<<" * ";
gotoxy(x,y + 2); cout<<"******";
Sleep(100);
Erase();
gotoxy(x,y); cout<<" * * ";
gotoxy(x,y + 1); cout<<"* * *";
gotoxy(x,y + 2); cout<<" * * ";
Sleep(100);
Erase();
gotoxy(x,y); cout<<"* *";
gotoxy(x,y + 1); cout<<" * * ";
gotoxy(x,y + 2); cout<<"* * *";
Sleep(100);
if(hp > 0){ // If you still have a heart or more
energy = 5;
}
else{ // If you don't
imDead = true;
}
}
void Move(){ // The main function of the spaceship
if(kbhit()){ // If you move the spaceship
Erase(); // Look I'm invisible
char key = getch(); // What did you type?
switch(key){ // Checks if the spaceship won't leave the game boundaries
case LEFT: if(x > 1) x -= 1; break;
case RIGHT: if(x + 4 < 77) x += 1; break;
case UP: if(y > 3) y -= 1; break;
case DOWN: if(y + 2 < 22) y += 1; break;
}
}
Draw(); // The spaceship is drawn regardless if you moved it or not, if you did then it will appear in it's new position.
}
};
class Bubble{
private:
int x;
int y;
public:
int X() { return x; }

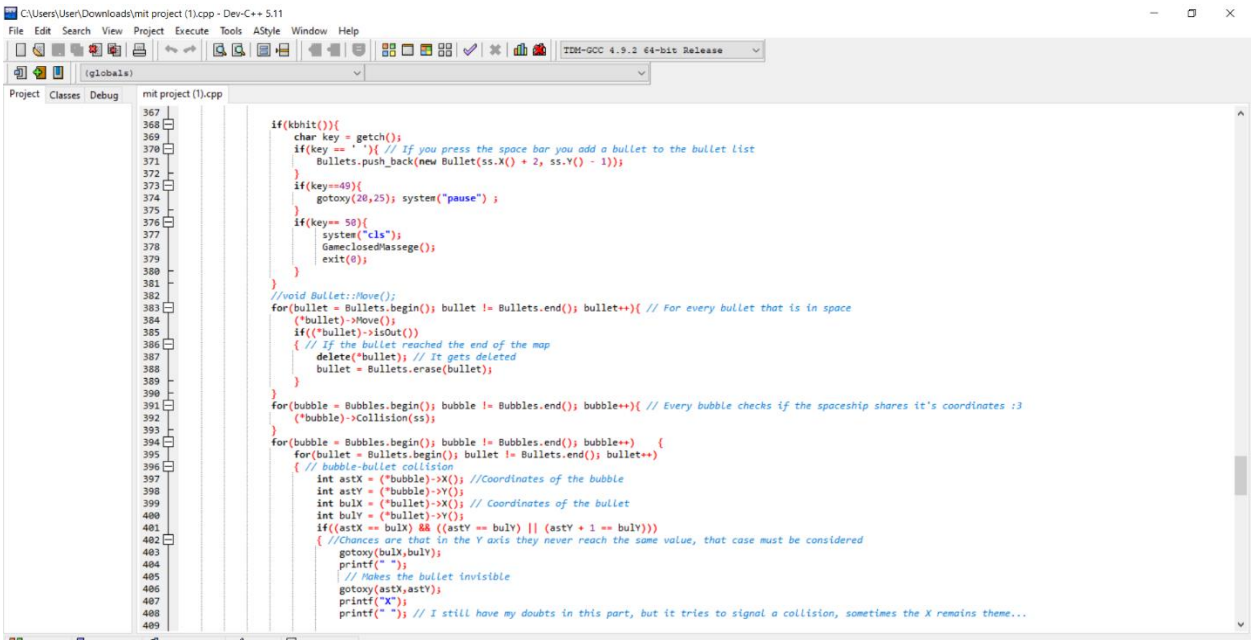

C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
mit project (1).cpp
class Bubble{
private:
int x;
int y;
public:
int X() { return x; }
int Y() { return y; }
Bubble(int _x, int _y){
x = _x;
y = _y;
}
void Draw(){
gotoxy(x,y); cout<<"O"; // Fear the Bubbles!!
