IT351 Assignment 3 - Report

Sachin Prasanna - 211IT058

March 7, 2024

1 Problem Statement

Implement the proposed Alphabet Learning System using the tools and software of your choice. Justify the same. Demonstration of working model of proposed Alphabet Learning System [10 Marks] Creativity and User Experience considered and its demo: [05 Marks] Presentation and report [05 Marks]

2 Programming Languages

The programming languages used for this assignment were HTML, CSS and Javascript.

2.1 HTML (HyperText Markup Language):

- HTML served as the backbone of the project, providing the structural framework for organizing the various elements of the learning application.
- It defined the layout of the user interface, including buttons, images, audio elements, and text content.
- Through HTML, the project created a cohesive and intuitive user experience, guiding children through the alphabet learning process with ease.

2.2 CSS (Cascading Style Sheets):

- CSS enhanced the visual presentation of the learning application, making it more engaging and appealing to its target audience.
- Various styles such as colors, margins, and padding were applied to different elements of the HTML markup.
- CSS animations like modals were employed to create interactive effects, enhancing user interaction and feedback within the application.

2.3 JavaScript:

- JavaScript added interactivity and dynamic behavior to the learning application.
- It facilitated features such as drawing on the canvas, playing audio files, handling user input, and controlling the flow of the application.
- JavaScript allowed the application to respond to user actions in real-time, providing immediate feedback and guidance to children.

3 Algorithm for Recognition of Alphabet Drawn

3.1 Description

The algorithm allows users to draw alphabets on a canvas using a mouse or touch input. As the user draws, the algorithm checks for intersections between the drawn lines and predefined dots representing key points of the alphabet. These dots serve as reference points for recognizing the alphabet. Upon completion of drawing, the algorithm verifies if all the required dots have been touched. If so, it determines that the alphabet has been correctly recognized. Otherwise, it prompts the user to try again.

3.2 Pseudocode

The following pseudocode outlines the main steps of the algorithm:

Algorithm 1 Alphabet Recognition Algorithm

```
Initialize canvas, context, dotRadius, and other variables
Define dotCoordinates representing key points of the alphabet
Initialize touchedDots and drawnCoordinates arrays
Listen for mouse events on the canvas
Upon mouse down event, start drawing and record coordinates
Upon mouse move event, continue drawing and record coordinates
Upon mouse up event, stop drawing and check for intersections
if Intersections found then
    Check if all required dots have been touched
    if All dots touched then
        Display success modal and play congratulatory audio
    else
        Display modal with retry option and play wrong audio
    end if
else
    Display modal with try again option and play wrong audio
end if
```

The proposed algorithm offers an interactive and engaging approach to alphabet recognition. By combining drawing exercises with real-time feedback, users, especially young children, can learn and recognize alphabets more effectively. Further enhancements and extensions to the algorithm can be parallelly computing the and comparing the pixel values, to reduce computation time.

4 The Canvas Tag

The <canvas> tag is a powerful HTML element that allows for dynamic, scriptable rendering of graphics, including shapes, images, and text, directly within a web page. It provides a versatile platform for creating interactive visualizations, games, and other graphical applications on the web.

The <canvas> tag is an HTML5 element that acts as a container for graphics, providing a drawing surface that can be manipulated using JavaScript. It has attributes for specifying the width and height of the drawing area, and it supports various drawing methods and properties through the associated JavaScript API.

4.1 Usage in Alphabet Learning Application

In the provided context, the <canvas> tag is used as the drawing area for the alphabet learning application. It is embedded within the HTML structure of the web page and configured with a width and height of 300 pixels each. Additionally, it is styled with a background color of "peachpuff" to provide a visually appealing backdrop for drawing.

The <canvas> tag serves as the interactive canvas where users can draw alphabets using mouse or touch input. JavaScript code interacts with the <canvas> element to facilitate drawing functionalities. It utilizes the Canvas API to manipulate the drawing context, draw lines and shapes, handle user input, and provide feedback based on the drawn content.

