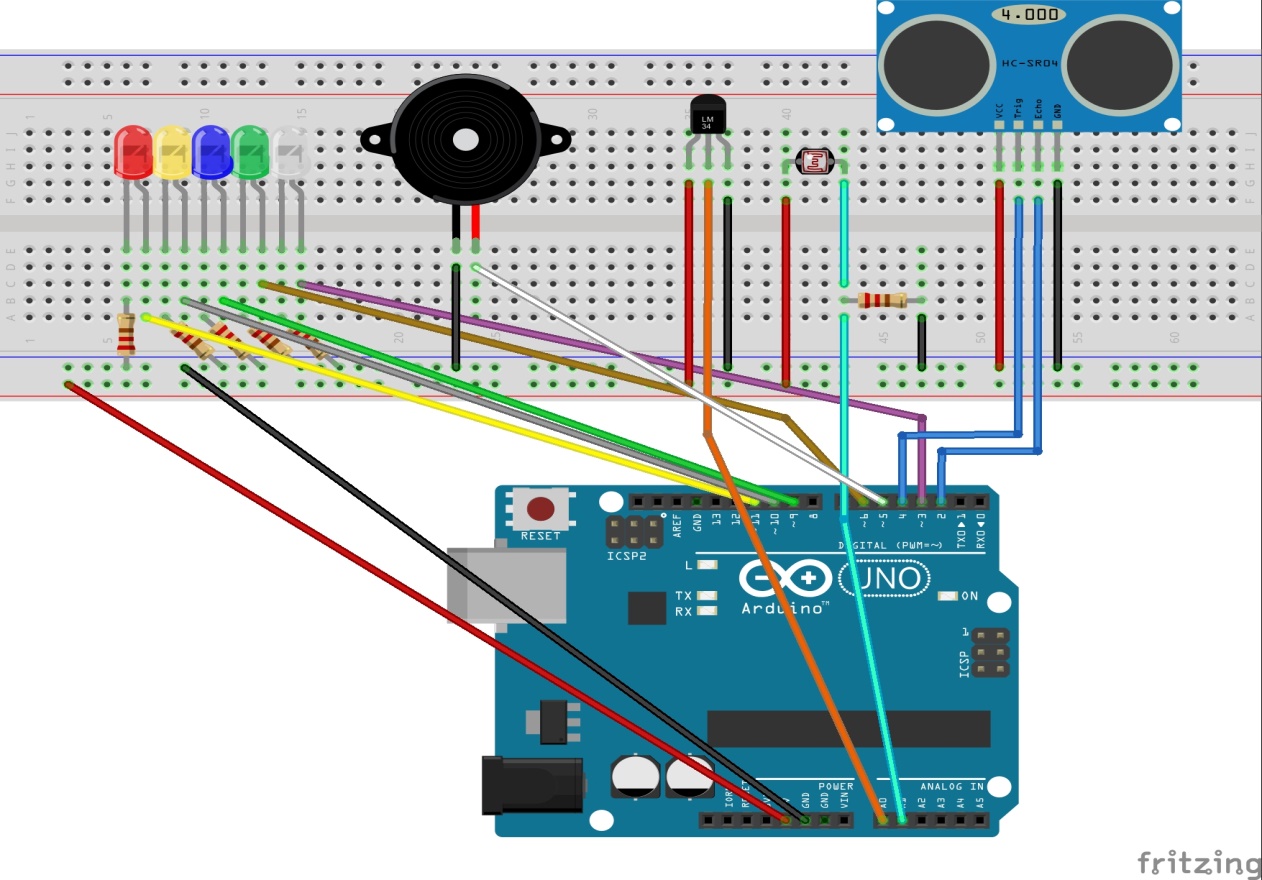
משחק אינטראקטיבי

במסגרת הפרויקט החלטנו לבנות משחק אינטראקטיבי המשלב מספר חיישני ומספר רכיבים אלקטרוניים.



במשחק יש 3 שלבים, בכל שלב משתמשים בחיישן אחר.

הרכיבים בהם השתמשנו:

חיישן מרחק – חיישן דיגיטלי המאפשר מדידת מרחק ע"י שליחת גל אולטראסוני ומדידת הזמן שעובר עד שהגל מתקבל חזרה במקלט. בעזרת חישוב, תוך שימוש במהירות הקול הידועה, ניתן לדעת את המרחק מהעצם בו פוגעים גלי הקול.

חיישן טמפרטורה – חיישן אנלוגי המודד את המתח על הרגל הנבדקת. בעזרת רזולוציית הדגימה של הארדואינו וידע מקדים שכל 10mV מהווים מעלת צלזיוס אחת, מחשבים את הטמפרטורה.

