**Flip & Find**

Project Proposal

LAYLA BADER

**Table of Contents**

[***Flip & Find 3***](#_Toc179045871)

[**Description 3**](#_Toc179045872)

[**User Story 3**](#_Toc179045873)

[**Pseudocode 5**](#_Toc179045874)

[**Grid Generation (Shuffling and Laying Out Cards) 5**](#_Toc179045875)

[**Card Flip Logic 5**](#_Toc179045876)

[**Matching Logic 5**](#_Toc179045877)

[**Track Progress 5**](#_Toc179045878)

[**End Game Condition 6**](#_Toc179045879)

[**Time Limit 6**](#_Toc179045880)

[**Restart Game 6**](#_Toc179045881)

[**Difficulty Levels 7**](#_Toc179045882)

[**Browser Game Psuedocode 7**](#_Toc179045883)

[**Define Constants and Variables 7**](#_Toc179045884)

[**Define the app’s state variables, but don’t assign values to them 7**](#_Toc179045885)

[**Select and save (cache) elements in variables that need to be accessed multiple times 7**](#_Toc179045886)

[**Add event listeners - use delegated event listeners to listen to multiple elements 8**](#_Toc179045887)

[**Invoke the init function used to initialize all state variables 8**](#_Toc179045888)

[**Invoke the primary render function that transfers all state variables to the DOM 8**](#_Toc179045889)

[**Wait for the user to click on a card 8**](#_Toc179045890)

[**checkForMatch Function 8**](#_Toc179045891)

[**Wait for the user to click the “Play Again” button 9**](#_Toc179045892)

[**startTimer Function 9**](#_Toc179045893)

Flip & Find

Description

*Flip & Find* is a web-based memory game designed to challenge players’ concentration and recall abilities. The game presents a grid of concealed images, where each image has an identical pair hidden within the grid. The objective is for the player to flip the cards, reveal the images, and successfully match all pairs within the grid.

