Magic Ring

Introducere

Proiectul constă în realizarea unui joc interactiv, în care mișcările jucătorului se bazează pe efectuarea anumitor activități fizice, folosind un inel de pilates care are atașat un microcontroler cu un accelerometru si un flex senzor. Jucătorul va trebui să imite mișcările prezentate de către un avatar. Cu cât va imita o mișcare cât mai exact, cu atât punctajul aferent acelei mișcări va fi mai mare.

În contextul social actual, jocul vine în ajutorul celor care doresc să facă sport ghidați și asitați de către cineva, în acest caz de către un asistent virtual.

Utilizatorul se va bucura de o interfață grafică placută, realizată în Unity, unde va putea urmării miscările avatarului

Descriere generală

Plăcuța cu ESP32, va prelua datele furnizate de către senzori (de la accelerometru si de la flex senzor). Aceste date mă vor informa despre poziția inelului de fitness, cât și despre modul în care ascesta a fost deformat (strâns sau extins). Aceste informații vor fi prelucrate corespunzător, iar rezultatul va fi trimis prin Wi-Fi, către server unde vor fi validate și i se va acorda un punctaj jucatorului. La rândul lui, ledul RGB conectat la plăcuță va lumina în concordanță cu punctajul. Întrucât plăcuța va fi atașată pe inelul de fitness, aceasta va fi cuplată la un modul de încărcare LiPo, pentru a asigura sursa de curent.

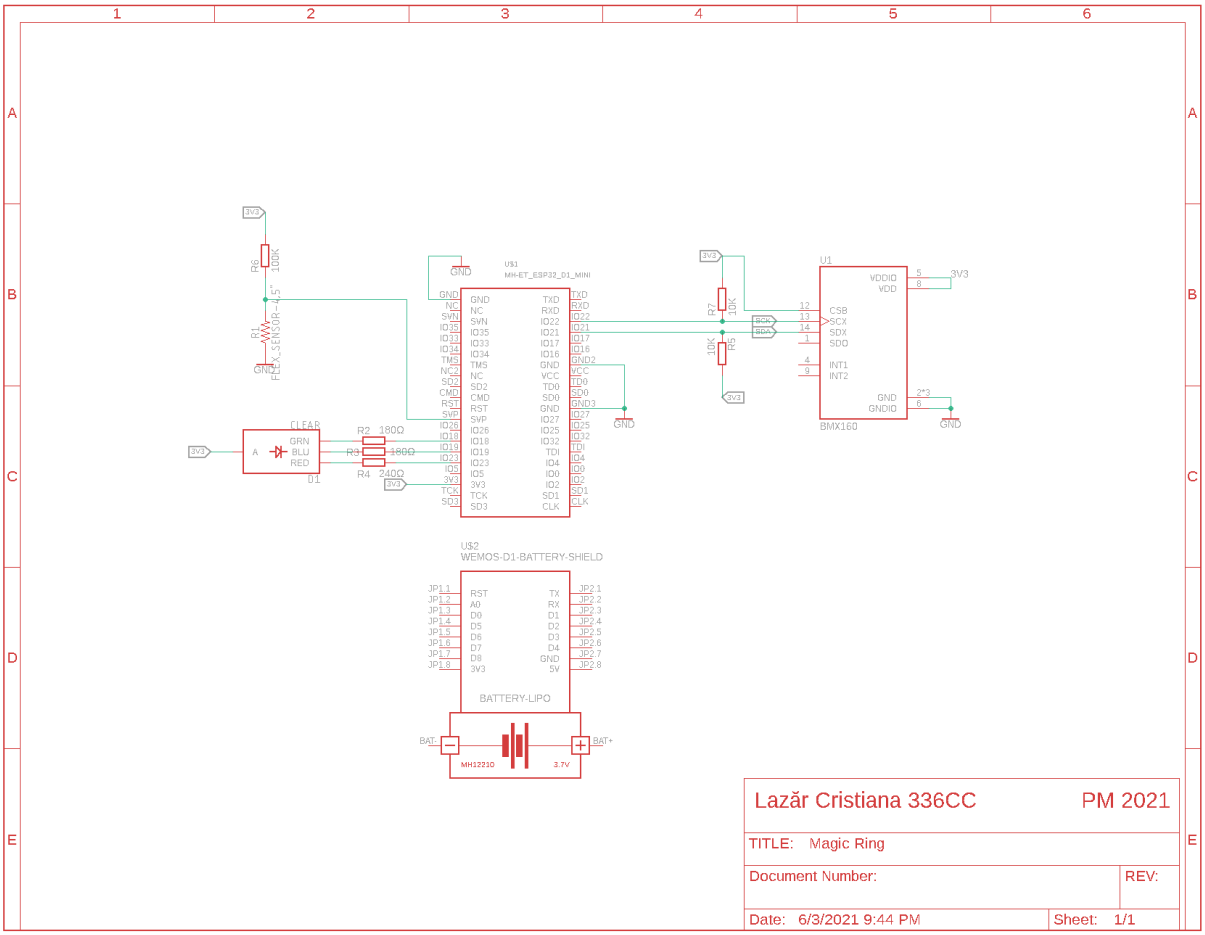
Schema bloc este următoarea:

Hardware Design

Componentele hardware folosite sunt următoarele:

- Wemos D1 ESP32 mini
- accelerometru BMX160 9-axis sensor
- flex senzor
- modul încărcare LiPo
- LiPo 18650 Cell
- RGB LED

Schema electrică este urmatoarea:



Software Design

Mediu de dezovltare folosit pentru programarea microcontrolerului a fost Arduino IDE. Pentru partea software, UI și interacțiune om-calculator am utilizat Unity.

Pentru a programa ESP32 am utilizat bibliotecile:

- WiFi.h - pentru a conecta plăcuța la rețeaua locală
- ArduinoWebsockets.h - pentru a realiza comunicarea dintre plăcuță și Unity - ESP32 se comportă ca un server, pe când programul realizat în Unity ca și un client. Astfel clientul Unity se conectează la serverul ESP32, iar după acesta, plăcuța va trimite datele recepționate de la senzori (10 pachete pe secundă) la programul din Unity, acesta urmând la recepționarea lor să le prelucreze.
- Adafruit_Sensor.h
- DPEng_BMX160.h
- Madgwick_BMX160.h - ultimele trei biblioteci au fost utilizate pentru a filtra datele recepționate de la BMX160

Pentru recunoașterea mișcărilor realizate de către jucător și validarea acestora, am realizat o rețea neuronală (Neural Network) utilizând Tensorflow, Keras pentru construirea acesteia. Rețeaua a fost realizată în Python, iar mai apoi convertită în ONNX, pentru a putea fi integrată în Unity folosind modulul Barracuda.

Arhitectura rețelei este următoarea:

```
Model: "sequential"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 64)                448
_________________________________________________________________
dense_7 (Dense)              (None, 16)                1040
_________________________________________________________________
dense_8 (Dense)              (None, 5)                 85
=================================================================
Total params: 1,573
Trainable params: 1,573
Non-trainable params: 0
_________________________________________________________________
```

Pentru a putea antrena rețeaua, a fost nevoie de un set de date care să conțină valorile recepționate de la senzori (datele de la accelerometru, respectiv giroscop) tagate corespunzător mișcării realizate. În momentul actual, sunt disponibile 5 exerciții care se pot realiza cu inelul:

- Front Bend 

- Left Bend 

- Right Bend 

- Left Twist 

- Right Twist 

Pentru fiecare astfel de exercițiu, am extras datele corespunzătoare, având proximativ 300 de date per sample. Acest lucru mi-a permis antrenarea rețelei pentru a atinge un accuracy de aproximativ 91% atât pe setul de validare cât și pe setul de antrenament.

