EMOJI MATCH

By Ethan Chung



Documentation – Project Management Report

TAble of  
Contents

Section A Project management Report 1

Problem DEFINITION 1

Ideas and target Audience 2

Screen Design Sketches 3

Gantt Chart – Development Progress 6

Section B Implementing the Solution 11

Software Approach chosen 11

Solution to the problem 14

Intrinsic Documentation 15

# Section A Project management Report

## Problem DEFINITION

The purpose is to develop and create a project using Visual Studio a game of boggle Yahtzee or memory puzzle.

Defining the problem involves identifying the **requirements** of the final solution

These requirements are outlined in the task notification, which were:

* It must be a game of Boggle, Yahtzee or Memory Puzzle
* It must use Visual Studio
* It must have calculations, case statements,
* It must have random number generation
* It must have a use of timer and have good graphic design

By understanding the problem, we have defined the problem by listing its requirements. The final version of my Memory Game has satisfied the problem by including all the requirements.

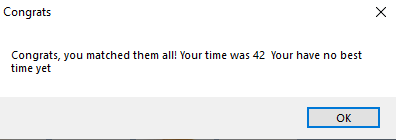
## Ideas and target Audience

There were several original ideas I had implemented in my memory game. Some examples are shown below

Original Memory Game

The Match the Emoji Memory game is entirely my creation. The concept is simple. The user gets to pick two cards and they are revealed to the user. The user can keep on picking 2 cards until they match all the cards. In easy mode, there are 20 cards, and the user needs to match 10 pairs. In medium mode, there are 36 cards, and the user needs to match 18 pairs. In hard mode, there are 56 cards, and the user needs to match 28 pairs. Every pair of cards revealed would be counted as a guess and displayed on the right-hand side. The goal is to match the cards as fast as possible.

Congratulations Message box

When the user finishes matching the cards, they would be greeted with a unique message box that tells the user their score. This also tells the user if they have beaten their high score if played more than once. This also applies if high score is not beaten and displays a message box saying the score and high score. In a very rare circumstance when the score is the same as the high score, there would also be a message box about how lucky it is.

Target Audience

* For this Memory Game, there is not really a targeted group of people that I would like playing this game, as this game could be suitable for people in all ages.
* The user interface was designed so even people that lack computing knowledge would be able to play the game

## Screen Design Sketches

Below are sketches of the windows forms both before I coded the game and after I coded the game. Note that I did not think I needed a starting form, but later decided it would be beneficial due to the game modes I would be implementing. Both are annotated to explain their functions and to note the differences before and after.

**Memory Game** – Initial Sketch (before coding)

**Matching Game**

pause

**Time taken:**

**Score:**

**Start button**

**Stop button**

The score and the time taken information about the current game. This remained there for the final version.

The start and quit button were originally planned to go at the bottom

The pause button was placed at the top originally

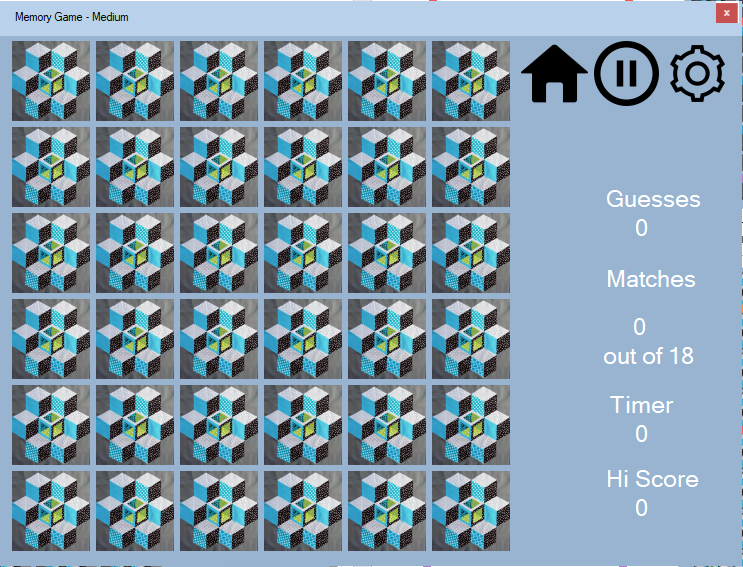


**Memory Game** – Final Version of easy mode

Added more navigation buttons, home and settings with pause button

There was no need for a start button as the starting form would start it already. The stop button became a cross seen on this form

The score and the time taken information about the current game. I added the number of matches and hi score time as well.



