

Software Development Report

by Benjamin Phillips

Submission date: 05-Apr-2019 06:04PM (UTC+0100)

Submission ID: 104230648

File name: 366845108.Software_Development_Report.docx (5.26M)

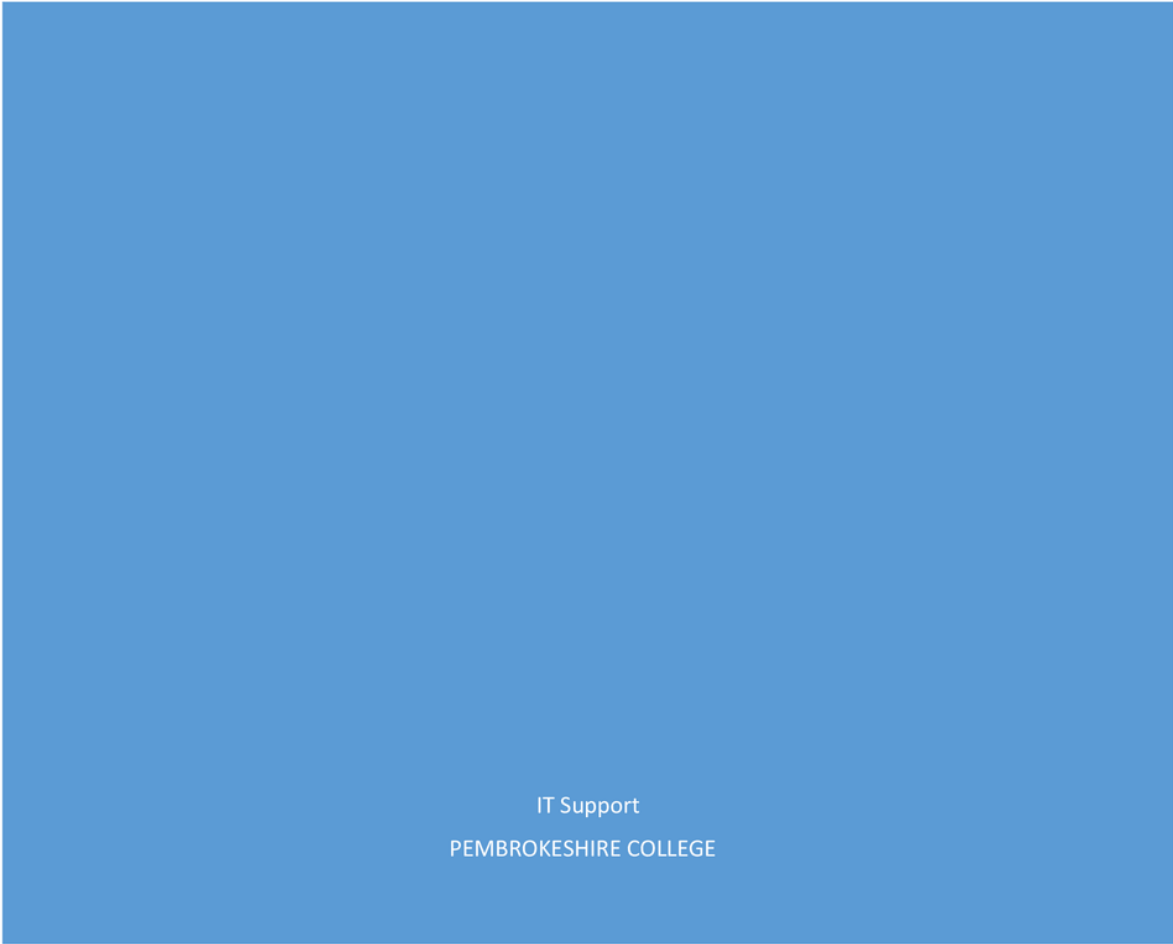
Word count: 3929

Character count: 17842



PROGRAMMING IN C

Software Development



IT Support
PEMBROKESHIRE COLLEGE

Table of Contents

Overview	2
Flowcharts	3
Problems and Code Development	4
Testing.....	7
Evaluation	17
Appendix	18

Overview

Here is an overview of the games that I have created:

High/Low Game

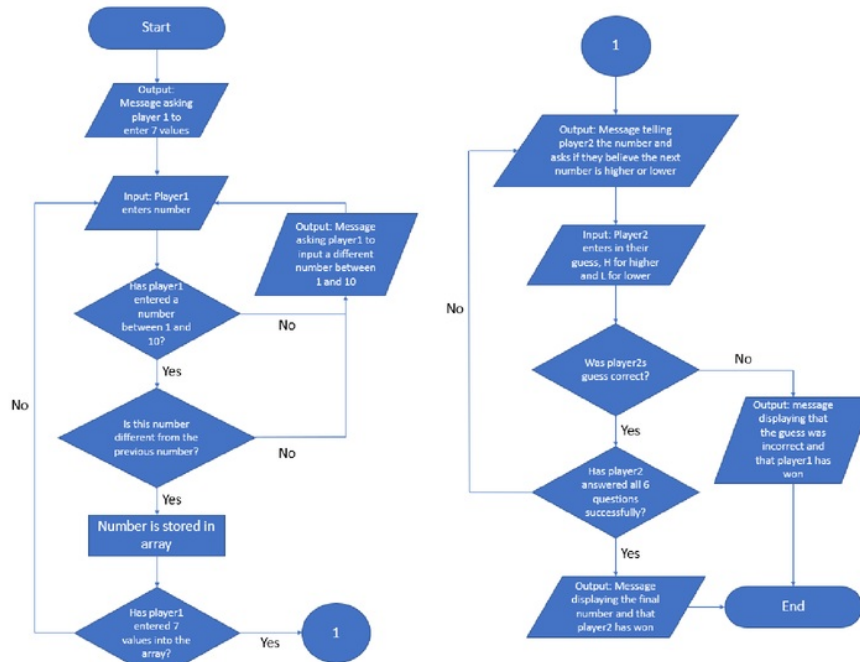
This is a simple 2 player game, the purpose of the game is to allow player 1 to input 7 numbers in a row, once they have done this player 2 will then be given the first of these numbers that player 1 inputted and must guess if the next number that was inputted is higher or lower than the first game. If the player guesses correctly then the 3rd number will be displayed and player 2 will again need to guess if the next number is higher or lower, player 2 wins if they make the 6 guesses correctly however if they make 1 mistake then player 1 wins the game.

