POST-STROKE ANALYSIS USING INTERACTIVE REHABILITATION

A PROJECT REPORT

Submitted by

N. BURUHANUDEEN -113118121005

D.GIRIPRASATH -113118121009

S. SURESH -113118121036

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
IN
BIOMEDICAL ENGINEERING

VEL TECH MULTI TECH Dr. RANGARAJAN Dr. SAKUNTHALA ENGINEERING COLLEGE, AVADI, CHENNAI

ANNA UNIVERSITY: CHENNAI- 600 025

JUNE-2022

BONAFIDE CERTIFICATE

Certified that this project report **POST-STROKE ANALYSIS USING INTERACTIVE REHABILITATION** is the bonafide work of**, BURUHANUDEEN.N, D.GIRIPRASATH**, and **S.SURESH**, who carried out the project work under my supervision.

SIGNATURE
Dr.G. Sasi M.E, Ph.D.
Professor
HEAD OF THE DEPARTMENT
Department of Biomedical Engineering
Vel Tech Multi Tech Dr.Rangarajan
Dr.Sakunthala Engineering College,
Avadi,Chennai-600062

SIGNATURE
Ms A.S Nivetha ,M.E
Assistant Professor
SUPERVISOR
Department of Biomedical Engineering
Vel Tech Multi Tech Dr.Rangarajan
Dr.Sakunthala Engineering College,
Avadi,Chennai-600062

This project report was submitted for viva voice held on_____at Vel Tech Multi Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi.

Signature of Examiners:

Internal Examiner **External Examiner**

Date : ……

ACKNOWLEDGEMENT

We wish to express our sincere thanks to almighty and the people who extended their help during the course of our project work.

We are greatly and profoundly thankful to our honourable Chairman **Colonel.Prof.Dr.Vel.R.Rangarajan B.E. (EEE), B.E. (MECH), M.S. (AUTO)** and **Vice Chairman Dr.Sakunthala Rangarajan, MBBS** for facilitating us with this opportunity. We also record our sincere thanks to our honourable **Principal Dr.V.Rajamani** for his kind support to take up this project.

We are thankful and extremely grateful to our **Head of the Department Dr.G. Sasi M.E., Ph.D.**, Professor, Biomedical Engineering Department for his moral support.

We take this opportunity to express our profound gratitude and deep regard to our internal guide **Ms A.S Nivetha ,M.E Assistant professor** and to our project coordinator **S. Sathish, M.TECH Assistant professor**, Department of Biomedical Engineering for their constant technical support, valuable suggestions and stupendous encouragement toward the completion of our project and guiding us throughout the work.

Further the acknowledgement would be incomplete if we would not mention a word if thanks to our most beloved Parents whose continuous support and encouragement all the way through the course has led us to pursue the degree and confidently complete the project.

N.BURUHANUDEEN
D.GIRIPRASATH
S.SURESH

ABSTRACT

Stroke is one of the leading causes of disability. The majority of survivors have mobility abnormalities, and many have persistent dexterity impairments. Patients can restore many of their lost functions with post-stroke rehabilitation therapy. Game-based methodologies have been proposed to keep patients motivated and achieve therapeutic goals during stroke rehabilitation therapy. We've created a movable actuated glove orthosis with a flex sensor that detects the fingers' flexing ability. By use of this interactive rehabilitation using a game which improves the upper extremity control and task performance. Unity 3D has been updated to include a hand-training game. The goal of this study is to see if activities that fostered more mobility, were more entertaining, and were less active may help patients with hand disabilities. It is a portable and cost-effective haptic controller system for use in games. The use of game-based rehabilitation led to an overall improvement in post-exercise outcomes.

Keywords: Post- Stroke, Rehabilitation, Treatment, Flex sensor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Stroke:

A stroke happens when a blood artery in the brain ruptures and bleeds, or when the blood supply to the brain is cut off. The rupture or blockage prevents blood and oxygen from reaching the tissues of the brain.

Motor disability in the hand is frequent following a stroke. One-third of persons acquire wrist and hand contracture (loss of passive joint range of motion) 6 months after a major stroke, and more than half of those with hand impairments do not regain function. Hand deficits in chronic stroke, for example, are associated with a diminished capacity to regulate voluntary muscle activity and aberrant activation of contralateral cortico-reticulospinal pathways.