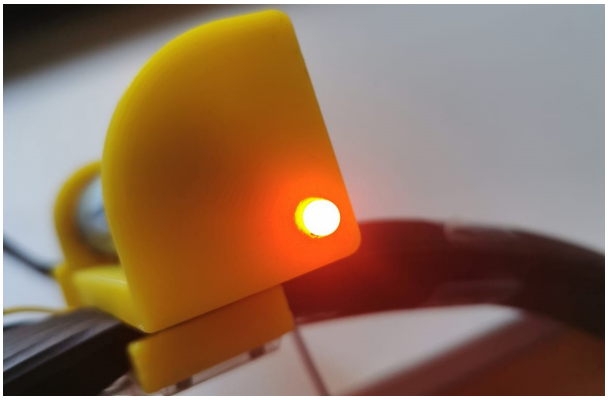
În Unity, fiecare dintre exercițiile enumerate anterior, la care se mai adaugă încă un exercițiu și anume Push (care se folosește de datele primite de la flex senzor) au câte o animație asociată. Această animație îi spune utilizatorului, modul în care trebuie sa manevreze inelul. Dacă utilizatorul îl manevrează în modul indicat, mișcare se va valida, iar utilizatorul va primi un punctaj. Utilizatorul trebuie să se sincronizeze cu animația pentru a primi punctaj.

Fiecare exercițiu are o durată de aproximativ 2 secunde. În momentul în care se anunță inceperea unui nou exercițiu, se vor înregistra mesajele primite de la placuță, iar după două secunde se va valida mișcarea corespunzător, urmând să se șteargă datele actuale, pentru a face loc unei noi serii/repetări.

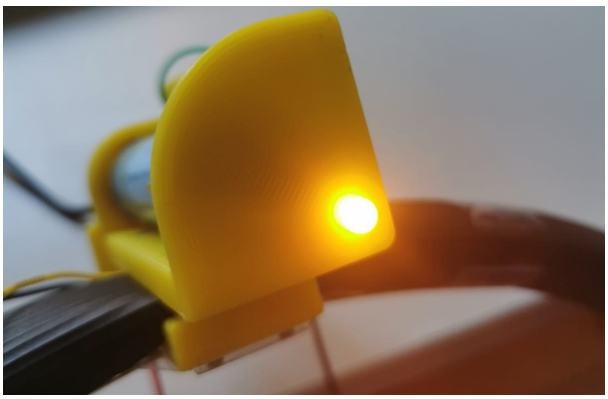
Rezultate Obținute

ESP32

În momentul în care plăcuța este conectată la alimentare, led-ul va lumina culoarea roșie pentru a semnaliza faptul că așteaptă conectarea la rețeaua locală



În urma conectării la Wi-Fi, led-ul va lumina culoare galbenă pentru a semnala faptul că așteaptă conectarea clientului Unity