Treasure Hunt Game

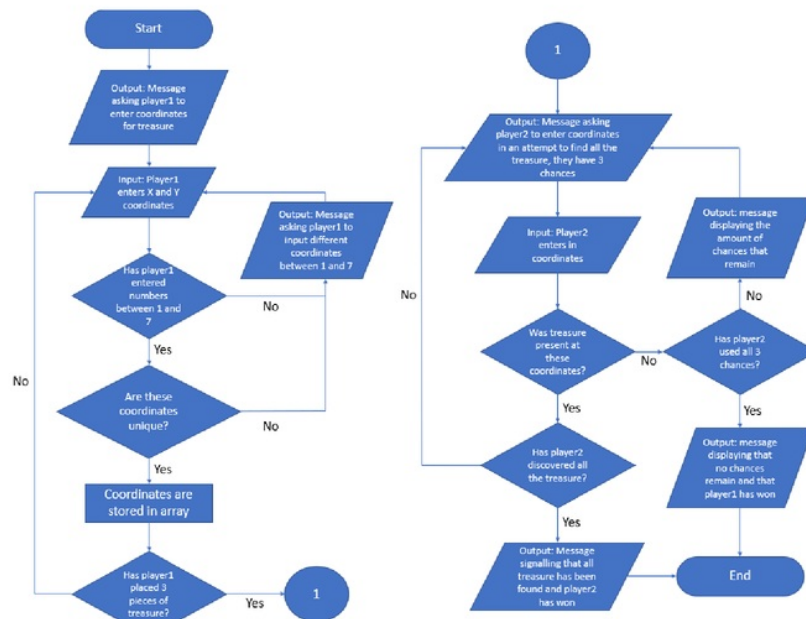
This is also a 2-player game where player 1 sets coordinated within an 8 x 8 grid, these set coordinates are where the treasure will be placed. Once the treasure has been placed player 2 will then have 3 guesses in order to find it, if player 2 finds the treasure then they win but if they use up all their lives then player 1 wins the game.

Flowcharts

Higher/Lower Game



Treasure Hunt Game



Problems and Code Development

High/Low Game

1. The problem I faced was trying to prevent the player from entering invalid characters (not numbers between 1 and 10), this was resolved using a while loop that prevents numbers higher than 10 and lower than 1.

```
while (number[0] > 10 || number[0] < 1) //While loop that ensures that the player only inputs number between 1 and 10
{
    cout << "Enter a value between 1 and 10" << endl; //Message that gets displayed when the player doesnt input a number between 1 and 10
    cin >> number[0];
}
cout << "Please insert your second value" << endl; //Outputs message that asks player to input their second number
cin >> number[1];

while (number[1] > 10 || number[1] < 1 || number[1] == number[0]) //While loop that prevents players from entering numbers over 10 or under 1
{
    cout << "Enter a different value between 1 and 10" << endl;
    cin >> number[1];
}
```

2. There was also a problem with the game allowing the same number twice in a row, this would obviously cause some issues, so I also resolved this with the same while loop.

```
while (number[0] > 10 || number[0] < 1) //While loop that ensures that the player only inputs number between 1 and 10
{
    cout << "Enter a value between 1 and 10" << endl; //Message that gets displayed when the player doesnt input a number between 1 and 10
    cin >> number[0];
}
cout << "Please insert your second value" << endl; //Outputs message that asks player to input their second number
cin >> number[1];

while (number[1] > 10 || number[1] < 1 || number[1] == number[0]) //While loop that prevents players from entering numbers over 10 or under 1
{
    cout << "Enter a different value between 1 and 10" << endl;
    cin >> number[1];
}
```

3. The problem of storing numbers was also an issue, this was resolved using an array.

```
int number[7]; //Produces variable array for numbers
```

```
cout << "Player 1, please enter a value between 1 and 10 7 times" << endl; //Produces text asking player 1 to input values
cout << "Please insert your first value" << endl; //Produces text asking player 1 for the first number
cin >> number[0]; //Inputs the number to the array

while (number[0] > 10 || number[0] < 1) //While loop that ensures that the player only inputs number between 1 and 10
{
    cout << "Enter a value between 1 and 10" << endl; //Message that gets displayed when the player doesnt input a number between 1 and 10
    cin >> number[0];
}
```

4. There was the problem of the requiring an appropriate response occurring within the game when certain criteria were met, such as if player2 made an incorrect guess the game would end and player1 would win, this problem was resolved using if statements and while loops.

```
cout << "The first number is " << number[0] << ". Is the second number higher (H) or lower (L)?" << endl; //Informs player 2 what the first number is and asks them if they think the next number is higher or lower
cin >> answer1; //Inputs player 2s first answer

if (number[0] < number[1]) //If statement for a less than number
{
    while (answer1 == 'L') //While loop for a less than answer
    {
        cout << "This answer was incorrect, Player 1 wins" << endl; //Outputs message informing player 2 that they were incorrect and that player 1 has won
        cin >> placeholder;
    }

    if (answer1 == 'H') //While loop for a more than answer
    {
        cout << "This answer was correct" << endl; //Outputs message informing player 2 that they were correct
    }
}

if (number[0] > number[1]) //If statement for a more than number
{
    if (answer1 == 'L') //While loop for a less than answer
    {
        cout << "This answer was correct" << endl; //Output message informing player 2 that they were correct
    }

    while (answer1 == 'H')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl; //Outputs message informing player 2 that they were incorrect and that player 1 has won
        cin >> placeholder;
    }
}
```

