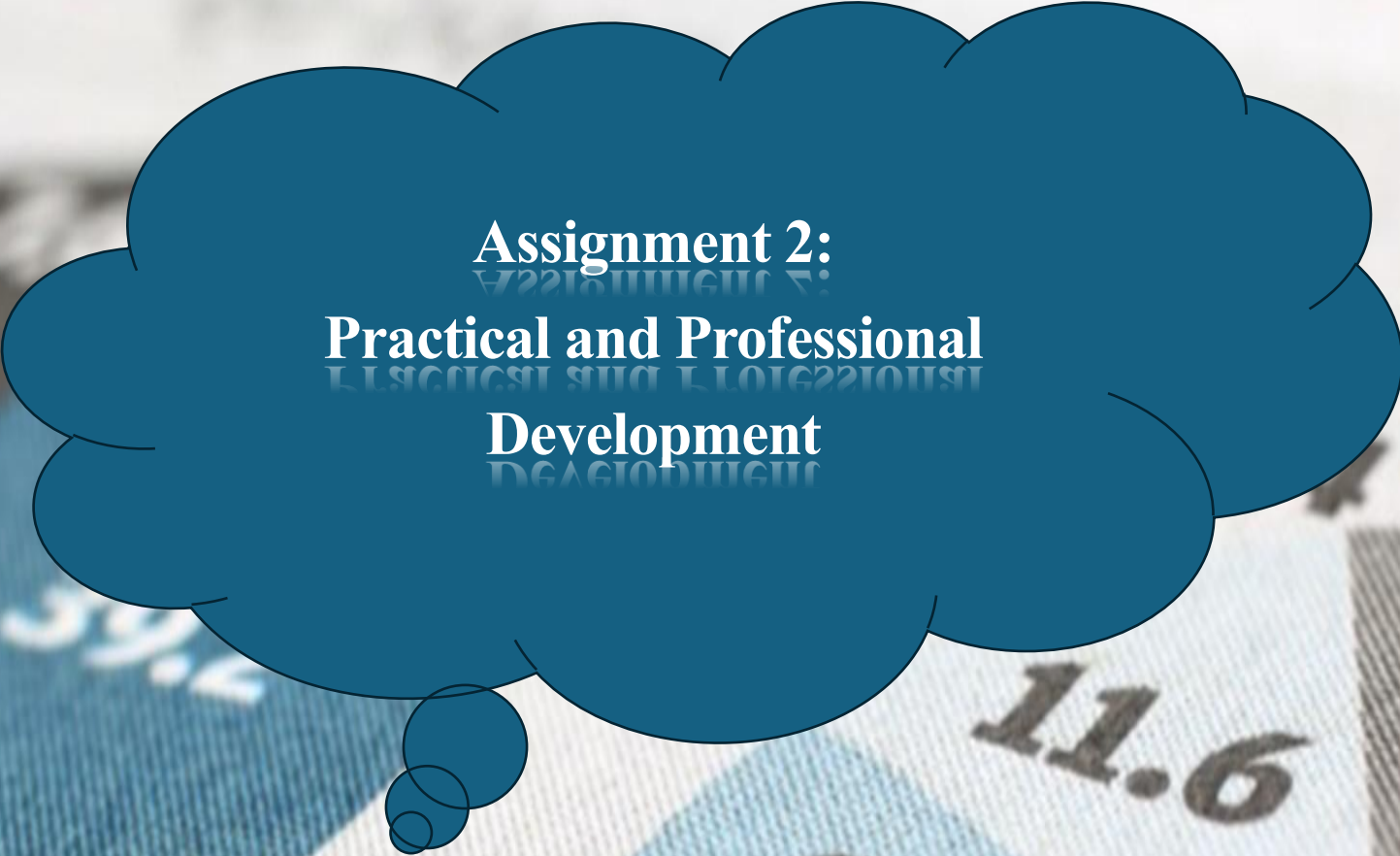


L5 - Changemakers: Creativity and Value Creation

(ACCA5033 BSc Sem 2)



Assignment 2: Practical and Professional Development

Module code: ACCA5033

Minefield Project

Assignment 2:

Practical and Professional Development

Table of Contents

Introduction	2
Purpose.....	2
Rules	2
Task 1: Project Implementation & Initial Feedback	3
Question 1: Initial Feedback.....	3
Question 2: Implementation	6
Task 2: T2: Innovation	9
Question 1: Concepts, Methods, Practices and Tools	9
Question 2: PEST Analysis	10
Task 3: Project Feedback and Reflection	11
Question 1: Practical Feedback.....	13
Test Results after practical feedback.....	14
Question 2: Project Progress Diary	16
Question 3: Skills Developed	16
Question 4: Positive Social Impact.....	16
Code Snippets:	20
Conclusion.....	24
References	25
Appendices'	26

Introduction

The minefield Project is a C# Windows Forms (.NET Framework) application that implements a simple minefield game. This documentation uses clear language and visual aids, including forms and screenshots, to explain the main components, rules, and functionalities. It is regularly updated to reflect any changes or improvements, ensuring an easy to understand overview of the project's design and implementation. [1]

Purpose

The primary purpose of this project is to demonstrate an understanding of game design principles, and user interaction, develop problem-solving skills by implementing search, randomisation, and recognition techniques. Showcase proficiency in programming languages (e.g., C#, Python, and JavaScript) and relevant frameworks. The project will focus on the following Key features:

- ✚ Grid Generation
- ✚ Mine Logic
- ✚ User Interface
- ✚ Game Flow
- ✚ Error Handling and Edge Cases

Rules

- ✚ The player begins in the upper-left corner of the grid.
- ✚ The objective is to navigate the grid without landing on a bomb.
- ✚ If a bomb is triggered, a message box display "Boooooooooom!!! Game Over" and reveals all bomb locations.
- ✚ The player can click a button to reveal all bomb locations at any time.