5 Creative Features of the Implemented Application

5.1 Interactive Drawing Canvas

The application includes an interactive drawing canvas where children can practice tracing letters of the alphabet. This feature allows for hands-on learning and helps improve fine motor skills.

5.2 Audio Feedback

Audio feedback is provided throughout the application almost everywhere as the application is used by 3-5 year old kids who cannot read. The list of audio cues in the website include:

- Clear audio feedback when the kid has finished the drawing alphabet correctly or wrongly. There are 3 different audios from which the system chooses (which is randomly selected) to play when the alphabet is drawn wrongly by the child.

- The question mark is designed such that when the kid clicks on it, instructions related to how to use the application will be given to the kid.

- Naturally, the kid will be more driven towards clicking or hovering over the big image in the middle of the screen. When the kid hovers over this image, an audio cue describing the alphabet is played. For example, for the alphabet A, ”A for aeroplane” is played, so that the kid can understand and relate more to the drawing.

- Celebration sounds, such as applause or cheering along with audio feedback, are played when children successfully complete a task or activity, reinforcing positive behavior.

- Error sounds alert children when they make a mistake along with verbal encouragements to the kid are provided, encouraging them to correct their actions and learn from their errors.

5.3 Visual Feedback

Visual feedback is provided throughout the application almost everywhere to enhance user experience. The list of visual cues/feedback in the website include:

- Big Pictures and colorful logos as pictures capture the attention of young children, making the learning experience more engaging and enjoyable.

- The website is fully functional without almost no text at all. This makes it a powerful visually equipped application, which can guide its users only through pictures and visual cues on screen.

- Error modals appear when the child has traced the alphabet wrong. A modal appears for 3 seconds with a catchy emoji and strong red background, indicating that the alphabet has been traced wrongly by the child. It is displayed in Figure 1.

- Reward animations, such as a smily emoji along with a proper green background modal signifying correctness, are displayed upon successful completion of tasks, providing positive reinforcement and encouraging continued participation. Henceforth, 3 buttons are provided for the child to navigate on what to do next. It is displayed in Figure 2.

- Interactive animations, such as the hover animation on the main picture at the centre of the website.

5.4 Gamification

Gamification elements such as images and congratulatory messages are incorporated to celebrate achievements and motivate children to continue learning make the application more of a game. This encourages the child to participate more and explore the app further. These elements add excitement and fun to the learning process.

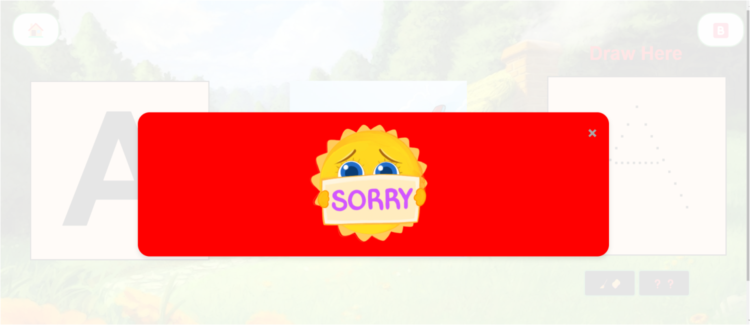

Figure 1: Error Modal

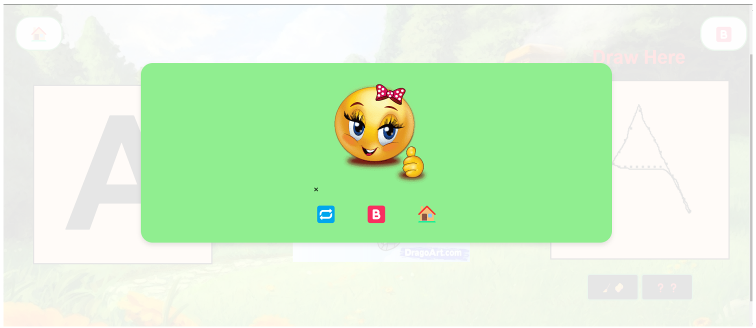

Figure 2: Correct Modal

5.5 Audio Novelty

A variety of different voices are used for different audio cues, making the application fresh and not repetitive.