5. The final issue that occurred was signalling that player2 had won if they successfully made all 6 guesses, this was done by simply adding the text "player 2 wins" once the final guess had been made successfully and then putting a placeholder so the message can be seen before the game closes.

```
cout << "The sixth number is " << number[5] << ". Is the third number higher (H) or lower(L)" << endl;
cin >> answer1;

if (number[5] < number[6])
{
    if (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    while (answer1 == 'H')
    {
        cout << "The seventh number is " << number[6] << ". Player 2 wins;" << endl; //Last output that is displayed if player 2 has guessed all the correct answers.
        cin >> placeholder;
    }
}

if (number[5] > number[6])
{
    while (answer1 == 'L')
    {
        cout << "The seventh number is " << number[6] << ". Player 2 wins;" << endl; //Last output that is displayed if player 2 has guessed all the correct answers.
        cin >> placeholder;
    }

    if (answer1 == 'H')
    {
        cout << "this answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }
}
```

Treasure Hunt Game

1. The first problem that occurred to me while developing this game was how to assign coordinates to certain parts of the grid, this problem was resolved using an array shown below.

```
int grid[8][8] = {
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 },
    { 0, 0, 0, 0, 0, 0, 0, 0 }; // grid array
```

2. After creating the variables required, I then began to form the structure required in order to allow player1 to place the treasure using x and y coordinates. The structure created is shown below. This process is done 3 times for the 3 treasures that need to be placed.

```
int main()
{
    cout << "Player 1, please enter coordinates for the first treasure location" << endl; //output message asking player 1 to enter first treasure location
    cout << "X: "; cin >> x; //output message asking for X coordinate
    cout << "Y: "; cin >> y; //output message asking for Y coordinate
    grid[x][y] = 1; // this inputs the coordinates into the array
    cout << "Treasure 1 has been placed" << endl; //output message saying the treasure has been set
}
```

3. I then wished the grid to be printed onto the screen as player2 should understand the size of the structure that they're trying to find the treasure in. This was done by simple printing out the grid.

```
cout << "  0  1  2  3  4  5  6  7" << endl;
cout << "0  -  -  -  -  -  -  -" << endl;
cout << "1  -  -  -  -  -  -  -" << endl;
cout << "2  -  -  -  -  -  -  -" << endl;
cout << "3  -  -  -  -  -  -  -" << endl;
cout << "4  -  -  -  -  -  -  -" << endl;
cout << "5  -  -  -  -  -  -  -" << endl;
cout << "6  -  -  -  -  -  -  -" << endl;
cout << "7  -  -  -  -  -  -  -" << endl; //displays the grid to the player
```

4. Player2 must also understand their objective and the amount of chances they have so a clear message is displayed before they're able to make any inputs. This message is shown below.

```
cout << "Player 2, please input coordinates in an attempt to find the treasure" << endl; //Outputs message asking player 2 for coordinates to attempt to find the treasure
cout << "3 mistakes are allowed" << endl; //Outputs message signalling that 3 chances are available
```

5. I then needed to develop a way of tracking if a treasure piece had been found or not using the coordinates that player2 had inputted, this is done using an if statement.

```
cout << "Player 2 please enter an X and Y coordinate" << endl; //Outputs message asking for coordinates
cin >> x; //Inputs the X coordinate
cin >> y; //Inputs the Y coordinate
if (1 == grid[x][y])
{
    cout << "Treasure was found here" << endl; //This message will be displayed if coordinates that contain treasure are inputted
}
else
{
    cout << "No treasure was found, a chance was lost" << endl; //This message will be displayed if coordinates that do not contain treasure are inputted
    chances--;
}
```

6. I then needed to ensure that if player2 had found all pieces of treasure then they would win the game so this was done using a score system, when player2 finds a piece of treasure they get 1 point and if they get 3 points (finding all the treasure) then they win the game, this was done using a nested if statement.

```
cout << "Player 2 please enter an X and Y coordinate" << endl; //Outputs message asking for coordinates
cin >> x; //Inputs the X coordinate
cin >> y; //Inputs the Y coordinate
if (1 == grid[x][y])
{
    cout << "Treasure was found here" << endl; //This message will be displayed if coordinates that contain treasure are inputted
    score++;
    if (score > 2)
    {
        cout << "All treasure was found, player 2 wins" << endl; //This will be displayed when player 2 finds all the treasure
        cin >> placeholder;
    }
}
```

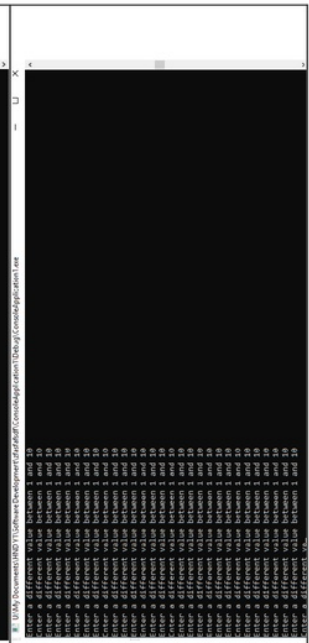
7. The final problem that needed resolving was giving player2 3 chances and if they do not find all the treasure before all their chances are used then player1 would win the game, this was done using a while loop. A placeholder was also used so the game doesn't immediately end before the final message can be witnessed.

