Name(s)_____ Period _____ Date _____

Project Guide - Emoji Race

Overview

Building a larger piece of software like a game can quickly get complex. Starting with a plan can help you stay organized and identify issues ahead of time. A lot of the work you do here will make it much easier to keep track of what you need to do once you begin writing your actual code.

Program Goal and Design

Start by thinking about what your game actually does. How will the user interact with it? How does it communicate information to the player? What will make it fun, interesting, or relevant to the player?

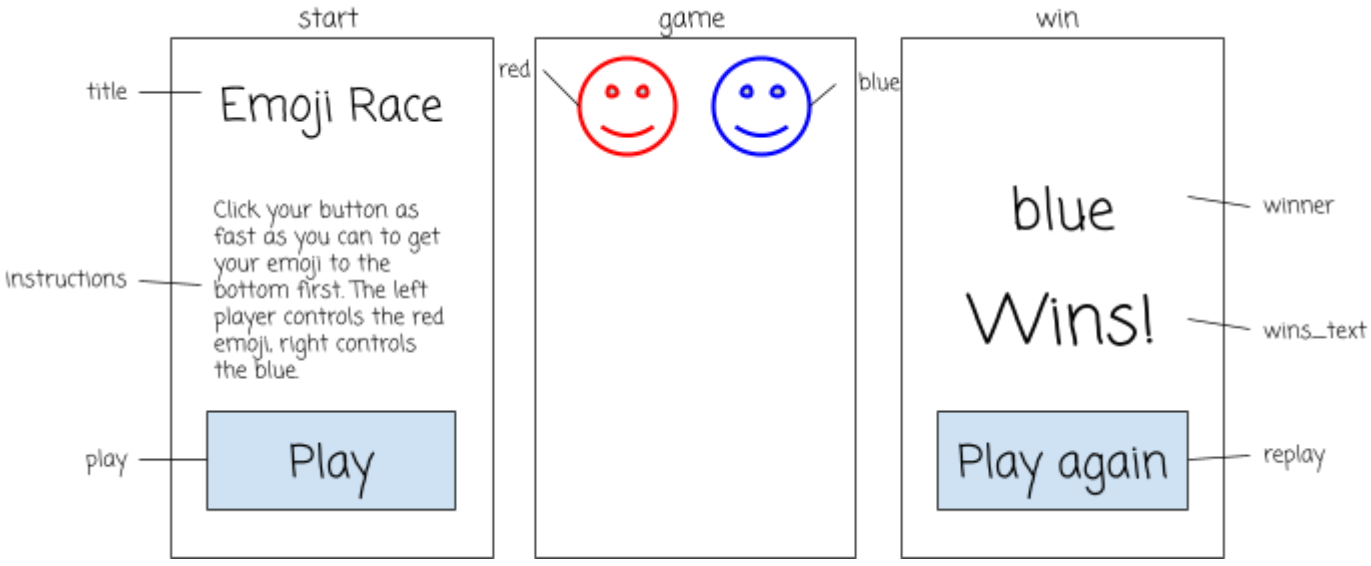
Describe Your Program

In a couple of sentences describe the program you are going to build and how it will work.

Two players are competing clicker race. Each player needs to click one of the Circuit Playground buttons as fast as possible to move their emoji to the bottom of the screen. Whoever gets to the bottom first. The buzzer plays a high note if the red player wins and a low note if the blue player wins..

Draw Your Screen(s)

Draw a quick sketch of the screen(s) you'll need. What design elements will you use? What should their IDs be?



Circuit Playground

Which components of the Circuit Playground does this program use? Make sure that you are using at least one input (eg buttons and sensors) and one output (eg LED or buzzer).

Board Component	What it is Used For
buttonL	Player one clicks buttonL to move the red player down
buttonR	Player two clicks buttonR to move the blue player down
buzzer	Plays when the game is over

Events and Functions

Using the description of your program above, figure out what events you'll need to respond to and which functions

Events

In the table below list information about all of the events that your program will use, and what will happen when they are triggered

Name or ID	Event Type (eg "click")	Description (What happens when this event occurs?)
buttonL	*press*	*Move the red emoji down the screen by 10 pixels and check to see if they have reached the bottom*
buttonR	*press*	*Move the blue emoji down the screen by 10 pixels and check to see if they have reached the bottom*
"play"	*click*	*Reset players to the top of the screen and change to the "game" screen*
"replay"	*click*	*Reset players to the top of the screen and change to the "game" screen*

Functions

Your events shouldn't have a lot of complex code. Instead, break your program up into the major steps you'll need for it to work. The different behaviors you described in your events should help you decide what these steps should be.

Function name	**Parameters** (Inputs to the function)	**How It Changes During the Program** (What's the starting value, when will it change?)
startGame()		Resets the players to the top of the screen and changes to the "game" screen
movePlayer()	player	Moves the specified player 10 pixels down the screen
checkWin()	player	Checks to see if the specified player has reached the bottom.. If so, set the text of "winner" to the player, switch to the "win" screen, and buzz

Additional Notes

Use this area to take any extra notes that you might need to complete the program. This could include any variables that you might need, hardware setup for the board, or resources that you'll need to find (like images, sounds, etc), or ideas for more features that you want to explore.

In the movePlayer() and checkWin() functions, I'm going to use a variable player_y to store the player's current y position so that I can add 10 to it and reset.

I still need to figure out how to keep the players from moving their emojis when they shouldn't (like when the "start" or "win" screens are showing).

It would be cool if the emojis started out frowning and then turned into meh and eventually smiling as they move down the screen.