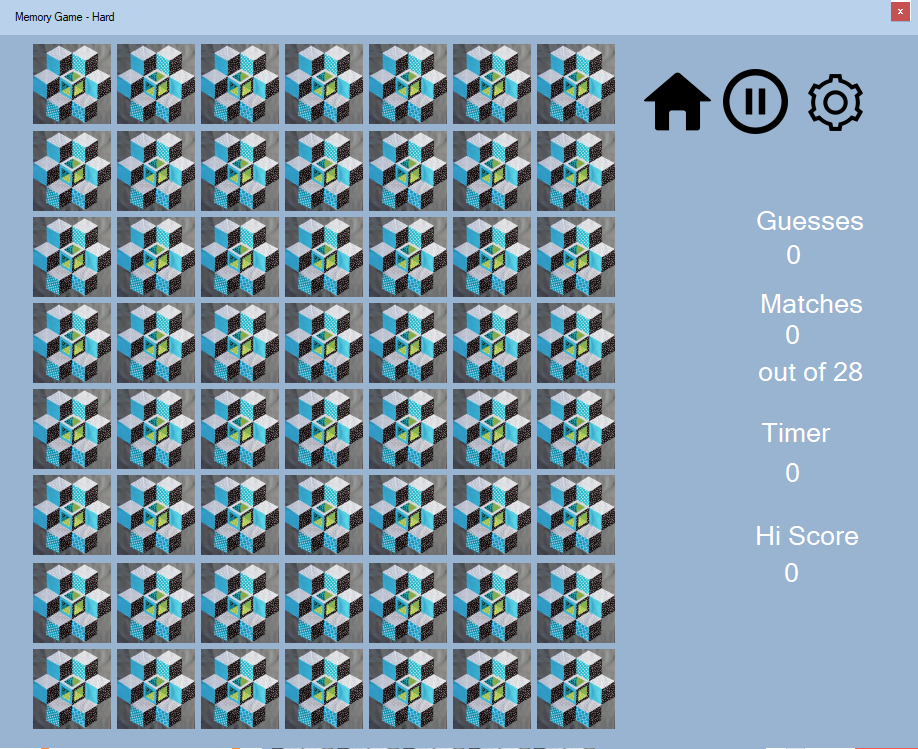
**Memory Game** – Final Version of medium mode

Similarly, the medium form has a consistent layout of game info like the easy form

Similarly, to the easy form, the cross is on the top right, and there is no start button

The nav buttons are in the same location as the easy form

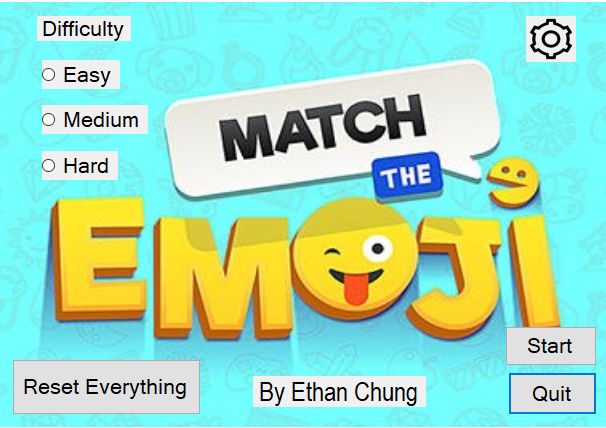
**Memory Game** – Final Version of Hard mode



Similarly, the Hard form has a consistent layout of game info like the easy form

Similarly, to the easy form, the cross is on the top right, and there is no start button

The nav buttons are in the same location as the easy form



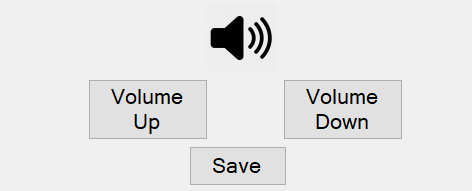
**Memory Game** – Final Version of starting screen

The difficulty was put in the starting screen to choose which form to launch.

The start and quit button locations were taken from original sketch of memory game, which is at the bottom right scorner, as seen in this form.