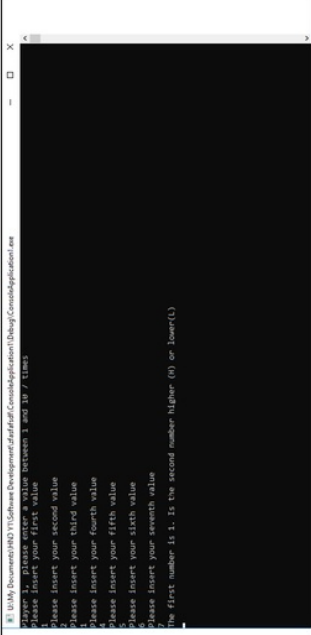
```
while (chances > 0)
{
    cout << "Player 2 please enter an X and Y coordinate" << endl; //Outputs message asking for coordinates
    cin >> x; //Inputs the X coordinate
    cin >> y; //Inputs the Y coordinate
    if (1 == grid[x][y])
    {
        cout << "Treasure was found here" << endl; //This message will be displayed if coordinates that contain treasure are inputted
        score++;
        if (score > 2)
        {
            cout << "All treasure was found, player 2 wins" << endl; //This will be displayed when player 2 finds all the treasure
            cin >> placeholder;
        }
    }
    else
    {
        cout << "No treasure was found, a chance was lost" << endl; //This message will be displayed if coordinates that do not contain treasure are inputted
        chances--;
    }
}
cout << "You are out of chances, player 1 wins" << endl; //This message will be outputted if the player runs out of chances
cin >> placeholder; //This is used to prevent the game from closing immediately after the last chance has been used, so the message above can be seen
```

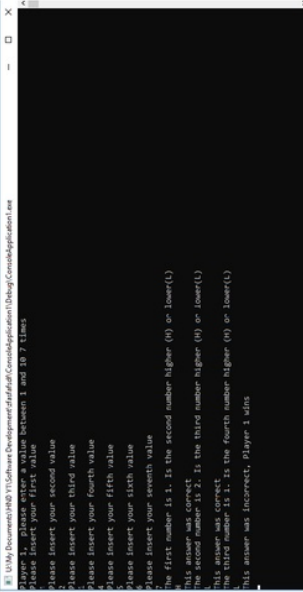
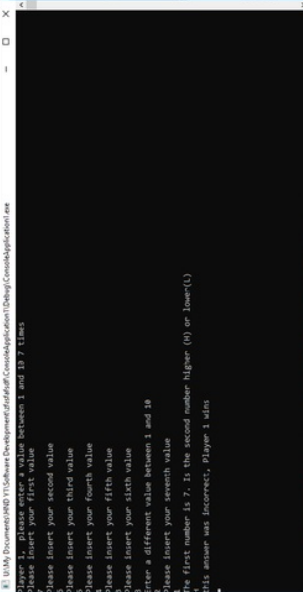
Testing

High/Low Game

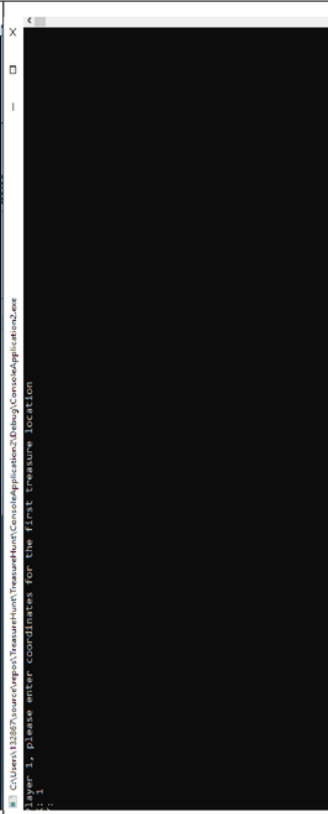
Number	Test	Test Input	Expected Output	Test Output	Evidence
1	Game opens with output asking player 1 for their first number	Start game	Player 1, Please enter a value between 1 and 10 7 times. Please insert your first value	Player 1, Please enter a value between 1 and 10 7 times. Please insert your first value	
2	Player 1 enters first number and game asks for second number	First Number	Number is stored and game asks for second number	Number is stored and game asks for second number	

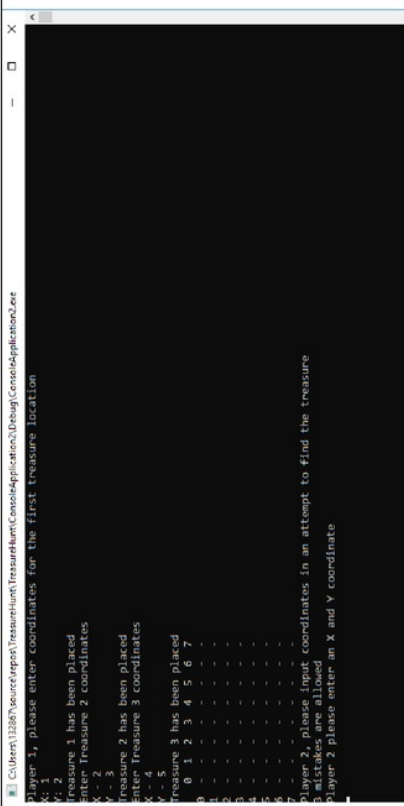
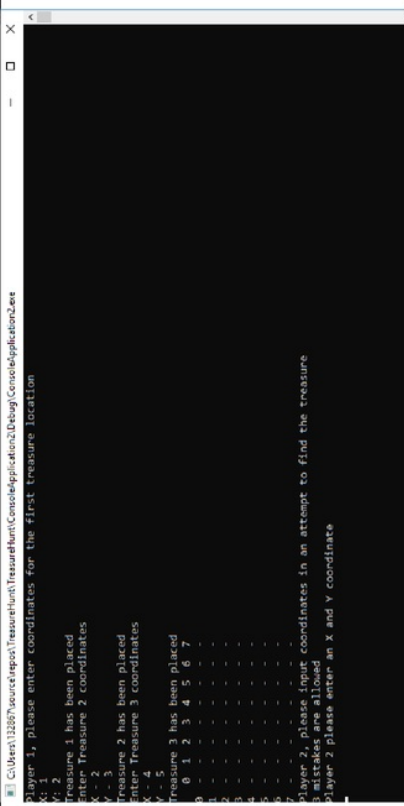
3	Game does not allow numbers below 1 and higher than 10	Number higher than 10 and lower than 1	Message is outputted asking player to input a number between 1 and 10	Message is outputted asking player to input a number between 1 and 10	
4	The same number isn't allowed 2 times in a row	Same number is inputted twice in a row	Message is outputted asking player to input a different number between 1 and 10	Message is outputted asking player to input a different number between 1 and 10	
5	Game does not allow letters	Entered a letter	Message is outputted asking player to input a number between 1 and 10, however this is done continuously	Message is outputted asking player to input a number between 1 and 10	