חיישן אור – נגד המשתנה על פי כמות האור שהוא קולט. נמדד בעזרת רגל אנלוגית. חיברנו לחיישן נגד של 10kΩ כדי למנוע מצב של קצר במקרה ויש חושך מוחלט והתנגדות קרובה לאפס (התנגדות אפסית אינה קיימת במציאות). סיבה נוספת היא למנוע מצב של "רגל צפה" ע"י זה שהמעגל סגור כל הזמן, ללא תלות בחיישן עצמו.

זמזם – רכיב פיאזואלקטרי המשנה את תכונותיו בהתאם למתח החשמלי המופעל עליו. במקרה שלנו – ישנה פונקציה מובנית בארדואינו (Tone) המקבלת מספר המהווה את התדר הרצוי, ומפעילה את הזמזם לפי התדירות הרצויה, ובכך נוצרים הצלילים.

לד (בצבעים שונים) – דיודה פולטת אור. רכיב המאפשר מעבר מתח בכיוון אחד בלבד, ומגיב למעבר המתח ע"י פליטת אור. הלד ניתן לחיבור בדרך אחת בלבד – האנודה מחוברת למקור המתח והקתודה מחוברת לאדמה. מכיוון שהלדים רגישים למתח, אפילו למתח נמוך יחסית, מחברים אליהם בטור נגדים בעלי ערכים נמוכים (220Ω-330Ω) בד"כ, כדי להגביל את המתח הנופל עליהם וכך להגן עליהם.

נגד – רכיב חשמלי בעל התנגדות למעבר המתח, ובעל מספר שימושים. נמדד ביחידות אוהם Ω ומסומן באות R.

הסבר על מהלך המשחק:

בשלב הראשון על השחקן לתמרן את חיישן הטמפרטורה עד שיגיע לקריאה של 28 מעלות.

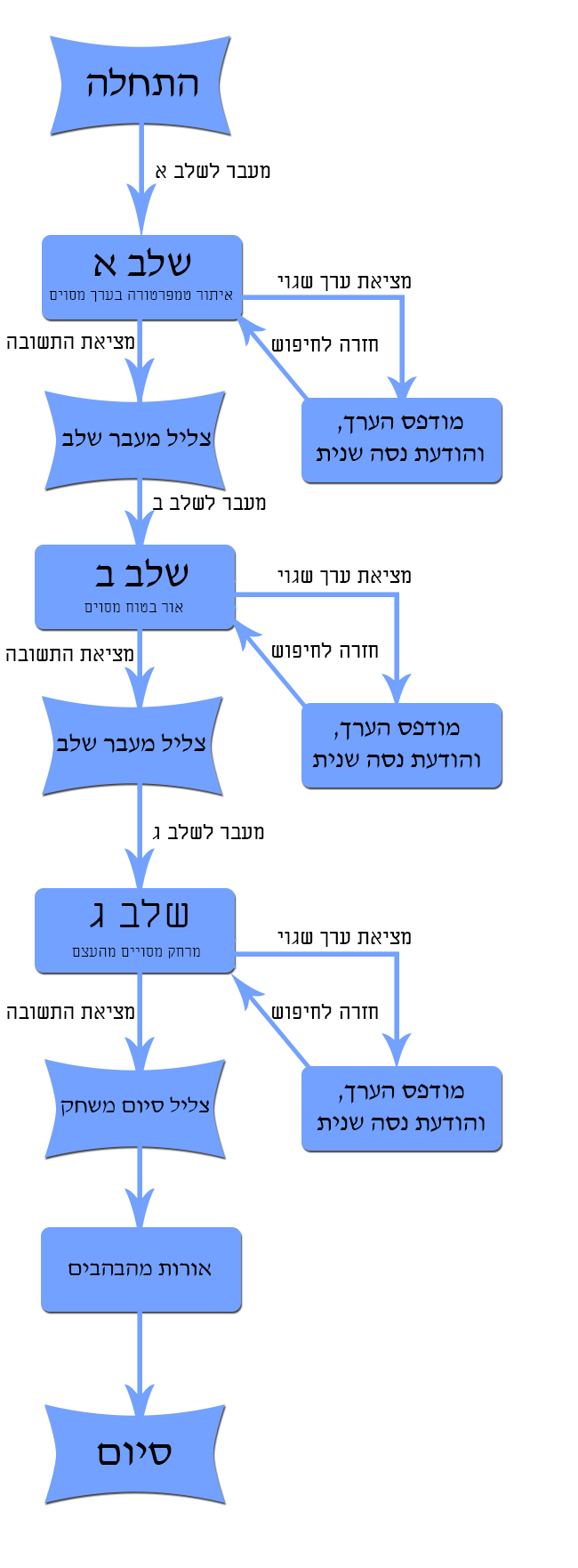
בשלב השני על השחק לתמרן את חיישן התאורה כך שיגיע לקריאה של בין 250 למ270 בקלט האנלוגי, כ1220-1318 מיליוולט.

