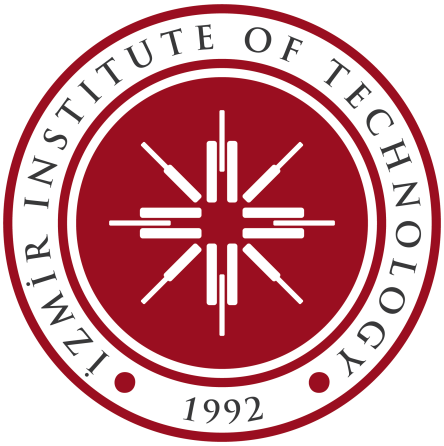
CENG424 Embedded Computer Systems
Spring 2022

Project Title: Wireless Game Controller and Game Design
Final Project Report

June 8, 2022



Group Members

- Emre Nedim Hepsağ & 250206012
- Bengisu Sağlam & 250206027
- Berke Eren & 250206008
- Ayşenur Karaca & 250206018
- Emre Korkmaz & 250201044
- Gazi Özgen & 250201051
- İsmail Batuhan Akbaş & 250201040

Abstract

Computer games are an important part of the entertainment industry. They have been enhanced in both graphics and the way how they are controlling since the early days of the computers. Game industry also plays a significant role in economy so that large investments have been made to increase reality of the games. In this project, wireless game controller with four push buttons and joystick were designed. Wireless communication between controller and receiver hub is maintained through HC-05 Bluetooth modules. Bluetooth has a shorter range than Wi-fi and Zigbee but consumes lower power while provide enough range for gaming purposes. Two modules are used, and they are interfacing with Arduino's via serial USART interface. AT commands are used to set one module as a master and the other as a slave. By pairing two modules, it is ensured that communication is maintained. Master module starts the communication if the slave module is available. Controller sent a data when one of the buttons are pressed or joystick movements. Data transferred between modules and received at the hub. Receiver hub is serially connected to the computer. Data transfer between hub and computer is maintained by using Phyton serial port. It is observed from the tests results that data which is sent from controller and received at computer experiences a delay. Delay between modules is relatively small. Most of the delays have occurred between receiver hub and serial port of the Phyton. Implemented controller circuit is placed inside an box which is cut with CNC machine.

1 Introduction

Game controllers make playing easier most of the time while offering a play comfort besides. Especially for game players, those are important issues. Wireless game controllers are the common controllers used by gamers recently with its feature of independancy of tangled cables which provides free movement. Although there are lots of game controller choices with lots of futures in the market, its getting harder to buy a new one because of the economic situations. Considering all of them, we aimed to design a wireless game controller which is comfortable and at low prices.

The design of our controller includes 4 buttons and a joystick which allows us to control most of the games. By using the Arduino features, the communication between the computer keyboard and the controller was provided.The system design and Bluetooth connections were done. Game was tested and tried on the controller. The design of the controller we have so far is not comfortable as we aimed to have yet.

2 Literature Review

In the first example project, [KumarKumar2022] proposed a joystick controller which implemented with Arduino Leonardo. Project implementation is very simple since only Arduino IDE is sufficient to implement the project. ATMega32U4 microcontroller can be detected as mouse or keyboard on the computer. But it is not available in UNO boards, UNO boards are using ATMega328 microcontroller. Solution that was evaluated in the first example is a basic first step for designing a game controller, but wired communication was preferred and only one joystick was used. Therefore, limited number of games can be played . In our approach wireless controller was designed with four buttons and one joystick which has advantage of playing various games. Our project can be implemented with all Arduino boards since Pyhton serial port was used to simulate keyboard functions.

In the second example, [DeuskarDeuskar2022] proposed a method for designing a game controller that can be implemented with any type of the Arduino. Since the project is not restricted to ATMega32U4 microcontroller which is available in Arduino Leonardo or Micro, implementation can be made with Arduino UNO. ATMega32U4 microcontroller can be connected to a computer as a mouse or keyboard. Therefore, there is no need for an additional software for connection, but it is costly . Second example project can be implemented by using any Arduino which is good for the feasibility, but wired communication was used. Our proposed technique is wireless which eases of handling or storage and more flexible than wired solution. Example implementation can be seen in Figure 1.