1.2 Risk Factors of Stroke:

- **High blood pressure:**

  Blood vessels (arteries) that feed blood to the brain can be damaged by blood pressure of 140/90 or greater. It's also a major cause of stroke. It happens when the blood pressure in your arteries and other blood vessels is too high. High blood pressure frequently has no symptoms.

- **Heart disease:**

  Heart disease is the second leading cause of stroke and the leading cause of mortality among stroke survivors. Many of the risk factors for heart disease and stroke are the same.

  Other cardiac disorders, such as heart valve anomalies, irregular heartbeat (including atrial fibrillation), and enlarged heart chambers, can cause blood clots to form and trigger a stroke.

- **Diabetes:**

  Diabetes mellitus increases the risk of stroke, particularly strokes caused by damage to tiny blood arteries. An HbA1c of 7% or a fasting blood sugar of 80-120 mg/dL is the normal aim for management.

- **Smoking and Alcohol:**

  Smoking almost doubles your risk for an ischemic stroke. Alcohol (more than one drink per day) is associated with stroke risk. Limit the amount of alcohol you drink.

- **Illegal drug use:**

  Street drugs including crack, cocaine, and marijuana have been linked to an increased risk of stroke. Some of these medicines have a direct effect on the blood arteries in the brain, resulting in a stroke.

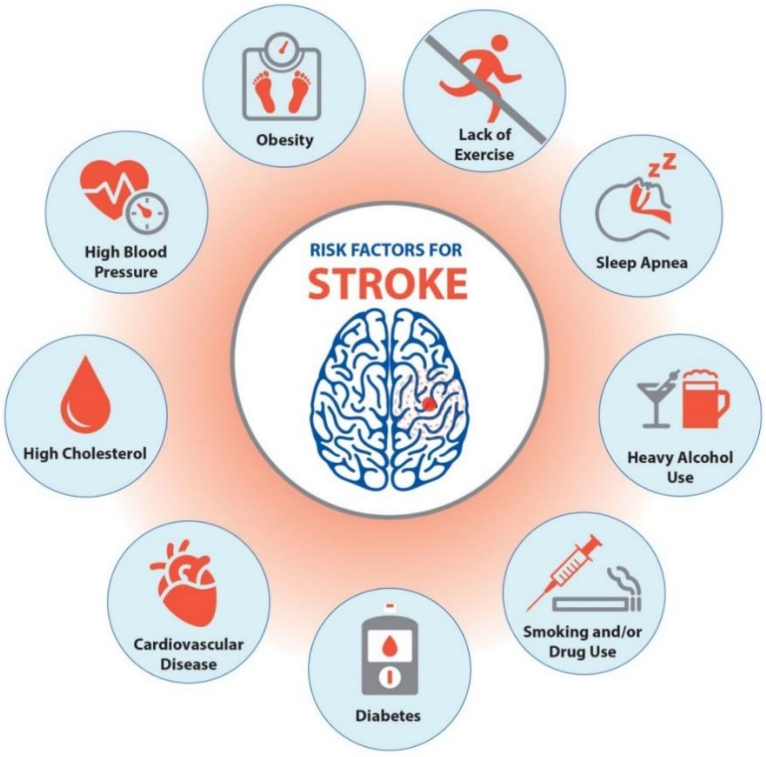


Figure 1- Risk Factors of Stroke

1.3 Symptoms of Stroke:

If a stroke is suspected, a rapid protocol known as FAST can be used to confirm it:

- **Face:** The individual should be advised to attempt smiling. A stroke should be detected if one corner of the lips droops to one side when smiling.

- **Arms:** The subject should be instructed to lift both arms over their heads. If a person is unable to lift one arm or if one arm begins to droop to one side or slides downward, a stroke should be considered.

- **Speech:** Ask the person to repeat simple words or phrases. Slurred speech or a change in speaking style should highlight the possibility of a stroke.

- **Time:** If any of these indicators are seen, it is critical to act quickly. Urgent medical attention should be sought. Particular attention should be paid to the period when symptoms first arise.