The reset button was put to reset all player data hi scores

The Settings button displays the volume adjustments for the game



**Memory Game** – Final Version of user control

The save button closes the control and the user continues to play the memory game

The volume Button is there so the user can mute their volume

The Volume up and down buttons are unique and is for the user to adjust the volume to their needs

## Gantt Chart – Development Progress

The tasks I needed to do include:

* Defining and understanding the problem
* Planning
* Designing the memory game
* Implementing (Coding)
* Testing the Solution (Debugging)
* Refinement
* Documentation creation

Below is the outline of the week-by-week progress of the creation of my memory game. The Gantt chart is at the end of the outline.

Week 1-2

**Planned tasks**

* Complete the defining and understanding the problem process and document it

**What was actually done in the week**

* Completed the defining and understanding the problem process and documented it
* Completed the planning of the features and planned the brief screen design

In this week, I successfully finished the Defining and Understanding the Problem stage as planned. I determined the requirements of the system and did some research on how much I could include in the software. I gathered a list of requirements and inputted into a text file for future reference. I spent less time than planned on Planning the features.

Week 2-3

**Planned Tasks**

* Complete the design of the memory game
* Complete the planning of the features and a brief plan of the screen design

**What was actually done**

* Completed first design of the memory game, but yet to be improved

This week was purely spent on creating an easy and simple screen design. I also created the 20 picture boxes yet to be coded in my first design, as well as all the visuals such as timer value, score and start/stop button. Due to other assignments and the fact that the design is complicated, I did not end up completing the design this week.

**Week 3-4**

**Planned tasks**

* Complete the second design of the memory puzzle
* Begin the basic coding for the game, start/stop buttons and the coding of arrays (implementing)
* Test the memory game by debugging

**What was actually done**

* Completed the second design of the memory puzzle
* Begun the basic coding for the game, start/stop buttons and the coding of arrays
* The testing was also begun as to test if the memory puzzle works and if it produces the required outcome

This week was spent on making a better design for my memory game that better suited the game. I also spent time starting the coding of the memory game, but just the simple things such as arrays and start/stop buttons. Testing (Debugging) was carried out this week, to test if the code works and if it produces the intended result.

**Week 4-5**

**Planned Tasks**

* Complete the design for the starting screen for the memory puzzle.
* Code the Starting screen buttons (implementing)
* Testing (Debugging) the solution

**What was actually done**

* I completed the design for the starting screen for the memory puzzle.
* I coded the starting screen buttons
* I created the user interface screen for the settings
* I also tested the solution to see if it meets the expected outcome

I decided to create the starting screen due to the fact that there would be different game modes: easy, medium and hard, and for the user to adjust the volume. I finished this start screen this week as planned, as well as coding the buttons. The user interface screen was created in advance, since I had more time to work on the assignment this week.

**Week 5-6**

**Planned Tasks**

* Complete and improve the code for the starting form and the memory puzzle easy form (refining)
* Create and code the medium level form for the memory game (implementing)

**What was actually done**

* I completed and improved the code for the starting screen for the memory
* Created the medium level form for the memory game and coded it

I completed and improved the code for the starting screen for the memory game as planned, and completed the medium form for the memory game, since it is really similar to the easy form. The coding was not yet finished though, so I planned to complete it next week.

**Week 6-7**

**Planned Tasks**

* Complete testing the solution
* Complete the coding for the medium level (implementing)
* Start coding the Hard level (implementing)
* Start documentation
* Refine the code for all the forms

**What was actually done**

* I completed the coding for the medium level
* I started to code for the hard level
* I started to create my documentation for the project
* I started to refine the code for all the forms
* I also tested the solution to see if the code created meets the outcome expected

I completed all my planned tasks as planned. I completed the code for the medium level and started to code the hard level. I also started the documentation of the project, as it is a crucial part of the project. I also went through each form to see what refinements I could make. This included making all of them to have a consistent layout and user interface.

**Week 7-8**

**Planned Tasks**

* Refining code for all forms
* Complete Hard level coding (implementing)
* Complete documentation
* Testing the solution

**What was actually done**

* I refined the code for all the forms
* I completed the code for the hard level of the memory game
* I finished the documentation
* I also tested the memory game to see if the requirements are met

I completed the code for the hard level of the memory game and finished the documentation as planned. I also refined the code for all the forms in the memory game, by testing the game and making improvements. I also asked users to try to play the game and asked for their feedback to improve my game. The feedback was taken in and the memory game was refined even more, producing the end result.