Below are some Rules that I have created to use

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Drawing.Text;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11 using static System.Windows.Forms.VisualStyles.VisualStyleElement.ProgressBar;
12
13 //The player begins in the upper-left corner of the grid.
14 //The objective is to navigate the grid without landing on a bomb.
15 //If the player lands on a bomb, a message box displays "Boooooooooom!!! Game Over" and reveals all bombs.
16 //The player can click a button to reveal all bomb locations at any time.
17
18 namespace Minefield_Project
19 {
```

Task 1: Project Implementation & Initial Feedback

Question 1: Initial Feedback

I designed a short feedback survey using for questions. The survey contained a mixture of closed and open-ended questions, focused on three main areas:

- ✚ Opinions on the project idea
- ✚ Feedback on design prototypes
- ✚ Feedback on the initial implementation
- ✚ Feedback Summary

Key Survey questions

- ❖ How useful do you think this Game would be for students?
- ❖ Is the interface design clear and easy to understand?
- ❖ Which feature do you think is the most useful?
- ❖ What improvements would you suggest?

1. How useful do you think this Game would be for students?

4 Responses

ID ↑	Name	Responses
1	Aaron Miles	It helps students relax and take short breaks from studying.
2	Daniel Leone	Good for a quick game to relax and have fun
3	Francois Riches	Is the game specifically for students? Is it an educational game?
4	Sanwar Ahmed	Classic game

2. Is the interface design clear and easy to understand?

4 Responses

ID ↑	Name	Responses
1	Aaron Miles	The game screen is organised and not confusing.
2	Daniel Leone	Mostly yes. Subjective opinion but I would reduce saturation of colours. maybe too bright?
3	Francois Riches	The game is a bit confusing and not entirely intuitive. It was easy while being helped through by the creator, but I think the game UI could be easier to follow.
4	Sanwar Ahmed	ok

3. Which feature do you think is the most useful?

4 Responses

ID ↑	Name	Responses
1	Aaron Miles	Level progression that increases difficulty.
2	Daniel Leone	Its good to be able to review the outcome of the each game and display where the bombs are.
3	Francois Riches	Hard to say at this point.
4	Sanwar Ahmed	Puzzle based

4. What improvements would you suggest?

4 Responses

ID ↑	Name	Responses
1	Aaron Miles	Include sound effects and background music.
2	Daniel Leone	Refine and improve visual design, Colours & Graphical Aspects.
3	Francois Riches	The game is a bit confusing and not intuitive, without guidance I would likely not know how to play. The graphical feedback does not keep track of the character. The object of the game is not clear.
4	Sanwar Ahmed	More variety

Summary of feedback

The feedback was generally positive, with most participants agreeing that the idea was useful for students and found the game easy to use and helpful for their studies.

Positive feedback including:

- 🎨 Simple interface
- 🎨 Clear task organization
- 🎨 Quick schedule generation

Suggested improvements included:

- Improving visual design and adding calendar integration

Planned Improvements. Based on this feedback I would Improve the user interface design.

Question 2: Implementation

Interface Design

The main form includes a panel representing the game grid, buttons for game control, and labels for each cell. Wireframes were created to plan layout and functionality, including button placement and grid size. This ensures an intuitive user experience and clear visual hierarchy.

Game Logic Implementation

The backend logic was implemented in C#, including:

- ✚ Random bomb placement
- ✚ Event driven response for clicks
- ✚ Game over and victory conditions when a bomb is triggered
- ✚ A feature allowing users to reveal bomb's location and functionality

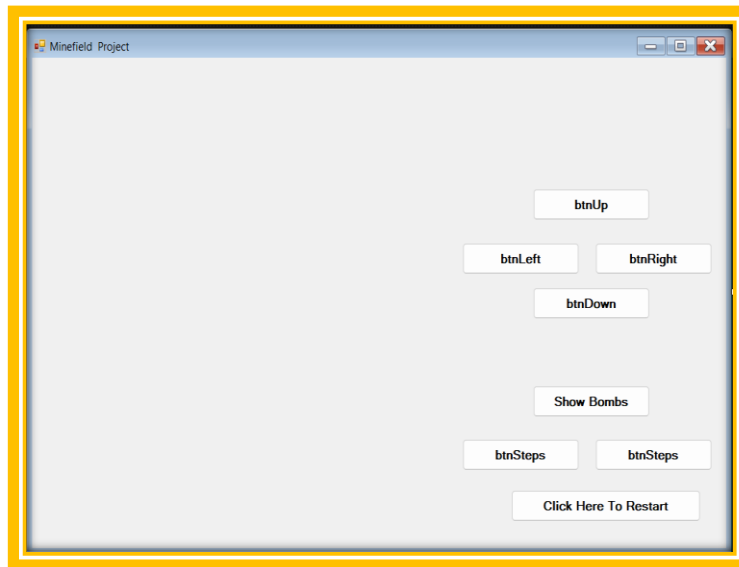
Troubleshooting included handling input validation to ensure correct processing of user action. These features demonstrate the use of conditional logic, loops, and event driven programming.

This design supports the project objective by providing a clear and intuitive interface, enabling users to easily navigate the game environment.

Interface / Design

Figure 1 Initial Form Setup

As shown in Figure 1, the interface layout includes the grid panel and control buttons.



Game Interface Layout

Figure 2 Added Panel with Blue color

A panel was added to represent the game grid.

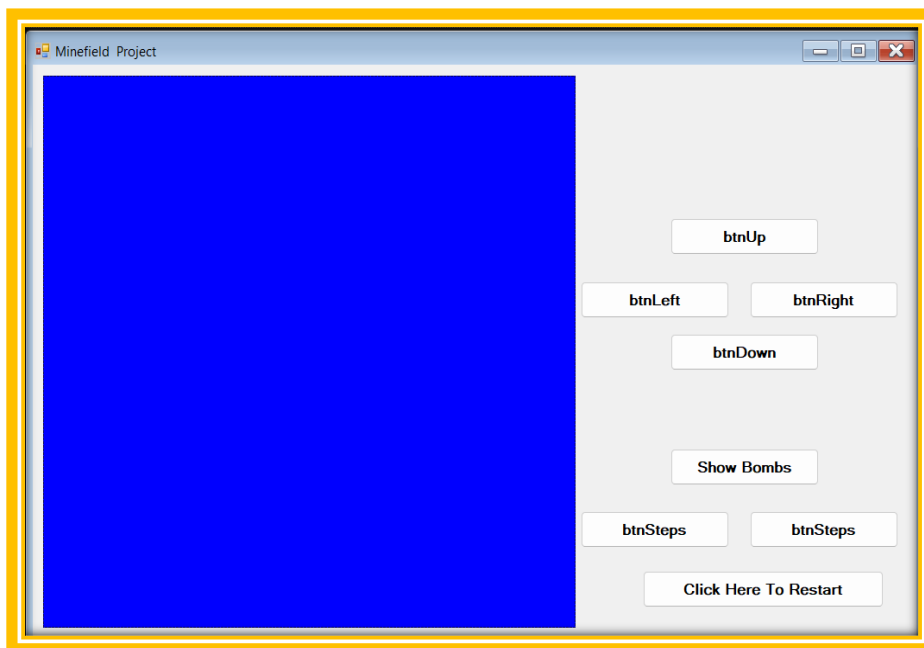
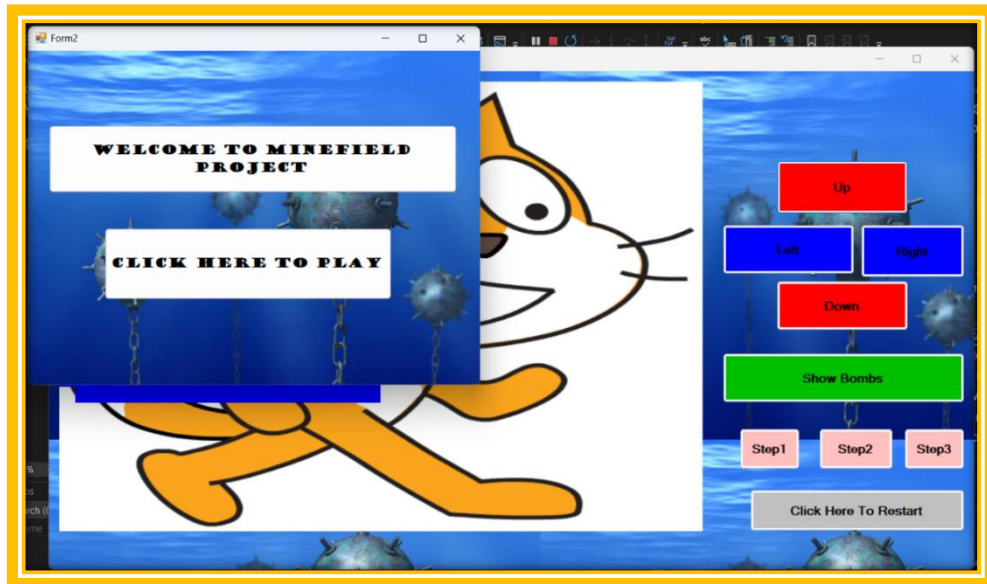


Figure 3 Added Welcome Page

This figure shows the welcome screen of the Minefield application.



- Troubleshooting

During development several issues were encountered:

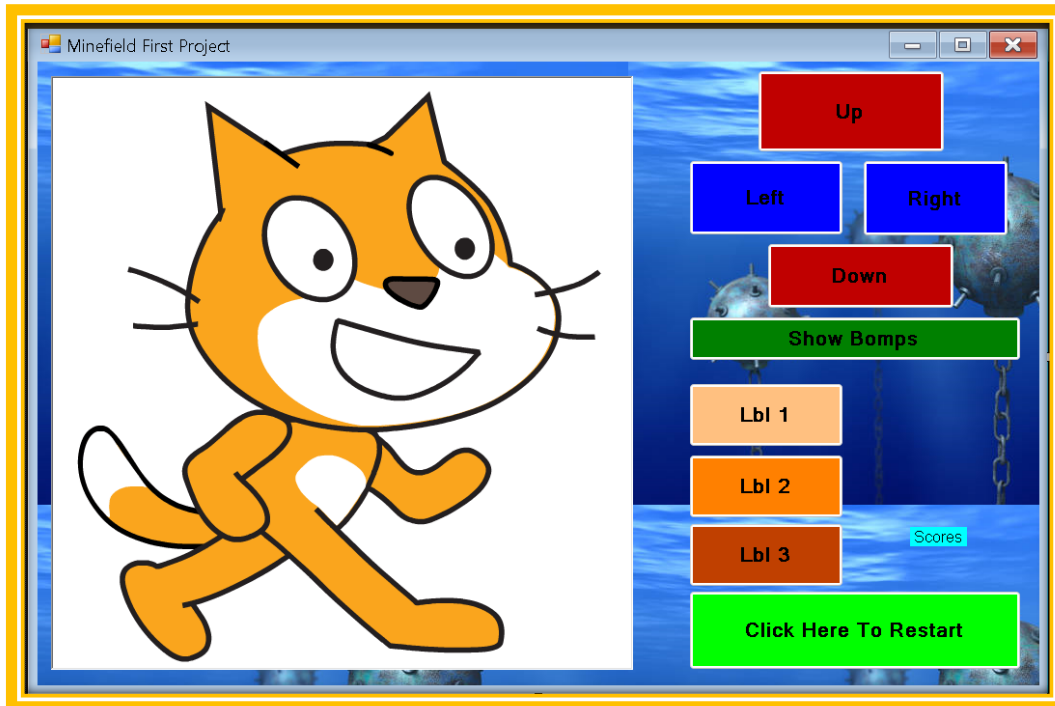
- Data Handling Errors

Some task inputs were not processed correctly due to missing validation

Solution: added input validation checks in the backed.

Step 2: Frontend Development

The user interface was developed using Windows Form, with labels representing grid cells and buttons controlling gameplay. Here is screenshot of task input interface that allows users to easily play the game.



Task 2: T2: Innovation

Question 1: Concepts, Methods, Practices and Tools

The project applied agile development practices with regular iteration, user feedback, and testing. Visual Studio and GitHub supported development and version control, while planning [2], [3]. Amazon Web Services would support development and cloud services such as continuous integration and continuous deployment CI/CD with user feedback to maintain and improve the game. [4]

Question 2: PEST Analysis

A PEST analysis evaluates the Political, Economic, Social, and Technological factors affecting a game.

✚ Political factors affecting:

Any platform collecting user data must ensure proper security and consent procedures. GDPR must be followed. [5]

✚ Economic Factors affecting

Student spending power can affect subscriptions, especially during economic downturns. Students may rely more on free versions of games.

✚ Social factors affecting:

Students increasingly use digital productivity tools and apps. There is also growing interest in tools that help manage workload and reduce stress.

