

Memory Game Project for UTM CSCI 352

Tony Anderson and Vrushank Mali

Abstract

This project is going to be a Memory Game that keeps score of how many matches the player got. The target for this game is really anyone that likes memory games or that likes to test their memory, it might even be used to study with. We have come up with the general idea of what its going to look like.

1. Introduction

With this project, we are going to try to make a fun but simple memory game and by the end have something that could maybe be turned into an app later on. If people play this game, we hope they just have fun and not compare it to one of those bad ad infested games on the app store. This is a first step into maybe creating games for a living or just to get a better understanding on how 2D video games are made.

1.1. Background

We decided to do this type of project because we play video games all the time and want to make our own and learn how complex or easy it is. We chose a memory game because puzzle games are more like disguised mental challenges that keep your brain alert and your memory active.

1.2. Impacts

The only impact that this project would really have is on health. Computer Eye Strain is a real and sometimes bad thing when looking at a computer screen for long periods of time. We highly recommend if you feel your eyes getting tired or worse start getting blurry even when you look away from the screen, take a break from your work, game, or whatever you are doing. Don't ruin your eye sight just for a silly game.

1.3. Challenges

The main challenges in our opinion will probably be getting all the interfaces to work together and interact with one another. More challenges would have to be animating the rotation of the pictures and randomly shuffling the pictures around when the game starts. We can't really think of any solutions right now for the main challenges, but we are thinking about fading in the pictures instead of animating the rotation. A fade in/out sounds easier to implement.

2. Scope

The goals for this project are as follows, keep up a good pace so we don't fall behind on coding it, have a working game that plays good and doesn't feel hacked together, have everything working 2 weeks before due date to go over bugs or problems, be happy with what we made and how it all came together

Stretch goals:

- taking away the matched pictures from the board
- the user can turn off the background music, and to have background music

We feel the project will be done when we are happy with how it all looks and plays, as well as being able to play an entire game and have no major hiccups (in a perfect world we would want it to have no hiccups at all, but this is a project for a semester of class and there is limited time to work every bug out). As long as we feel like it plays well and seems fun to play, it will be done. As far as time goes, we would like to have at least two weeks before the project is due to go over bugs and problems with code.

2.1. Requirements

These will not be all the requirements by the time the project is done, but these are some of the core requirements that we will have in the project. Seeing as this project is a game, there are going to be minimal functional and non-functional requirements.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Normal difficulty is selected	Player	Med	1
2	Pause	Player	Med	1
3	Restart	Player	Med	1

TABLE 1. MEMORY GAME USE CASE TABLE

2.1.1. Functional.

- Game lets user click on all pictures
- Game has ability to let the user restart from any point in the game
- Game has 3 levels of difficulty
- Game lets user turn off background music if wanted

2.1.2. Non-Functional.

- Game has nice menu layouts and backgrounds
- Game should load within a second
- Each screen transition should be smooth

2.2. Use Cases

These use cases are brief examples of how our game is expected to run and be used.

Use Case Number: 1

Use Case Name: Normal difficulty is selected

Description: A player in our game has selected normal difficulty. They will click on the “Normal” button. This will start the process of loading up the normal game screen.

- 1) Player starts game.
- 2) Player left-clicks on the “Normal” button.
- 3) The normal game screen is loaded up, and the pictures are shown for a set amount of time so the user can memorize them.
- 4) The player can now play the normal difficult game.

Use Case Number: 2

Use Case Name: Pause

Description: A player in our game has paused the game. They will click the “Pause” Button. This will stop the timer and show a pause menu.

- 1) Player is playing game.
- 2) Player clicks the “Pause” button.
- 3) The timer stops, and the background is faded to show the pause menu.
- 4) The player can either resume playing the game, restart the game, turn on/off music, or quit the game.

Use Case Number: 3

Use Case Name: Restart

Description: A player in our game has decided to restart their game mode. They will click on the “Restart” button in pause menu. This will restart the game mode, erasing the previous game.

- 1) Player hits restart from pause menu.
- 2) The game mode will be restarted.
- 3) The timer and score will be reset back to 0 and the pictures will all be randomized again and be shown to the player again.