# Section B Implementing the Solution

## Software Approach chosen

My aim of this project was to create an engaging and fun experience when playing the memory game, while being easy to use.

The software approach I used for the development of my memory game is more like the prototyping approach, as I used multiple properties that is similar to the prototyping approach, such as:

A model or mock-up version of a software system that enables evaluation of features in an operational environment.

* My memory game had many versions of the system that I tested in an operational environment
* Many versions after testing, raised issues which lead to further development of the memory game

This approach is less formal to the structured approach.

* My memory game had no clear steps except for the software development cycle.
* It was mainly based on improving the current versions of the memory game

Once complete, the prototype is used as a tool to redefine the problem, making improvements.

* The memory game was used as a tool to further develop future versions
* It was less to do with redefining the problem, unlike the prototyping approach

Implement operational solution

* The solution was tested by users which was similar to distributing it to user’s machines

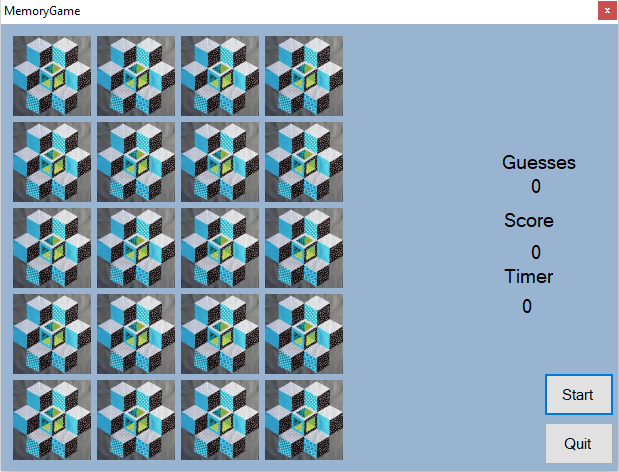
The first prototype is called “Initial Prototype” the with every new version of the model is called “Second iteration” and so on

My project had different versions, seen below:

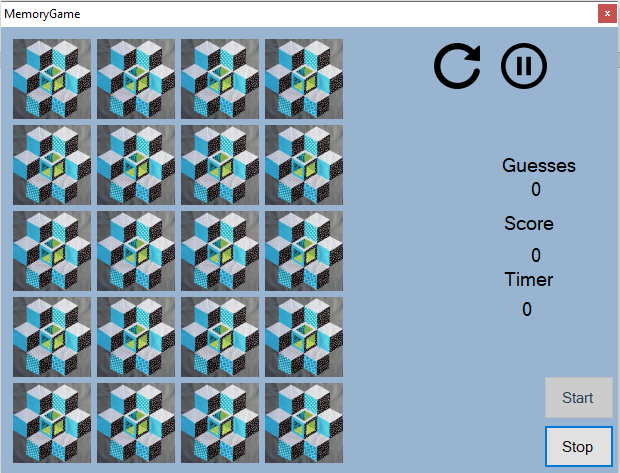
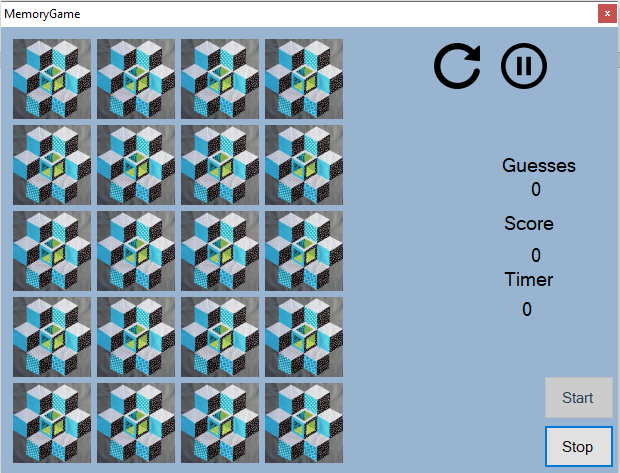
Initial prototype (I made this off visual basic 2019 and wasn’t compatible with other versions such as 2010, so I remade the design)