}
void Collision(SpaceShip &ss){ // The Bubble finds the spaceship
if(((x >= ss.X()) && (x <= ss.X() + 5)) && ((y >= ss.Y()) && (y <= ss.Y() + 2))){ // Depending on the shape of the spaceship you have to tinker when the Bubble really hits you
ss.Damage(); // The Bubble hurts
gotoxy(x,y);
cout<<" "; // And the Bubble is "destroyed"
x = rand()%74 + 3; // The truth is it just teleports to the top of the map
y = 4;
}
else{
gotoxy(x,y); cout<<" ";
y++;
if(y > 22){ // If the Bubble goes too down in the map
x = rand()%74 + 3; // It will be teleported to the top
y = 4;
}
Draw();
}
}
};
class Bullet{
private:
int x;
int y;
public:
int X() { return x; }
int Y() { return y; }

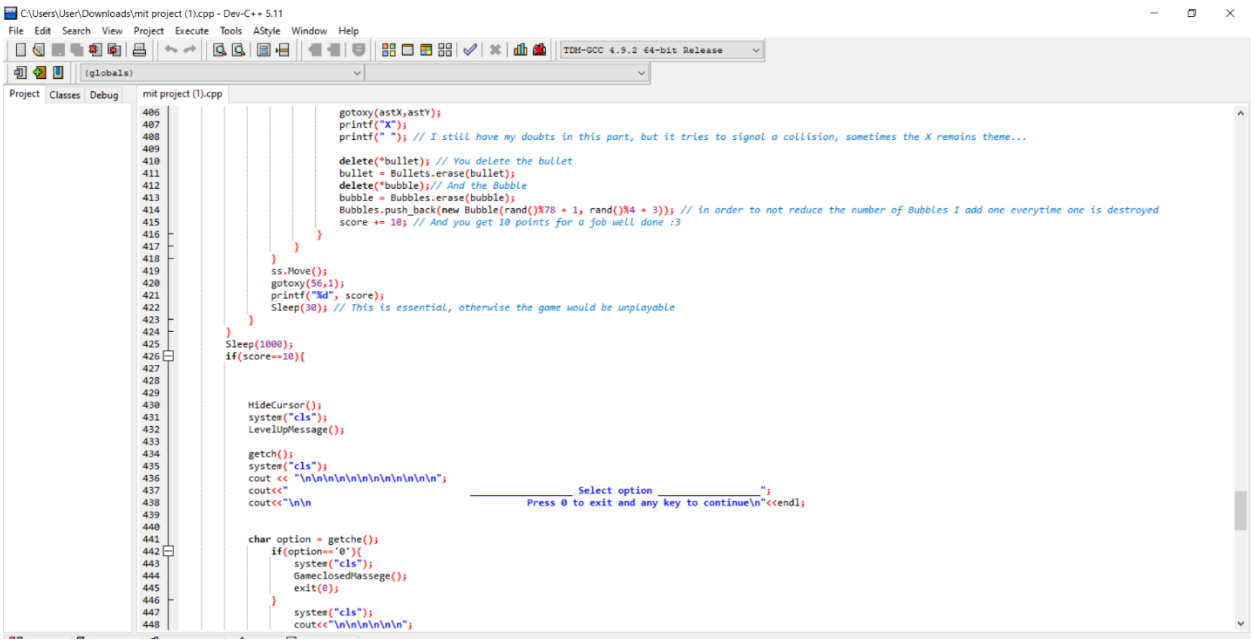


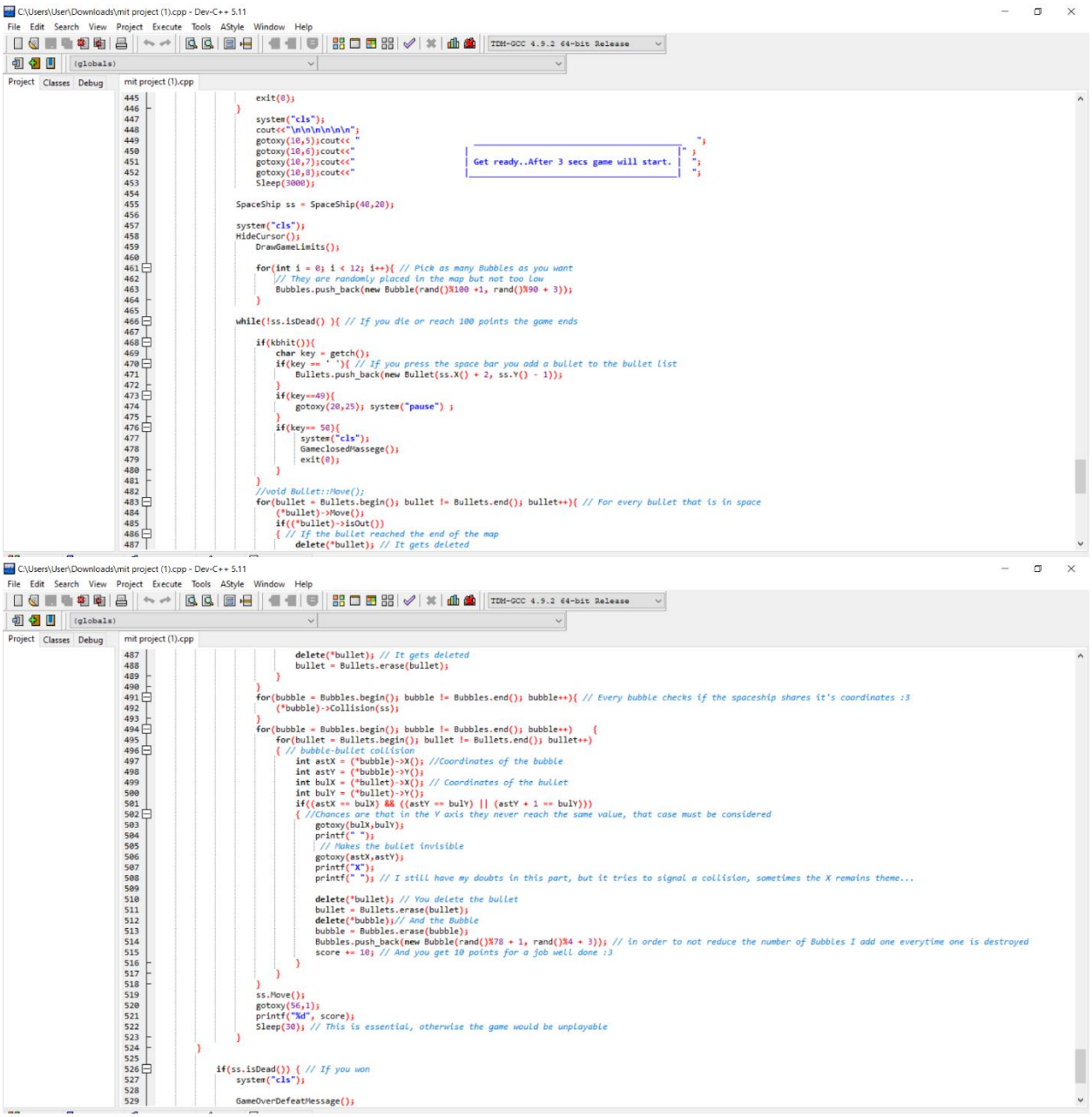
```
    }
};
class Bullet{

    private:
    int x;
    int y;
    public:
    int X()  { return x; }
    int Y()  { return y; }
    Bullet(int _x, int _y){
        x = _x;
        y = _y;
    }
    bool isOut(){
        if(y <= 3){ // If the bullet reaches the top of the map
            gotoxy(x,y); cout<<" "; // It disappears
            return true; // And informs the ame that it should no longer exist :c
        }
        else{
            return false;
        }
    }
    void Move(){
        gotoxy(x,y);
        cout<<" ";
        y--;
        gotoxy(x,y);
        cout<<"|"; // The shape of the bullet
    }
};
int main(){
    int score;
    HideCursor();

    srand((unsigned)time(NULL));
    do{
        system("cls");
        gotoxy(15,5); cout<<"                ------------------------------------ ";
        gotoxy(15,6); cout<<"                |       Bouble shooter Game        | ";