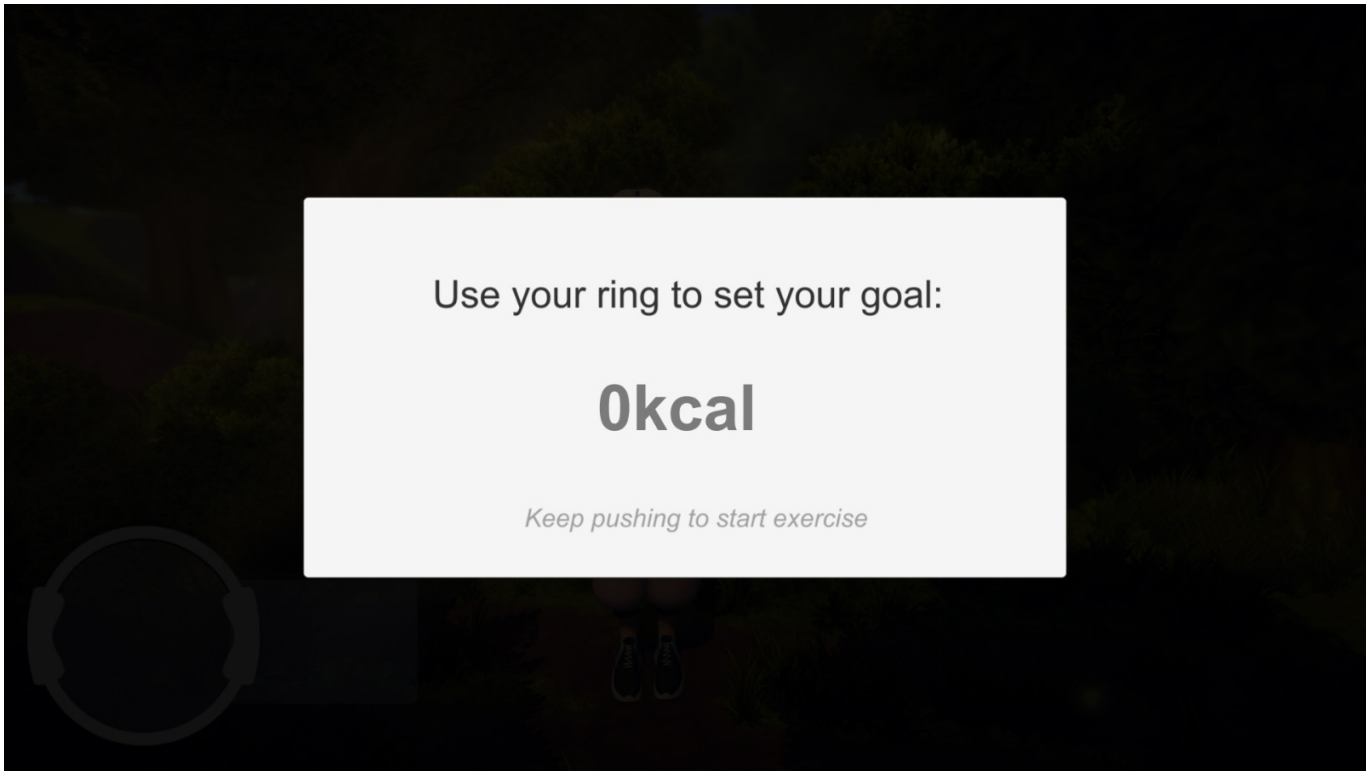


La conectarea clientului, led-ul va lumina albastru.

Dacă clientul se deconectează, led-ul va deveni din nou galben.

Unity

Jocul are o pagină de start, unde folosind inelul, poți seta un target de câte calorii ar putea utilizatorul să ardă, dacă realizează exercițiile propuse.



Utilizatorul trebuie să preseze inelul, iar target-ul va crește din 10 în 10 unități kilocalorice.

Dacă utilizatorul realizează o mișcare corectă, acest lucru va fi semnalizat vizual prin apariția unor particule pe inel. Dacă utilizatorul realizează o serie de 5 exerciții corecte, marginea ecranului va lumina, iar led-ul conectat la placuță va lumina puternic.

Concluzii

Magic Ring este un joc funcținal care avut ca surse de inspirație două jocuri celebre, Ring Fit Adventure de la Nindento și Just Dance de la Ubisoft.

Utilizarea inteligenței artificiale oferă atât posibilitatea de valida datele intr-un mod cât mai precis, cât și posibilitatea extinderii funcționalității jocului prin adăugare unor noi exerciții.