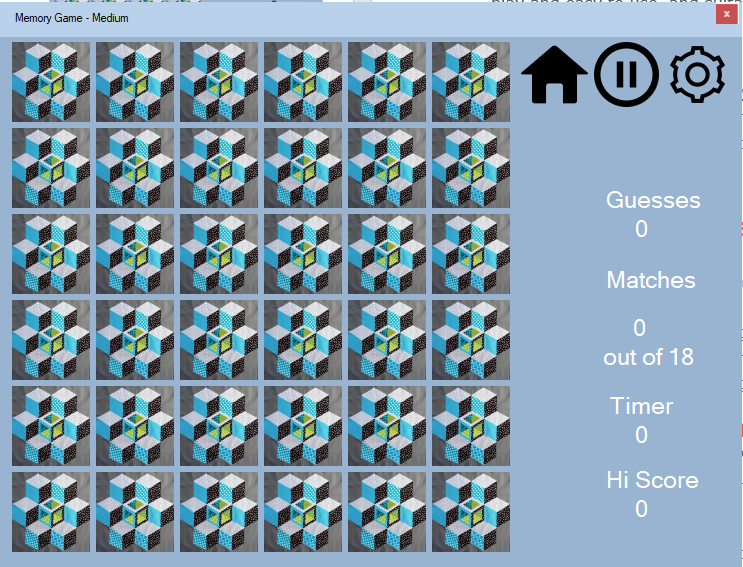
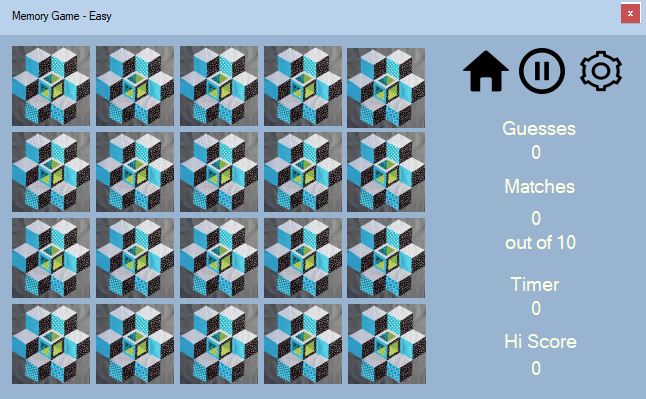
Second iteration (I then continued with this design further)



Third iteration (the starting screen was not related with the game, so I redesigned it)