        gotoxy(15,7); cout<<"                ------------------------------------";
        gotoxy(15,9); cout<<"     Select Option >";
        gotoxy(15,11); cout<<"     1. Start";
```



```
    do{
        system("cls");
        gotoxy(15,5); cout<<"                ------------------------------------ ";
        gotoxy(15,6); cout<<"                |       Bouble shooter Game        | ";
        gotoxy(15,7); cout<<"                ------------------------------------";
        gotoxy(15,9); cout<<"     Select Option >";
        gotoxy(15,11); cout<<"     1. Start";
        gotoxy(15,12); cout<<"     2. Instructions";
        gotoxy(15,13); cout<<"     3. Quit ";
        char op = getche();

        if( op=='1'){
            HideCursor();
            system("cls");
            WelcomeMessage();
            getch();

            list<Bullet*> Bullets; // We will use a dynamic list for the bullets in the game
            list<Bullet*>::iterator bullet; // And an iterator for the list

            list<Bubble*> Bubbles; // The same goes for the Bubbles
            list<Bubble*>::iterator bubble;

            score = 0; // Your score :3


             // Here our adventure begins
            SpaceShip ss = SpaceShip(40,20);

        if(score==0){
                DrawGameLimits();

                for(int i = 0; i < 10; i++){ // Pick as many Bubbles as you want
              // They are randomly placed in the map but not too low
                Bubbles.push_back(new Bubble(rand()%100 +1, rand()%90 + 3));
            }


            while(!ss.isDead() && score != 10){ // If you die or reach 100 points the game ends