✚ Technological factors

Advances in AI and cloud computing make it easier to develop smart applications, but companies must keep updating technology to stay competitive.

✚ Legal and Ethical:

Businesses must ensure data privacy, responsible AI use, prevent algorithm bias, and maintain accessibility for all users.

Task 3: Project Feedback and Reflection

Key Survey questions

- ❖ How do you evaluate the current implementation?
- ❖ Are there any additional features that would improve it?
- ❖ What are your thoughts on the UI, UX, output, gameplay, and overall environment?
- ❖ How effective is the color scheme, design, and visual environment?

Respondent

<

1

Aaron Miles

01:46

>

Time to complete

...

1. How do you evaluate the current implementation?

It helps students relax and take short breaks from studying.

2. Are there any additional features that would improve it?

The game screen is organised and not confusing.

3. What are your thoughts on the UI, output, gameplay, and overall environment?

Level progression that increases difficulty.

Respondent

<

2

Daniel Leone

05:12

>

Time to complete

...

1. How do you evaluate the current implementation?

Good for a quick game to relax and have fun

2. Are there any additional features that would improve it?

Mostly yes. Subjective opinion but I would reduce saturation of colours. maybe too bright?

3. What are your thoughts on the UI, output, gameplay, and overall environment?

Its good to be able to review the outcome of the each game and display where the bombs are.

Respondent

< 3 Francois Riches 07:18 >

Time to complete

...

1. How do you evaluate the current implementation?

Is the game specifically for students? Is it an educational game?

2. Are there any additional features that would improve it?

The game is a bit confusing and not entirely intuitive. It was easy while being helped through by the creator, but I think the game UI could be easier to follow.

3. What are your thoughts on the UI, output, gameplay, and overall environment?

Hard to say at this point.

Respondent

< 4 Sanwar Ahmed 01:08 >

Time to complete

...

1. How do you evaluate the current implementation?

Classic game

2. Are there any additional features that would improve it?

ok

3. What are your thoughts on the UI, output, gameplay, and overall environment?

Puzzle based

respondent

<

6

Hasan Sivas

06:05

Time to complete

>

...

- How do you evaluate the current implementation?

fun to use
- Are there any additional features that would improve it?

make show mines debug feature into gameplay e.g. allow the user to reveal mines in a certain radius at the cost of points once they've accumulated a suitable amount
- What are your thoughts on the UI, output, gameplay, and overall environment?

gameplay -easy to understand and use UI - Visually readable but could be refined - sound effects for triggering a mine would also be good

Question 1: Practical Feedback

User feedback evaluated usability and functionality. Most users found the interface simple and intuitive. Positive aspects included:

- Clear interface, Easy navigation and Functional game mechanics.

