Leo Durante

CMPT220

Pictionary

Abstract

This project is intended as a tool to be used by multiple users as an aid in playing the game pictionary. The users receive a prompt to draw on a surface that they have already ready and the game times their turns. The program acts a prompt generator, timer, and scorekeeper. The players are only required to give the program simple inputs and the game can last for exactly as many rounds as the player's desire.

Introduction

The purpose of this project is to aid players in the playing of the game pictionary. This program reduces the amount of peripherals required to play the game and allows for easier customization of the game. It can easily be edited to add more prompts and players. The program is also intended to broaden the appeal of the game, as the users can individually decide the difficulty of the subjects that are to be drawn. Score is automatically kept and displayed regularly. The general gameplay portion of the program loops for as many rounds as the players desire it too, and at the termination of the program displays the final scores as well as the winning player.

Detailed System Description

The program uses the user's input at each point in the game in order to tailor the experience to the individuals that are participating at that time. The beginning of the game prompts the user to input the difficulty that they desire as an integer numbers one to three. This number is stored as a value which serves multiple purposes. If the number is larger than three

it displays an error message as there are no prompts dedicated to numbers other than one two and three. When one, two , or three are entered the program displays a random prompt from a set list. There are three lists of prompts that are separated by the difficulty of the subjects ability to be drawn. The higher the number the more difficult. Higher difficulties also reward players with higher amounts of points, the points being equal to the difficulty value inputted by the users. If statements are used to hand out the prompts as well as to reward the players with points. After the prompt is given the program is instructed to wait for 30 seconds in order to give the players time to create their drawings and for the other players to guess. Once this timer terminates the program alerts the users to the end of the drawing period and asks if any of the players were successful in guessing what the prompt was. This is done with a boolean value, and the user is instructed to input a true or false. If the user inputs false the program asks the users if they wish to play another round. If the users say true to this the program restarts from the beginning. If the user's input true for if a player was able to successfully guess within the allotted time the the program asks for the player that made the drawing as well as the player that made the successful guess. Both of these players are rewarded points equal to the difficulty modifier chosen earlier. The score is then displayed and the users are asked if they wish to play for another round. If the users decline by entering false then the while statement terminates as the boolean that kept it running has been set to false. The program then displays the final score and terminates. If the players opt to play for additional rounds then the program restarts, keeping track of the scores each time and adding to the players's totals each round until the program is terminated.

Requirements

The program is geared to improve the playability, portability, and customizability of the game pictionary. The original game requires the use of cards with prompts, a physical timer and

manual score keeping. The program also allows the use of difficulty categories to broaden the appeal of the game by increasing the competitive nature. The timer, prompts, point values, and most any aspects of the game can be changed or added onto with ease with the use of the program. It is easy to organise and setup.

User Manual

The user is prompted to enter an integer, one two or three. This integer is stored as a difficulty value. The program displays a prompt and waits for the players to play the game. Once the wait period terminates the program prompts the players to input if they were successfully able to guess. If true, the player inputs the current turn's player and the guesser. It rewards these two players with points equal to the difficulty value. If false the program skips the point rewarding and moves directly to prompting the user for if they wish to play another round. If the player inputs true then the program restarts from the difficulty input and retains all score values, as well as displaying the score. If false the program displays the final scores and terminates.

Conclusion

The project currently functions as intended. It keeps scores for players, rewards based on difficulty, and its list of prompts can be easily manipulated. Its functions are clearly explained to the user as they occur in the program. It is easy to manipulate the prompts, timer, and other aspects of the program without much difficulty or technical proficiency.