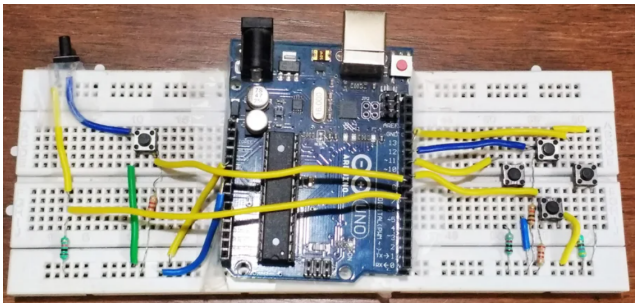


Figure 1: The Arduino Controller Project Example 2 [DeuskarDeuskar2022]

In the third example, [RajRaj2022] proposed a method for a game controller which was designed with Arduino UNO . Although external joystick was designed and implemented, increases the comfort for playing. Our implementation is more cost effective since it is based on soldering of the components on the perforated plate. Example implementation can be seen in Figure 2.

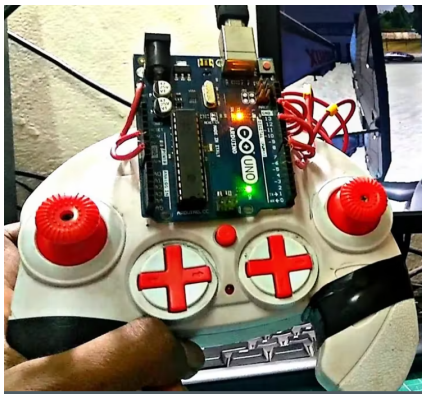


Figure 2: The Arduino Controller Project Example 3 [RajRaj2022]

3 Methodology

3.1 Joystick:

The purpose of the xy joystick module is to transmit 2-axis motion to the Arduino. This is achieved by hosting two independent 10K potentiometers, one on each axis. These potentiometers are used as dual adjustable voltage dividers providing 2-Axis analog input in control stick form. The basic working principle of a joystick is to translate the position of the stick on two axes - X-axis and Y-axis to electronic information that an Arduino can process. The way of construction may be a little difficult as an idea, but thanks to the joystick design consisting of two potentiometers and a Gimbal Mechanism, it is handled very easily.

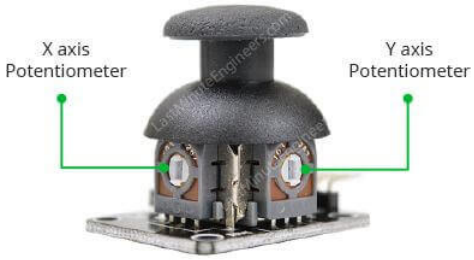


Figure 3: Joystick Module [lastminuteengineerslastminuteengineers2022]

To read the physical position of the joystick, we need to measure the change in resistance of a potentiometer. This change can be read by an Arduino analog pin using the ADC. Since the Arduino board has a 10-bit ADC resolution, the values on each analog channel can range from 0 to 1023. Therefore, if the bar is moved from one end to the other on the X-axis, the X values will change. A similar value movement from 0 to 1023 occurs when moved along the Y axis. The value is around 512 when the joystick remains in the middle position [lastminuteengineerslastminuteengineers2022].

3.2 Buttons and Pull Up Method:

When the button is not pressed, the input pin of the Arduino is at 5 volts. When the button is pressed, the current directly reaches ground instead of the Arduino's input pin. Arduino sees 0 when the button is pressed and 1 when the button is not pressed. The purpose of using a pull-up resistor is to prevent the ground and supply line from being directly short-circuited when the button is pressed. Pull-up resistors are usually 10K ohms.
In Arduino, the INPUT_PULLUP libary is used with pinMode(). It monitors the status of the button by establishing serial communication between Arduino and computer via USB.

3.3 Communication between Joystick and Computer:

In order to transfer the commands received from the joystick into the game, a bluetooth connection must be established. There should be bluetooth sensors on two arduinos for this. The reason for this is to make the joystick independent by creating a serial communication channel. Since we use Arduino UNO, the computer's bluetooth is not used for bilateral communication.

3.4 Serial Communications:

In order for the sent message to be received by the Bluetooth module, it must communicate in series with the Arduino. Serial communication between devices allows messages to be sent using a single connection. To establish a serial connection between two devices, the devices must share the baud rate at which data transmission will occur, each "number of data bits in". For Ardunio , serial communication uses the TTL protocol. Data is sent in eight-bit or one-byte chunks, as shown in Figure 4. The least significant bit of the byte (LSB) is sent first. The start bit and end bits are low (0 V) and there is no parity bit.