The suggested improvements enhance visual design and introduce additional interactive elements. These will be implemented in future iterations.

- Feedback Summary:

Most users found the game was easy to plays and positive feedback included:

- ✚ Simple interface
- ✚ Quick schedule generation
- ✚ Clear task organisation

Suggested improvements included:

Improving visual design, adding calendar integration.

Based on this feedback I would improve user interface design.

Test Results after practical feedback

Objective: Evaluate the functionality of the Minefield Project game with a focus on user experience and functionality without delving into the internal code structure.

Loading and Running:
Observation: The game successfully loaded and run without issues.
Status: Pass
User Interface:
Observation: The welcome message and form displayed correctly, enhancing user experience.
Status: Pass
Functionality:
Observation: The test button operated as intended, and scores were saved properly.
Status: Pass
Audio:
Observation: The sound played appropriately during gameplay, if a bomb is hit, contributing to immersion.
Status: Pass

After feedback tested and improvements

All key features worked as expected, ensuring that players can enjoy the game without any technical issues. Final Minefield Game Simulation with welcome page you can see below screen shot as evident. No further actions are required currently

Figure 4 welcome page

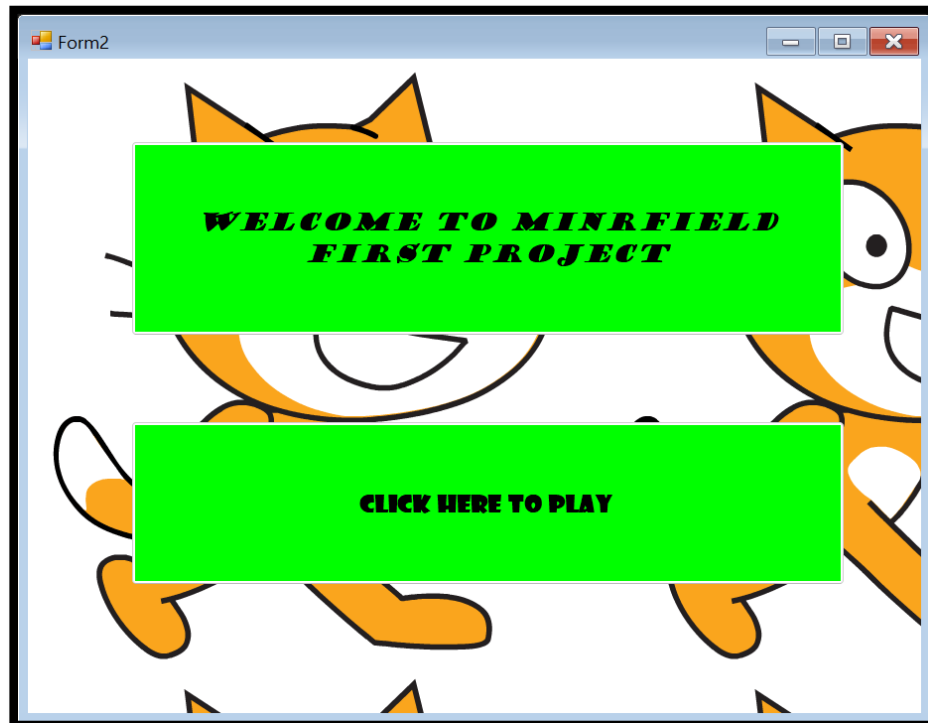
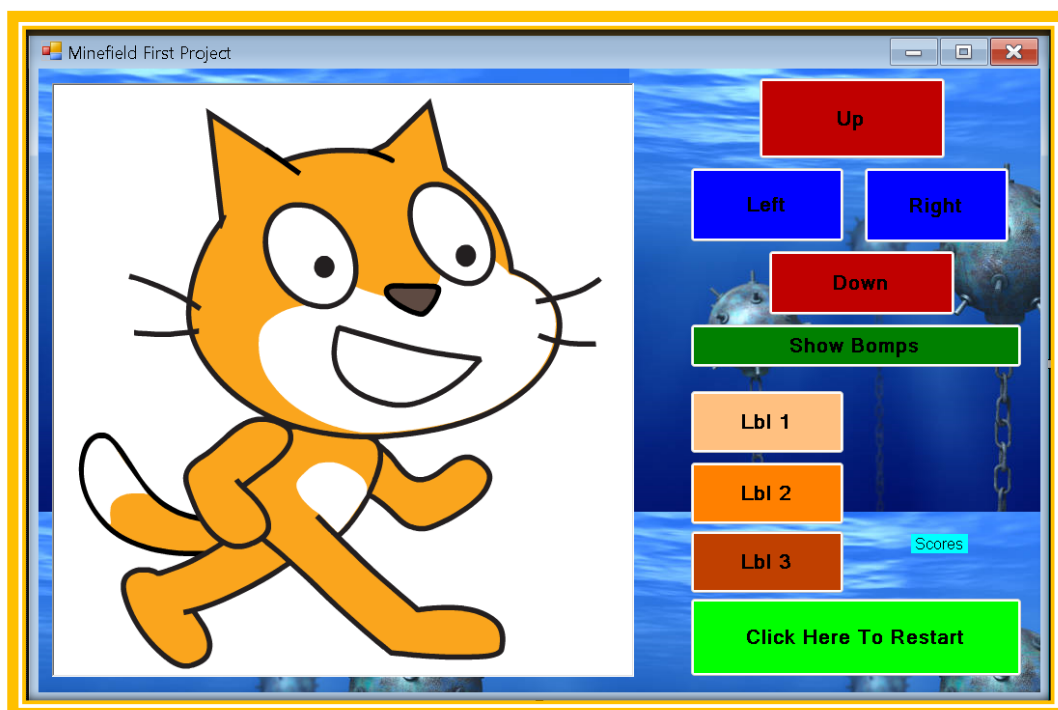


Figure 5 Final Minefield Game Simulation



Question 2: Project Progress Diary

I maintained weekly progress sheets to track development milestones and tasks, this helped manage deadlines, break down large tasks, and identify delays early, ensuring consistent Agile workflow. The weekly progress sheets are included in the appendices at the end of the document.

Question 3: Skills Developed

During this module I developed several technical and professional skills including:

❖ Technical skills include:

- ✚ C# programming and Windows Forms development
- ✚ Event driven programming and debugging
- ✚ GitHub version control

❖ Professional skills include:

Project planning, time management and documentation and responding user feedback analysis. Problem solving, technical writing and analytical thinking. These skills are highly relevant for careers in software development and AI systems design.

Question 4: Positive Social Impact

This project could have a positive impact on society by helping students manage their time more efficiently. Poor time management causes stress and academic problems. This game helps students plan their study time efficiently.

- ✚ Reduce stress and academic issues
- ✚ Boosts productivity and learning.
- ✚ Develops AI and software engineering skills.

User Interactions

Figure 6 Final Minefield Gameplay Simulation

As shown in Figure 6, Label one with 30 random Bomps and one sprite image.



Figure 7 Label Two

Label two with 50 Bomps and one sprite image.

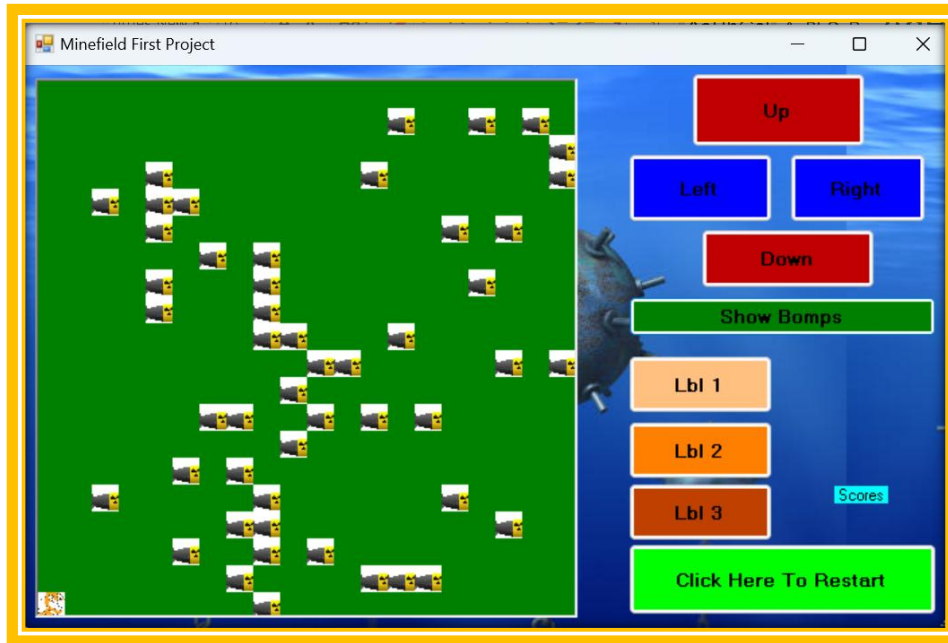
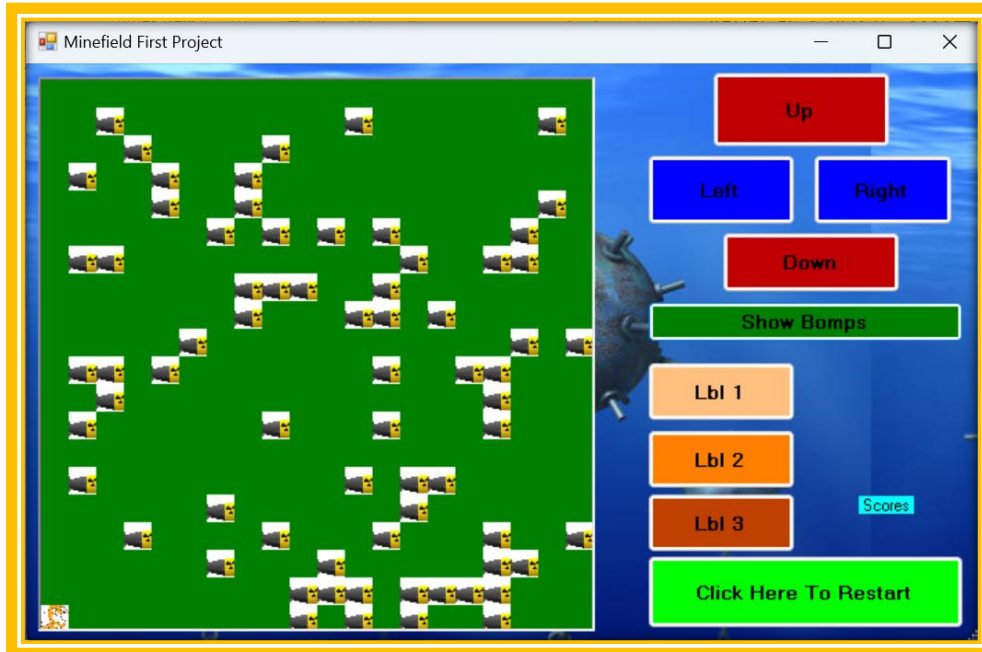


Figure 8 Label Three