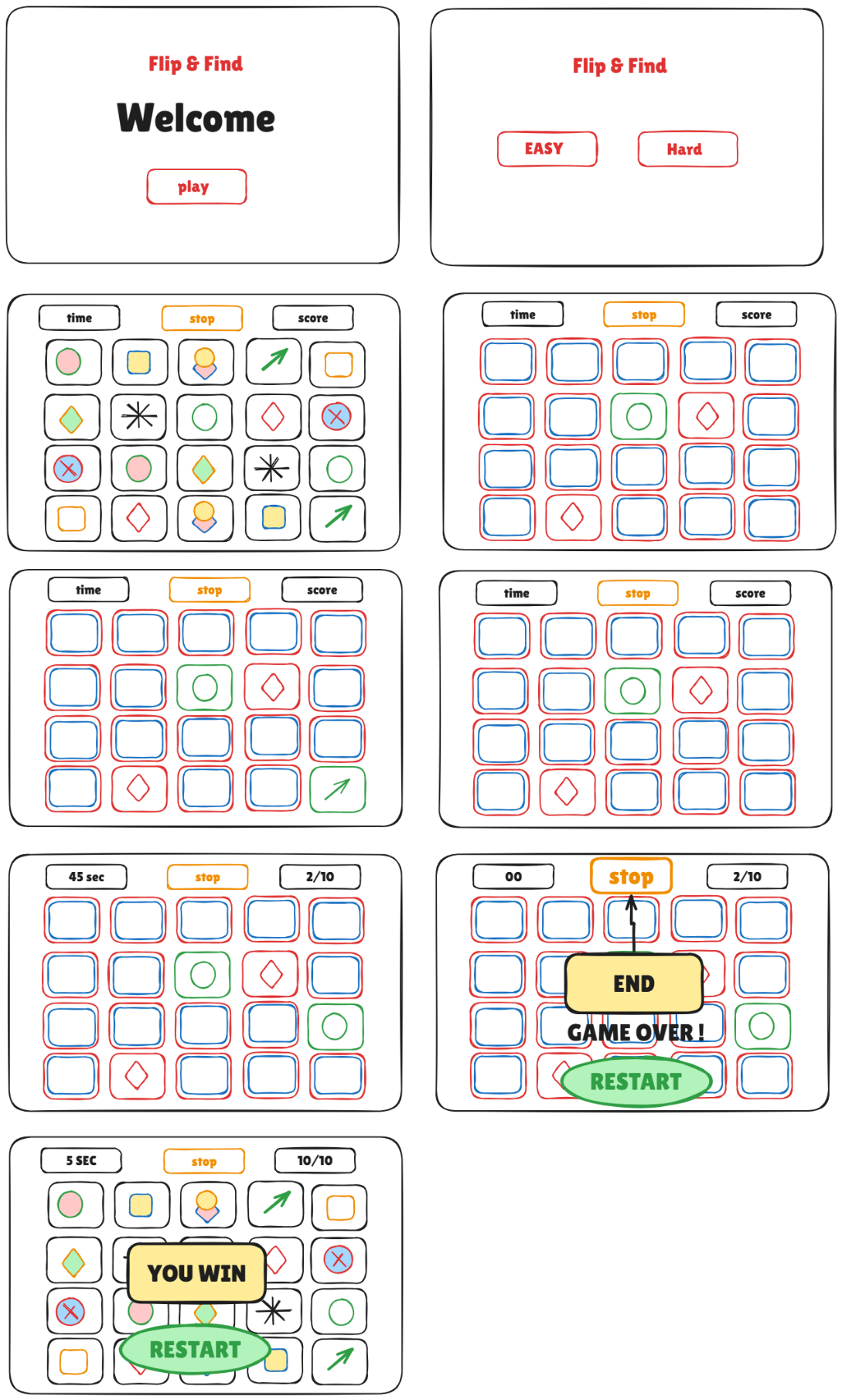
Upon starting the game, all images are hidden, and the player can select two cards at a time to reveal their content. If the two cards display the same image, they remain visible, indicating a match. If the images do not match, the cards will automatically flip back over after a short delay, and the player will need to continue searching for the matching pairs.

Players are given the option to select from multiple difficulty levels, with varying grid sizes, to adjust the complexity of the game. Additionally, the game features a countdown timer, which imposes a time limit on completing the matching task. If the player is unable to match all pairs before the timer reaches zero, the game will end, and the player will be notified of the loss. Once the game is completed or the time runs out, players are provided with the option to restart the game.

The game’s design is responsive, allowing for smooth gameplay across a variety of devices, including desktop, tablet, and mobile platforms.

User Story

1. **As a user**, I want to see a clear, welcoming landing page when I open the game so that I know I'm in the right place.
2. **As a user**, I want to see a grid of flipped cards on the game screen so that I can start selecting cards to match.
3. **As a user**, I want to be able to click on a card to flip it and reveal the image so that I can identify it.
4. **As a user**, I want the cards to flip back if I don’t find a match so that I can try again.
5. **As a user**, I want a visual indication when I successfully match two cards so that I know the pair has been found.
6. **As a user**, I want to track my score or number of attempts so that I can see how well I’m doing.
7. **As a user**, I want to see my remaining pairs or time on the screen so that I know how much longer I have to complete the game.
8. **As a user**, I want to see a timer counting down during the game so that I know how much time I have left to match all the cards.
9. **As a user**, I want the game to stop when the timer reaches zero, and I want to be notified that I’ve lost so that I know the game has ended.
10. **As a user**, I want to be unable to flip any more cards once the time is up so that it’s clear that the game is over.
11. **As a user**, I want the option to restart the game after finishing (whether I win or lose) so that I can play again.
12. **As a user**, I want to select different difficulty levels with varying grid sizes so that I can challenge myself.
13. **As a user**, I want the game to work smoothly on my mobile device so that I can play it anywhere.

