Software Design and Development – Major Project

Defusal Doyen

[Edward Liu]

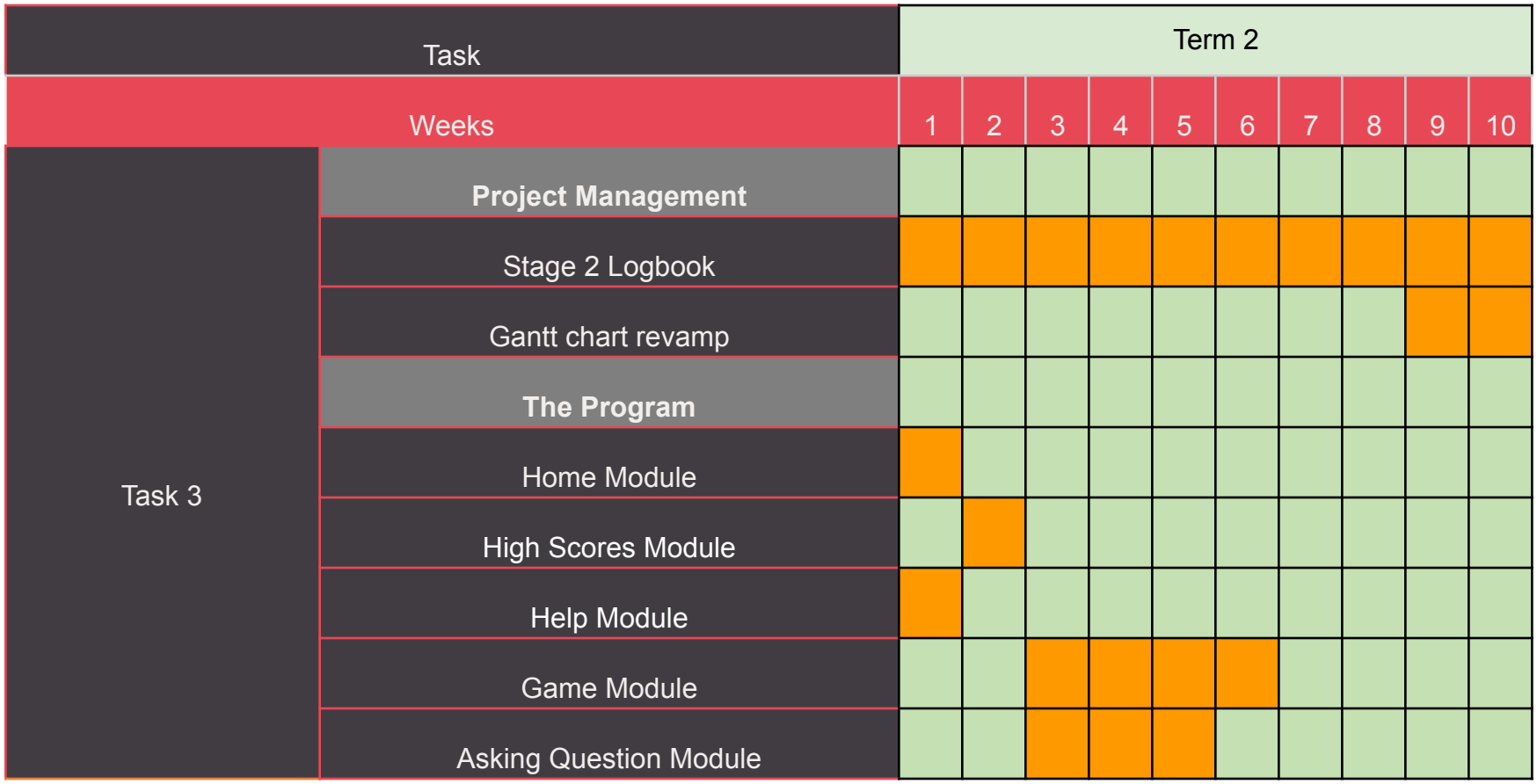
Contents

Contact us at defusaldoyen@gmail.com

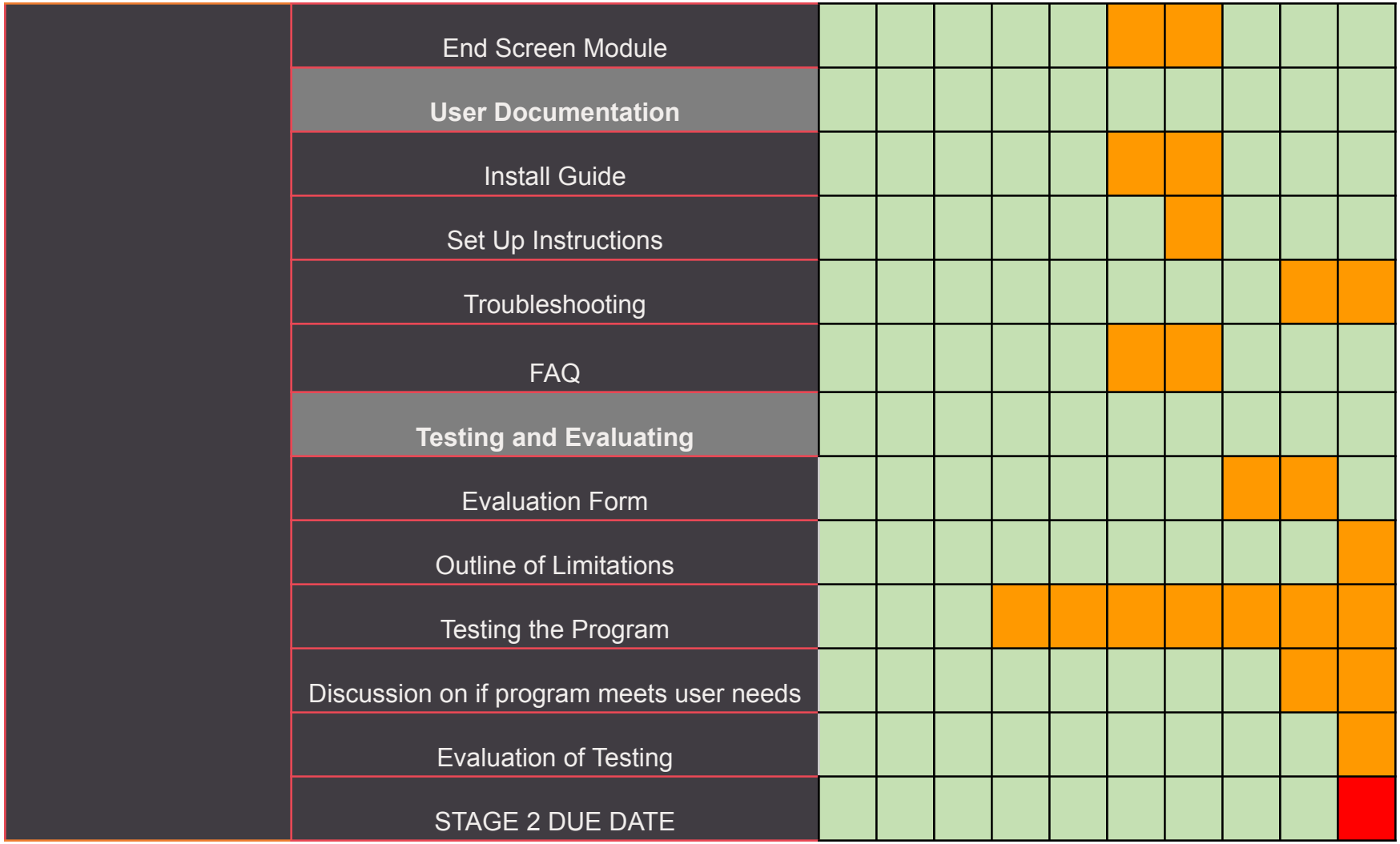
Project Management

Gantt Chart

Initial:

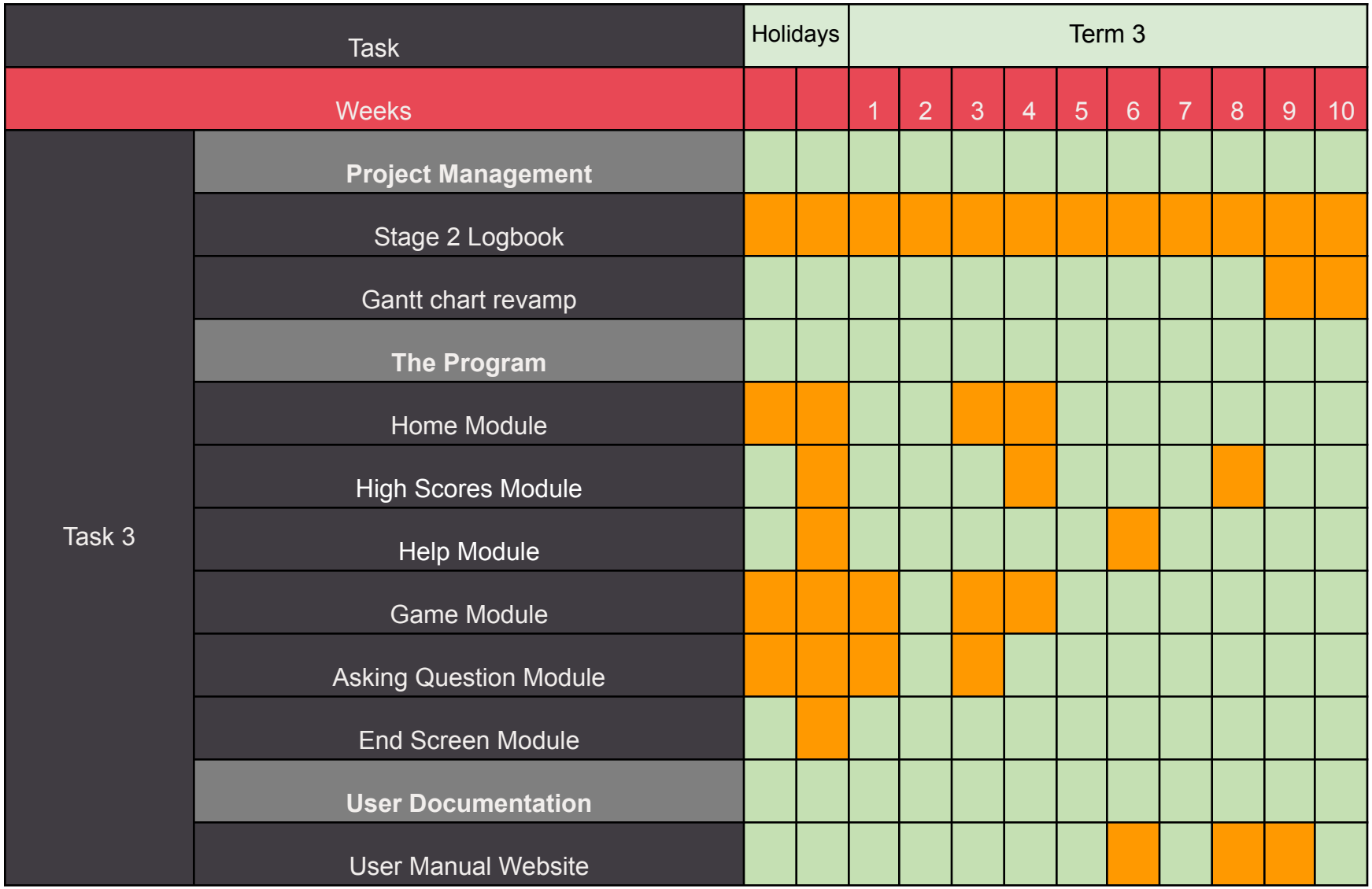


Contact us at defusaldoyen@gmail.com

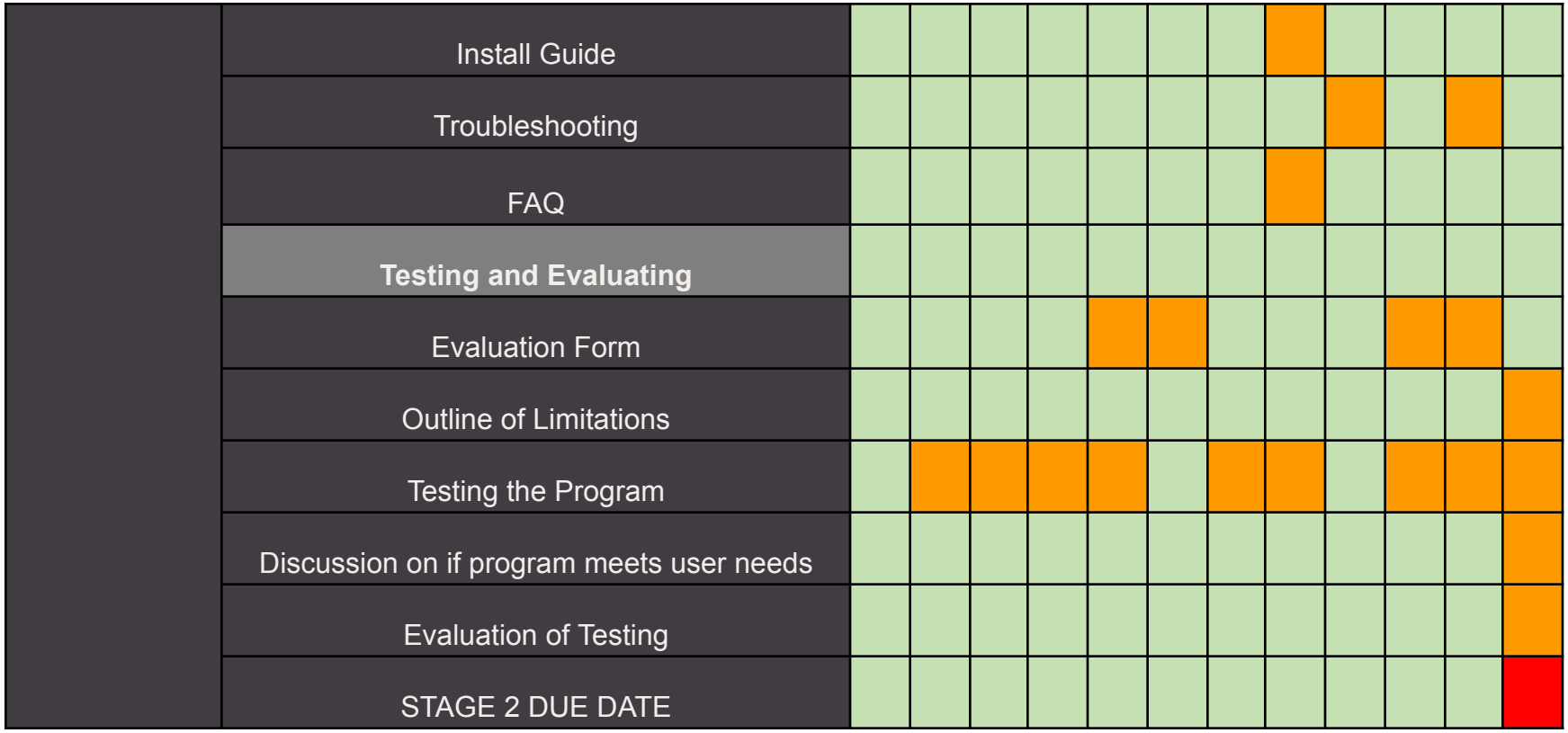


Contact us at defusaldoyen@gmail.com

Final:



Contact us at defusaldoyen@gmail.com


Install Guide
Troubleshooting
FAQ
Testing and Evaluating
Evaluation Form
Outline of Limitations
Testing the Program
Discussion on if program meets user needs
Evaluation of Testing
STAGE 2 DUE DATE

Logbook

Date	Progress
Throughout Week 7 to Week 10 (The previous Project)	Finished an initial version of the project that created a board and allowed for opening and displaying the tiles along with the game rules. This was to get screenshots of the game for the previous section.
Week 11 9/04/24 (In Class)	Worked on moving functionality of opening tiles and flagging tiles to the Game Manager object instead of doing it locally in the Tile Object Script, as Game Manager object is essentially the code for the main game.
Week 11 11/04/24 (In Class)	Created a function that checks the win condition of the game, and made buttons for the powerups and created code to label how many of each power up the user has.
Week 11 13/04/24	Worked on functionality for displaying the timer and changing the color of the tile if the user is hovering over it for easier visibility
Week 1 Holidays 17/04/24	Figured out how to have 0 bombs next to the tile the user first clicked by changing the initial board loading such that the bomb positions and numbers on tiles are generated after the user clicks first. Completed power up functionality for bomb flagger and defusing report, with added buttons for the powerups Initial “bomb flagger” code had flagged to many bombs since when in the for loop, there was no end condition to how many bombs were flagged in a row.
Week 1 Holidays 21/04/24	Finished functionality on the anti bomb button, with disabling other buttons when anti bomb is being used. Also implemented changing color on tiles to tell the user which tiles the anti bomb will affect. Some of the tiles weren't getting opened, specifically in the rows right and columns below the tile selected, however it was because the for loop needed an equals sign. Attempted to work on saving data going between the question page/scene and the game scene There is a challenge of changing the program to static data (This is global data for C#), and having less reliance on instantiated or created game objects in the game scene for storing data.

Week 2 Holidays 22/04/24	Finished converting the information on tiles (bombs on it, if it's revealed etc), timer, whether the user has opened their first tile and the number of powerups the users have to be stored as static data to allow it to be kept constant between scenes in unity. I had to learn how to manage static variables and referencing them in the program. Also replacing the existing structure of the code that didn't use static variables led to errors showing in the program initially, with my initial approach being bugged with errors in converting reference to the StaticData class. Figured out how reading files and creating files works in C# and also made a design for the question page, with reading questions from file implemented, and displaying question and question responses as buttons also implemented
Week 2 Holidays 23/04/24	Implemented full functionality of the question page in giving visual response to whether the user got the question right and wrong, with adding a power up to the user's inventory after answering a question correctly. Also designed the question page to fit the description given in the first task. Implemented Help Menu in the game based on the screen designs (No issue, as it is just text and no logic) Learning how to delay a function by a certain time in unity, as unity uses in-built methods to delay the application.
Week 2 Holidays 24/04/24	Added glowing effects to the game through universal pipeline.
Week 2 Holidays 25/04/24	Added audio to the game through pixabay with taking into account their license policies. Also learnt about how to add audio to Unity, and added a result to losing the game. Audio manager wasn't playing consistently across scenes, I added a do not destroy on load function to not destroy the audio manager between scenes. Added a camera shake effect to the game when opening a tile with no bombs next to it, and added more audio clips for the questioning page, as well as designing the endscreen pages. Also optimized the code by reducing the number of times the Check Win Condition function was called, and found errors in the code when a random chance to go to the question page was still generated even when the user won. Added text to tell users how

