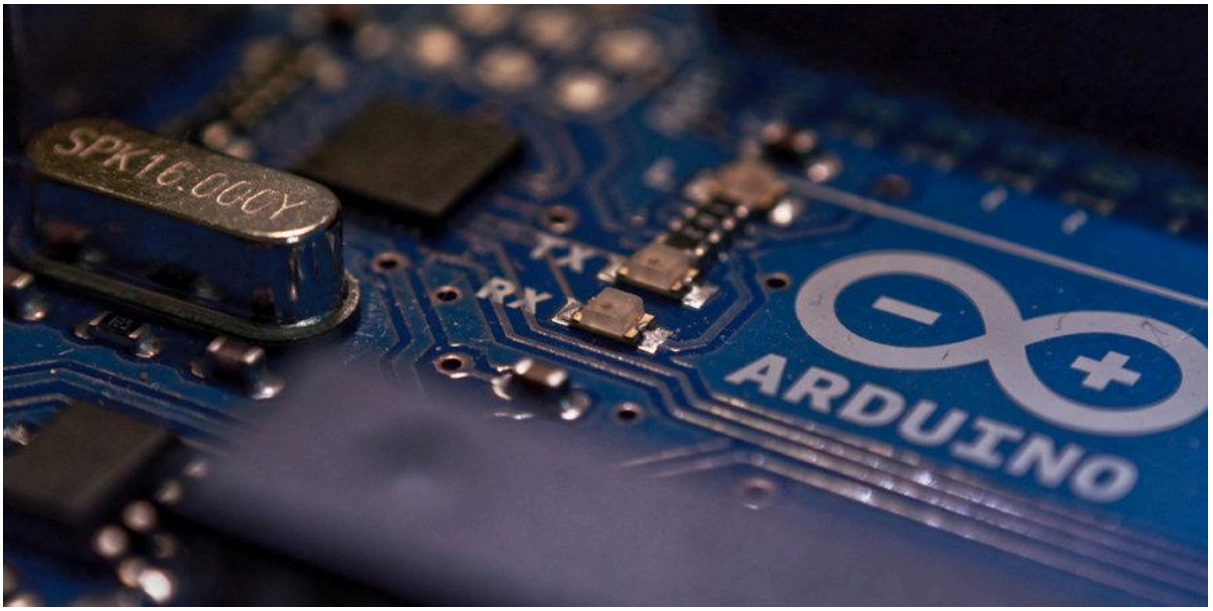
PROYECTO ARDUINO



Materia: Organización y arquitectura

Año: 2021

Docente: Roman, Alejandro Bond

Integrantes: Juarez Lucas y Godoy Thiago

Universidad Nacional
ARTURO JAURETCHE

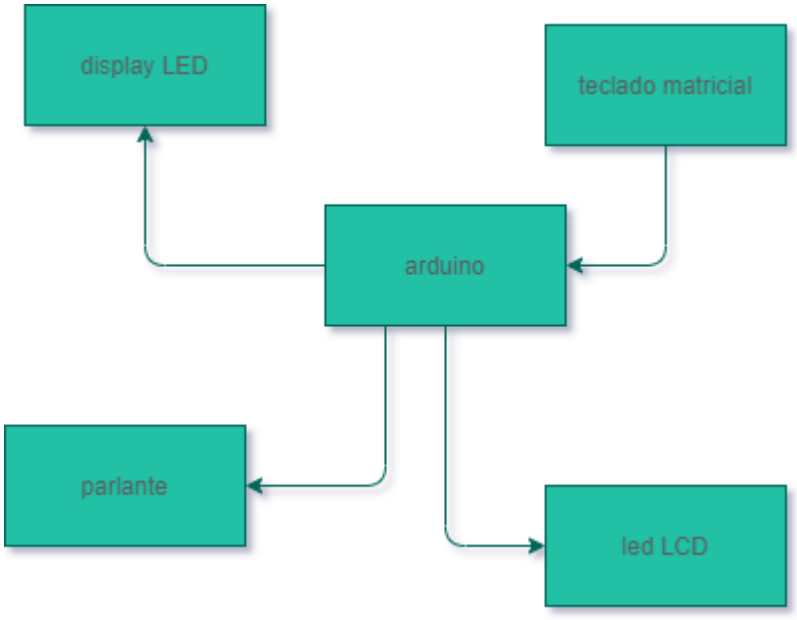
Introducción

El presente informe pretende describir el desarrollo de una aplicación en Arduino Uno. La aplicación seleccionada fue el tradicional juego conocido como El Ahorcado. El mismo será una versión simplificada dado que nos vemos limitados por el teclado disponible, por lo cual, solo se deberá averiguar las vocales de una serie de palabras que varían en dificultad. Utilizando los distintos recursos disponibles se pretende desarrollar el juego y que logre realizar una comparación exitosa del ingreso de vocales con la palabra a descubrir, devolver un tipo de respuesta ante un acierto o error y derrota o victoria.