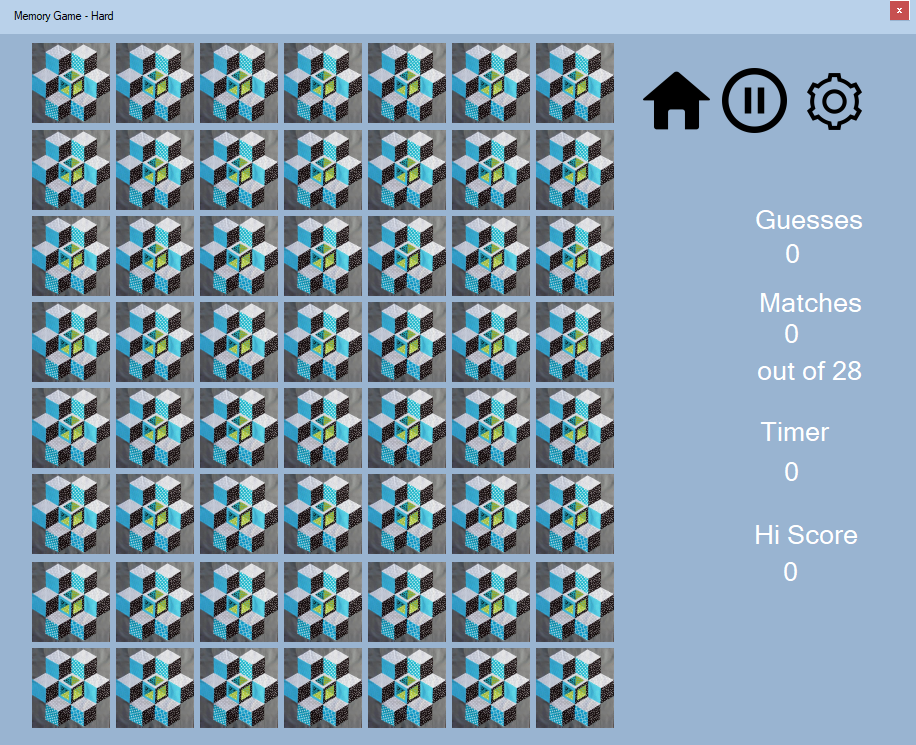
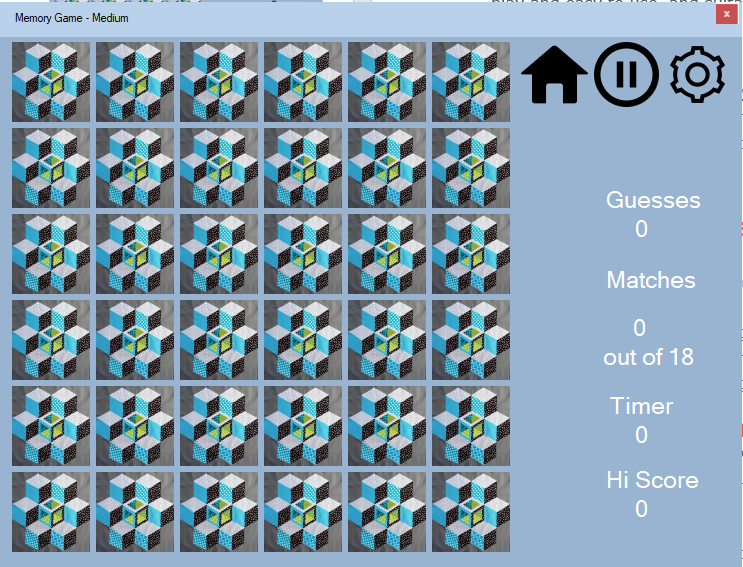
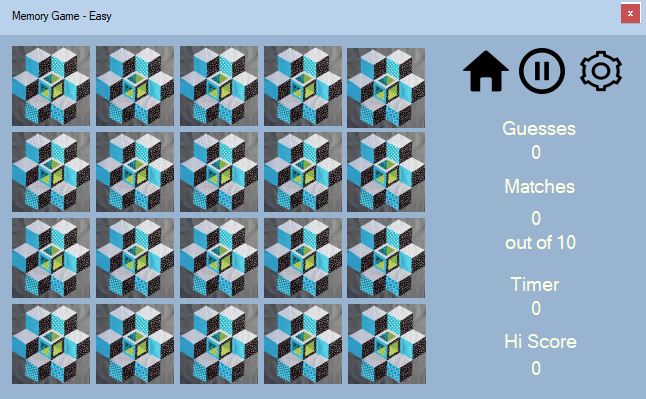






Fourth iteration (added Medium difficulty)

Fifth iteration (Added hard difficulty)



## Solution to the problem

My goal for this software project was to create something fun to play and easy to use, and suitable for people of all ages. I wanted something that I created that I was proud of. My project revolves around a few main points:

* User interface that is easy to use and consistent
* Gives a sense of achievement
* Multiple game modes to choose from

User interface is easy to use and consistent

Developing a user interface that users find enjoyable, and easy to use is important, as this is what leads them to play the game. If the instructions are not clear enough, the user would not know what to do. My memory game has clear large buttons that help the user navigate through my memory game.

Gives a sense of achievement

My memory game gives a sense of achievement to users, as the game score goes up when the user matches a pair. This encourages the user to finish matching the remaining cards, and to try other harder game modes. When the user finishes matching the cards, the high score would be given to the user and stored. This would be the user’s goal to try to beat, and the sense of achievement would be felt again when the user beats the score.

Game modes

My Memory game has 3 different game modes to choose from: easy, medium and hard. As I have explained before, in easy mode, there are 20 cards, and the user needs to match 10 pairs. In medium mode, there are 36 cards, and the user needs to match 18 pairs. In hard mode, there are 56 cards, and the user needs to match 28 pairs. The goal is to match the cards as fast as possible.

Problems encountered

I had come across many problems that arise from the game including:

* Project does not work when run with visual basic 2010

This was due to the project being made in a later version of visual basic and is not compatible with the older versions

* Sound file does not play when in a different user

This was due to the fact that I used the file path of the sound file that uses the user’s name. This was solved using the resources folder in the memory game path, which is the same for every user

* Application does not run

This was due to the syntax error I would have made in the code, and can be quickly identified by double clicking the error.