בשלב השלישי על השחקן לתמרן את חיישן המרחק כך שיקבל קריאה של 90-100 ס"מ.

לאחר הצלחה בכל שלב מושמע קטע מוזיקלי קצר.

בסיום המשחק ולאחר הצלחה בשלב 3, מתנגנת נעימה נוספת והלדים מהבהבים.

תרשים זרימה:



קוד: (260 שורות, 4 עמודים)

#include <Tone.h> //the external Tone library is used

#include <rtttl.h>

//pins for the different sensors and leds

#define tmp\_sensor A1

#define buzzer 11

#define dist\_echo 13

#define dist\_trig 12

#define light\_sensor A0

#define octave 0

#define whiteled 9

#define greenled 3

#define blueled 10

#define yellowled 5

#define redled 6

//variables for storing data

float temprature **=** 0**;**

float temp\_sample **=** 0**;**

int distance **=** 0**;**

long duration **=** 0**;**

float cm **=** 0**;**

int light **=** 0**;**

//array of chars for the music

const char popcorn**[]** PROGMEM **=**

"Popcorn:d=4,o=5,b=160:8c6,8a#,8c6,8g,8d#,8g,c,8c6,8a#,8c6,8g,8d#,8g,c,8c6,8d6,8d#6,16c6,8d#6,16c6,8d#6,8d6,16a#,8d6,16a#,8d6,8c6,8a#,8g,8a#,c6"**;**

const char pink\_panther**[]** PROGMEM **=**

"PinkPanther:d=4,o=5,b=160:8d#,8e,2p,8f#,8g,2p,8d#,8e,16p,8f#,8g,16p,8c6,8b,16p,8d#,8e,16p,8b,2a#,2p,16a,16g,16e,16d,2e"**;**

//object for the music player

Rtttl player**;**

//array of lights

int lightarr**[**5**]** **=** **{**whiteled**,** greenled**,** blueled**,** yellowled**,** redled**};**

void setup**()** **{**

// put your setup code here, to run once:

Serial**.**begin**(**9600**);**

//declare all the pins

pinMode**(**tmp\_sensor**,** INPUT**);**

pinMode**(**light\_sensor**,** INPUT**);**

pinMode**(**dist\_echo**,** INPUT**);**

pinMode**(**buzzer**,** OUTPUT**);**

pinMode**(**dist\_trig**,** OUTPUT**);**

pinMode**(**whiteled**,** OUTPUT**);**

pinMode**(**greenled**,** OUTPUT**);**

pinMode**(**blueled**,** OUTPUT**);**

pinMode**(**yellowled**,** OUTPUT**);**

pinMode**(**redled**,** OUTPUT**);**

player**.**begin**(**buzzer**);**

//welcome the player

Serial**.**println**(**"Hello! welcome to our game!"**);**

Serial**.**println**(**"our game has 3 Levels. Let's start!"**);**

delay **(**3000**);**

**}**

Code 1

void loop**()** **{**

//play level 1

Level1**();**

Serial**.**println**(**"Great! you finished level 1!. let's keep going"**);**

delay **(**3000**);**

//play level 2

Level2**();**

Serial**.**println**(**"Great! you finished level 2!. let's keep going"**);**

delay **(**3000**);**

//play level 3

Level3**();**

delay **(**3000**);**

Serial**.**println**(**"Great! you finished the game!"**);**

//flashing lights when you finish the game

lights**();**

**while** **(**1**)** **{}**

**}**

void Level1 **()** **{**

Serial**.**println **(**"let's start level 1!"**);**

Serial**.**println **();**

Serial**.**println **(**"in this level, you need to reach 28 degrees celsius sample in the temrature sensor"**);**

Serial**.**println **();**

delay **(**3000**);**

//sample the tempratures

temprature **=** Sample\_temp**();**

**while** **(**temprature **<** 28 **)**

**{**

Serial**.**print **(**"the temprature is: "**);**

Serial**.**print **(**temprature**);**

Serial**.**println **(**". not there yet. keep trying."**);**

delay **(**1000**);**

temprature **=** Sample\_temp**();**

**}**

//if sucsseded - print the result, play a little music and continue

Serial**.**print **(**"the temprature is: "**);**

Serial**.**println **(**temprature**);**

Serial**.**println**(**"you made it!"**);**

delay**(**1000**);**

Buzz**(false);**

**}**

void Level2 **()** **{**

Serial**.**println **(**"let's start level 2!"**);**

Serial**.**println **();**

Serial**.**println **(**"in this level, you need to reach the level between 250 to 270 in the sample of the light sensor"**);**