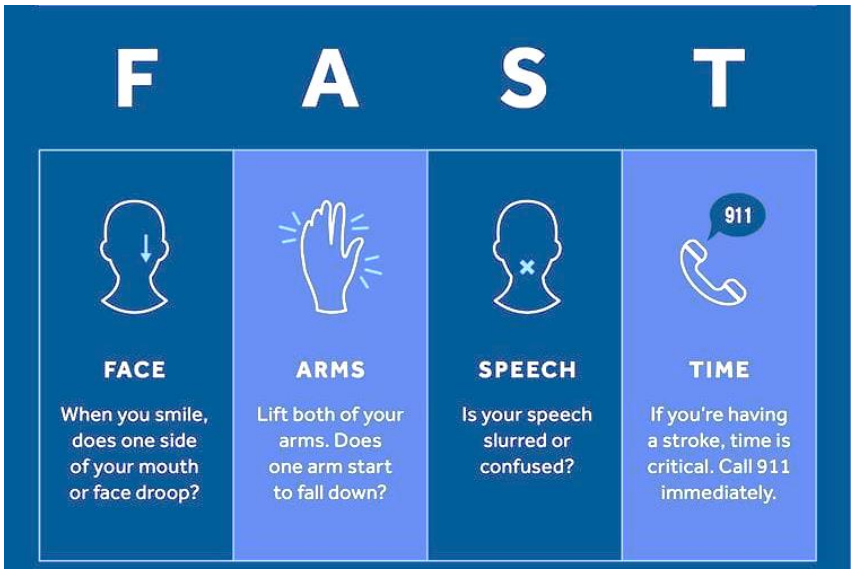


Figure 2: Symptoms of Stroke

1.4 Post stroke:

The term "post stroke patient" refers to someone who has experienced or is experiencing depression as a result of a stroke. Because they have different impairment pathways, stroke survivors with more reduced motor capacities can be challenging to treat.

Motoneuronal hyperexcitability, exhibited as spasticity, increased coactivation, or extended muscle relaxation time, is likely to affect them. They also show significant weakness as a result of motoneuronal activation abnormalities and a restricted ability to control activation patterns properly.

1.5 Symptoms of Post stroke:

1. The patients' hands are impaired.
2. Weakness, paralysis, and balance or coordination issues.
3. Numbness, pain, or tingling and burning sensations.
4. Fatigue, which may persist once you go home.
5. Neglect or inattention to one side of the body; in extreme circumstances, you may be unaware of your arm or leg.

1.6 Treatment of post stroke:

Proper medical evaluation and prompt treatment are vital to recover from stroke.

Treatment of stroke depends on the type of stroke.

- Clot breaking drugs:
- Mechanical thrombectomy
- Stents
- Surgery
- Medications

1.7 Recovering from stroke:

Because stroke is a long-term impairment, it is critical that healing and therapy begin as soon as feasible.

A medical team can stabilize your condition and assess the impact of a stroke in a hospital. They can help you discover issues and begin therapy to help you restore some of the skills that have been lost.

1.8 Rehabilitation:

The integrated and coordinated use of medical, social, educational, and vocational measures focuses on the existing capacities of the stroke afflicted patient and takes him to the optimum level of his or her functional capacity. It makes a person's life more meaningful and productive, hence lengthening their years.

Because of its originality and comprehensiveness, rehabilitation is growing in popularity and demand around the world. The ultimate goal of critical illness care is to restore the patient to a functioning level where they can achieve quality of life and quality adjusted life years.

There are many types of rehabilitation therapy, each designed to address specific ranges of issues.

- Physical Therapy – This therapy works to improve movement dysfunction. Therapists work with patients to restore movement, strength, stability and/or functional ability
- Occupational Therapy – This form of therapy focuses on restoring an individual's ability to perform necessary daily activities. This may mean working to improve fine motor skills, restore balance, or assist patients in learning how to increase their functional ability

1.9 Benefits of Rehabilitation:

Rehabilitation can reduce the impact of a broad range of health conditions, including diseases (acute or chronic), illnesses or injuries.

It can also complement other health interventions, such as medical and surgical interventions, helping to achieve the best outcome possible.

For example, rehabilitation can help to reduce, manage or prevent complications associated with many health conditions, such as spinal cord injury, stroke, or a fracture.

Rehabilitation helps to minimize or slow down the disabling effects of chronic health conditions, such as cardiovascular disease, cancer and diabetes by equipping people with self-management strategies and the assistive products they require, or by addressing pain or other complications.

Rehabilitation is an investment, with cost benefits for both the individuals and society.

It can help to avoid costly hospitalization, reduce hospital length of stay, and prevent re-admissions.

Rehabilitation also enables individuals to participate in education and gainful employment, remain independent at home, and minimize the need for financial or caregiver support.

1.10 Post- stroke rehabilitation:

Rehabilitation helps someone who has had a stroke relearn skills that are suddenly lost when part of the brain is damaged.