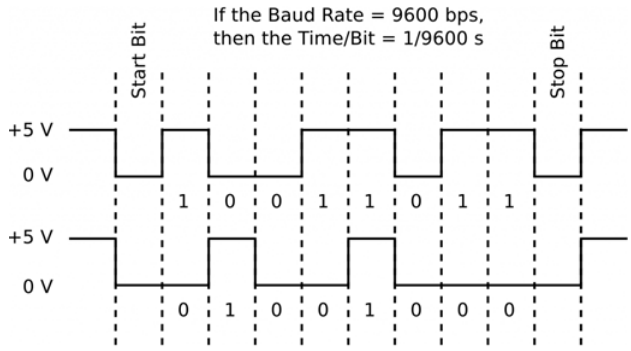


Figure 4: Example of What an Actual Arduino TTL [fiz ixfiz ix2022]

As shown in Figure 5 , attention was paid to the decoupling of the RX and TX pins between the Arduino and Bluetooth for serial communication. Arduino UNO has a TX (pin 1) pin and an RX (pin 0) pin. If the Arduino is connected to a device via USB, these TX and RX pins are disabled. The reason for this is that these pins are used for serial communication with the device. For this reason, virtual RX and TX pins were created for the Bluetooth module to work in all conditions. Virtual pins were created using the SoftwareSerial library. The purpose of this library was developed to allow serial communication on the designated digital pins of Arduino. Then the baud rate was set to 9600.

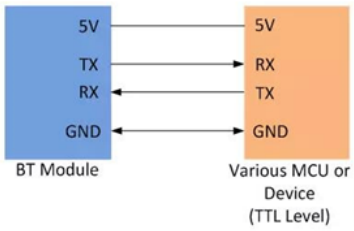


Figure 5: Decoupling of the RX and TX Pins [feasywififeasywifi2019]

3.5 Setting Up Bluetooth Modules:

In this project, the HC-05 Bluetooth module was preferred. This is because these modules can be used in both master and slave mode. If used in master mode, it can initiate communication. If the both devices are available, communication is maintained and modules are operating as a transreceiver.The bluetooth module added to Arduino on the joystick is set to master mode. The arduino on the receiver side is set to slave mode. The HC-05 module has a 2.4 GHz communication frequency and a synchronous speed of 1 MBps. As seen in the Figure 6, there is a KEY pin whose modes can be adjusted manually, RX and TX pins required for communication, and a STATE pin that determines the connection status. It also has voltage and ground pins [components101components1012018].

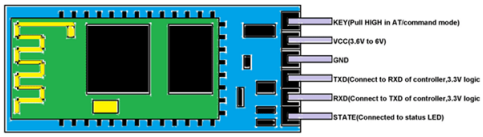


Figure 6: Schematic of HC-05 [components101components1012018]

3.6 Setting The Mode of The Bluetooth Module:

AT commands, short for ”ATtention”, are used to control a bluetooth module. The main AT commands are as follows:

- AT : Check the connection
- AT+NAME: To check the default name of the module
- AT+ADDR: Address of Bluetooth module
- AT+BAUD: To check the default baud rate of the Bluetooth module
- AT+ROLE: To obtain the mode of the module
- AT+NAME’DESIRED NAME’: Changes the default name of the module to the DESIRED NAME
- AT+CMODE: Set and Check connect mode
- AT+BIND: Fixed address

A voltage divider circuit must be installed in order to be able to access the settings of an HC-06 bluetooth module, namely the AT command window, without interruption.

3.7 Timer and Interrupts

Interrupt is quite fatal feature of any computational hardware. Briefly, when a device raises an interrupt request, which may be from hardware or software, the current program is being stopped, interrupted, and desired or dedicated action is being done. When interrupt routine finishes, the program continues from where it stopped [GeeksforGeeksGeeksforGeeks2022]. The switch button of our personal computers is a great example of interrupts. While operating system is running, if we press the button, the PC instantly stops doing what it was doing and shot itself down. This is what we want for our sampling process, but we need one more feature which is the timer.
A timer is a piece of hardware that basically works as a counter. There are different timers in a single chip for multiple and various usages, such as pulse width modulation, delays and so on. Also, we can program our Arduino to interrupt the program when the timer reaches our sampling period. There are different types of parameters to use for our purposes such as modes and Prescaler. In short, the timer ticks at every clock cycle when Prescaler is 1, but when it is 8, it ticks at every 8 cycle. So, we need to adjust our parameters to reach our desired sampling period without coming across counter overflow. When it reaches that level, we can make it to call an interrupt request.

3.8 Pulse-Code Modulation

There are several steps that are supposed to be applied to do Pulse-Code Modulation, which are Low pass filtering to reduce noise, sampling to make the signal discrete in time, quantize to digitize signal amplitudes, and encoding. After those steps, our signal became ready to send with a significant rate of efficiency and accuracy. At the receiver side, the signal is supposed to be decoded according to the encoding method, then applied Low-pass filter as a reconstruction filter. After these steps which are shown in Figure 7, we can observe the original signal with a relative amount of error according to parameters of modulation.