2.3. Interface Mockups

These pictures are rough mockups of what we want the screens and main layout of the game to look like. These were drew on an iPad using procreate.

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

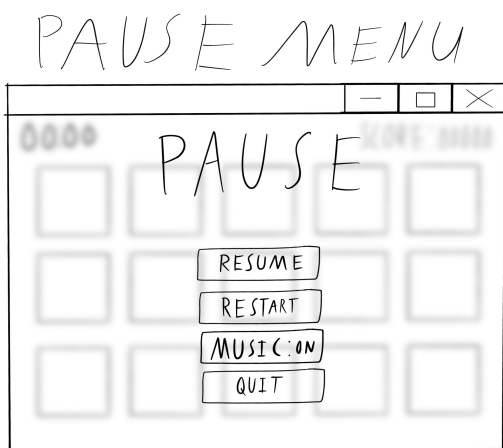
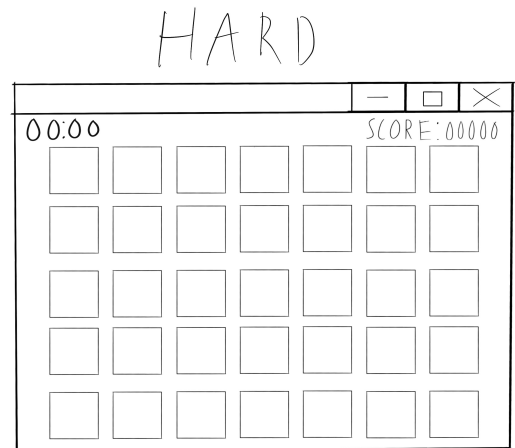
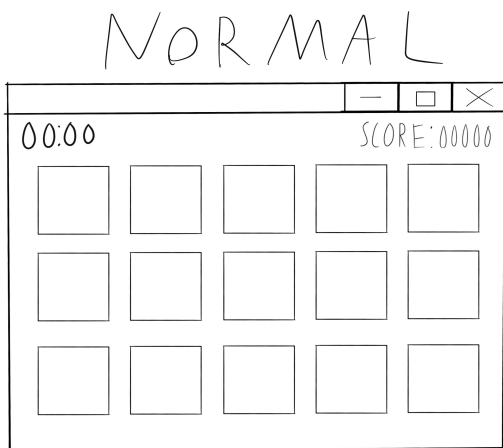
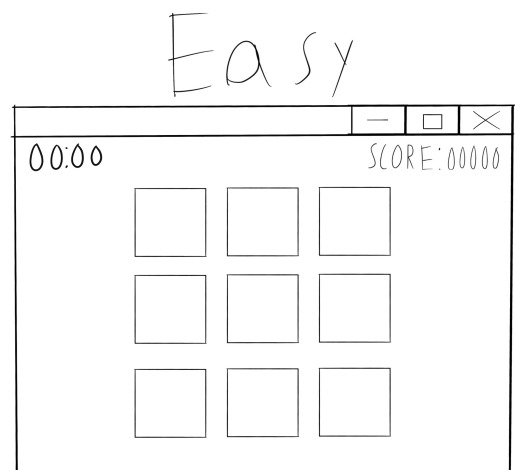
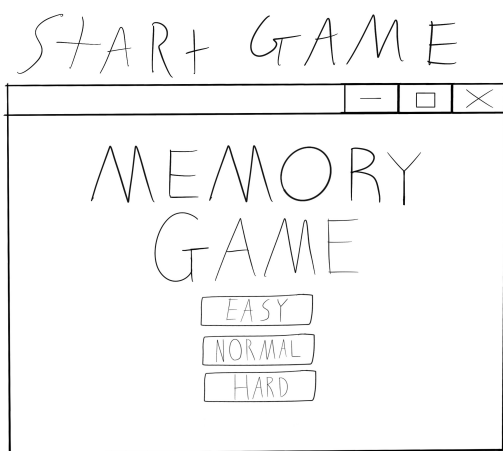


Figure 1. The main screen layouts for our memory game

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.