            if(kbhit()){
                char key = getch();
                if(key == ' '){ // If you press the space bar you add a bullet to the bullet list
```


C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
mit project (1).cpp
if(kbhit()){
char key = getch();
if(key == ' '){ // If you press the space bar you add a bullet to the bullet list
Bullets.push_back(new Bullet(ss.X() + 2, ss.Y() - 1));
}
if(key==49){
gotoxy(20,25); system("pause") ;
}
if(key== 50){
system("cls");
GameclosedMassege();
exit(0);
}
}
//void Bullet::Move();
for(bullet = Bullets.begin(); bullet != Bullets.end(); bullet++){ // For every bullet that is in space
(*bullet)->Move();
if((*bullet)->isOut())
{ // If the bullet reached the end of the map
delete(*bullet); // It gets deleted
bullet = Bullets.erase(bullet);
}
}
for(bubble = Bubbles.begin(); bubble != Bubbles.end(); bubble++){ // Every bubble checks if the spaceship shares it's coordinates :3
(*bubble)->Collision(ss);
}
for(bubble = Bubbles.begin(); bubble != Bubbles.end(); bubble++) {
for(bullet = Bullets.begin(); bullet != Bullets.end(); bullet++)
{ // bubble-bullet collision
int astX = (*bubble)->X(); //Coordinates of the bubble
int astY = (*bubble)->Y();
int bulX = (*bullet)->X(); // Coordinates of the bullet
int bulY = (*bullet)->Y();
if((astX == bulX) && ((astY == bulY) || (astY + 1 == bulY)))
{ //Chances are that in the Y axis they never reach the same value, that case must be considered
gotoxy(bulX,bulY);
printf(" ");
// Makes the bullet invisible
gotoxy(astX,astY);
printf("X");
printf(" "); // I still have my doubts in this part, but it tries to signal a collision, sometimes the X remains theme...