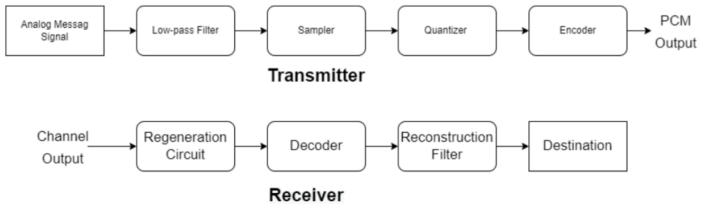


Figure 7: Pulse-Code Modulation Process

3.9 Game Design

Game design can be considered the planning arm of the entire process for making a video game. No video game gets made without a plan, and video game design is, more or less, the process of making that plan. The field is somewhat a hybridization of creativity and technical skills that combine into a cohesive, fleshed out idea that people can work with using concrete and actionable tasks. According to our Input Connection Test, our controller has almost 1000ms delay thus our game has to be slow and a player must do only 1 action at the time. So we designed a puzzle game to reach this goal. In this game player pushes blocks around, trying to get them to certain locations.

3.10 Game Art

Game art is the visual elements you see while playing a game so it is a very important component for the game to be understandable and interesting. We made our game map using the primitive mesh types given to us in Unity. And we used built-in material types in unity. We used a 3D design program called Blender to use meshes other than primitive meshes in Unity. With Blender, we made the animated cubes that we use frequently in the game. While making our basic cube, we first made the 3D modeling and then the skeleton system for the animations.Then we evenly distributed the weight vectors of the bones over the mesh. We used a vector design program called Illustrator for the textures we will use. Finally, we select a color palette so that the colors we use in the game create enough contrast and are pleasing to the eye.

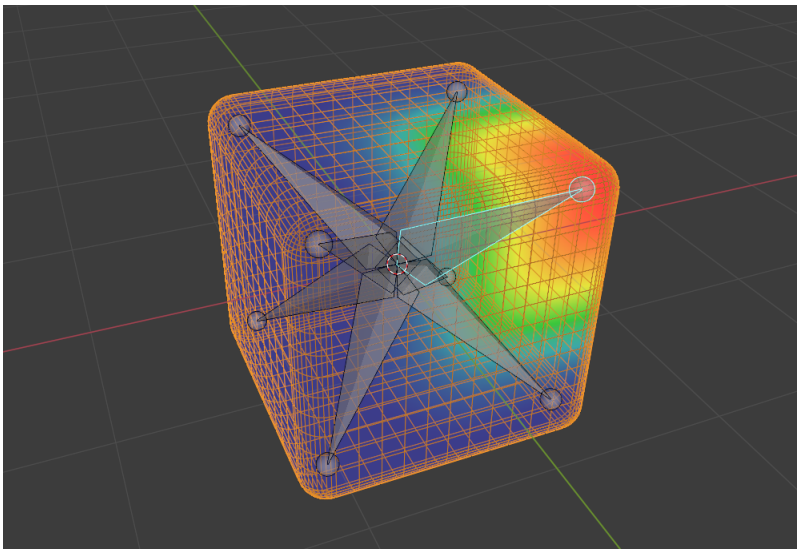


Figure 8: Animated Cube Mesh From Blender

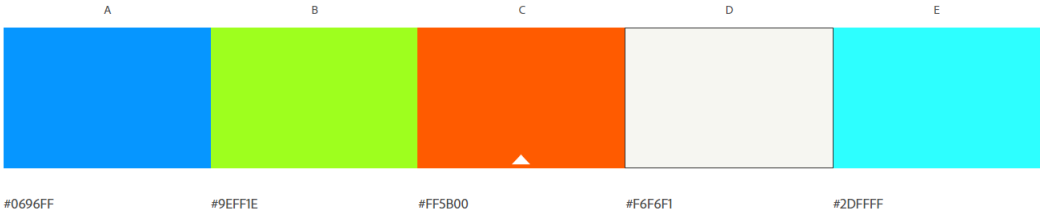


Figure 9: Color Palette

3.11 Game User Interface

A good game UI helps players navigate through the game, find the right information and complete challenges. This interface must be right down to the smallest detail. It should be visually appealing but above all functionally sound. It can be shown on screen space or world space.

3.12 Game Engine Configurations

Game engine configurations are important for having good performances on different devices. Draw call batching is one of the key elements of good performance. Draw call batching is a draw call optimization method that combines meshes so that Unity can render them in fewer draw calls. Unity provides two built-in draw call batching methods: static batching and dynamic batching. With static batching, Unity combines all static game objects and render them together. In dynamic batching, Unity transforms their vertices on the CPU, groups similar vertices together, and renders them in one draw call. We used static batching for the game board and dynamic batching for the blocks.