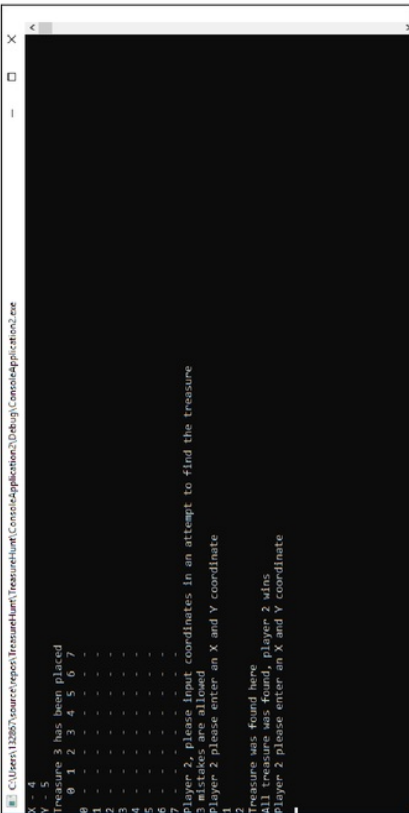
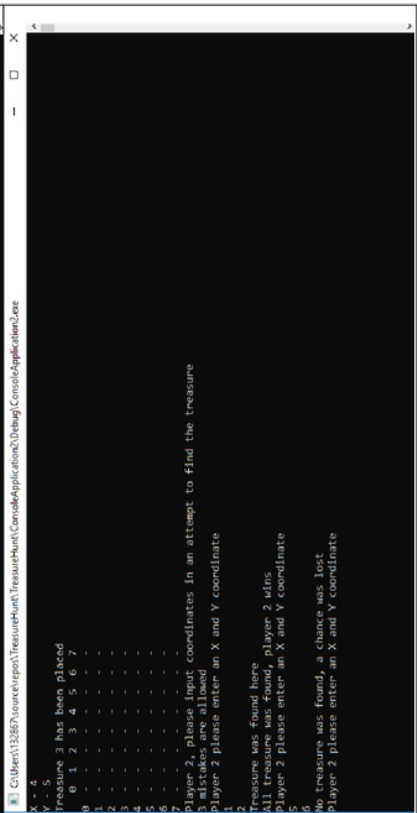
6	When 7 number have been entered, the game asks player 2 if they believe the next number is higher or lower	Seventh number is inputted	Message is outputted asking player 2 if they think the next number is higher or lower	Message is outputted asking player 2 if they think the next number is higher or lower	 <pre> // UVA Document#19215 Software Development at UVA All Copyrights Reserved! // Player 1, please enter a value between 1 and 10 7 times // Please insert your first value // Please insert your second value // Please insert your third value // Please insert your fourth value // Please insert your fifth value // Please insert your sixth value // Please insert your seventh value // The first number is 1. Is the second number higher (H) or lower(L) </pre>
7	Second number is higher than first number so H is correct answer	H is inputted	The answer was correct and game asks if the next number is higher or lower	The answer was correct and game asks if the next number is higher or lower	 <pre> // UVA Document#19215 Software Development at UVA All Copyrights Reserved! // Player 1, please enter a value between 1 and 10 7 times // Please insert your first value // Please insert your second value // Please insert your third value // Please insert your fourth value // Please insert your fifth value // Please insert your sixth value // Please insert your seventh value // The first number is 1. Is the second number higher (H) or lower(L) // This answer was correct // The second number is 2. Is the third number higher (H) or lower(L) </pre>
8	Third number is lower than second number so L is correct answer	L is inputted	The answer was correct and game asks if the next number is higher or lower	The answer was correct and game asks if the next number is higher or lower	 <pre> // UVA Document#19215 Software Development at UVA All Copyrights Reserved! // Player 1, please enter a value between 1 and 10 7 times // Please insert your first value // Please insert your second value // Please insert your third value // Please insert your fourth value // Please insert your fifth value // Please insert your sixth value // Please insert your seventh value // The first number is 1. Is the second number higher (H) or lower(L) // This answer was correct // The second number is 2. Is the third number higher (H) or lower(L) // This answer was correct // The third number is 1. Is the fourth number higher (H) or lower(L) </pre>

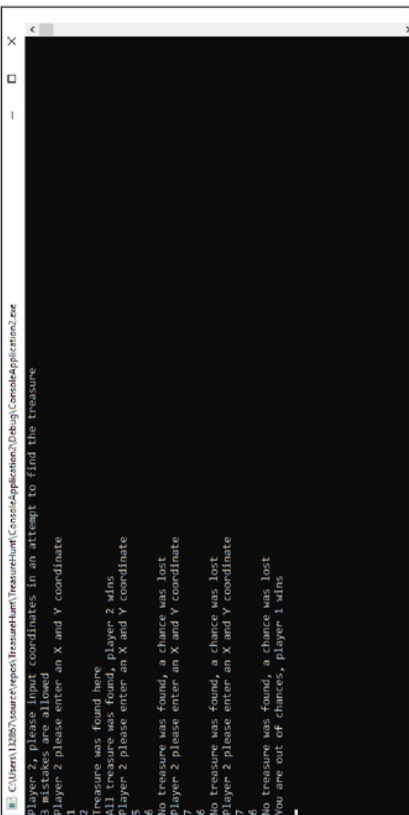
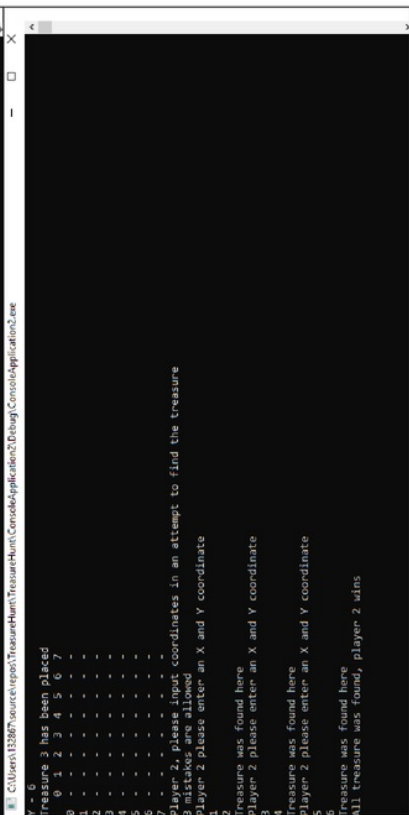
9	Fourth number was higher than 3 rd number so L was the incorrect answer	L is inputted	The answer was incorrect so game says player 1 wins and game ends	The answer was incorrect so game says player 1 wins and games ends	
10	Second number was lower than first number so H was the incorrect answer	H is inputted	The answer was incorrect so game says player 1 wins and game ends	The answer was incorrect so game says player 1 wins and game ends	
11	Player 2 guesses all of the answers correctly and a message is displayed saying player 2 won	Last correct answer is inputted	Message is displayed saying player 2 won	Message is displayed saying player 2 won however the game ends immediately after last answer is inputted	