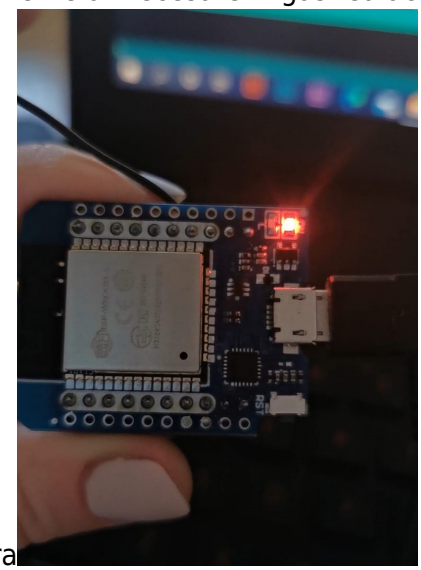
Download

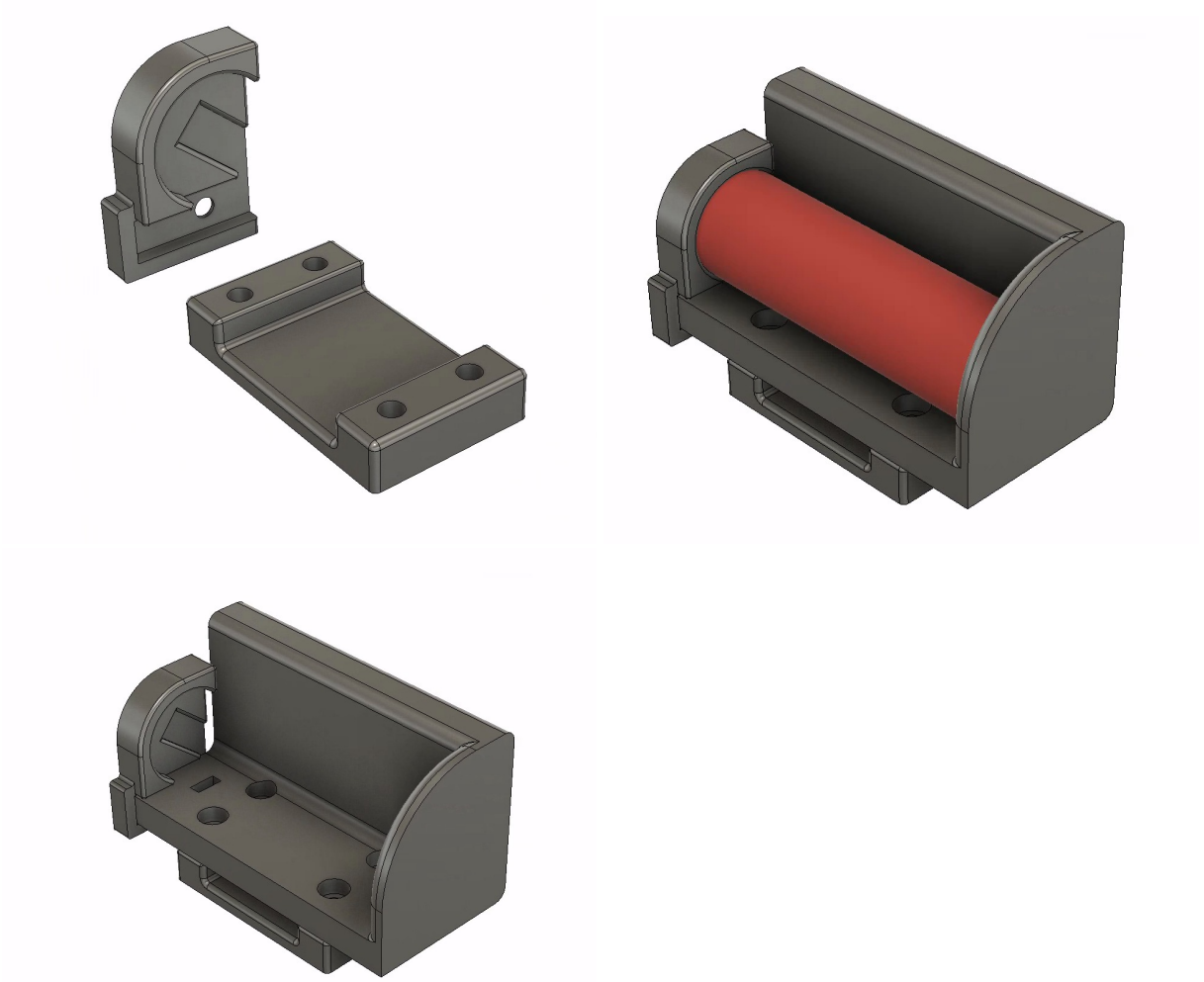
magicringpm2021cristianalazar336cc.zip

Notebook colab - pentru antrenarea retelei
https://colab.research.google.com/drive/1ZnOTPeVc7nCOq5lnstSDZ9ISPZ30W1_c