Descripción

El trabajo fue realizado con las respectivas herramientas dadas por el profesor.
Dicho trabajo cuenta con:

- Display LCD: en el cual presentaremos las reglas del juego y los mensajes de acierto o error así como también la palabra incógnita.
- Teclado matricial: que nos permite la selección de palabras y el de vocales.
- Parlante: utilizado para devolver un sonido ante derrota o victoria.
- Led RGB: del cual solo podremos utilizar el led rojo para indicar si hubo un error.



Resultados

Nuestra primera tarea fue encontrar la manera de plasmar la aplicación, en nuestro caso fue el clásico juego ahorcado (contemplando las limitaciones del lenguaje de programación de arduino y las librerías a usar) o una caja musical con diversas canciones. Luego de días

pensando nos inclinamos por el juego. Este, debiera tener tres palabras a adivinar únicamente rellenando las vocales faltantes.
A la hora de programar nos encontramos con muchos problemas, uno de ellos fue cómo lograríamos comunicarnos con el jugador, ya que este debería tener una manera de poder ver las palabras y letras que iba completando. Nuestra idea principal fue imprimir texto por la terminal que nos ofrece arduino. Luego de días trabajando sin poder llegar a nuestro objetivo (poder imprimir un menú y la selección de una palabra aleatoria con la creación de un método y el comando .random) comentamos al docente los inconvenientes que estaban surgiendo, por lo cual nos recomendó que trabajemos con el display LCD y no con la terminal ya que esta cuenta con muchas limitaciones.
Nos dispusimos a volver al código y las pruebas, esta vez con el display LCD. Nuestro plan ahora era otro, dejamos atrás la selección de palabra aleatoria y optamos por un sistema donde el usuario elegiría, mediante teclado, la palabra en base a la dificultad deseada (siendo este el que quedaría en el modelo final) dicho sistema estaría conformado por niveles asignados a teclas numéricas y que corresponden a las palabras:

1. Fácil: hilo
2. Medio: caduco
3. Difícil: vislumbrando

El código en la parte loop es:

```
char key = keypad.getKey();
if (key) {
  switch (key) {
    case '1':
      problema = problemas[0];
      solucion = soluciones[0];
      lcd.print("Su palabra es:");
      delay(2500);
      lcd.setCursor(0,1);
      lcd.print(problemas[0]);
      delay(3000);
      juego = true;
      break;

    case '2':
      problema = problemas[1];
      solucion = soluciones[1];
      lcd.print("Su palabra es:");
      delay(2500);
      lcd.setCursor(0,1);
      lcd.print(problemas[1]);
      delay(3000);
      juego = true;
      break;
```

```
      case '3':
        problema = problemas[2];
        solucion = soluciones[2];
        lcd.print("Su palabra es:");
        delay(2500);
        lcd.setCursor(0,1);
        lcd.print(problemas[2]);
        juego = true;
        delay(3000);
        break;
    }
  }
```

Una vez lograda la selección de dificultad y por consiguiente las palabras, nuestra atención estaba en crear la base, la estructura del juego.
Ésta se basa en los principios básicos de vidas y/o oportunidades para perder o ganar, si el participante, acierta las palabras y la completa, ganará. Si el participante falla tres veces en el intento de descubrir qué vocales conforman la palabra, éste perderá. Una idea simple, pero compleja de programar, ya que nos llevó más de cincuenta líneas de código.
Inicialmente, no pudimos plantearnos la manera de resolver una forma eficaz de lograr la comparación de palabras con el ingreso de vocales. Pero, con la guía del docente creamos dos listas de strings que serían la base de nuestro programa. Las mismas contenían las mismas palabras pero, en una estableceríamos las palabras “solución” que no tenían nada distinto a cualquier lista de strings. Y por otra parte, tendríamos la lista con palabras “problema” que a diferencia de la primera, reemplazamos las vocales con guiones bajos, los cuales serían la incógnita a resolver.

```
String problemas[] = {"h_l_", "c_d_c_", "v_sl_mbr_nd_"};
String soluciones[] = {"hilo", "caduco", "vislumbrando"};
```

Trabajando nuevamente en la parte loop para poder repetir el proceso cada vez que el participante ingresa una nueva vocal procedimos a escribir la parte del código que nos permitirá realizar la comparación de estas dos listas y la forma de operar con las mismas.
Resolvimos trabajar utilizando:

1. Un while que tome como parámetro un booleano que bajo ciertas condiciones cambie su valor.
2. Dentro del while un condicional “if” al cual solo se ingresa si se presiona una tecla.
3. A su vez, dentro del while, creamos un controlador de selección o switch case para cada vocal de cada palabra.