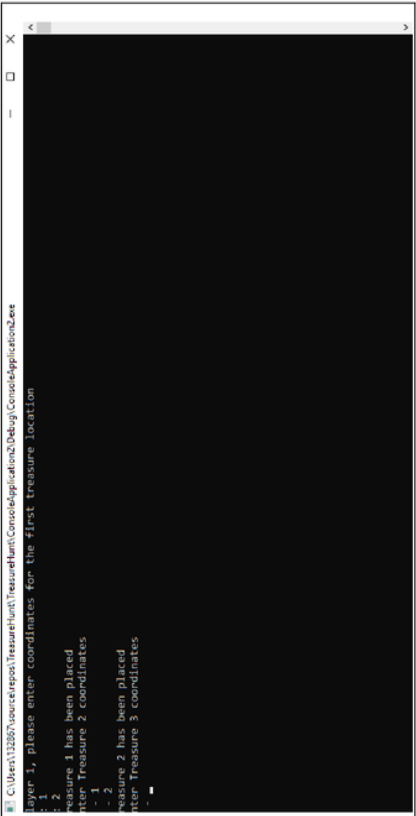
Treasure Hunt Game

Number	Test	Test Input	Expected Output	Test Output	Evidence
1	Game opens with output asking player 1 for their first coordinate	Start game	Player 1, Please enter coordinate for the first treasure location	Player 1, Please enter coordinate for the first treasure location	
2	Player 1 enters X coordinate and then asks for Y coordinate	X coordinate	Coordinate is stored and game asks for Y coordinate	Coordinate is stored and game asks for Y coordinate	

5	After the 3 treasure locations have been set the grid will appear	Enter X and Y coordinates 3 times	After the coordinates are entered the grid appears	After the coordinates are entered the grid appears	
6	After the grid appears instructions to player 2 are present	Enter X and Y coordinates 3 times	After the grid instructions are shown as well as the giving ability for player 2 to enter their first X coordinate	After the grid instructions are shown as well as the giving ability for player 2 to enter their first X coordinate	

7	Player 2 enters coordinates that contain a treasure	Enter X and Y coordinates that contain treasure	A message will appear that signals to the player that a treasure has been found	A message will appear that signals to the player that a treasure has been found	
8	Player 2 enters coordinates that does not contain a treasure	Enter X and Y coordinates that do not contain treasure	A message will appear telling the player that there was no treasure present and that they've lost a chance	A message will appear telling the player that there was no treasure present and that they've lost a chance	

9	The player enters coordinates that do not contain treasure 3 times	Enter X and Y coordinates that do not contain treasure 3 times	A message will appear saying "you are out of chances, player 1 wins"	A message will appear saying "you are out of chances, player 1 wins"	
10	The player successfully guesses the location of the 3 treasures	Enter X and Y coordinates that contain treasure 3 times	A message will appear saying "you have found all the treasure, player 2 wins"	A message will appear saying "you have found all the treasure, player 2 wins"	

11	Player 1 is unable to place the treasure in the same location twice	Enter the same coordinates for the treasure twice	The game will identify that the player has already placed treasure in that location	The game places the treasure in the same location twice	
----	---	---	---	---	--

Evaluation

High/Low Game

The final product of the high/low game was quite successful in my opinion, the game is fully functional in its current form and will allow the two players to complete the game without many game breaking issues. Despite the functionality of the game there are still a few issues that can be amended in future versions of the game, the main issue that is present is when player 1 enters a letter instead of a number, this results in a message being repeated indefinitely resulting in the game requiring a restart. The code of the game could have also been a lot more refined, it is quite functional however it does appear quite messy with large amounts of repeated code, this can be fixed when my knowledge in this field increases and I can develop more refined solutions to problems.

Treasure Hunt Game

The final product of the high/low game is useable in my opinion, it goes through the steps of the game quite well and with a bit of refinement to reduce the amount of repeated code present. Despite the games ability to be performed successfully there are a issues that need to be fixed in future versions, these include the possibility that player 1 is able to place the treasure in the same coordinates several times and that numbers above 8 can be entered which does cause issues and makes the game unplayable. Despite the issues signalled above the game is playable and able to be enjoyed by a pair of players.