	many bombs they flagged and how many bombs are on the board to make the defusing report more useful.
Week 2 Holidays 26/04/24	Played the game for a bit to debug it, and fixed a bug where it can't detect when the player has won. Also changed the screen size to be 1600x900 to fit on smaller screen sizes. Looked at a project when searching up "Minesweeper with powerups" and found https://www.youtube.com/watch?v=sRE6Axt8Z3Q and discussed new features to add to my game with a friend
Week 2 Holidays 27/04/24	Added particle effects and a description on hover to the power ups, and learnt using github terminal to back up my project.
Week 2 Holidays 28/04/24	Changed up the UI of the home page and end pages to use circular buttons instead with on hover description, with Unity's inbuilt lighting system to make them visible to the user. Designed new button layouts in Piskel Also changed the type of light on the tiles to make it more aesthetic. Also added high scores functionality
Week 1 29/04/24	Researched on why distributing to mac not working with error message of "File corrupted", and found out apple automatically adds a "Quarantine" flag to every app when downloaded off the internet, and this flag cannot be removed unless through user local commands or if the app is notarised. Notarising the app costs $100 per yr, making it unviable. Hence I am researching deployment onto WebGL. Discovered there are more complexities to a WebGL build regarding the access of files (in Web Pages, they use local storage). Hence, for MacOS, I will instruct the user to delete the quarantine flag through commands. Added lighting effects for anti-bomb to tell users where the anti-bomb affects.
Week 1 2/05/24	Changed the way that powerup distribution works, where instead the user is gifted a random amount of power ups instead of just 1. It also negates the need for there to be a high probability to be sent to the questions page. However, this brings the issue of balancing into the game, as the user getting multiple power ups is unfair with completion of the game being limited in time. Changed the board to a 20x20 grid and played the game to determine the correct amount of bombs to make the game fair (I ended up with 60). However, the Defusing Report has always been a useless

	power-up that I have never used in any play through of the game. Changed the defusing report power up to a bomb tester power up instead, that allows the user "The next tile clicked will be flagged if it is a bomb, and opened if safe".
Week 1 4/05/24	Designed a sprite and added a new powerup called "Plane Cho", which would open a column up, flagging tiles that are bombs. Also reformed the changing light into a function (modularising the code). There were issues of the effect of showing where the power-up would hit in that the lights were not activated, however through debugging it, it was found that there hadn't been a statement to make the lights active. Additionally when modularising the code, a problem arose when flagging a tile would make the sprite appear to flicker. The issue was in the lighting effect of the tile, and by disabling that for a flagged tile it solved the issue.
Week 1 7/05/24	Added features on telling users on what power ups they received in the question page, and also reformed variable names.
Week 2 9/05/24	Added comments to the classes in my code. (Sound, StaticData, TileData, DescOnHover, DisplayTime, FollowMe, TileScript)
Week 3 13/05/24	Made a feedback form for people to do after playing the game. Designed an icon for the mute button, and also added a mute functionality into the game (Mute button) based on feedback response, and returned back to the tile sprites that had more visible numbers based on user feedback
Week 3 15/05/24	Received feedback from classmates on the game. Allowed the plane cho power up to be used in a row and column, and fixed the glow effect for changing between a row and a column, and removed possibility of being asked duplicate questions in a single game.
Week 4 21/05/24	Added the difficulty setting into the game - Added Difficulty Selection Page (Addition to home page) - Added Board Generation of different sizes

Week 4 23/05/24	Changed the style of elements in the page (Text Orientation to centered), Fixed some of the colliders on the buttons and toggled with chances to receive a power up.
Week 6 4/06/24	Changed help menu to include the tiles that are opened and unopened, and finished installation guide and FAQs.
Week 6 6/06/24	Finished FAQ section and started on Google Site help menu, added 1st page of game rules.
Week 7 6/12/24	Started on troubleshooting section, added the case where the question page didn't display any information
Week 8 17/06/24	Revised code comments for modules of StaticData, TileScript, IsHighScore and GameManager to account for the added difficulty setting and switching between row and column for using power up of Plane Cho. Fixed help menu in game for Power Up sections for better clarity on how to use the power ups on the board.
Week 8 20/06/24	Worked on the introduction page formatting as well as descriptions in the website to improve user navigation
Week 9 24/06/24	Finished the home page of the Defusal Doyen help website, and finished the section on the basic rules of minesweeper as well as the controls along with images of the game to support it. Also got extra feedback from a friend. Fixed up some wording in the power ups help page in the description of the anti bomb, as well as finishing pages for Using the Question Page and starting How to use the Power Ups page
Week 9 25/06/24	Finished making the video tutorial/walkthrough of the game, and finished the "How to use the power ups page"
Week 9 26/06/24	Finished editing the website to add file access in the website, and added details on the types of errors that can be received in the game and how to troubleshoot them. Edited the user manual to include the files for the game and the google site. Also resent an email to gain more user feedback.

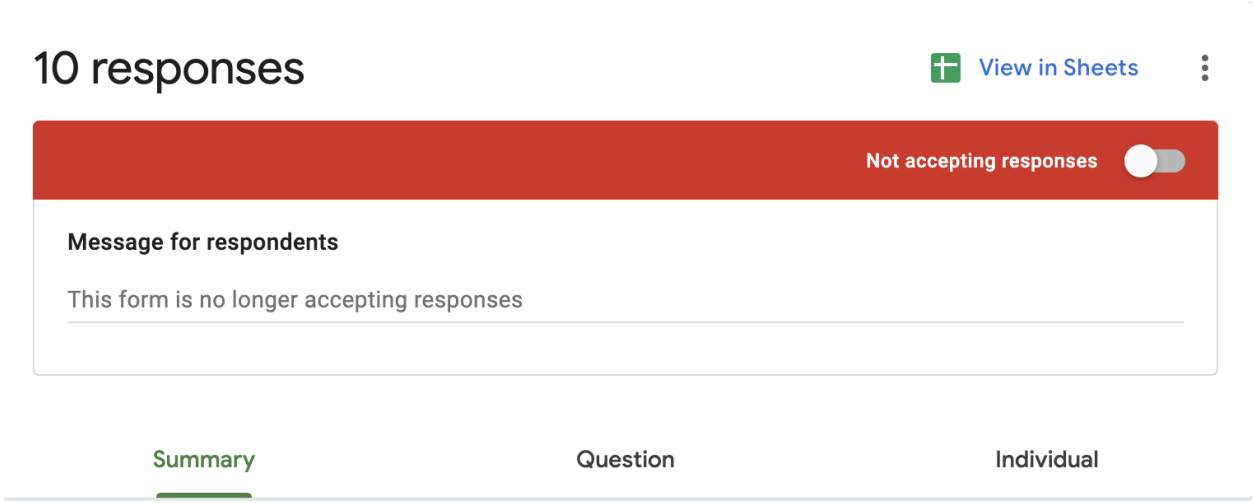
Week 9 27/06/24	Finished formatting the code into the documentation, and showed Mr Beilby the game for feedback.
Week 10 1/07/24	Pasted form logbook responses onto the documentation logbook, and also started on the evaluation of user feedback section with adding Q1-4 and adding paragraphs below to explain how each question identified areas of concern. Finished evaluation of user feedback on positive and negative responses to the game. Also finished the self evaluation of the game with Language Used, System Limitations, Changes to Initial Concept, User Needs.
Week 10 2/07/24	Made formatting on the document online such that on the document, you can jump between organized sections, and revised the contents page of the google doc. Didn't know how to initially set headings, but learnt from a friend. However, an issue occurred on having to manually rechange all of the fonts back as they defaulted to arial, without underline or bolding Finalized self-evaluation of system and the testing report with 3 testing methods (Performance Testing, Black Box Functionality Testing, Debugging Output Statement)

Evaluation of System

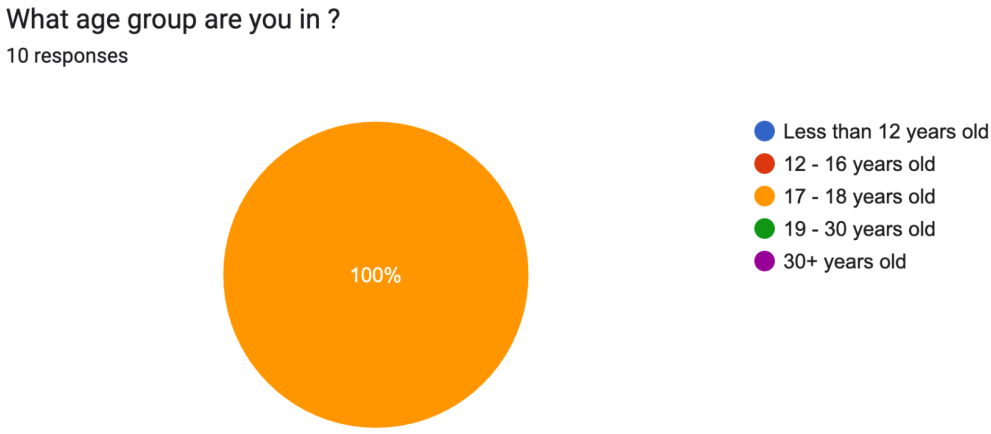
Evaluation of User Feedback:

This is from the form: (https://forms.gle/azWnqLG622EuL2WM8), a feedback form for the game.

In total: 10 responses were collected, which is an adequate amount of feedback to allow for better design of the product to fulfill user needs.