Wire Frame

Pseudocode

Grid Generation (Shuffling and Laying Out Cards)

// CREATE a list of image pairs

// SHUFFLE the list to randomize the order of images

// LAY OUT the shuffled images in a grid formation

// DISPLAY the grid with all cards face down

Card Flip Logic

// WHEN the user clicks on a card:

IF the card is already revealed or matched, DO NOTHING.

OTHERWISE, flip the card to reveal the image.

IF this is the first card flipped, REMEMBER the card.

IF this is the second card flipped:

CHECK if the images on the two cards match.

IF the images match, KEEP both cards face-up.

IF the images do not match, WAIT a short time, then flip both cards back over.

Matching Logic

// IF the two flipped cards have the same image, CONSIDER them matched

// KEEP them face-up and make them non-clickable

Track Progress

// KEEP TRACK of how many pairs the player has matched

// IF all pairs are matched, the game ends, and the user WINS

End Game Condition

// CHECK after each match if all pairs have been matched

// IF all pairs are matched, DISPLAY a message indicating the player has won

Time Limit

// START a countdown timer when the game begins

// DISPLAY the remaining time on the screen

// IF the time reaches zero and not all pairs are matched:

STOP the game.

DISPLAY a message indicating that time is up and the player lost.

PREVENT the player from clicking any more cards.

Restart Game

// IF the user clicks the "Restart" button:

SHUFFLE the image pairs again.

RESET the grid with all cards face down.

RESET the timer.

ALLOW the player to start a new game.

Difficulty Levels

// ALLOW the player to select a difficulty level before starting the game

// FOR easy mode, use a small grid with fewer image pairs.

// FOR hard mode, use a larger grid with more image pairs.

// ADJUST the timer based on the difficulty level chosen.

Browser Game Psuedocode

Define Constants and Variables

DEFINE a constant for the number of rows in the grid

DEFINE a constant for the number of columns in the grid

DEFINE a constant for the total number of pairs

DEFINE a constant for the time limit

DEFINE a variable for the game timer

DEFINE a variable for the user's score

DEFINE a variable for the game state (e.g., playing, won, lost)

Define the app’s state variables, but don’t assign values to them

DEFINE a variable for the image pairs array

DEFINE a variable for the shuffled image array

DEFINE a variable for the revealed cards array

DEFINE a variable for the matched pairs count

Select and save (cache) elements in variables that need to be accessed multiple times

SELECT and cache the grid element

SELECT and cache the timer display element

SELECT and cache the score display element

SELECT and cache the play again button

Add event listeners - use delegated event listeners to listen to multiple elements

ADD an event listener to the grid element to handle card clicks

ADD an event listener to the play again button

Invoke the init function used to initialize all state variables

INITIALIZE image pairs array with image sources

SHUFFLE the image pairs array to randomize the order

INITIALIZE matched pairs count to 0

INITIALIZE revealed cards array as empty

RESET the score and game timer

RENDER the grid with all cards face down

Invoke the primary render function that transfers all state variables to the DOM

UPDATE the grid display with the current state of cards

UPDATE the timer display with the remaining time

UPDATE the score display

Wait for the user to click on a card

ON click of a card:

IF the card is already revealed or matched:

DO NOTHING

ELSE:

FLIP the card to reveal the image

ADD the card to revealed cards array

IF revealed cards count is 2:

CALL checkForMatch() function

checkForMatch Function

IF both revealed cards have the same image:

INCREMENT matched pairs count

IF all pairs are matched:

SET game state to 'won'

DISPLAY win message

ELSE:

WAIT a short time

FLIP both cards back down

CLEAR the revealed cards array

Wait for the user to click the “Play Again” button

ON click of the play again button:

CALL init() function to reset all state variables

CALL render() function to refresh the UI

startTimer Function

SET game timer to the predefined limit

WHILE timer is greater than 0:

DECREMENT timer every second

CALL render() to update the timer display

IF timer reaches 0:

SET game state to 'lost'

DISPLAY lose message

PREVENT further card clicks