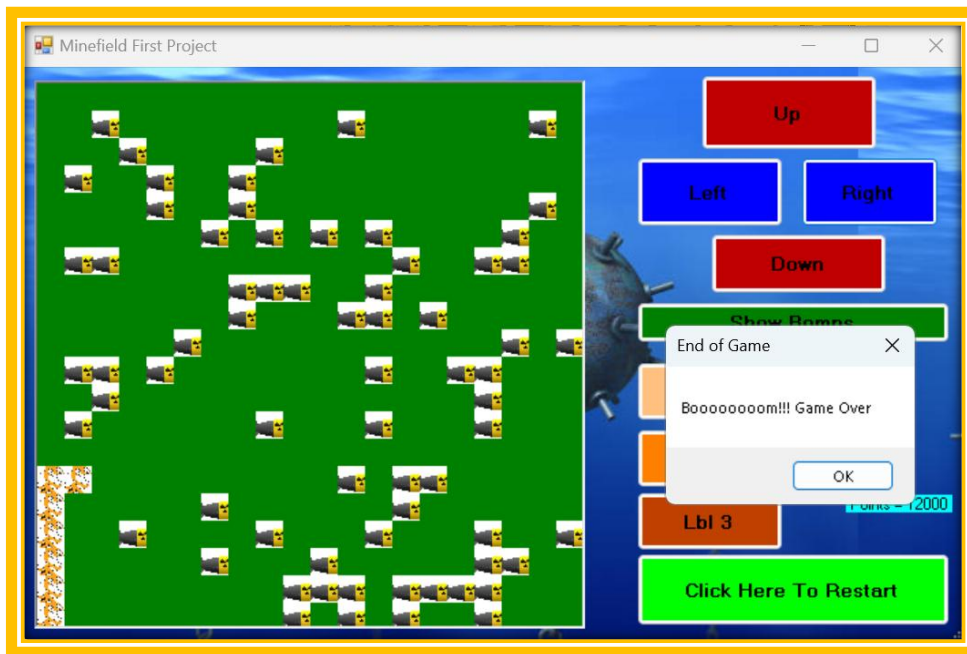
Label three with 70 Bomps and one Sprite image.



Game Over Screen

Figure 9 Game Over Notifications

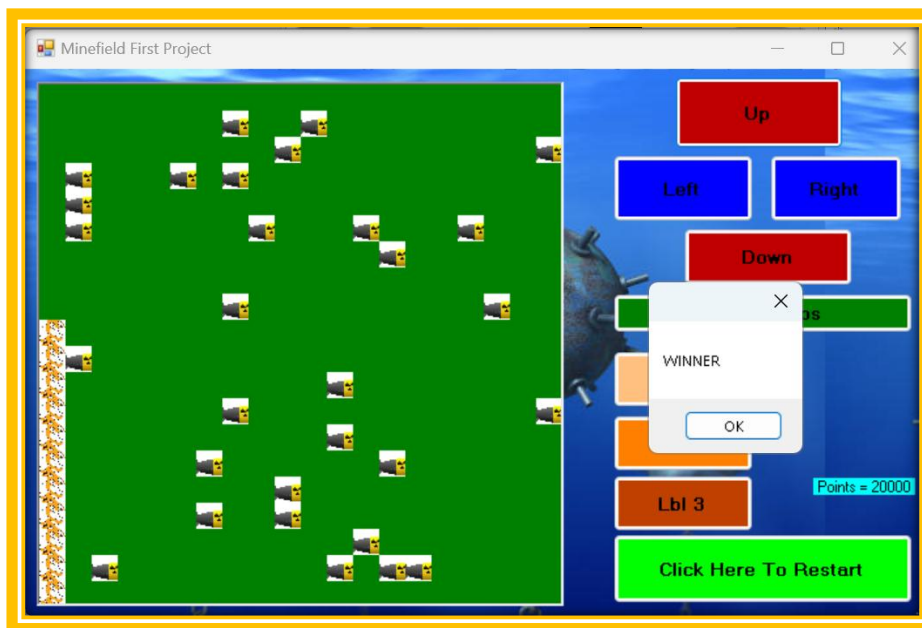
End of game I had hit the Bombs and the message below shown as “Boooooom”! Game Over



Winning State

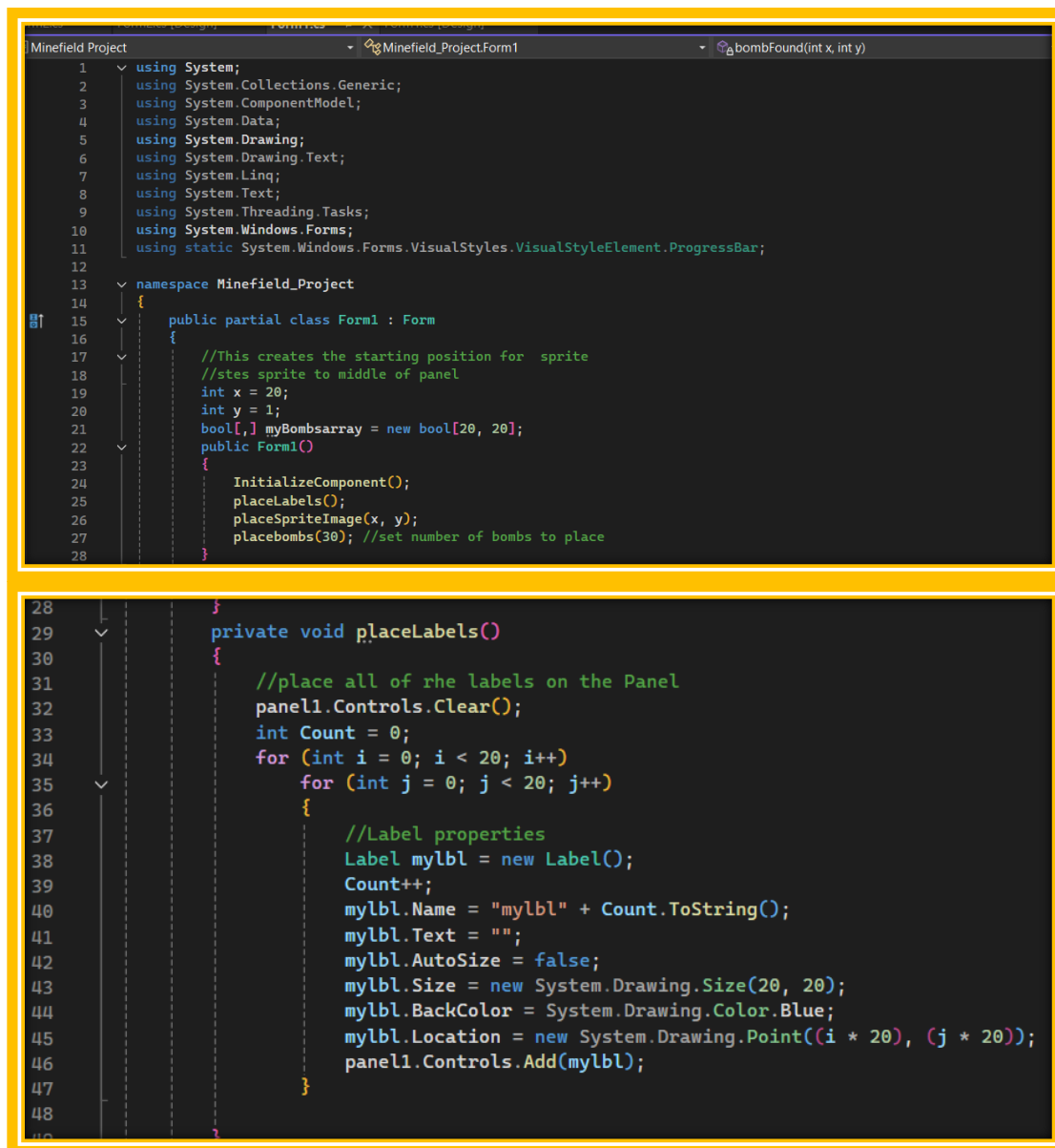
Figure 10 The player successful Game completed the game.

The game was easy to use and understand.



Code Snippets:

These key sections of C# code snippets illustrate the implementation of methods, objects, and features in a Windows Form application using C#. To understand and utilize these components effectively, see the screenshots below.



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Drawing.Text;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11 using static System.Windows.Forms.VisualStyles.VisualStyleElement.ProgressBar;
12
13 namespace Minefield_Project
14 {
15     public partial class Form1 : Form
16     {
17         //This creates the starting position for sprite
18         //stes sprite to middle of panel
19         int x = 20;
20         int y = 1;
21         bool[,] myBombsarray = new bool[20, 20];
22         public Form1()
23         {
24             InitializeComponent();
25             placeLabels();
26             placeSpriteImage(x, y);
27             placebombs(30); //set number of bombs to place
28         }
29
30         private void placeLabels()
31         {
32             //place all of rhe labels on the Panel
33             panel1.Controls.Clear();
34             int Count = 0;
35             for (int i = 0; i < 20; i++)
36                 for (int j = 0; j < 20; j++)
37                 {
38                     //Label properties
39                     Label mylbl = new Label();
40                     Count++;
41                     mylbl.Name = "mylbl" + Count.ToString();
42                     mylbl.Text = "";
43                     mylbl.AutoSize = false;
44                     mylbl.Size = new System.Drawing.Size(20, 20);
45                     mylbl.BackColor = System.Drawing.Color.Blue;
46                     mylbl.Location = new System.Drawing.Point((i * 20), (j * 20));
47                     panel1.Controls.Add(mylbl);
48                 }
49         }
50     }
51 }
```

```

49     }
50     private Label Getmylbl(int x, int y)
51     {
52         //obtaining label position within the panel
53         int mulblNumber = (y - 1) * 20 + x;
54         string mylblName = "mylbl" + mulblNumber.ToString();
55
56         foreach (Control ctrl in panel1.Controls)
57         {
58             if (ctrl.GetType() == typeof(Label) && ctrl.Name == mylblName)
59             {
60                 return (Label)ctrl;
61             }
62         }
63         return null;
64     }
65     private void placeSpriteImage(int x, int y)
66     {
67         //This draws the spriteImage within the panel
68         Label mylbl = Getmylbl(x, y);
69         mylbl.Image = Properties.Resources.SpriteImage;
70     }
71     private void Form1_Load(object sender, EventArgs e)
72     {
73         MessageBox.Show("Welcome to Minefield");
74     }