4. Si la tecla presionada corresponde a una vocal dentro de la palabra “solución”. En primer lugar se averigua la posición de la misma y luego se procede a reemplazar la vocal seleccionada dentro de la palabra “problema”. Luego, se imprime por pantalla un mensaje que indica que hubo un acierto y se vuelve a la selección de vocal.
5. En caso de error, se muestra un mensaje indicando que hubo un error y se activa el led rojo 3 veces. A su vez el participante pierde una vida.

Adjuntamos el código correspondiente a una de las vocales y mostramos las instrucciones en caso de error:

```
while (juego) {
      char key_uno = keypad.getKey();
      if (key_uno) {
      switch (key_uno) {
      case 'a':
            //corroborar si la "a" es solucion del problema
            if (solucion.indexOf(key_uno) != -1) {
            problema.setCharAt(solucion.indexOf(key_uno), key_uno);
            lcd.clear();
            lcd.print(" ACERTASTE");
            delay(1500);
            lcd.clear();
            lcd.print(problema);
            delay(1500);
            }
            else {

            lcd.print(" ERROR");
            //SE ACTIVA LED ROJO
            for (int i = 0 ; i < 3 ; i++) {
            prenderRojo(255);
            delay(500);
            prenderRojo(0);
            delay(250);
            lcd.clear();
            lcd.print(problema);
            delay(1500);
            }
            delay(1500);
            vidas--;
            }
            break;
      }
    }
```

Una vez terminada la estructura de la dificultad y selección de palabras comenzamos a trabajar en los casos de victoria o derrota.
En este caso, utilizamos un condicional “if” para comparar la palabra “problema” y “solución” y otro más en caso de que sus vidas se hayan agotado. Si las palabras coinciden, se imprime por pantalla “ganaste” y si se terminan las vidas se imprime “game over”.

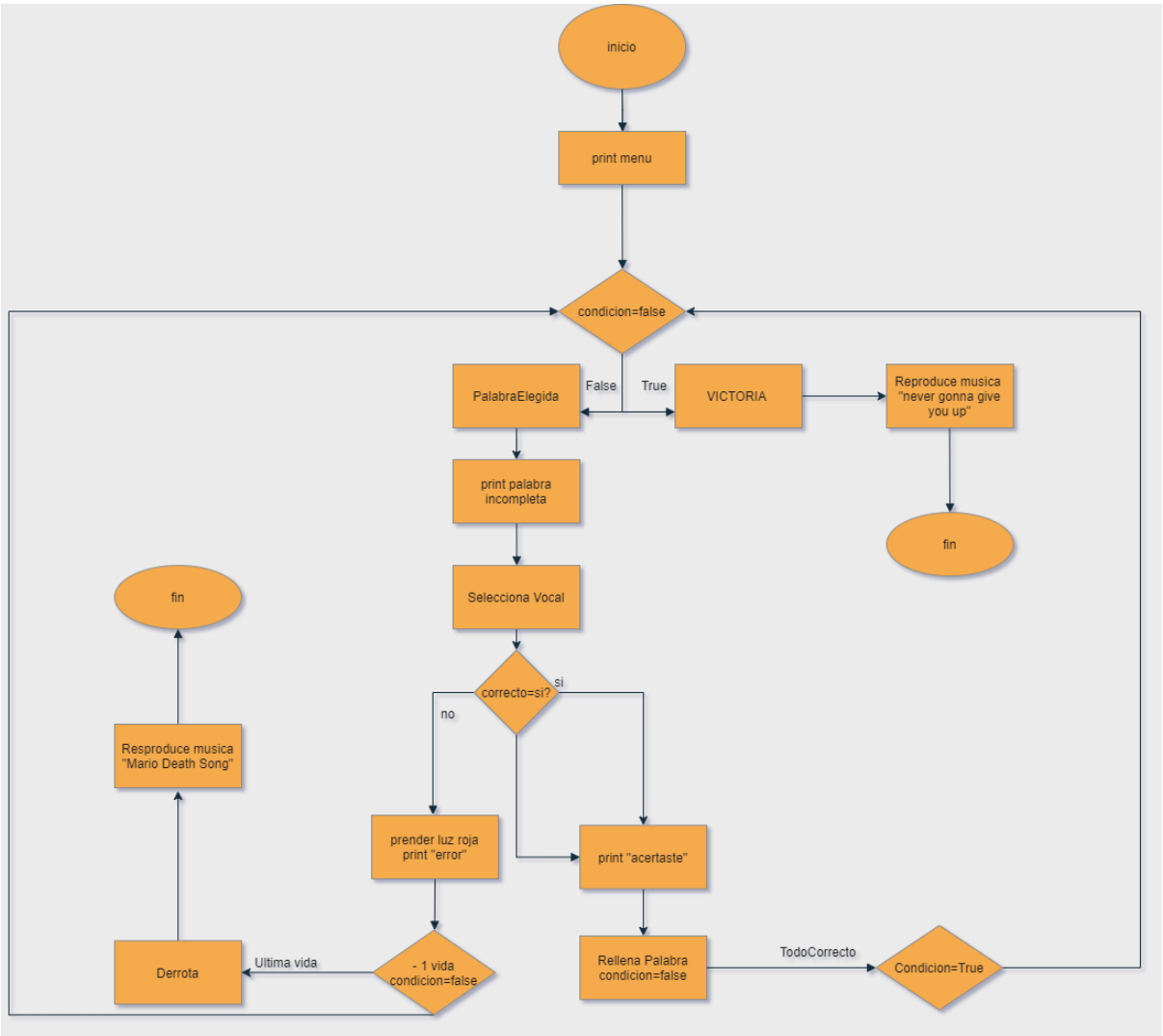
```
////////////////////////////////////////////
//CONDICION DE VICTORIA ROYALE
////////////////////////////////////////////

    if (problema == solucion) {
      lcd.clear();
      lcd.print("    GANASTE");
      juego = false;
      delay(1500);
      win();
    }

////////////////////////////////////////////
//CONDICION DE DERROTA
////////////////////////////////////////////

    if (vidas == 0) {
      lcd.clear();
      lcd.print("    GAME OVER");
      delay(1000);
      game_over();
      juego = false;
    }
  }
}
```