Equally important in rehabilitation is to protect the individual from developing new medical problems, including pneumonia, urinary tract infections, injury due to fall, or a clot formation in large veins.

The goals of rehabilitation are to optimize how the person functions after a stroke and the level of independence, and to achieve the best possible quality of life.

Rehabilitation also teaches new ways to compensate for any remaining disabilities.

For example, one might need to learn how to bathe and dress using only one hand, or how to communicate effectively with assistive devices if the ability to use language has been affected.

1.11Rehabilitation techniques:

Mental Exercise

Another stroke recovery therapy option is mental practice, which is suitable for people with extremely limited movement or paralysis. It entails mentally imagining oneself moving. You can imagine yourself completing your rehab activities or moving in different ways, such as walking through a lawn or along a sandy beach.

Mental practice is easy, non-invasive, free, and widely available. It's an excellent supplement to any stroke recovery programmed, especially for those recovering from paralysis.

Mirror therapy

Mental practice is another stroke recovery therapy option that is appropriate for patients who have severely limited movement or paralysis.

It entails visualizing oneself moving in one's mind. You can visualize yourself finishing your rehab activities or moving in various ways, such as walking through a grass or along a sandy beach.

It is simple, non-invasive, free, and generally available to engage in mental practice. It's a great addition to any stroke recovery programme, especially for those who have been paralysed.

Constraint-Induced Movement Therapy (CIMT)

Constraint-Induced Movement Therapy (CIMT) is a high-intensity stroke rehabilitation treatment. It involves restricting your unaffected side and forcing you to use your afflicted side.

It's also worth noting that CIMT can help people who have a hard time ignoring their wounded arm. CIMT is, in fact, a more effective method of minimising taught nonuse.

Figure 3: Rehabilitation Techniques

Electrical stimulation

Electrical stimulation is a rehabilitation treatment that can help with severe movement issues and even post-stroke paralysis. It works by placing electrodes on the skin and sending electrical impulses to the muscles that are afflicted

CHAPTER 2

LITERATURE REVIEW

1.Department of Computer Science, AIS-Lab Laboratory of Applied Intelligent Systems, Università degli Studi di Milano, 20133 Milan, Italy; jacopo.essenziale@unimi.it (J.E.); renato.mainetti@unimi.it (R.M.)

"Hand Rehabilitation and Telemonitoring through Smart Toys"

The hand, both morphologically and functionally, represents one of the main elements that characterize humankind and serves as the main tool by which to interact with the environment in daily life. High finger mobility and opposable thumbs allow us to pinch, grab, manipulate, and interact with objects, enabling us to effectively explore the world, proving to be an essential element of the learning process throughout our life cycle, from childhood to old age. However, to achieve this goal, a long rehabilitation period is required. Rehabilitation is also required, although for a shorter time, for those children who, after domestic or outdoor accidents, suffer from injuries to their hands (from fractures to finger amputation), requiring clinical intervention.

In conclusion, we provide evidence that patients in the chronic phase following stroke can reduce hand impairment and improve finger dexterity. This was observed by direct quantitative measurement of dexterity following training on a multiday skill task focused on moving fingers against abnormal synergies without allowing other compensatory movements (i.e., from proximal upper extremity or trunk). Since this is a single cohort group, we cannot determine whether this effect was purely the result of the training paradigm or other nonspecific effect, such as passage of time or research participation.

2.Department of Computer Science, California State University, Fullerton, Fullerton, CA, USA "Designing a Unity Game Using the Haptic Feedback Gloves, VMG 30 Plus"

At its most basic level, Virtual Reality (VR) can simply be a simulation of a virtual environment that is experienced through a user's sense of vision. In such a setup, users can only passively experience the environment through keyboards, mice, joysticks, and game controllers, but not directly interact with it. This requirement can break the user's sense of "presence" within the virtual environment by compartmentalizing complex, dynamic, multifaceted behaviors into a short series of button presses.

It takes a step towards that direction by implementing a proof-of-concept game allowing users, with the help of smart gloves equipped with motion sensors, to interact with the game environment.

3.Jacques Foottit, Dave Brown, Stefan Marks and Andy M. Connor Auckland University of Technology, Auckland, New Zealand "A WEARABLE HAPTIC GAME CONTROLLER"

There are two significant developments that have arisen which have done a great deal to influence how people interact with machines. One of these developments comes from the area of sociology, where the Actor-Network Theory has been challenging the divide between the social and the technical. The other significant development is the rise of wearable technology to the point where it has left the realm of high-tech labs and made inroads into classrooms, fashion, arts, and hobbyist garages. With this definition in mind, a wearable haptic feedback device can be viewed as an orthotic that adds the capability of interaction with machines that is not present naturally.