## Intrinsic Documentation

Intrinsic documentation is important, as it allows for easier interpretation of the code. This makes it easier to correct the code when something is not working as expected. In my memory game, I had intrinsic documentation in all my forms. This includes:

Meaningful variable names

On every form had been named to represent what it is used for. This assists the people who read the code to identify what it is.

For example:

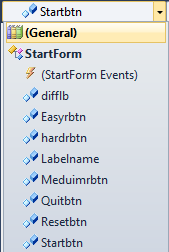
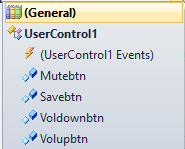
Pausebtn

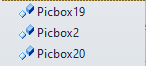
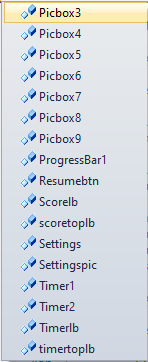
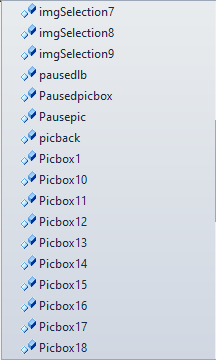
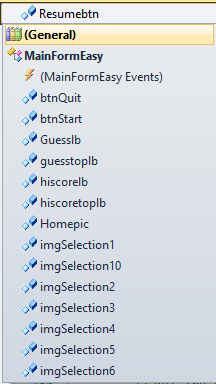
This name references that it is the button that pauses the game when pressed by the user. It is a meaningful name, as it represents the button pressed when user wants to pause.

StartForm

The name of this form implies that the form is displayed when the user starts the memory game.

Below are forms and their controls in them. All the controls are meaningful variable names.

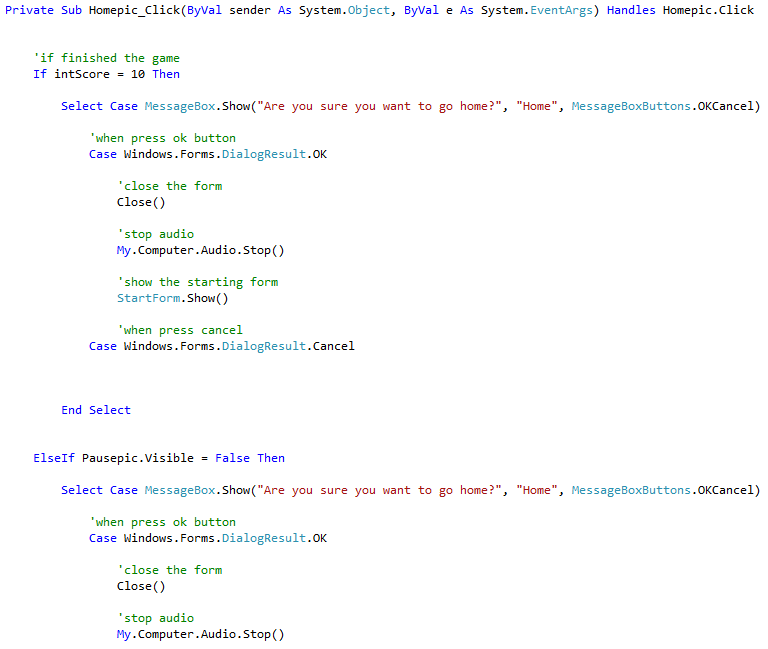
StartForm MainFormEasy, MainFormMeduim, MainFormHard



UserControl1

Use of spaces between each different piece of code/Indentations

In my memory game, there are spaces between each piece of code that does something different to the one before or after. It makes the code better to read and understand. For example, below is a segment of code:



This code is spread out and has indentations. This is important, as there are embedded case statements in the if statement. If a programmer were to review this code, they could instantly know where the case and if statements start through the spacing and indentations.

Use of comments

Referring again to the above code, there are clear comments saying which lines execute what things. These aid the creation of the overall game, as if there are problems or the code doesn’t work, the code can be easily referred back to. Changes can be easily made as well, more efficiently and less tedious in some cases.