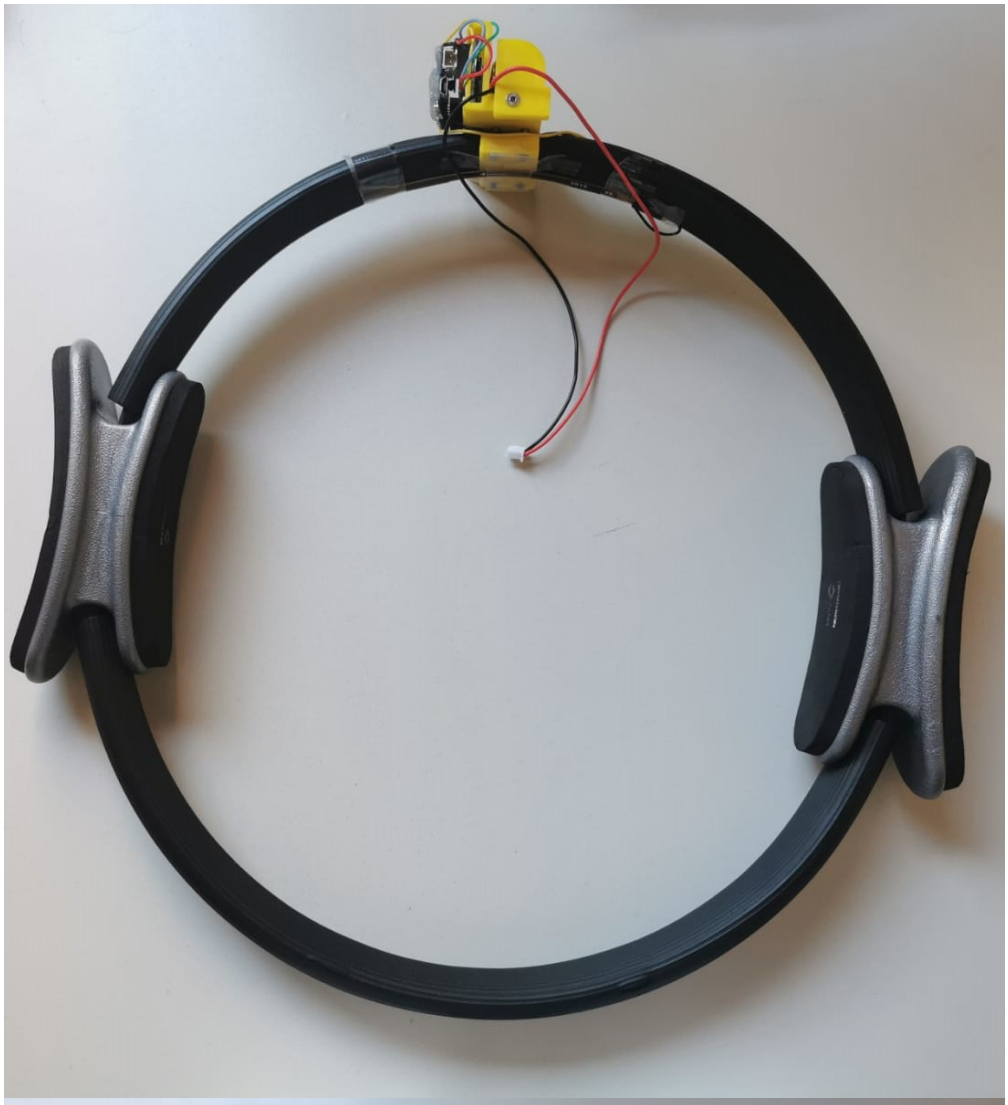
Demo: https://youtu.be/xma1qHbcmXw

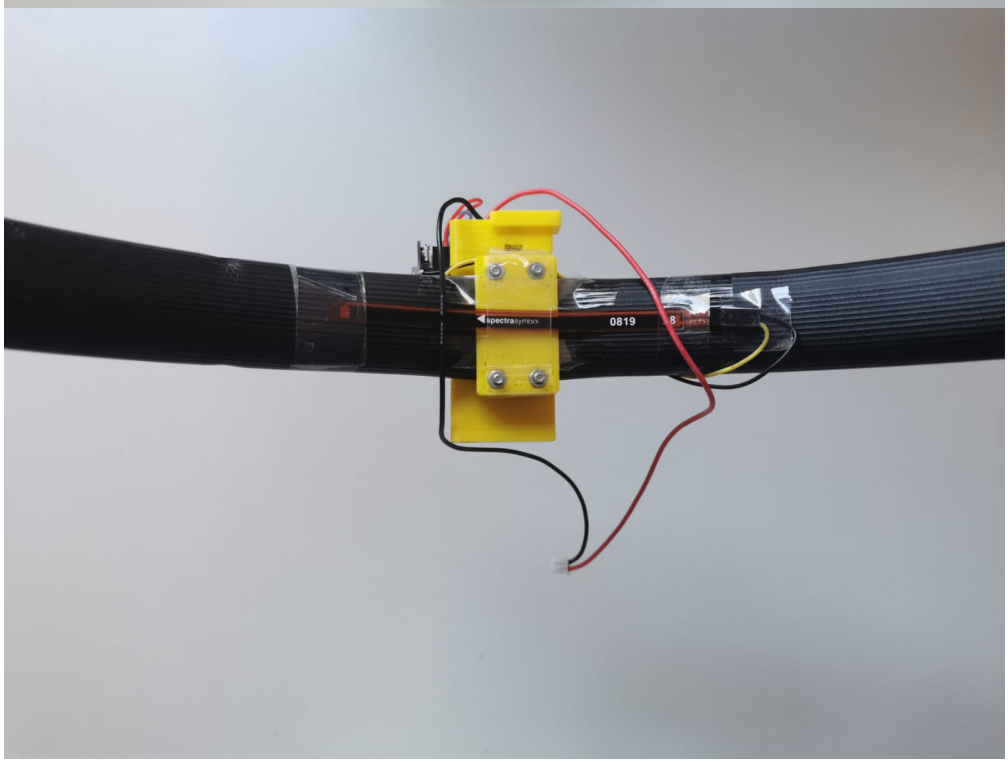
Jurnal

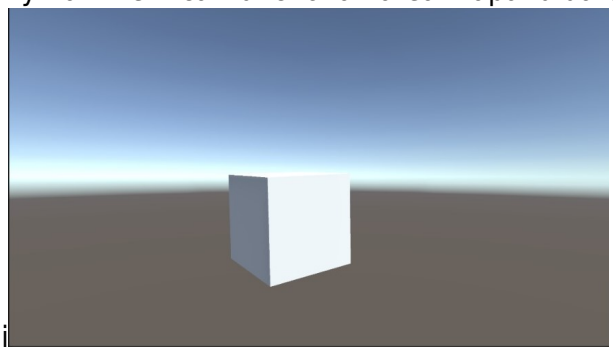
1. Stabilirea temei proiectului
2. Stabilirea componentelor necesare + găsirea acestora
3. Testarea acestora 
4. Realizarea unui suport pentru plăcuță și baterie care să fie atașat pe inel