4 Experimental Results

4.1 Test 1 - Input devices test:

We checked the input devices, as seen in the Figure 10, if they work properly, and if Arduino captures the data on time.

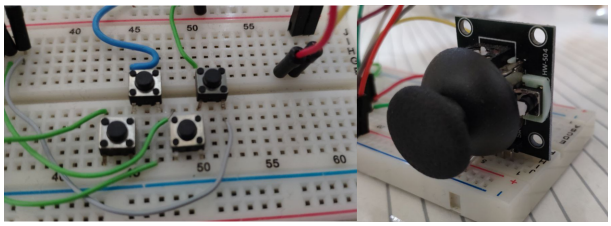


Figure 10: Buttons and joystick test

Joystick did not work as expected. Typical joystick has 2 axis potentiometer, so when we change its position, we should observe results just like Figure 11.

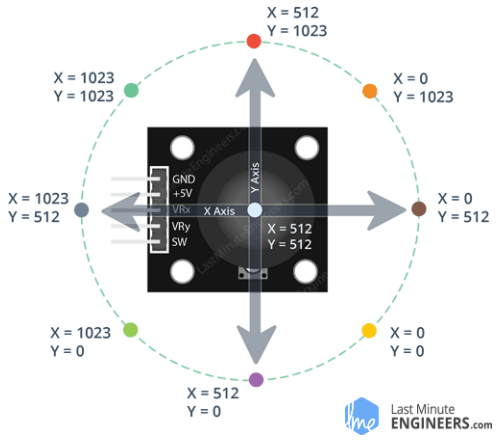


Figure 11: Joystick positions [lastminuteengineerslastminuteengineers2022]

However, our joystick did not work like that. Instead the x and y values were as seen in the Figure 12. Therefore, we manipulated the input signal with codding on Arduino.

X = 500 Y = 700	X = 350 Y = 500	X =< 100 Y =< 100
X = 800 Y = 1000	X = 500 Y = 500	X = 250 Y = 15
X = 1000 Y = 1000	X = 700 Y = 500	X = 500 Y = 50

Figure 12: Real position values of the joystick

On the other hand, the buttons worked properly with no error.

4.2 Test 2 - Bluetooth Test:

We tested Bluetooth modules by first setting their modes to connect each other automatically. Thereby in AT command mode, we did the adjustments for each module. Then, we both connected the modules for different Arduino boards, as seen in the Figure 13, and tested if we can send and receive message for both sides.

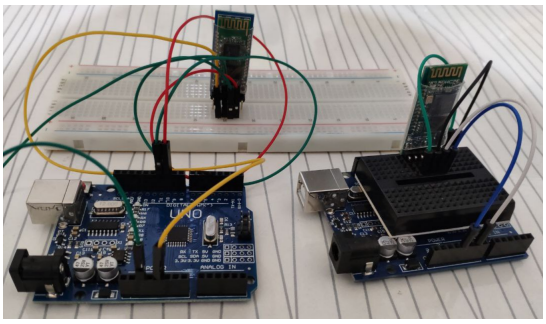


Figure 13: Bluetooth test set up

As a result of the tests, we saw that the Bluetooth modules can only connects with each other and the baud rates

are the same. With this configuration, we can easily connect the modules and start the communication rapidly.

4.3 Test 3 - Communication Test:

In this project, we tried to replicate Pulse Code Modulation as seen in the Figure 14. We first needed to sample the input data. In default set of Arduino, sampling is being done inherently, but it not reliable because it may change due to some random delays. To make sampling precise and constant, we used Timer and Interrupt feature of Arduino. As a result, we found out that with 256 prescaler value and 2000 compare match value, which causes sampling period of 64ms, we could establish fastest reliable madulation also compatible for bluetooth module.

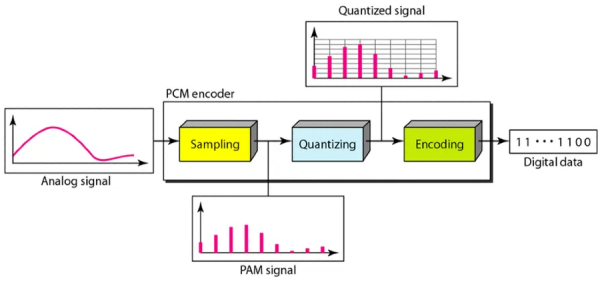


Figure 14: Pulse Code Modulation [SpeakerformSpeakerform2022]

Then, we needed to quantize the input signals. Since, button inputs are already digital, we did not have to do anything additional. On the other hand, there are 1024*1024 values for joystick, so we aimed to quantize it into 9 levels which represents every distinct directions.