Serial**.**println **();**

delay **(**3000**);**

//sample the temprature

light **=** analogRead**(**light\_sensor**);**

**while** **(** light **<** 250 **||** light **>** 270 **)**

**{**

Serial**.**print **(**"the light level is: "**);**

Serial**.**print **(**light**);**

Serial**.**println **(**". not there yet. keep trying!"**);**

delay **(**1000**);**

analogRead**(**A3**);**

light **=** analogRead**(**light\_sensor**);**

**}**

//if sucsseded - print the result, play a little music and continue

Serial**.**print **(**"the light level is: "**);**

Serial**.**println **(**light**);**

Serial**.**println **(**"you made it!"**);**

delay**(**1000**);**

Buzz**(false);**

**}**

void Level3 **()** **{**

Serial**.**println **(**"let's start level 3!"**);**

Serial**.**println **();**

Serial**.**println **(**"in this level, you need to reach to distance between 90 to 100 centimeters"**);**

Serial**.**println **();**

delay **(**3000**);**

//sample the distance

distance **=** Sample\_dist **();**

**while** **(**distance **<** 90 **||** distance **>** 100 **)** **{**

**if** **(**distance **<** 300**)** **{**

Serial**.**print **(**"the distance is: "**);**

Serial**.**print **(**distance **);**

Serial**.**println **(**". not there yet. keep trying!"**);**

delay **(**1000**);**

**}**

distance **=** Sample\_dist **();**

**}**

//if sucsseded - print the result, play a little music and continue

Serial**.**print **(**"the distance is: "**);**

Serial**.**println **(**distance**);**

Serial**.**println **(**"you made it!"**);**

delay**(**1000**);**

Buzz **(true);**

**}**

//play music with the buzzer

void Buzz **(**boolean finish**)** **{**

**if** **(!**finish**)**

player**.**play\_P**(**pink\_panther**,** octave**);**

**else** player**.**play\_P**(**popcorn**,** octave**);**

**}**

//flash the lights when you finish the game

void lights **()** **{**

**for** **(**int j **=** 0**;** j **<** 3**;** j**++)** **{**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** **{**

digitalWrite**(**lightarr**[**i**],** HIGH**);**

delay **(**50**);**

digitalWrite**(**lightarr**[**i**],** LOW**);**

delay **(**50**);**

**}**

delay**(**50**);**

**for** **(**int i **=** 4**;** i **>=** 0**;** i**--)** **{**

digitalWrite**(**lightarr**[**i**],** HIGH**);**

delay **(**50**);**

digitalWrite**(**lightarr**[**i**],** LOW**);**

**}**

**}**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** 256**;** j**++)** **{**

analogWrite**(**lightarr**[**i**],** j**);**

delay**(**6**);**

**}**

**}**

delay**(**1000**);**

**for** **(** int i **=** 0**;** i **<** 5**;** i**++)**

digitalWrite**(**lightarr**[**i**],** LOW**);**

**}**

//sample the temprature

float Sample\_temp**()** **{**

//the arduino samples the sensor and gets the result in milivolts. because the arduino has 10bit a2d, the resolution is 5000/1024, which is 4.9 mV.

//every 10 mV is 1 degree

float temp **=** analogRead**(**tmp\_sensor**);**

float temperatureC **=** **(**5000 **\*** temp **/** 1024**)** **/** 10**;**

**return** temperatureC**;**

**}**

//sample the distance

float Sample\_dist**()** **{**

// The speed of sound is 340 m/s or 29.1 microseconds per centimeter.

// The ping travels out and back, so to find the distance of the

// object we take half of the distance travelled.

digitalWrite**(**dist\_trig**,** LOW**);**

delayMicroseconds**(**10**);**

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite**(**dist\_trig**,** HIGH**);**

delayMicroseconds**(**10**);**

digitalWrite**(**dist\_trig**,** LOW**);**

// Reads the echoPin, returns the sound wave travel time in microseconds

duration **=** pulseIn**(**dist\_echo**,** HIGH**);**

// Calculating the distance

cm **=** **((**duration**)** **\*** **(**1 **/** 29.1**)** **/** 2**);**

**return** cm**;**

**}**