Appendix

High/Low Game

```
8
#include "stdafx.h"
using namespace std;
#include <iostream>
int main()
{
    int number[7]; //Produces variable array for numbers
    char answer1; //Produces variable array for answers
    char answer2;
    char answer3;
    char answer4;
    char answer5;
    char answer6;

    char placeholder; //Produces variable for placeholder

    cout << "Player 1, please enter a value between 1 and 10 7 times" << endl;
    //Produces text asking player 1 to input values
    cout << "Please insert your first value" << endl; //Produces text asking player
1 for the first number
    cin >> number[0]; //Inputs the number to the array

    while (number[0] > 10 || number[0] < 1) //While loop that ensures that the
player only inputs number between 1 and 10
    {
        cout << "Enter a value between 1 and 10" << endl; //Message that gets
displayed when the player doesnt input a number between 1 and 10
        cin >> number[0];
    }

    cout << "Please insert your second value" << endl; //Outputs message that asks
player to input their second number
    cin >> number[1];

    while (number[1] > 10 || number[1] < 1 || number[1] == number[0]) //While loop
that prevents players from entering numbers over 10 or under 1
    {
        cout << "Enter a different value between 1 and 10" << endl;
        cin >> number[1];
    }

    cout << "Please insert your third value" << endl;
    cin >> number[2];

    while (number[2] > 10 || number[2] < 1 || number[2] == number[1])
    {
        cout << "Enter a different value between 1 and 10" << endl;
        cin >> number[2];
    }

    cout << "Please insert your fourth value" << endl;
    cin >> number[3];

    while (number[3] > 10 || number[3] < 1 || number[3] == number[2])
    {
        cout << "Enter a different value between 1 and 10" << endl;
        cin >> number[3];
    }
}
```

```

}
cout << "Please insert your fifth value" << endl;
cin >> number[4];

while (number[4] > 10 || number[4] < 1 || number[4] == number[3])
{
    cout << "Enter a different value between 1 and 10" << endl;
    cin >> number[4];
}

cout << "Please insert your sixth value" << endl;
cin >> number[5];

while (number[5] > 10 || number[5] < 1 || number[5] == number[4])
{
    cout << "Enter a different value between 1 and 10" << endl;
    cin >> number[5];
}

cout << "Please insert your seventh value" << endl;
cin >> number[6];

while (number[6] > 10 || number[6] < 1 || number[5]==number[6])
{
    cout << "Enter a different value between 1 and 10" << endl;
    cin >> number[6];
}

}

cout << "The first number is " << + number[0] << ". Is the second number higher
(H) or lower(L)" << endl; //Informs player 2 what the first number is and asks them if
they think the next number is higher or lower
cin >> answer1; //Inputs player 2s first answer

if (number[0] < number[1]) //If statement for a less than number
{
    while (answer1 == 'L') //While loop for a less than answer
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        //Outputs message informing player 2 that they were incorrect and that player 1 has
        won
        cin >> placeholder;
    }

    if (answer1 == 'H') //While loop for a more than answer
    {
        cout << "This answer was correct" << endl; //Outputs message
        informing player 2 that they were correct
    }
}

if (number[0] > number[1]) //If statement for a more than number
{
    if (answer1 == 'L') //While loop for a less than answer
    {
        cout << "This answer was correct" << endl; //Output message
        informing player 2 that they were correct
    }
    while (answer1 == 'H')
    {

```

```

        cout << "this answer was incorrect, Player 1 wins" << endl;
//Outputs message informing player 2 that they were incorrect and that player 1 has
won
        cin >> placeholder;
    }
}
cout << "The second number is " << + number[1] << ". Is the third number higher
(H) or lower(L)" << endl;
cin >> answer1;

if (number[1] < number[2])
{
    while (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    if (answer1 == 'H')
    {
        cout << "This answer was correct" << endl;
    }
}
if (number[1] > number[2])
{
    if (answer1 == 'L')
    {
        cout << "This answer was correct" << endl;
    }
    while (answer1 == 'H')
    {
        cout << "this answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }
}
cout << "The third number is " << +number[2] << ". Is the fourth number higher
(H) or lower(L)" << endl;
cin >> answer1;

if (number[2] < number[3])
{
    while (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    if (answer1 == 'H')
    {
        cout << "This answer was correct" << endl;
    }
}
if (number[2] > number[3])
{
    if (answer1 == 'L')
    {
        cout << "This answer was correct" << endl;
    }
    while (answer1 == 'H')
    {

```

```

        cout << "this answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }
}
cout << "The fourth number is " << +number[3] << ". Is the third number higher
(H) or lower(L)" << endl;
cin >> answer1;

if (number[3] < number[4])
{
    while (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    if (answer1 == 'H')
    {
        cout << "This answer was correct" << endl;
    }
}

if (number[3] > number[4])
{
    if (answer1 == 'L')
    {
        cout << "This answer was correct" << endl;
    }
    while (answer1 == 'H')
    {
        cout << "this answer was incorrect Player 1 wins" << endl;
        cin >> placeholder;
    }
}
cout << "The fifth number is " << +number[4] << ". Is the third number higher
(H) or lower(L)" << endl;
cin >> answer1;

if (number[4] < number[5])
{
    while (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    if (answer1 == 'H')
    {
        cout << "This answer was correct" << endl;
    }
}

if (number[4] > number[5])
{
    if (answer1 == 'L')
    {
        cout << "This answer was correct" << endl;
    }
    while (answer1 == 'H')
    {
        cout << "this answer was incorrect, Player 1 wins" << endl;
    }
}

```

```

        cin >> placeholder;
    }
}
cout << "The sixth number is " << +number[5] << ". Is the third number higher
(H) or lower(L)" << endl;
cin >> answer1;

if (number[5] < number[6])
{
    if (answer1 == 'L')
    {
        cout << "This answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }

    while (answer1 == 'H')
    {
        cout << "The seventh number is " << +number[6] << ". Player 2
wins;" << endl; //Last output that is displayed if player 2 has guessed all the
correct answers.
        cin >> placeholder;
    }
}
if (number[5] > number[6])
{
    while (answer1 == 'L')
    {
        cout << "The seventh number is " << +number[6] << ". Player 2
wins;" << endl; //Last output that is displayed if player 2 has guessed all the
correct answers.
        cin >> placeholder;
    }
    if (answer1 == 'H')
    {
        cout << "this answer was incorrect, Player 1 wins" << endl;
        cin >> placeholder;
    }
}
}