The capability it adds is the ability to communicate with machines using gesture and to sense the virtual world with tactile sensations.

In this paper a proof of concept of core game mechanics for glove integration. As a result for performing this work, this prototype can demonstrate the use of gloves in a virtual environment. Since this game is a first of its kind, it only performs fundamental actions such as punching, opening doors, activating/deactivating switches, and knocking objects around.

4.E.KEERTHIKA, MUHAMMADU SATHIK RAJA Department of Medical Electronics Sengunthar college of Engineering "HAND THERAPIST: A REHABILITATION APPROACH BASED ON WEARABLE TECHNOLOGY AND VIDEO GAMING"

In this paper the wired process has been made with the interfacing module. So that data flow cannot lose due to the connectivity problem. Here two flexibility sensors are fixed in the two hand fingers to identify the force associated with it and accelerometer sensor is fixed in the hand to detect the movement of it. All the sensors are interfaced with the microcontroller through the amplifier and interfacing circuit. Microcontroller is already programmed for the task needed for the proposed work. From where PC is interfaced with it through the RS232 serial communication.

In PC we can easily monitor the whole hand movements and force associated with it. And also, in PC video game is created in Unity 3D for hand training in which the user must grab, hold, transport and drops a cube in several increasingly difficult puzzle levels.

This has the potential to improve the immersion of virtual experiences, as well as reducing errors and improving input speed for a range of tasks. There is even the potential for this technology to allow for new methods of learning that are yet to be discovered.

5.Judith E. Deutsch, Brittany Hoehlein, Marisa Priolo, Joshua Pacifico Rivers Lab, Dept. of Rehab & Movt Sci SHP, Rutgers University Newark, NJ. "Custom game paced video games played by persons post-stroke have comparable exercise intensity but higher accuracy, greater enjoyment and less effort than off-the-shelf game"

Active video games, also called serious games, have been shown to improve motor performance of persons post-stroke. These games may be customized for rehabilitation or adapted from their original purpose of recreation and applied therapeutically. It has been shown that recreational games that are bundled with commercially available game consoles such as the Nintendo Wii and Microsoft Kinect may address body function and structure, and activities as well as provide elements of motor learning such as knowledge of performance and knowledge of results.

The principle of the development of science is that "nothing is impossible". So we shall look forward to a bright & sophisticated world.

6.Paul Tamayo-Serrano, Samir Garbaya, Saida Bouakaz, and Pierre Blazevic IEE SYSTEMS"A Game-Based Rehabilitation Therapy for Post"

Based on the study of the previous research, this article presents a new approach for the rehabilitation of poststroke patients. This approach consists of using gamification to increase patients' motivation for and engagement with the rehabilitation program. It includes virtual exercises based on methods validated at the hospital by physiotherapists and practiced by patients at their homes. Playing a custom game paced video game consisting of stepping and marching movements met the recommended guidelines for moderate activity exercise that were comparable to those executed when playing an off-the-shelf game.

While the stepping frequency was greater for the off-the-shelf game, this came at the cost of lower accuracy. These findings support the use of custom video games over off-the-shelf games for promotion of physical activity while maintaining desirable movement accuracy.

7.Heidi C. Fischer, Kristen M. Triandafilou, Kelly O. Thielbar, José M. Ochoa, Emily C. Lazzaro, Kathleen A. Pacholski, and Derek G. Kamper, Member, IEEE "Use of a Portable Assistive Glove to Facilitate Rehabilitation in Stroke Survivors with Severe Hand Impairment"

While numerous devices have been developed to assist hand rehabilitation, for a review of exoskeletons see, the X-Glove is one of the few to independently actuate each digit while also allowing free movement of each joint, portability, and interaction with real objects. Cables serving as external extensor tendons run through cable guides attached to the dorsal side of a modified batting or driving glove . The guides form a bridge over each joint to allow joint flexion but to prevent joint hyperextension.

Our results suggest that use of an actuated device, which can be incorporated directly into clinical therapy, may be beneficial for facilitating rehabilitation. Stroke survivors with severe hand impairment may benefit from a combination of passive cyclical stretching and targeted assistance of active movements. This paradigm affords participants and their therapists the opportunity to maximize rehabilitation of motor control by providing skilled, task-oriented therapy in the clinic that may not otherwise be attempted for stroke survivors with this level of impairment.