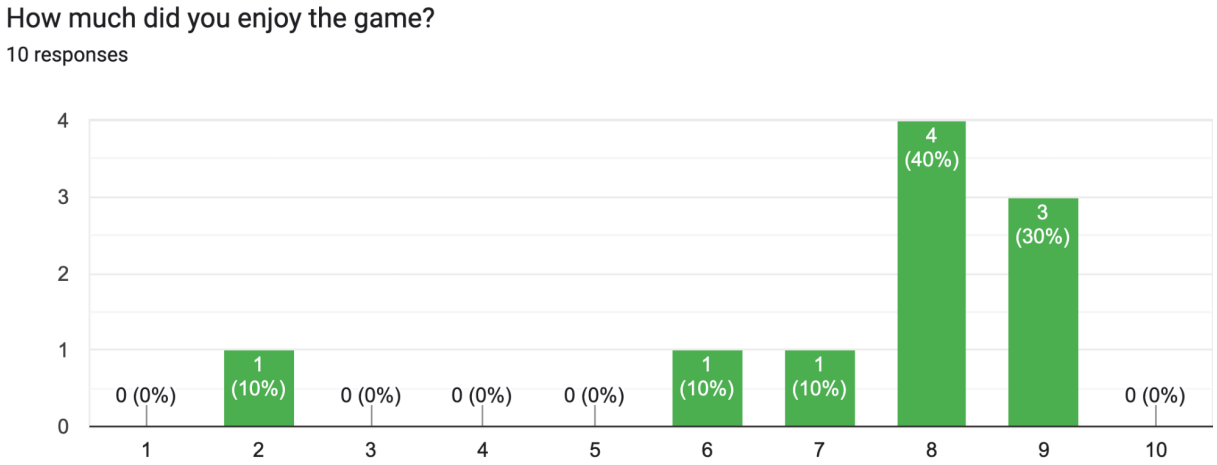
Question 1:



Unfortunately, there was a lack of responses from any other ages other than 17-18 years old, as I was only able to get responses mainly from the Y12 Cohort rather than younger or

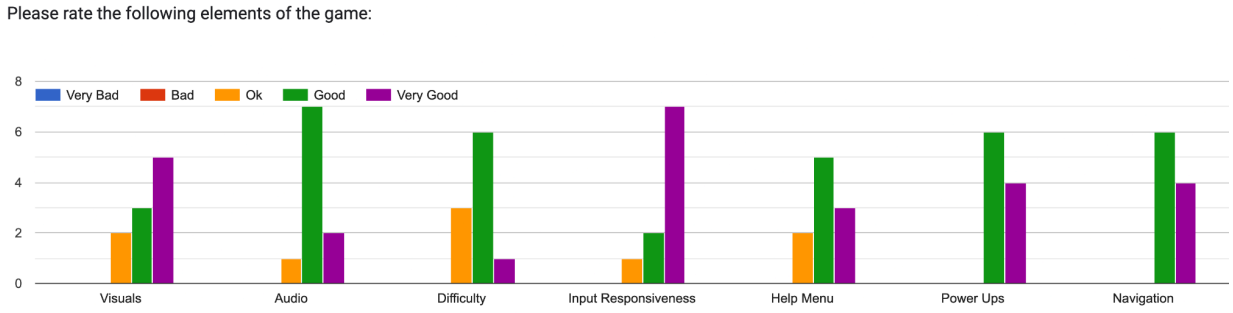
older ages. However, considering that the target audience is Y11-Y12 as a consolidation of knowledge of topics, this was acceptable in getting feedback from this specific age range.

Question 2:



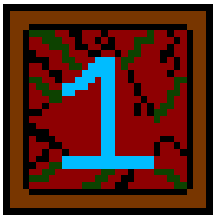
As seen above, most people who completed the feedback form enjoyed the game with an outlier value of a 2. The form responses are for both before and after the feedback for the game was implemented. Hence, response to the game was mostly positive, suggesting that it fulfilled most user requirements.

Question 3:



This question identifies the areas of concern to be the difficulty, help menu and visuals. This reflects the further feedback of the difficulty level of the game being excessively high, and the help menu being relatively unformatted and had some information missing.

Additionally, there were initial concerns on the small size of the tiles initially, where users complained about the difficulty of seeing what number was on the tiles. This was eventually replaced with tile sprites such that the user would be able to see the number on the tiles.



Questions 5, 6: Asking the user what they liked about the game and what they didn't like about the game:

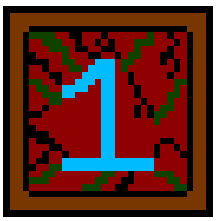
Summary of Positive Feedback:

- Most people found the game fun
- Users liked the dynamic lighting and particle effects
- Visuals/Graphics were good
- Liked the use of power ups in the game
- People liked the questions as an inbetween
- Help menu was helpful

Summary of Negative Feedback and Responses to Feedback:

- The game could add a difficulty setting: (Very common among users)
    - This was addressed by adding a difficulty setting of Easy vs Hard in later releases of the game.

- Numbers in the tiles were hard to read:
    - This was addressed by redesigning the tile sprites such that the numbers were larger, and hence more visible.



- The game was too difficult:
    - This was addressed by changing the power up distribution and adding a difficulty setting.
    - Power Ups:
        - Rather than only rewarding the user with one power up when they answered the question correctly, it was changed such that users can receive multiple power ups.
        - This made the game more enjoyable and put the spotlight onto power up functionality.
    - Difficulty Setting:
        - This allowed for differentiation between users' skill level, such that they can play at a difficulty that is more enjoyable and not too difficult.

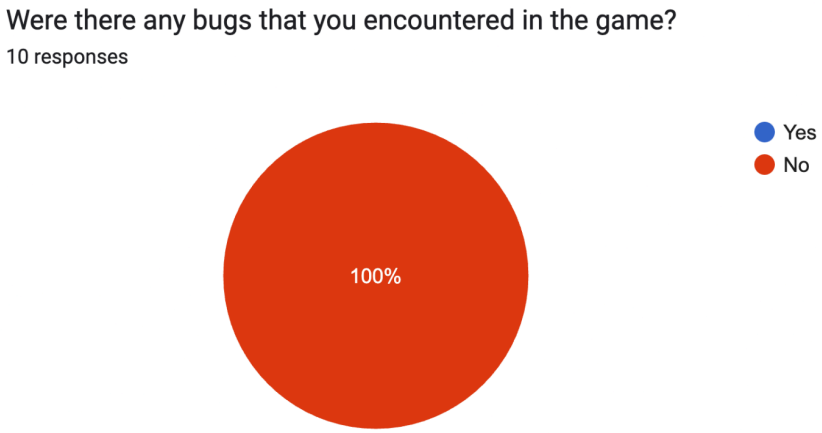
- Encryption of High Scores File:
    - This was unable to be addressed, as encryption proved to be difficult to implement in Unity.
- Questions would sometimes repeat themselves during playthroughs
    - This was fixed, where the game would mark questions previously answered by users and not ask them again in one playthrough
    - However, there was still the concern where over multiple playthroughs, the game would repeat the same questions.
        - This unfortunately couldn’t be solved as it would require developing a substantial amount of time.

- Visibility and accessibility issues (Consideration for disabilities):
    - Unfortunately, within the small timeframe given for the project, accessibility features were unable to be developed in time for the submission for the project.

Question 7:

With the 80% approval on consolidation of knowledge of high school topics suggests that the game achieved its purpose of being educational and testing people on their knowledge of various topics.

Question 8:



There were no issues with bugs in the game, fulfilling user requirements in providing an error-free product. This shows the high effectiveness of the evaluation of testing in reducing bugs in the game.

Self Evaluation:

Language Used:

The decided programming language was kept constant, using C# paired with Unity due to its intuitive UI and interactable elements in the editor. However, there are limitations with Unity with its steep learning curve when compared to other engines like GameMaker or Godot. However, as I already had experience with Unity's UI and programming language, I decided to choose Unity. Using this, less time was spent on learning an unfamiliar environment and instead used on creating a better quality solution to satisfy user needs.

System Limitations:

As Unity has the option to export to MAC OS, it would be expected that a MAC release for the game would be possible. However, this was limited due to Apple's implementation of the "Quarantine Flag", where applications downloaded off the internet without being approved by Apple's check for malware (which costs money) would be flagged, and upon opening the application it would appear that it is damaged. This heavily restricts the user base of the game, and also limits the game's ability to fulfill the users' needs of playing the game. Additionally, due to the use of particle and lighting effects in Unity's Editor, it limits the effectiveness of running on low-end Windows PCs without significant lag-time issues. While this was partially addressed by reducing particle effects, it may still have difficulties running on low-end PCs.

Changes to Initial Concept:

- Initially, there were no particle and lighting effects in the game:
    - However, after seeing Google's version of minesweeper that used confetti, I decided that it would be an enjoyable element to include in my game.
    - Additionally, the lighting effects were added to improve the visual aspect of the game, and make it visually appealing.
- Changes to the UI of the Page:
    - Initially, the page had used square buttons with text in the initial concept.
    - However, after seeing the video of another similar game, "DemonCrawl" through this video: https://www.youtube.com/watch?v=sRE6Axt8Z3Q
        - I took inspiration for the picture icons on the home page into my game, making elements/buttons stand out more and the game environment more lively.
- Addition of difficulty setting:
    - This was added due to user feedback suggestions for a difficulty setting, as well as complaints of the high difficulty level of the game
    - This allowed users' skill sets to be paired to the respective difficulty level, enhancing user enjoyment in playing the game and hence fulfilling user needs.
- Addition of Plane Cho power up:
    - Alongside the changes to the UI of the page, the number of power ups inside the game "DemonCrawl" inspired me to develop the power up of "Plane Cho"
- Changes to tile sprite design:
    - The tiles were initially difficult to see in the current state, so the numbers on the tiles were made to be bigger.