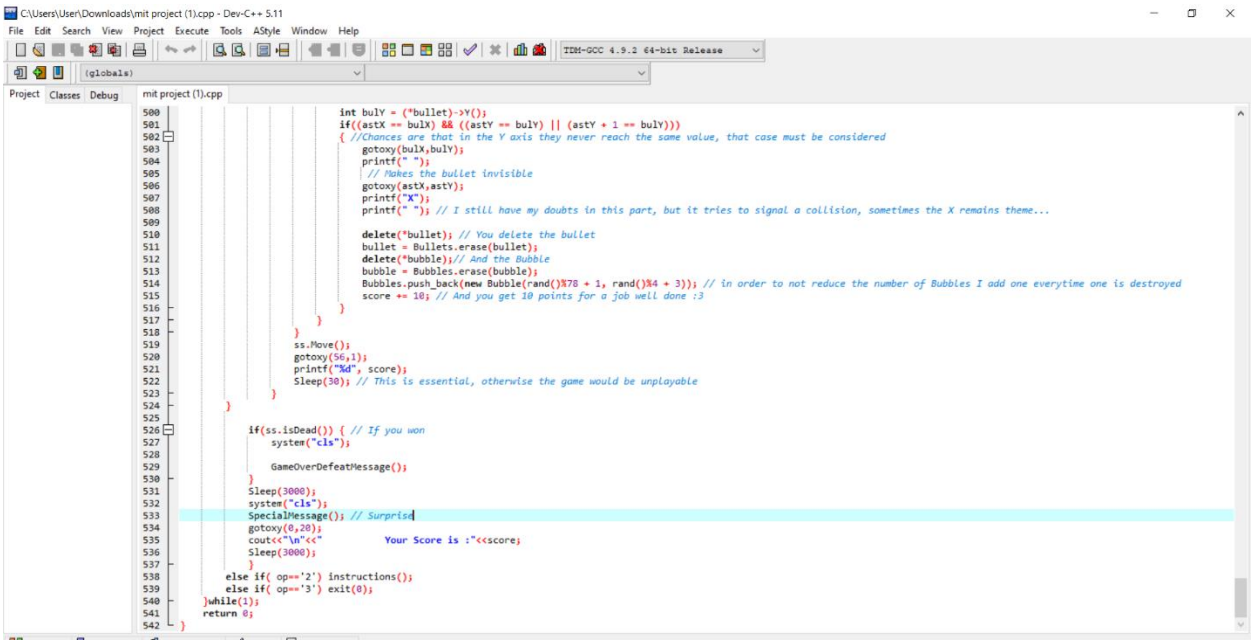

C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
mit project (1).cpp
gotoxy(astX,astY);
printf("X");
printf(" "); // I still have my doubts in this part, but it tries to signal a collision, sometimes the X remains theme...
delete(*bullet); // You delete the bullet
bullet = Bullets.erase(bullet);
delete(*bubble);// And the Bubble
bubble = Bubbles.erase(bubble);
Bubbles.push_back(new Bubble(rand()%78 + 1, rand()%4 + 3)); // in order to not reduce the number of Bubbles I add one everytime one is destroyed
score += 10; // And you get 10 points for a job well done :3
}
}
}
ss.Move();
gotoxy(56,1);
printf("%d", score);
Sleep(30); // This is essential, otherwise the game would be unplayable
}
}
Sleep(1000);
if(score==10){
HideCursor();
system("cls");
LevelUpMessage();
getch();
system("cls");
cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n";
cout<<" _____ Select option _____";
cout<<"\n\n Press 0 to exit and any key to continue\n"<<endl;
char option = getche();
if(option=='0'){
system("cls");
GameclosedMassege();
exit(0);
}
system("cls");
cout<<"\n\n\n\n\n\n\n";



```
                exit(0);
            }
                system("cls");
                cout<<"\n\n\n\n\n\n";
                gotoxy(10,5);cout<< "  _____________________________________  ";
                gotoxy(10,6);cout<<"                                        | ";
                gotoxy(10,7);cout<<"| Get ready..After 3 secs game will start. | ";
                gotoxy(10,8);cout<<"|_____________________________________| ";
                Sleep(3000);

            SpaceShip ss = SpaceShip(40,20);

            system("cls");
            HideCursor();
                DrawGameLimits();

                for(int i = 0; i < 12; i++){ // Pick as many Bubbles as you want
                    // They are randomly placed in the map but not too low
                    Bubbles.push_back(new Bubble(rand()%100 +1, rand()%90 + 3));
                }

            while(!ss.isDead() ){ // If you die or reach 100 points the game ends

                if(kbhit()){
                    char key = getch();
                    if(key == ' '){ // If you press the space bar you add a bullet to the bullet list
                        Bullets.push_back(new Bullet(ss.X() + 2, ss.Y() - 1));
                    }
                    if(key==49){
                        gotoxy(20,25); system("pause") ;
                    }
                    if(key== 50){
                        system("cls");
                        GameclosedMassege();
                        exit(0);
                    }
                }
                //void Bullet::Move();
                for(bullet = Bullets.begin(); bullet != Bullets.end(); bullet++){ // For every bullet that is in space
                    (*bullet)->Move();
                    if((*bullet)->isOut())
                    { // If the bullet reached the end of the map
                        delete(*bullet); // It gets deleted
```



```
                        delete(*bullet); // It gets deleted
                        bullet = Bullets.erase(bullet);
                    }
                }
                for(bubble = Bubbles.begin(); bubble != Bubbles.end(); bubble++){ // Every bubble checks if the spaceship shares it's coordinates :3