```

```

75     private void btnRight_Click(object sender, EventArgs e)
76     {
77         //Getting the spriteImage to move with a trail
78         if (y < 20)
79         {
80             y = y + 1;
81             placeSpriteImage(x, y);
82             bombFound(x, y);
83         }
84     }
85
86     private void btnLeft_Click(object sender, EventArgs e)
87     {
88         //Getting the spriteImage to move with a trail
89         if (y > 1)
90         {
91             y = y - 1;
92             placeSpriteImage(x, y);
93             bombFound(x, y);
94         }
95     }
96
97
98

```

```

98
99  private void btnUp_Click(object sender, EventArgs e)
100  {
101      //Getting the spriteImage to move with a trail
102      if (x > 1)
103      {
104          x = x - 1;
105          placeSpriteImage(x, y);
106          bombFound(x, y);
107      }
108  }
109
110
111  private void btnDown_Click(object sender, EventArgs e)
112  {
113      //Getting the spriteImage to move with a trail
114      if (x < 20)
115      {
116          x = x + 1;
117          placeSpriteImage(x, y);
118          bombFound(x, y);
119      }
120  }
121

```

```

122  private void placebombs(int bullseye)
123  {
124      Random r = new Random();
125      //ste up variables
126      int x;
127      int y;
128      int k = bullseye;
129
130      //clear the current bombs list
131      Array.Clear(myBombsarray, 0, myBombsarray.Length);
132
133      //loop to fill with desired number of mines
134      do
135      {
136          x = r.Next(1, 20);
137          y = r.Next(1, 20);
138          if (!myBombsarray[x, y])
139          {
140              myBombsarray[x, y] = true;
141              k--;
142          }
143      } while (k > 0);
144
145  }
146