User Needs:

While the game does make use of particle effects and lighting engine in Unity which puts stress on low-end computers, for most modern computers on the market, the game is able to be run with limited lag time, reflected in the lack of negative user feedback on lag time. Furthermore, the questions designed in the game test various areas of the HSC course, satisfying the need for the game to be educational and relevant towards their study towards the HSC. Moreover, the combination of the aspects of power ups, the game "Minesweeper" and questions on various HSC topics allowed for a more appealing way to inspire students to continue their study on the HSC topics. Additionally, the added lighting and particle effects, revised UI and navigation elements allowed for ease of access and playability, enhancing user satisfaction with the game. With all of these elements combined, the users' needs have been adequately satisfied in the design of the product, where it is enjoyable to users and runs mostly smoothly without bugs.

Areas of Concern and Limitations:

- Copyright: This means the program must respect the intellectual property of others. This was done by using non-copyrighted music for the game's audio effects, and developing the sprites by hand. However, even with it being non-copyrighted, if a commercial release of the game is published, some artists may believe that their intellectual property is being infringed on. As of now, the game is non-profit, with no commercial release being planned. If a commercial release is published, then artist permission for their works must be obtained. If not, a new source must be discovered.
- Inclusivity: To be inclusive to all audiences, the game uses non-discriminatory and non-sexist language, with simple mechanics allowing a majority of audiences of all ages to engage with the game and enjoy. Unfortunately, the game currently doesn't support accessibility features for individuals with visual impairments (e.g. epilepsy, color blindness). This is due to the large amount of time to research and revise graphics/visual elements in the game to be more accessible to visually impaired audiences that couldn't be done within the limited timeframe of the project.
- Malware/Viruses: While the game available to download online doesn't contain any malware intentionally, unfortunately distributions of the copy of the game not from the main source (e.g. Transfer by USB) may have malware/viruses uploaded into it by others. However, distribution of the game outside the official game files is out of my control.
- Capability of Running on Low End: While the game can run smoothly on many of the PCs available on the market, lower end PCs may find it difficult to run the game without lag time due to the use of Unity's particle engine and lighting engine that may put stress on its operation. This was partially addressed by lowering the amount of particle effects released in the main game, and making them short-lasting.
- Not enough Questions: As there are a lack of questions in the game, it diminishes the user's need for the educational aspect of the game, and hence limits the program's ability to satisfy user needs. This is mainly due to the extensive amount of time required to develop more varieties of questions for the game. Even if more questions were added, eventually all of the questions will be asked to the user among several playthroughs of the game, and eventually limits the educational aspect of the game.

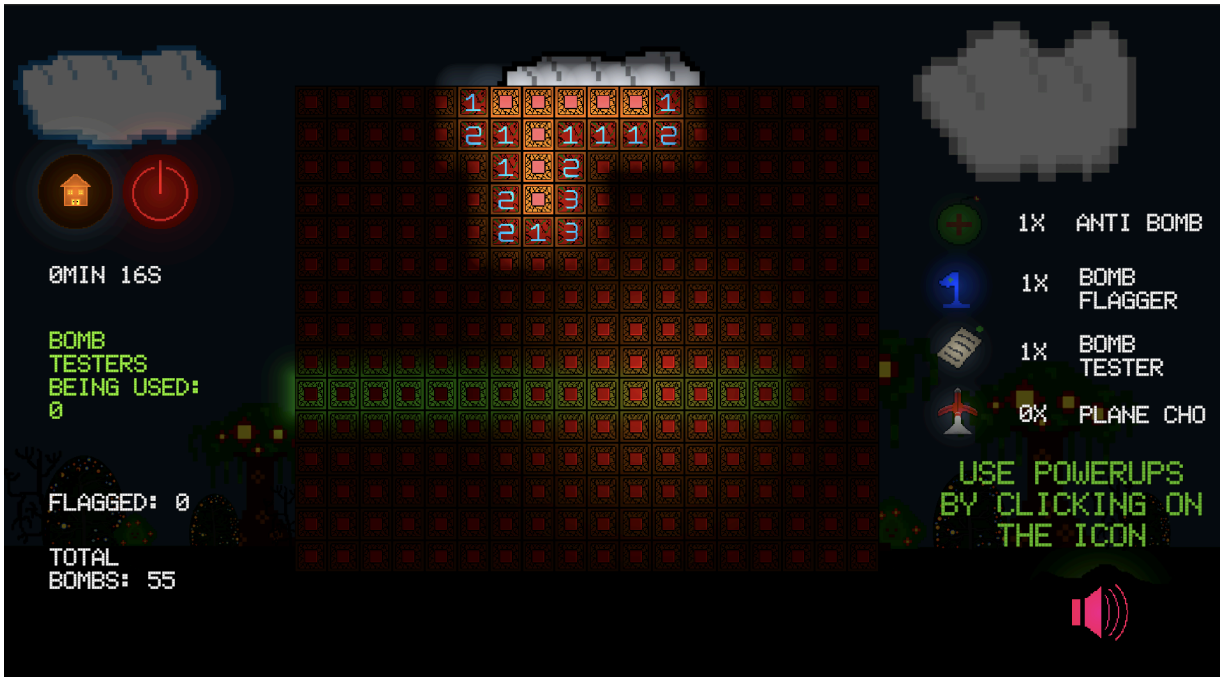
Evaluation of Testing:

Testing is an integral part of the solution, where it ensures that the product is released error free, and is necessary to satisfy user requirements of the game's stability and reliability. In this section, three different testing methods that were used will be detailed.

Debugging Output Statements:

Debugging output statements is used by adding temporary log statements to the code (in Unity, this is Debug.Log("Something")) to display the value of certain variables in areas of the program that have errors in them. This provides instant feedback on what the code is doing at areas of concern, and where specifically it is failing as well as providing information on how the data is being incorrectly manipulated.

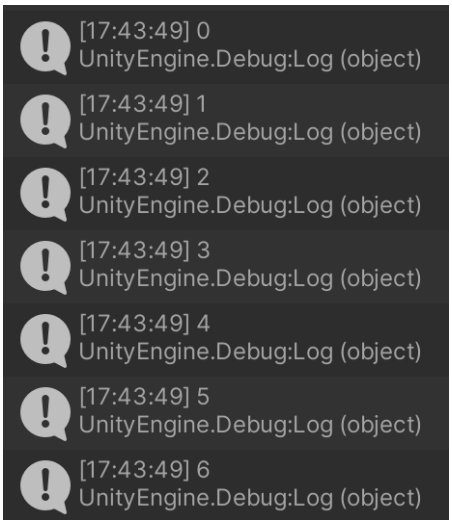
An issue that was found in the game was that when using the rotation of the power up "Plane Cho" on the Easy Mode board, it was discovered that the lighting effects didn't reach the end of the board (Specifically only for lighting up a column).

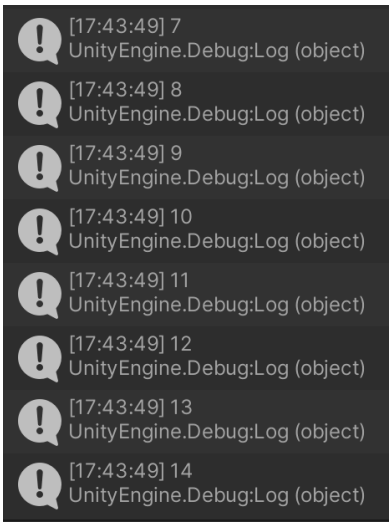


This issue was eventually resolved through the use of output statements. As seen below, this code was implemented to log which columns that the board would light up, and why some columns were excluded

from the lighting effect.

```
for (int col = 0; col < height; col++) {
    Debug.Log(col);
    // For every non-revealed tile in the column
    // Change the tile light to be green and other effects
    if (!StaticData.TileArr[useRow, col].revealed) {
        TileObjRef[useRow, col].ChangeTileLight(Color.green, 0.9f, 2, 3);
    }
}
```





As seen in the log statements, it was discovered that the lighting effect only reached up to the 15th tile (aka the height of the board). It was quickly realized that the for loop was only counting by the height rather than the width, which is evident within the code above.

After this, the for loop was changed such that the for loop changed to go to the width rather than the height (changed condition to col < width)

```
for (int col = 0; col < width; col++) {
```

Black Box Functional Testing (Testing by Expected Output):

Black Box Functional Testing looks at the functionality of the application without looking into the internal functions and processes of the code. The tests aren't looking at the internal code, instead looking at just the inputs and expected outputs of the program to see if the code is functioning as expected.

This method was implemented in the testing of the win condition of the game. As there was several different ways to win the game:

- Manually Opening/Flagging a tile
- Opening the tile by power ups

- Plane Cho
- Bomb Tester
- Bomb Flagger
- Anti Bomb

It was discovered that the end condition was activated when winning the game by manually opening/flagging a tile, however when it came to using the power ups to win, the game didn't move to the win page as expected, and left the user on the completed game board.

