Dino Jump

Introduction

My project is a copy of the popular Google Chrome dinosaur jump mini-game that uses the Arduino UNO R3 as the “brains”.

The inspiration to make this project came when I realized that my PC had no internet connection and I started playing the mini-game out of boredom.

My aim in creating the game was to replicate the addictive and fun game that we all played at least once when we had no internet. I believe that this project is a great way to enhance your skills in Arduino programming and in electronics.

General Description

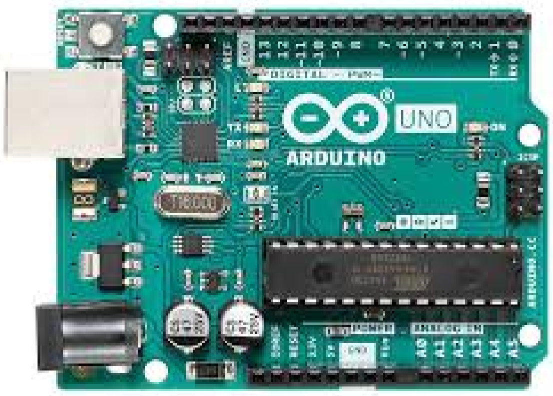
My project is a copy of the Google Chrome dinosaur jump mini-game replicated using an Arduino UNO R3 to run the program, a 16×2 LCD with I2C module to display the game, a button to control the dinosaur's jumps and a piezo buzzer to play certain sounds during the game in order to keep the player updated on certain events. The game starts the moment that the player presses the button and then, as the game progresses, the player must correctly time the jumps and dodges in order to avoid cactuses on ground and birds on air. In order to increase the difficulty, as time passes by, the game's speed increases and therefore its challenge level. The software treats the first row of the LCD as the sky and the second one as the ground. The buzzer is used to play jump sounds, death sounds, and score sounds, giving to the player a retro style “soundtrack”.

Hardware Design

List of parts used in this project:

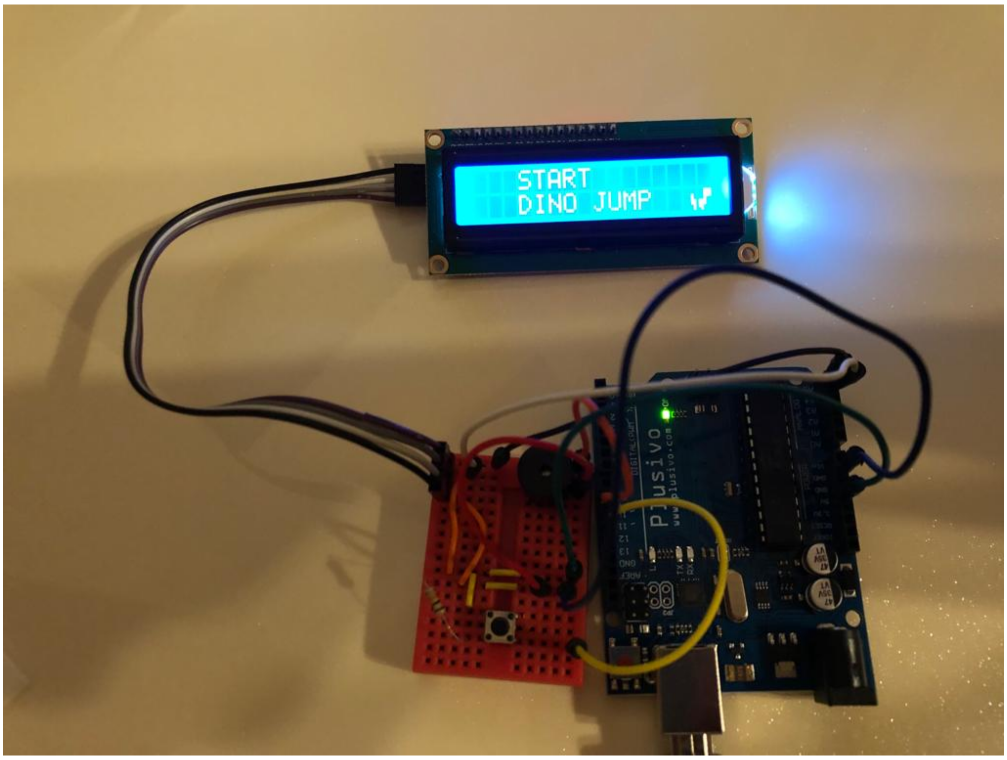
- **16×2 LCD with I2C**

- **Arduino UNO R3**



Other small components:

- Push button
- Buzzer
- 10KΩ resistor
- Wires



Software Design

The game code is based on the project by the youtube channel: Max Imagination.

I used the latest version of the Arduino IDE for the software development of this game, making use of its compilation error checking and nice black background .

The code includes several libraries that provide additional functionality to the Arduino system:

- LiquidCrystal_I2C.h: This library allows communication with I2C LCD displays.
- avr/io.h: This library provides access to AVR microcontroller I/O ports and registers.
- avr/interrupt.h: This library provides functions to work with interrupts on the AVR microcontroller.
- Wire.h: This library enables I2C communication on the Arduino.

Of course, The code uses various global variables that are used throughout the program for tracking certain events, updating the points, raising the difficulty of the game, etc.

All the user defined functions that the program uses are:

1. start_screen(): Displays the start screen of the game on the LCD display and waits for the button press to start the game.
2. run(): Controls the running animation of the dinosaur character by switching between different frames.
3. cleaner(): Clears previously displayed characters on the LCD display to keep the display clean and remove old elements.
4. dino_down(): Updates the display of the dinosaur character when it is in the air, showing the appropriate animation frames.
5. obstacle_generator(): Generates different types of obstacles (cacti or birds) at random positions on the LCD display.
6. collider(): Checks for collisions between the dinosaur character and obstacles, and handles game

over scenarios accordingly.

7. accelerate(): Handles the acceleration of the game over time by reducing the time period between obstacle movements.
8. jump(): Handles the jump action of the dinosaur character based on button presses.
9. jump_sound(): Generates a sound effect when the dinosaur character jumps.
10. points_tracker(): Tracks and updates the game score based on the time passed and obstacle crossings.
11. game(): Executes the main game loop, calling different functions to handle game logic.

Interrupt Service Routine (ISR) The code defines an Interrupt Service Routine (ISR) for handling a timer interrupt. The timer interrupt is used to control the timing and execution of the points system, as well as running animation for the dinosaur while he is on the ground.

Results

The result of this project is a pretty nice and smooth, but most importantly playable, copy of the google dinosaur jump game. While it does not have the graphics of the browser game due to the limitations of the LCD and personal skill, I am quite proud of the result, even though there is always something that can be improved!

Conclusions

At first I thought this project will be quite straight forward and honestly, quite boring. But as soon as I found a project that really called out to me and decided to improve upon the original code, the more I started to enjoy working on this project. It is certainly is a challenge to make sure the code you develop is easy to understand, works as intended and most importantly, does not have redundant pieces of code. The hardware side was quite straight forward considering that I use a I2C module for the LCD and the other components are a passive 3.3V buzzer and a button.

Download

The source code for the project in case you want to play the game yourself!

dino_jump_timisica_andrei_1221ea.txt

Bibliography/Resources

- Youtube channel of Max Imagination: https://www.youtube.com/@MaxImagination
- Create custom character for LCD: http://omerk.github.io/lcdchargen/
- Arduino_datasheet: https://docs.arduino.cc/static/32b4941b81a2c6a5308e0e9bd348d0e6/A000066-datasheet.pdf

Export to PDF