```

```

147 private void showbombs()
148 {
149     for (int i = 1; i <= 20; i++)
150     {
151         for (int j = 1; j <= 20; j++)
152         {
153             Label l = Getmylbl(i, j);
154             if (myBombsarray[i - 1, j - 1] == true)
155             {
156                 //set label to bomb image
157                 l.Image = Properties.Resources.BOMB;
158             }
159             //set label to different colour
160             l.BackColor = Color.Green;
161         }
162     }
163 }
164
165 private void bombFound(int x, int y)
166 {
167     int bombCount = 0;
168     int newX;
169     int newY;
170     if (myBombsarray[x - 1, y - 1] == true)
171     {
172         MessageBox.Show("Boooooooooom!!! Game Over", "End of Game", MessageBoxButtons.OK);
173         showbombs();

```

```

174     }
175     else
176     {
177         newX = x - 1;
178         if (newX > 0)
179         {
180             if (myBombsarray[newX - 1, y - 1] == true)
181             {
182                 bombCount++;
183             }
184         }
185         newX = x + 1;
186         if (newX < 21)
187         {
188             if (myBombsarray[newX - 1, y - 1])
189             {
190                 bombCount++;
191             }
192         }
193         newY = y - 1;
194         if (newY > 0)
195         {
196             if (myBombsarray[x - 1, newY - 1] == true)
197             {
198                 bombCount++;
199             }

```

```

193 |         newY = y - 1;
194 |         if (newY > 0)
195 |         {
196 |             if (myBombsarray[x - 1, newY - 1] == true)
197 |             {
198 |                 bombCount++;
199 |             }
200 |             newY = y + 1;
201 |             if (newY < 21)
202 |             {
203 |                 if (myBombsarray[x - 1, newY - 1])
204 |                 {
205 |                     bombCount++;
206 |                 }
207 |             }
208 |         }
209 |     }
210 | }
211 |
212 | private void btnShowBombs_Click(object sender, EventArgs e)
213 | {
214 |     showbombs();
215 | }
216 |
217 | }

```

Conclusion

The Minefield project successfully demonstrates the application of programming, design, and problem-solving skills through the development of an interactive Windows Forms Application. The project achieved its objectives by implementing a functional grid-based game using event driven programming and user interface design principles.

User feedback and testing confirmed that the application is functional, intuitive, and engaging. Additionally, the project highlights the importance of planning, iterative development and continuous improvement.

Overall, this project provides a strong foundation for future development and demonstrates key technical and professional skills relevant to software development or engineering.

References

- [1] learn.microsoft.com. [Online]. Available: <https://learn.microsoft.com/en-us/cpp/?view=msvc-170>. [Accessed 2026 02 12].
- [2] [Online]. Available: <https://learn.microsoft.com/en-gb/visualstudio/?view=visualstudio>. [Accessed 22 02 2026].
- [3] [Online]. Available: <https://docs.github.com/en>. [Accessed 12 04 2026].
- [4] [Online]. Available: <https://aws.amazon.com/>. [Accessed 03 23 2026].
- [5] [Online]. Available: <https://gdpr.eu/>. [Accessed 23 02 2026].
- [6] B. S, 15 12 2025. [Online]. Available: <https://www.nimblework.com/agile/agile-methodology/>. [Accessed 17 02 2026].

Appendices'

Appendix A: Weekly Project progress Sheets

Project Progress Sheets:

Completed this during each weekly tutorial to track and manage the implementation part of the project and the completion of assignment 2.

Project progress form

Name: Ibrahim Yahia Mohamed (2010957)

Programme: Changemakers Assignment 2: Practical & Professional

Development: Changemakers, Creativity and Value Creation

Project Aim:

To design and develop a Minefield game using C# Windows Forms that demonstrates event driven programming and user interaction, and problem-solving techniques.

Project Objectives:

- | | |
|--|---|
|  To design a functional grid-based game interface |  To ensure user-friendly design and clear feedback messages |
|  To implement random bomb placement logic | |
|  To develop interactive gameplay using event-driven programming |  To test and evaluate the functionality of the game |

Weekly Project progress forms:

Date: Wednesday & 11/03/2026

Progress:

- Tutor show Assignment 2
- Started practical implementation

Actions:

State priority actions to undertake over the next week below.

1. Continuous practical implementation of my project
2. Documented

Next review date: Wednesday & 18/03/2026.

Date: Wednesday & 18/03/2026

Progress:

- Continuous practical implementation

Actions:

1. Continuous practical implementation of my project
2. Documented

Next review date: Wednesday & 25/03/2026.

Date: Wednesday & 25/03/2026

Progress:

- Continuous implementation

Actions:

Update Gantt Chart

Undertake over the next week below.

- Continuous practical implementation of my project
- Documented

Next review date: Wednesday & 01/04/2026.

Date: Wednesday & 01/04/2026

Progress:

- Continuous practical implementation

Actions:

- Continuous practical implementation and Documented

Next review date: Wednesday & 08/04/2026.

Date: Wednesday & 08/04/2026

Progress:

- Continuous practical implementation

Actions:

- Continuous practical implementation of my project
- Documented

Next review date: Wednesday & 1/04/2026.

Date: Wednesday & 22/04/2026

Progress:

- Practical implementation

Actions:

- Continuous practical implementation of my project
- Documented

Next review date: Wednesday & 29/04/2026.

Date: Wednesday & 29/04/2026

Progress:

- Practical implementation

Actions:

- Final practical implementation and documented and **submission**

Final review date and submission: Wednesday & 29/04/2026.