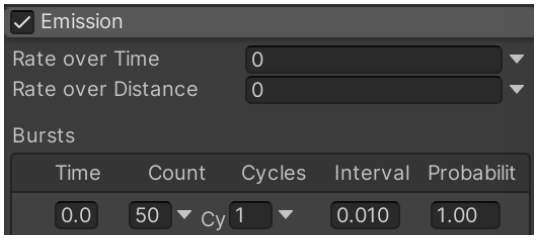
The main issue was that the function CheckWinCondition was only activated on the function of opening a tile without using a power up, with the function call being not present in the functions of using the power ups. This was quickly resolved by adding that function call. (CheckWinCondition() below)

```
public void UseBombTester(int useRow, int useCol) {
    // Tracks that user is using 1 less bomb tester
    bombTestersUsed -= 1;

    // Changes text to show that one more bomb tester is being used
    BombTestersUsedText.text = "Bomb Testers being used: " + bombTestersUsed.ToString();

    // If user is no longer using a bomb tester, set power up buttons to be active
    if (bombTestersUsed == 0) {
        for (int i = 0; i < 4; i++) {
            PowerUpButtons[i].interactable = true;
        }
    }

    // If the tile selected has a bomb, then flag it
    if (StaticData.TileArr[useRow, useCol].hasBomb) {
        FlagTile(useRow, useCol);
    } else {
        // If it doesn't have a bomb, then open it
        OpenTile(useRow, useCol);
    }

    // Checks if user has won
    CheckWinCondition();
}
```

Performance Testing:

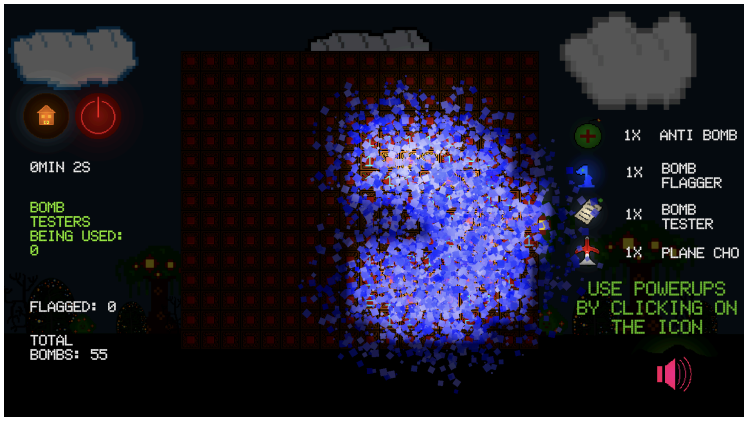
Performance testing evaluates how responsive and reliable a system is under a particular workload by measuring its usability under normal and extreme conditions with various hardware requirements.

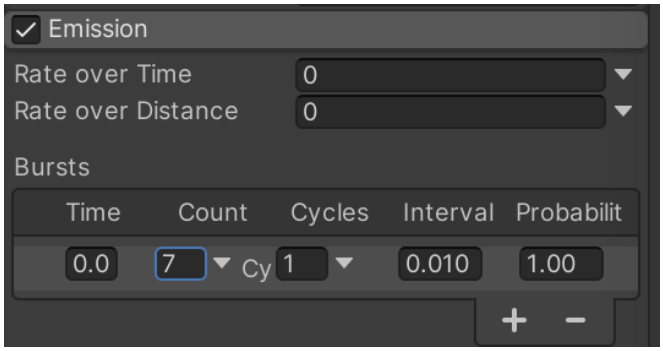
This test was done when considering the particle effects and lighting effects in Unity's engine, as this was known to be the primary source of lag time and overuse of the GPU.



Before the test was conducted, each tile opened was meant to release 50 particles as clicked. However, this brought the issue when multiple tiles were opened at once.

This unfortunately emitted a lot of particles at once due to the chain reaction of opening a tile with no bombs next to it. It created a lag-spike in the program, and on the hardware requirements of the school computer (in extreme cases) froze the app temporarily.





This issue was resolved by reducing the number of particles released on opening to 7, allowing the functionality of the program to run smoothly on the required hardware (aka the school computers)

Effectiveness of testing methods:

Through the success of these testing methods, I was able to create an application with no errors and bugs, as well as allowing the program to run smoothly on the hardware requirements (school computers), allowing it to maximize user enjoyment within their playthrough of the game. Hence, the testing methods of Performance Testing, Black Box Functional Testing and Debugging Output statements were highly effective.

Contact us at defusaldoyen@gmail.com

User Documentation

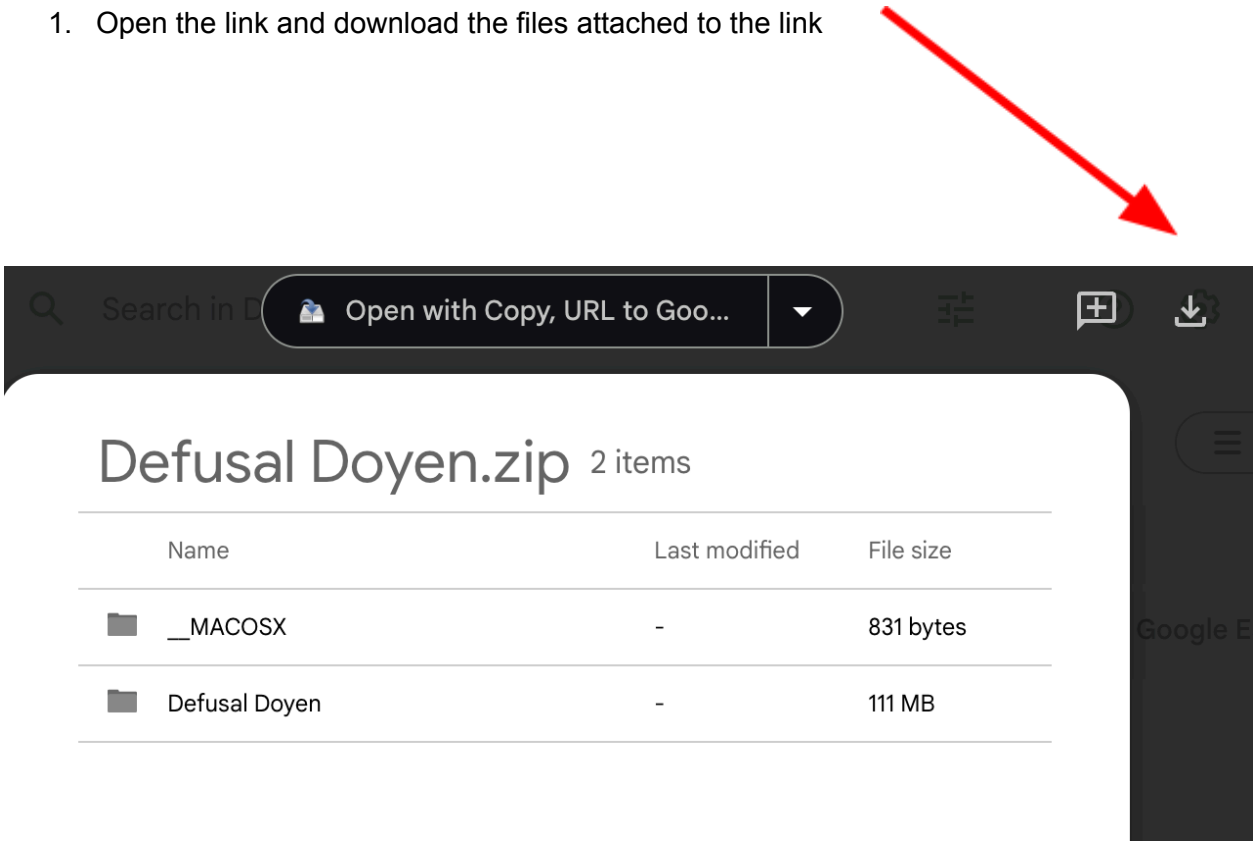
Link to the Website User Manual (With Help Manual as well):
https://sites.google.com/education.nsw.gov.au/defusaldoyenhelppage

Link to the files of Defusal Doyen:
https://drive.google.com/drive/folders/1L1-0VZT13ElJB6vriZdGb6FzlwKu2CWl?usp=drive_link

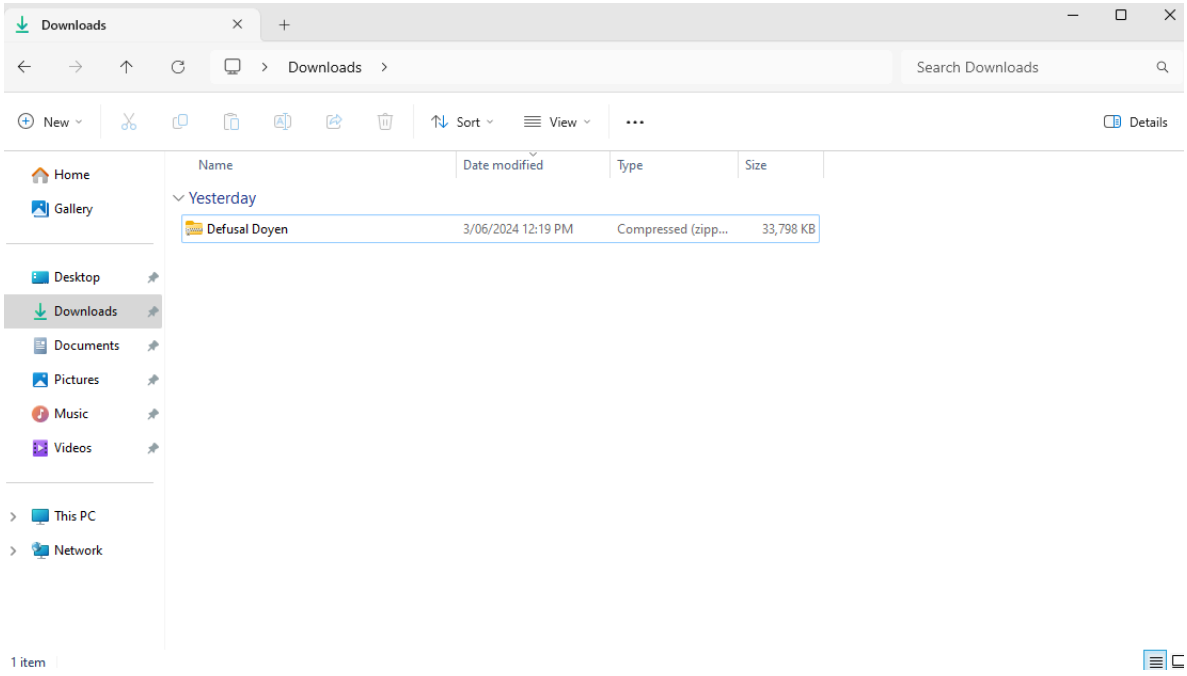
Installation Guide:

https://drive.google.com/file/d/1GHEVU4p2yMXxF3-8m8yPvEH-SeZy1Xsx/view?usp=sharing
(Link for the game files)