```

Treasure Hunt Game

```

#include "stdafx.h"
#include <string>
#include <iostream>
#include <stdlib.h>

using namespace std;

char x; // variable for x coordinate
char y; // variable for y coordinate
char treasure;
char chances; //variable for the player chances
char score;
char index; // counter for the game
char placeholder; //Produces variable for placeholder
bool correct = true;

```

```

int grid[8][8] = {
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, },
{ 0, 0, 0, 0, 0, 0, 0, 0, }}; // grid array

int main()
{
    cout << "Player 1, please enter coordinates for the first treasure location" <<
endl; //output message asking player 1 to enter first treasure location
    cout << "X: "; cin >> x; //output message asking for X coordinate
    cout << "Y: "; cin >> y; //output message asking for Y coordinate
    grid[x][y] = 1; // this inputs the coordinates into the array
    cout << "Treasure 1 has been placed" << endl; //output message saying the
treasure has been set

    cout << "Enter Treasure 2 coordinates" << endl; //output message asking player
1 to enter first treasure location
    cout << "X - "; cin >> x; //output message asking for X coordinate
    cout << "Y - "; cin >> y; //output message asking for Y coordinate
    grid[x][y] = 1; // this inputs the coordinates into the array
    cout << "Treasure 2 has been placed" << endl; //output message saying the
treasure has been set

    cout << "Enter Treasure 3 coordinates" << endl; //output message asking player
1 to enter first treasure location
    cout << "X - "; cin >> x; //output message asking for X coordinat
    cout << "Y - "; cin >> y; //output message asking for Y coordinate
    grid[x][y] = 1; // this inputs the coordinates into the array
    cout << "Treasure 3 has been placed" << endl; //output message saying the
treasure has been set

    cout << "   0  1  2  3  4  5  6  7" << endl;
    cout << "0  -  -  -  -  -  -  -  -" << endl;
    cout << "1  -  -  -  -  -  -  -  -" << endl;
    cout << "2  -  -  -  -  -  -  -  -" << endl;
    cout << "3  -  -  -  -  -  -  -  -" << endl;
    cout << "4  -  -  -  -  -  -  -  -" << endl;
    cout << "5  -  -  -  -  -  -  -  -" << endl;
    cout << "6  -  -  -  -  -  -  -  -" << endl;
    cout << "7  -  -  -  -  -  -  -  -" << endl; //displays the grid to the player

    cout << "Player 2, please input coordinates in an attempt to find the treasure"
<< endl; //Outputs message asking player 2 for coordniates to attempt to find the
treasure
    cout << "3 mistakes are allowed" << endl; //Outputs message signalling that 3
chances are available

    while (chances < 3)
    {
        cout << "Player 2 please enter an X and Y coordinate" << endl; //Outputs
message asking for coordinates
        cin >> x; //Inputs the X coordinate
        cin >> y; //Inputs the Y coordinate
        if (1 == grid[x][y])
        {

```

```

        cout << "Treasure was found here" << endl; //This message will be
displayed if coordinates that contain treasure are inputted
        score++;

        if (score > 2)
        {
            cout << "All treasure was found, player 2 wins" << endl;
//This will be displayed when player 2 finds all the treasure
            cin >> placeholder;
        }
    }
    else
    {
        cout << "No treasure was found, a chance was lost" << endl;
//This message will be displayed if coordinates that do not contain treasure are
inputted
        chances++;
    }
}
cout << "You are out of chances, player 1 wins" << endl; //This message will be
outputted if the player runs out of chances
    cin >> placeholder; //This is used to prevent the game from closing immediately
after the last chance has been used, so the message above can be seen
}

```

Software Development Report

ORIGINALITY REPORT

5%

SIMILARITY INDEX

2%

INTERNET SOURCES

1%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Pembrokeshire College

Student Paper

1%

2

Submitted to University of Wales, Lampeter

Student Paper

1%

3

"The Global Paleomagnetic Database",
Springer Nature America, Inc, 1991

Publication

<1%

4

cmosedu.com

Internet Source

<1%

5

Yuwei Peng, Mingliang Yue. "A Zero-Watermarking Scheme for Vector Map Based on Feature Vertex Distance Ratio", Journal of Electrical and Computer Engineering, 2015

Publication

<1%

6

empublic.dcs.warwick.ac.uk

Internet Source

<1%

7

www.coursehero.com

Internet Source

<1%

stackoverflow.com

9

Fan, F., G. B. Nie, and X. D. Zhi. "Retracted: A Constitutive Model of Circular Steel Tubes and Its Applications in Analysis of Space Structures", Journal of Structural Engineering, 2011.

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

Software Development Report

GRADEMARK REPORT

FINAL GRADE

56/100

GENERAL COMMENTS

Instructor

Exercise 1 34/ 50%

Flow chart shows logical planning of task and conforms to conventions taught, and is logically correct. The original brief did not specify that player one could not repeat a number, only that they numbers they entered were between 1 and 10, however you have successfully implemented this feature. A counter could have been used with a for loop which could have saved you writing so many while loops, however you would have had to change the way you checked for repeated values. Also a loop for checking would have been more efficient rather than repeating in effect the same code 6 times. Code is annotated. You have included a formal testing table with evidence of testing, although should show checks at boundaries to ensure that conditions work.

Exercise 2 22/ 30%

Flow chart shows logical planning of task and conforms to conventions taught, it is logically correct although you may also have considered testing player 2's input was also within the valid range (as you have already implemented this for player 1 you can reuse code- particularly if this feature is implemented as a function). You should show you would be storing the location of the guess in the array as well as location of found treasure.

It would be more efficient to initialise your array with a nested loop!

Again as for previous exercise consider the use of functions.

With the testing, check at boundaries for invalid inputs, more thorough testing and evidence of simplifying problem and building it up would have gained further marks in this section.

Exercise 3 0/20%

You have not attempted this exercise.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23