Lastly, we prefered to do binary encoding. So, for 5 buttons, we assigned 5 bits, and for 9 level of joystick we assigned 4 bits. Thus, we ended up with 9 bit symbols for every transmission.

4.4 Test 4 - Communication between the controller and computer:

We wanted to replicate keyboard keystrokes with the controller. Therefore, we wanted to build a system that since the receive Arduino is connected to the computer, it establish serial communication on COM port. Than, another software captures the data and replicates keystrokes. Some Arduino models can do it without any external software, such as Leonardo, Zero and Due [ArduinoArduino2022]. However, Arduino Uno is not capable for this purpose. Hence, we used python to do it for us with pyserial and pyautogui libraries. As a result, we saw that by pressing the buttons and changing the position of the joystick, we can press different keys of the keyboard. However, we there was a bit delay in the communication due to additional process we added to the communication.

4.5 Test 5 - Game performance test:

We have tested the game on several devices both on mobile and desktop.The game was very smooth on every devices we have tested. The average frames per seconds are shown on the table below.

Device	Device Screen Frame Rate	Average FPS
iPhone 6	60	60
iPhone 7	60	60
iPhone 12	60	60
Xiaomi Poco X3 Pro	120	120
Realme 7 Pro	60	60
Xiaomi Redmi Note 7	60	60

Figure 15: Average FPS results on mobile devices

Device	Average FPS
ASUS Rog 512LWS	875
Monster Abra a5 v12.1	405
Monster Abra a5 v11.1	424
Dell 7567	673
Macbook Air 2020	398

Figure 16: Average FPS results on desktop devices

4.6 Test 6 - Input connection test:

We have tested the game with a keyboard and our controller. Keyboard has 10ms delay but our controller has almost 1000 ms delay. Our game is designed to overcome this problem, so it does not affect the gameplay much.

4.7 Test 7 - Art test:

We showed the current state of the game to 27 people. 24 of them said that the game has enough graphics compared to a puzzle game, and except for one person, everyone said that the game was designed in a way that could be understood at first glance.

4.8 Test 8 - Level generator test:

We wrote a level generator script so that the game can generate its own levels. We have verified that the levels produced by the script are sufficient in terms of logic and design.

4.9 Test 9 - Playtest:

We had the game tested by a sample group. No bugs affecting gameplay were encountered. The game was understandable by the players.

5 Final Product

Controller circuit was built and soldered on the perforated plate to increase durability of the controller. To achieve desired design constraints and to increase safety, circuit was put inside a plastic box. Solidworks design software was used to sketch desired pattern on the upper side of the box. Finally, holes was drilled on the box by using CNC machine and necessary mountings were completed. Design steps of the controller can be explained as follows.

Firstly, controller circuit was designed and set up on the breadboard. Necessary testes were conducted and it was observed that controller was working properly. Since building circuit on a breadboard is not an durable solution, same circuit was built and soldered on the perforated plate. Soldered circuit can be seen in Figure 17.

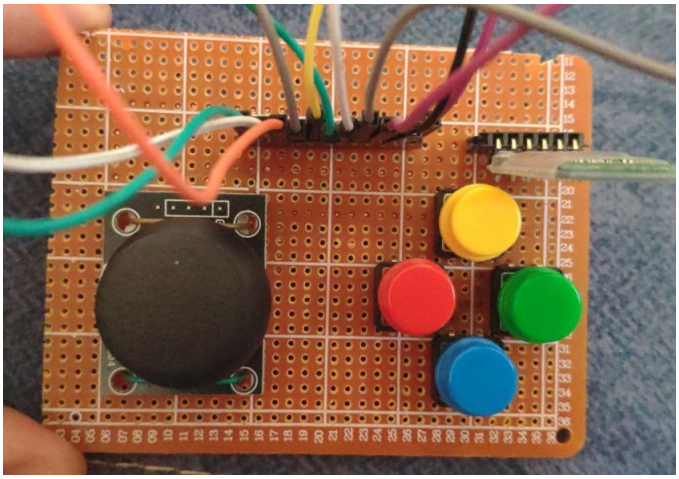


Figure 17: Perforated Plate Soldered Circuit

We aimed to provide comfortable playing while conducting this project. Therefore, it is necessary to put circuit together in some plastic box. In addition to provide comfort, we protected the components and achieve better visualization by placing the circuit into the box. Solidworks software was used to sketch patterns for drilling. After the box was selected, it was drilled according to sketch by using the CNC machine. Drilled box can be seen in Figure 18.