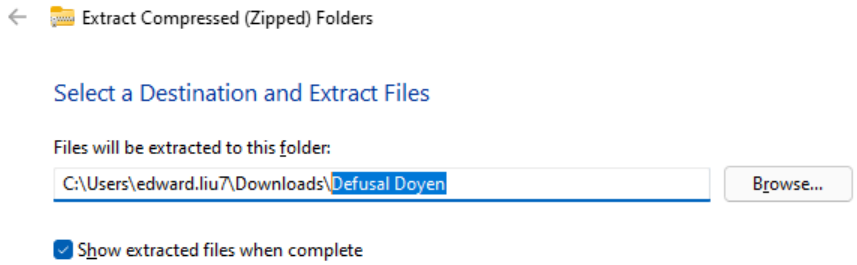
1. Open the link and download the files attached to the link



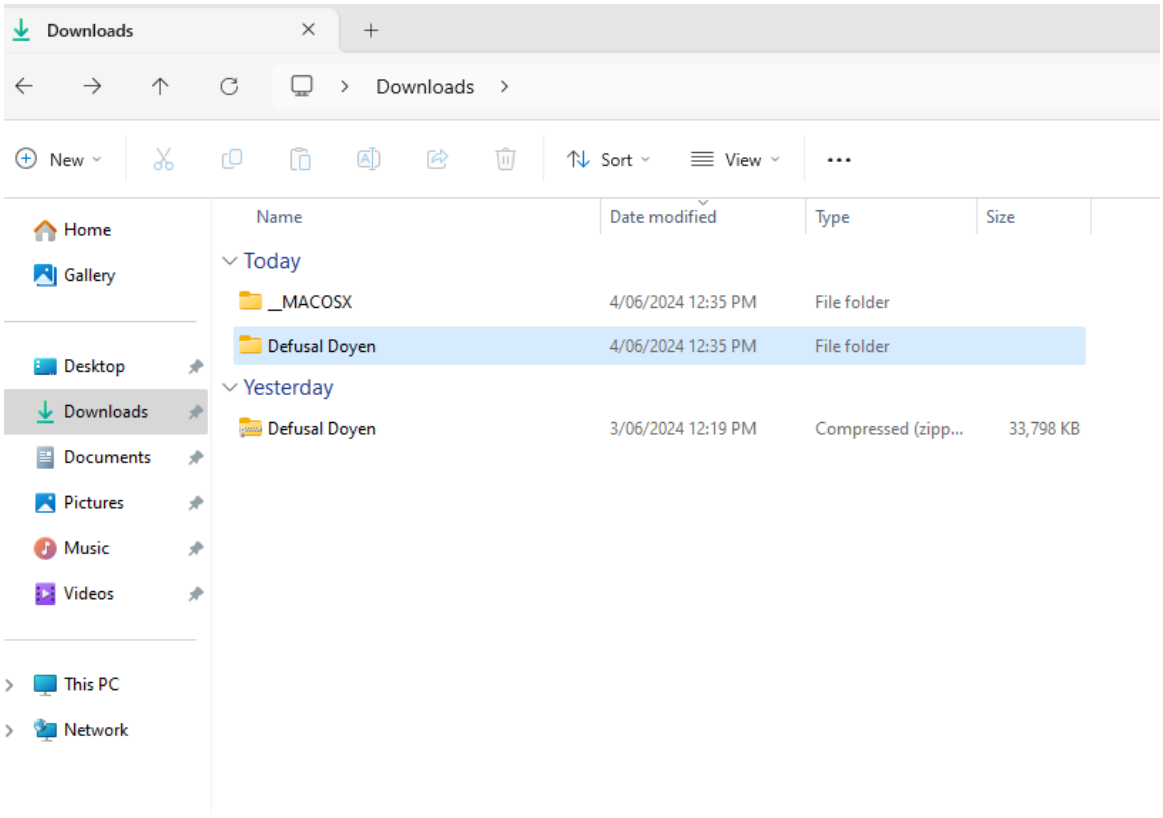
2. Open the downloads folder of your computer, and there should be a file called “Defusal Doyen” as a zip.



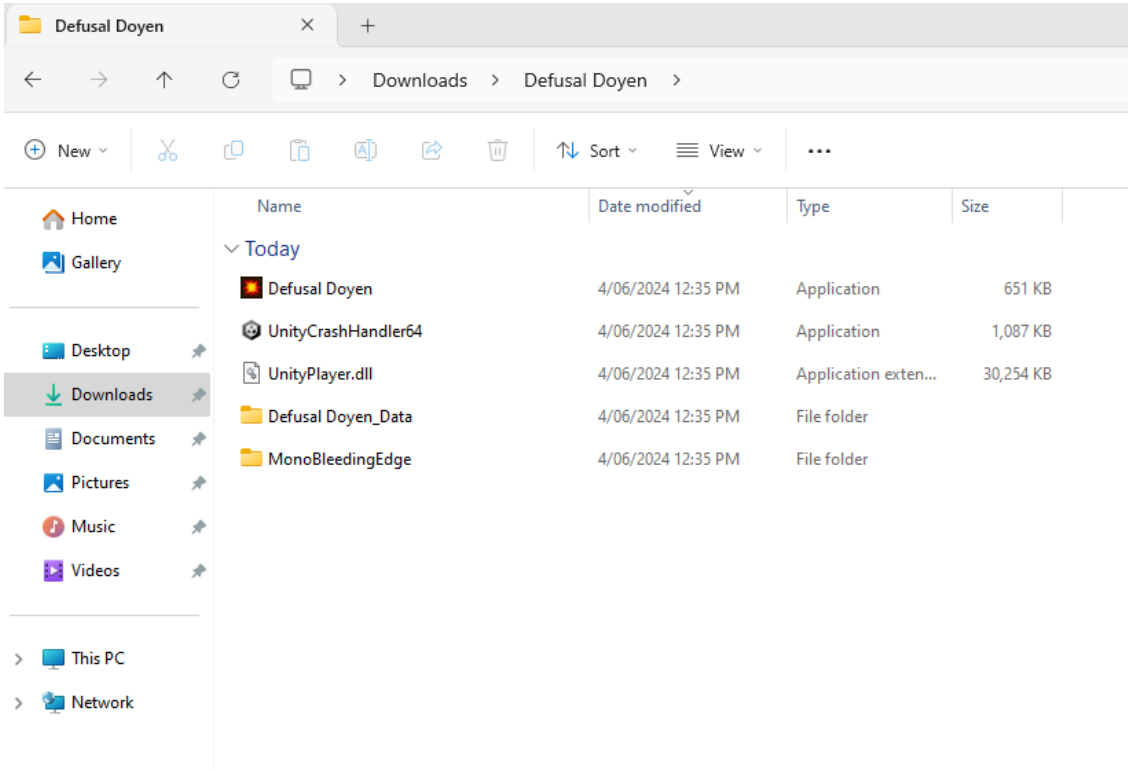
3. Left Click the file once, and then right click the file, and then click the option of “Extract All”
4. You will be prompted with this type of screen on your computer. You can press Browse to put the game in a folder that you want it to be in. If putting it in downloads, then delete the tag of “Defusal Doyen” as highlighted (Not necessary, more for convenience)



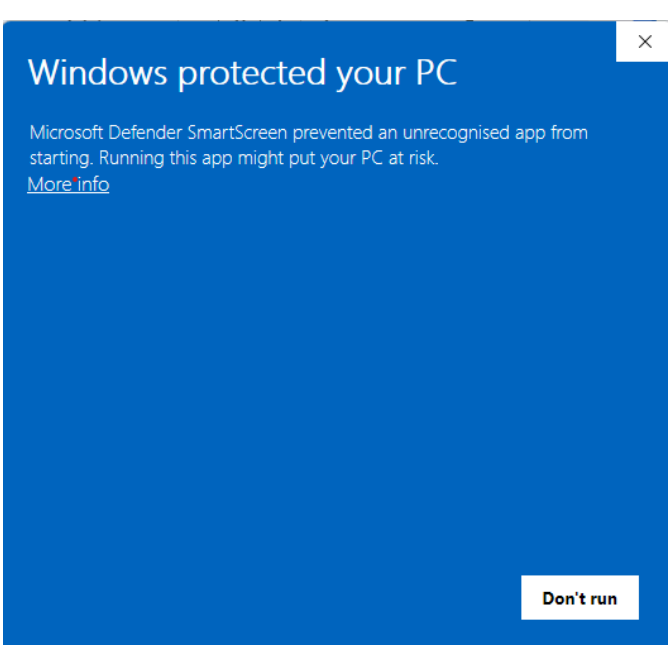
5. There will be a folder now called “Defusal Doyen” in your chosen folder. (Ignore the _MACOSX folder)