Por último, pero no menos importante, restaba realizar un menú práctico que explicara las reglas del juego y los botones de los que dispondremos para jugar. Nada muy complejo para este punto del trabajo. Se resolvió mediante el uso de diversos “print”. Pero una vez finalizado el menú, se nos ocurrió buscar alguna manera de utilizar de manera más eficaz el espacio disponible en pantalla por lo que nos pareció conveniente buscar un método de utilizar ambas filas de la pantalla en nuestra versión final. Esto se logró mediante el uso de la declaración:

```
lcd.setCursor(0,1);
```

Una vez finalizado todo sentimos que la sensación de victoria a la hora de completar las palabras no era lo suficientemente gratificante, una palabra impresa por pantalla no servía,

necesitábamos algo más. Pensamos: “¿por qué no música?”. Optamos por “Never Gonna Give You Up” de Rick Astley para la condición de victoria y “Death Tab” de Super Mario Bros para la condición de derrota, finalizando así nuestro proyecto.



Conclusiones

Este trabajo fue un gran desafío personal para ambos, dedicamos mucho tiempo a poder pulirlo y que sea realmente divertido de jugar. Más allá de ser un trabajo para una materia de la facultad nuestro plan principal fue que nos divirtiéramos haciéndolo y mostrándolo al punto de sentirnos representados en cada aspecto del proyecto.
Lamentablemente a la hora de llevar a cabo nuestras ideas nos vimos forzados a limitarnos en varios aspectos como por ejemplo la música, esta misma no fue compuesta ni programada por nosotros ya que llevaría mucho tiempo hacerlo y no poseemos el conocimiento musical que esta requiere, así que, optamos por utilizar una pequeña parte de código fuente de terceros para esta parte. Otra limitación fueron las palabras, ya que hubiéramos deseado colocar palabras más complejas con vocales repetidas pero la

solución que hallamos a la hora de programar la comparación de palabras solo nos permite reemplazar la primera vocal, en caso de que hubiese varias el programa no responde correctamente. Sumado a esto, el teclado matricial es un buen recurso pero igualmente nos limita a la hora de adivinar consonantes sin hacer tediosa la interacción del usuario con el juego.
En conclusión, fue una buena experiencia, ligeramente frustrante pero muy satisfactoria tras lograr nuestro objetivo.

Bibliografía

https://github.com/robsoncouto/arduino-songs
https://www.youtube.com/watch?v=R-CRIthB7ZY
https://forum.arduino.cc/t/variable-or-field-declared-void/201442
https://www.arduino.cc/reference/en/language/structure/control-structure/for/
https://www.arduino.cc/en/Tutorial/LibraryExamples
https://www.arduino.cc/reference/es/language/variables/data-types/string/functions/indexof/
https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/setchara/