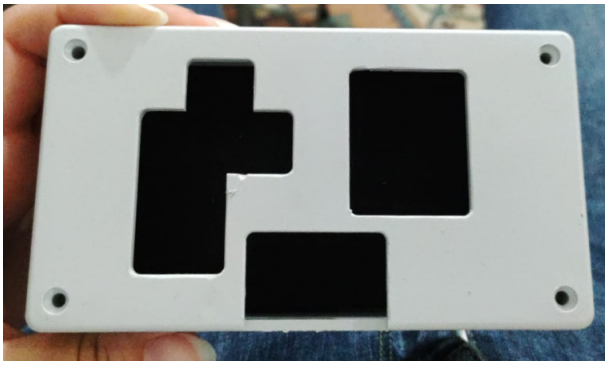


Figure 18: Drilled Box

Finally, necessary mountings were done into the box. Sponge was used to provide isolation between Arduino and circuit to prevent short circuiting that may occur. Also, a battery was placed inside the box to feed the Arduino, thus powering the overall controller circuit. Final product can be seen in Figure 19.

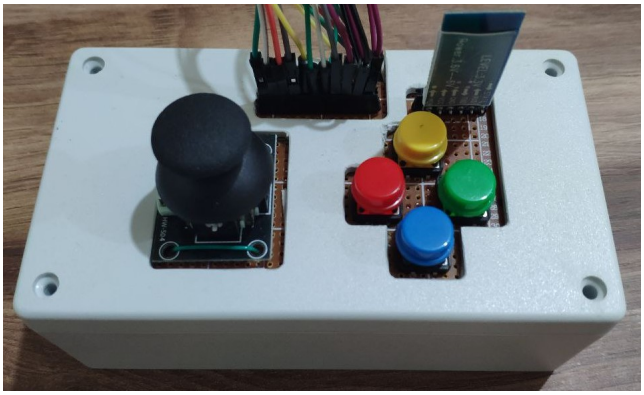


Figure 19: Final Product

6 Conclusions and Future Works

In this project, a wireless game controller was designed by using several electronic components and a game software. Arduino UNO was used as a processor. HC-06 Bluetooth modules were used to provide the wireless communication. Unity 2020.3.30 was used for the game and Blender 2.8.3 was used for 3D models. Necessary tests were performed in both electronic and software parts like the Bluetooth communication, the working mechanism of the controller on a real game, etc.

The future works were done as they planned to. The components were soldered on the perforated plate to provide the stability of the components. The design of the controller was improved by using a specially cut plastic box. So, the play comfort was improved. Performance of the controller was upgraded by making several changes in the coding. Game design was improved by improving the visualization.

The need of inexpensive and useful controller for players make this project valuable for us. Because of the changing economy in our country, the basic electronic devices are getting harder to buy. Especially for the beginner players, our controller offers a beginning with its main and necessary features for a game before spending tons of money for just a controller. Also, for the players who are interested in embedded systems and electronic devices, our documents carry the value of being a guide. It allows you to make your own controller by yourself.

To conclude, the schedule of the project was followed. Planned tasks were done at the planned times and future plans were realized.

7 Weekly Schedule/Project Plan

Weekly project schedule in a Gantt Chart as seen in the figure 20.

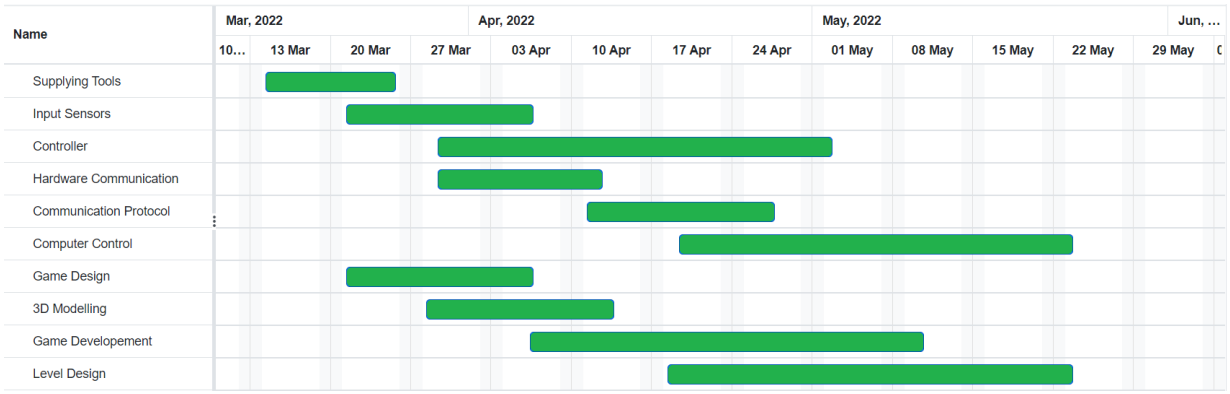


Figure 20: Gantt Chart

- Week 1 (02.03.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Created.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Created.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Created.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Created.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Created.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Created.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Created.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Created.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Created.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Created.
- Week 2 (09.03.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: In Progress. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Created. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Assigned. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Assigned. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Created. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Assigned. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Assigned. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Assigned. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.
- Week 3 (16.03.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.

Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: In progress. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
Task Name: Controller. Definition: Building and testing the game controller. Current status: Assigned. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Assigned. Responsible Person: Berke Eren, Ayşenur Karaca.
Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Assigned. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Assigned. Responsible Person: Emre Hepsağ, Berke Eren.
Task Name: Game design. Definition: Designing a convenient game for controller . Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Assigned. Responsible Person: Gazi Özgen.
Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Assigned. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 4 (23.03.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: In progress. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Assigned. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Assigned. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Assigned. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Assigned. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Assigned. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 5 (30.03.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: In progress. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: In progress. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Assigned. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Assigned. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: In progress. Responsible Person: Gazi Özgen.

Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Assigned. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 6 (06.04.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: In progress. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Assigned. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Assigned. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: In progress. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: In progress. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 7 (13.04.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: In progress. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: In progress. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: In progress. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Assigned. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 8 (20.04.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards.

Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: In progress. Responsible Person: Emre Hepsağ, Berke Eren.
Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: In progress. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 9 (27.04.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: In progress. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: In progress. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: In progress. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 10 (04.05.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Completed. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: In progress. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: In progress. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 11 (11.05.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Completed. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: In progress. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Completed. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş.
- Week 12 (18.05.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Completed. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Completed. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Completed. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: In progress. Responsible Person: Gazi Özgen, Batuhan Akbaş.
- Week 13 (25.05.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Completed. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Completed. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed.

Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Completed. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş.

- Week 14 (1.06.2022):
  Task Name: Supplying tools . Definition: Ordering and finding tools, devices we need. Current status: Completed. Responsible Person: Bengisu Sağlam, Berke Eren.
  Task Name: Input Sensors. Definition: Testing and controlling input sensors. Current status: Completed. Responsible Person: Bengisu Sağlam, Ayşenur Karaca.
  Task Name: Controller. Definition: Building and testing the game controller. Current status: Completed. Responsible Person: Emre Hepsağ, Ayşenur Karaca.
  Task Name: Hardware Communication. Definition: Building a Bluetooth Communication System for boards. Current status: Completed. Responsible Person: Berke Eren, Ayşenur Karaca.
  Task Name: Communication Protocol. Definition: Adjusting sampling rate and data precision. Current status: Completed. Responsible Person: Emre Hepsağ, Bengisu Sağlam.
  Task Name: Computer Control. Definition: Controlling computer with Arduino connected to PC. Current status: Completed. Responsible Person: Emre Hepsağ, Berke Eren.
  Task Name: Game design. Definition: Designing a convenient game for controller . Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş, Emre Korkmaz.
  Task Name: 3D modelling. Definition: Creating 3D models for the game, creating a color palette. Current status: Completed. Responsible Person: Gazi Özgen.
  Task Name: Game development. Definition: Defining the object oriented model, creating prefabs, creating player movements, building the game logic, programming a character life-cycle. Current status: Completed. Responsible Person: Batuhan Akbaş, Emre Korkmaz.
  Task Name: Level design. Definition: Designing levels for the game, improving the user experience, creating the maps. Current status: Completed. Responsible Person: Gazi Özgen, Batuhan Akbaş.

References

[ArduinoArduino2022] Arduino (2022). Keyboard.

[components101components1012018] components101 (2018). Hc-06 bluetooth module.

[DeuskarDeuskar2022] Deuskar, T. (2022). Video game controller with arduino.

[feasywififeasywifi2019] feasywifi (2019). kablosuz İletişimde tx ve rx nedir?

[fiz ixfiz ix2022] fiz ix (2022). introduction to serial communication.

[GeeksforGeeksGeeksforGeeks2022] GeeksforGeeks (2022). Interrupts.

[KumarKumar2022] Kumar, P. (2022). Joystick game controller using arduino leonardo.

[lastminuteengineerslastminuteengineers2022] lastminuteengineers (2022). how 2-axis joystick works interface with arduino + processing.

[RajRaj2022] Raj, P. (2022). Arduino game controller for pc and android.

[SpeakerformSpeakerform2022] Speakerform (2022). what is pcm audio? pulse code modulation (2022).