                    (*bubble)->Collision(ss);
                }
                for(bubble = Bubbles.begin(); bubble != Bubbles.end(); bubble++)    {
                    for(bullet = Bullets.begin(); bullet != Bullets.end(); bullet++)
                    { // bubble-bullet collision
                        int astX = (*bubble)->X(); //Coordinates of the bubble
                        int astY = (*bubble)->Y();
                        int bulX = (*bullet)->X(); // Coordinates of the bullet
                        int bulY = (*bullet)->Y();
                        if((astX == bulX) && ((astY == bulY) || (astY + 1 == bulY)))
                        { //Chances are that in the Y axis they never reach the same value, that case must be considered
                            gotoxy(bulX,bulY);
                            printf(" ");
                            // Makes the bullet invisible
                            gotoxy(astX,astY);
                            printf("X");
                            printf(" "); // I still have my doubts in this part, but it tries to signal a collision, sometimes the X remains theme...

                            delete(*bullet); // You delete the bullet
                            bullet = Bullets.erase(bullet);
                            delete(*bubble);// And the Bubble
                            bubble = Bubbles.erase(bubble);
                            Bubbles.push_back(new Bubble(rand()%78 + 1, rand()%4 + 3)); // in order to not reduce the number of Bubbles I add one everytime one is destroyed
                            score += 10; // And you get 10 points for a job well done :3
                        }
                    }
                }
                ss.Move();
                gotoxy(56,1);
                printf("%d", score);
                Sleep(30); // This is essential, otherwise the game would be unplayable
            }
        }

        if(ss.isDead()) { // If you won
            system("cls");

            GameOverDefeatMessage();
```


C:\Users\User\Downloads\mit project (1).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
mit project (1).cpp
int bulY = (*bullet)->Y();
if((astX == bulX) && ((astY == bulY) || (astY + 1 == bulY)))
{ //Chances are that in the Y axis they never reach the same value, that case must be considered
gotoxy(bulX,bulY);
printf(" ");
// Makes the bullet invisible
gotoxy(astX,astY);
printf("X");
printf(" "); // I still have my doubts in this part, but it tries to signal a collision, sometimes the X remains theme...
delete(*bullet); // You delete the bullet
bullet = Bullets.erase(bullet);
delete(*bubble);// And the Bubble
bubble = Bubbles.erase(bubble);
Bubbles.push_back(new Bubble(rand()%78 + 1, rand()%4 + 3)); // in order to not reduce the number of Bubbles I add one everytime one is destroyed
score += 10; // And you get 10 points for a job well done :3
ss.Move();
gotoxy(56,1);
printf("%d", score);
Sleep(30); // This is essential, otherwise the game would be unplayable
if(ss.isDead()) { // If you won
system("cls");
GameOverDefeatMessage();
Sleep(3000);
system("cls");
SpecialMessage(); // Surprise
gotoxy(0,20);
cout<<"\n"<<" Your Score is :"<<score;
Sleep(3000);
else if(op=='2') instructions();
else if(op=='3') exit(0);
}while(1);
return 0;

Thank You