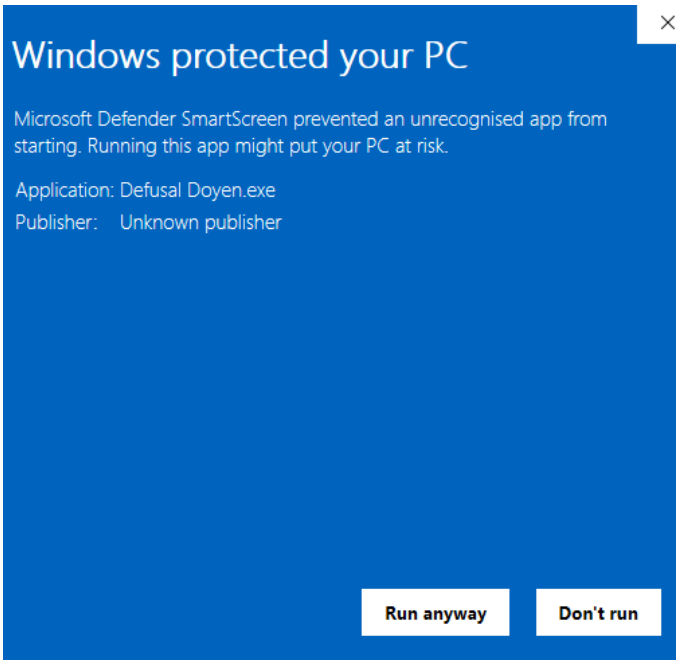


6. Open the folder, and there will be an EXE file located in it.



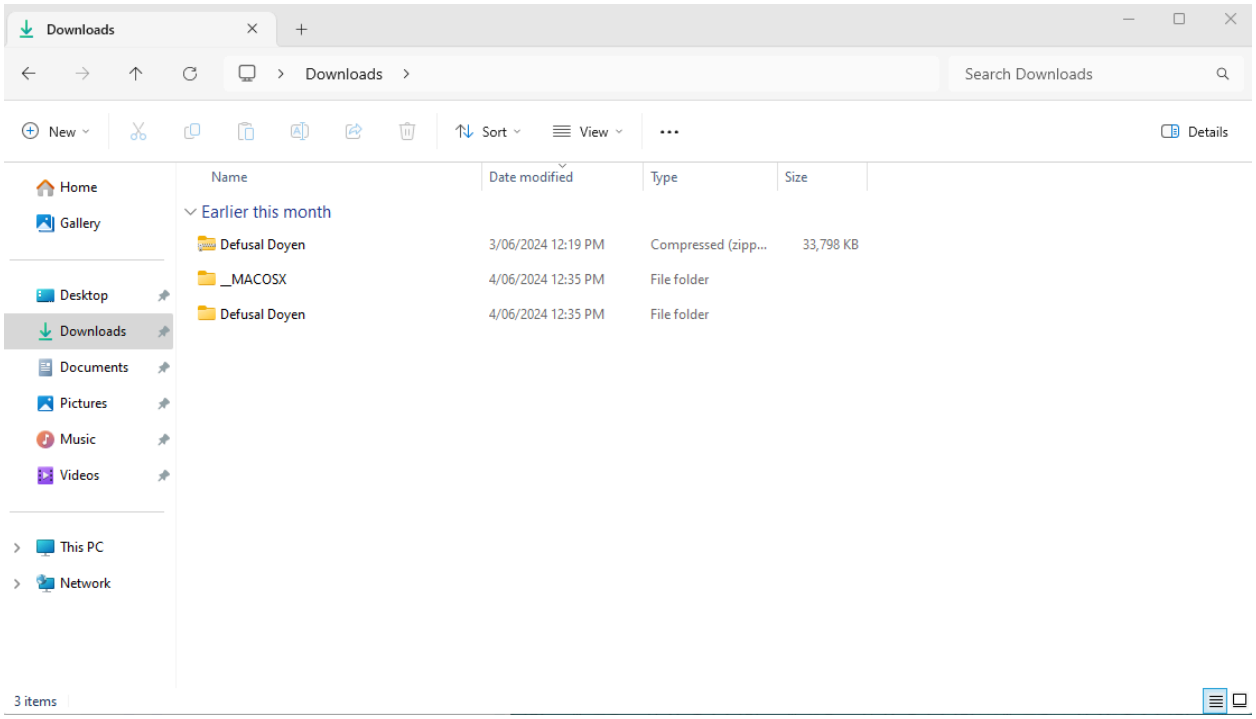
7. If you get this message: Click More Info and Click Run Anyways and enjoy the game!





Deleting the Game:

Delete the folder that the game is located in (aka 'Defusal Doyen' folder)



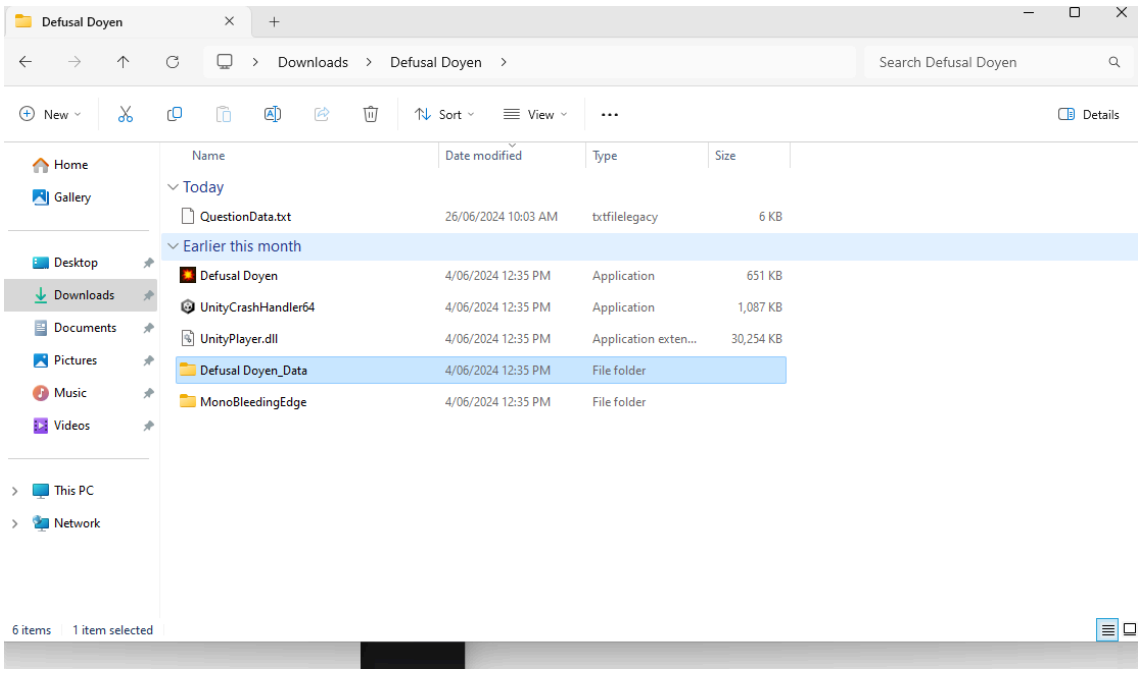
Contact us at defusaldoyen@gmail.com

Troubleshooting:

1. When getting to the Question Page, the game doesn't show any questions or responses.

Solution:
The Question Data text file is missing. You must reinstall the file QuestionData.txt via this link:
https://drive.google.com/file/d/19hlNcDvg1G3hQWB__9VhHidKvqjkwn-f/view?usp=sharing

After downloading the text file, move it into the Defusal Doyen folder onto your computer. Then move it into the Defusal Doyen_Data folder inside of it!



Contact us at defusaldoyen@gmail.com

2. The game won’t open, and shows me an error message when I click the application, what should I do?

Solution:

You have most likely deleted some of the files in the game folder, and the game cannot access these files any more. Delete the game folder and reinstall the game by the installation guide above.

3. The game is laggy, and won’t respond when I press it, what should I do?

Solution:

Try restarting the game by quitting the application via Task Manager or pressing the close button. If the problem persists, then restart your computer and see if the game runs better.

Frequently Asked Questions (FAQs):

Q: Can this game be run in windowed mode?

A: No, due to limitations with Unity's aspect ratio fitter for this game, the game is a fully full screen game.

Q: Are my highscores saved across computers?

A: No, the highscores files are stored locally on your computer, and aren't transferred to other computers.

Q: Is the game free to play?

A: Yes, as a part of NESA's STEAM project, the game is free to play.

Q: Can the game run on MAC?

A: No, as there is an introduced measure of a quarantine flag that stops applications from running unless the developer has a certification from Apple (which costs money).

Q: I deleted some files in the game, and the game won't work anymore.

A: Redownload the file from the link above and follow the install guide to reinstall the game.

Q: What's the age rating for the game?

A: The age rating of the game is 15+ as the audience would have sufficient knowledge to answer the questions prompted in the game.

Q: Where can I contact for support on issues with the game?

A: Contact the email address defusaldoyen@gmail.com for further support on the game.