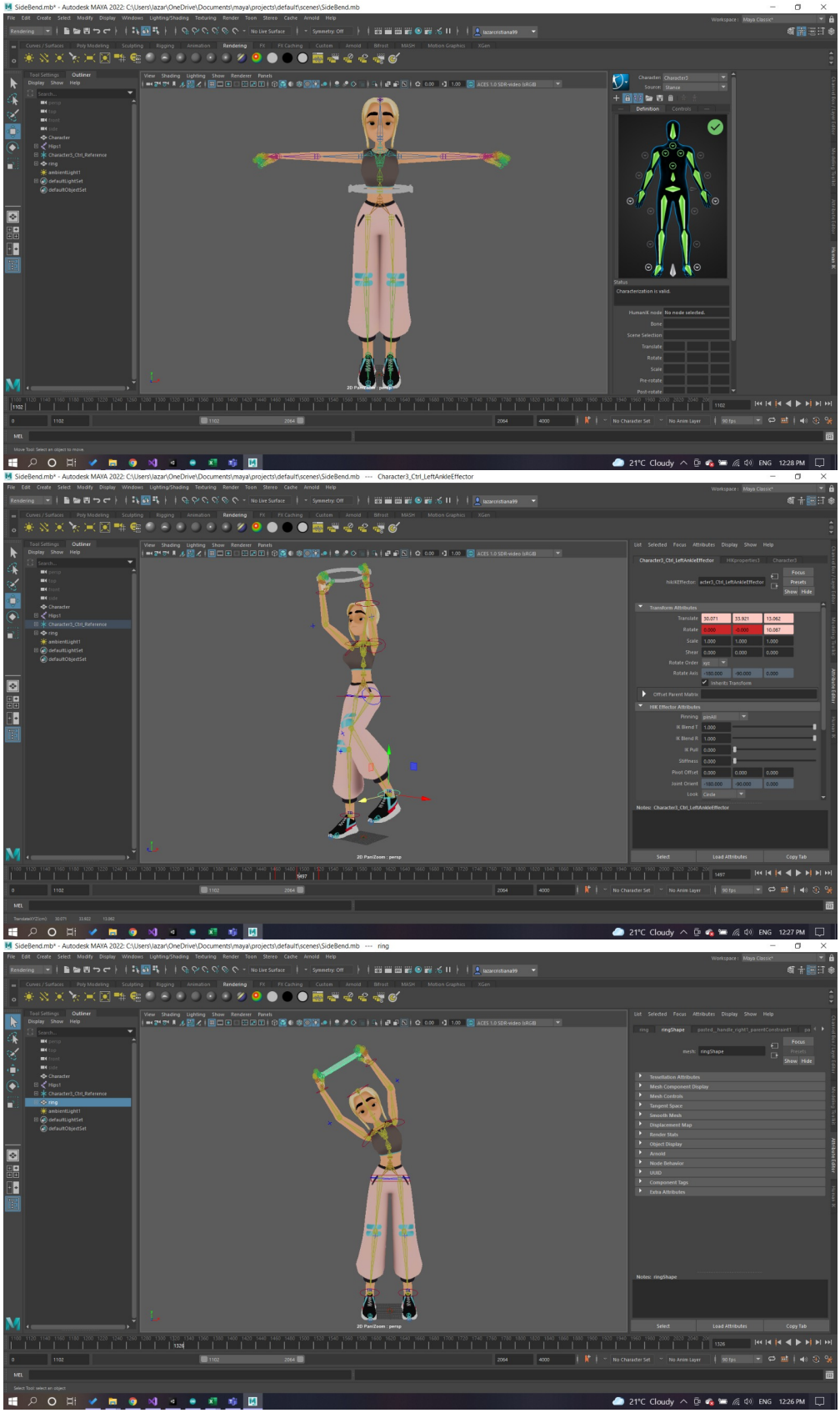


5. Atașarea suportului pe inel cu toate componentele




0819

6. Realizarea comunicație esp32-unity - am verificat functionalitatea mapand datetele primite pe un cub, cubul imita miscarea inelului 
7. Realizarea modelului 3D a inelului și ajustarea modelului 3D a caracterului (caracterului are la baza un model 3D care poate fi găsit și downloadat gratuit pe Mixamo) Am modificat topologia caracterul pentru a se potrivi tematicii alese
8. Realizarea scheletului caracterului si animarea acestuia



9. Inegrarea în Unity și implementarea logicii de joc

Bibliografie/Resurse

- https://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/magicring?do=export_pdf

- https://github.com/drcpattison/BMX160
- https://github.com/gilmaimon/ArduinoWebsockets
- https://eu.mouser.com/datasheet/2/783/BST_BMX160_DS000-1509610.pdf
- https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- https://www.mixamo.com/#/
- https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-free-109565
- https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361
- https://assetstore.unity.com/packages/3d/vegetation/the-illustrated-nature-sample-161188