2.2 Disadvantages Of Existing Methods:

A long rehabilitation period is required. Rehabilitation is also required, although for a shorter time, for those children who, after domestic or outdoor accidents, suffer from injuries to their hands (from fractures to finger amputation), requiring clinical intervention. [1]

It only performs fundamental actions such as punching, opening doors, activating/deactivating switches, and knocking objects around.

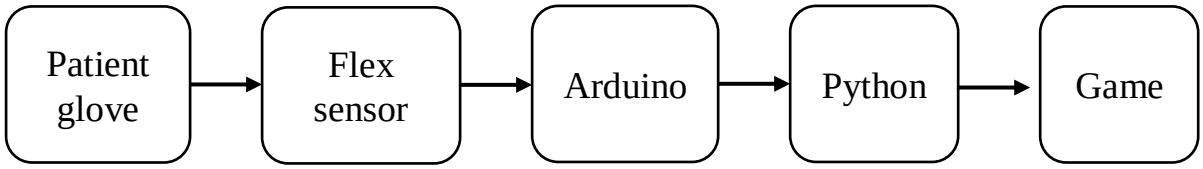
There are limitations to gesture approaches in that they lack the ability to provide any form of haptic feedback to the user and therefore do not necessarily provide any higher degree of engagement than traditional interfaces. [3]

The wired process has been made with the interfacing module. So that data flow cannot lose due to the connectivity problem. [4]

CHAPTER 3
MATERIALS AND METHODOLOGY

3.1 Block Diagram:

Our project's block diagram is presented below. It consists of a glove, a flex sensor, an Arduino UNO, Python, and a Unity-developed game. On the next pages, we'll go through each block in detail. The patient wearing the glove is the first block. The hardware (Arduino) is connected to the game using Python.



BLOCK DIAGRAM OF THE ORTHOSIS GLOVE

3.1.1. Patient glove:

It is the glove the used to identify the flexion and extension of the finger movements of post-stroke patients. The glove is attached with flex sensor to take the input form the user.

3.1.2 Flex Sensor:

A flex sensor is a device that detects the amount of bending or deflection. It takes the input of user to Arduino.

3.1.3 Arduino:

It was mainly used for the conversion of analog readings to the digital input

3.1.4 Python:

It was need to run the game based on the movements of finger of the user. It uses some python packages to run without the error

3.1.5 Game:

The game was developed in the unity editor software. It was basically jumping game by pressing the space key button**.**

3.2 Hardware:

It includes an Arduino Uno, a flex sensor USB cable, a breadboard, and wiring. In the next pages, we will go through each product in great detail.

3.2.1 Arduino Uno

The purpose of Arduino is to provide a simple way for software developers to learn how to program microcontrollers.

The Arduino Uno is an open-source microcontroller board designed by Arduino.cc and based on the Microchip ATmega328P microprocessor

Figure 4: Arduino uno

The board has several digital and analog input/output (I/O) pins that can be used to connect to different expansion boards (shields) and other circuits. The board features 14 digital I/O pins (six of which are capable of PWM output), and 6 analog I/O pins, and is programmable via a type B USB cable using the Arduino IDE (Integrated Development Environment). It can be powered by a USB cable or an external 9-volt battery, with voltages ranging from 7 to 20 volts. It's comparable to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

3.2.2 Specifications

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Pins	6(Pin 3, 5, 6, 9, 10, and 11)
Analog Input Pins	6
Communication protocol	UART x 1, SPI x 1, I2C x 1
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
ICSP Header	2

Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by the bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Power Sources	Power Jack, USB port, Vin pin

Table 1-Specifications of Arduino UNO

3.2.3 Pin Configuration Of Arduino Uno

1. **Vin:** This is the input voltage pin of the Arduino board used to provide input supply from an external power source.
2. **5V:** This pin of the Arduino board is used as a regulated power supply voltage and it is used to give supply to the board.
3. **3.3V:** This pin of the board is used to provide a supply of 3.3V which is generated from a voltage regulator on the board
4. **GND:** This pin of the board is used to ground the Arduino board.
5. **Reset:** This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.
6. **Analog Pins:** The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.
7. **Digital Pins:** The pins 0 to 13 are used as a digital input or output for the Arduino board.

8. **Serial Pins:** These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiver pin number 0 is used to transmit and receive the data resp.
9. **External Interrupt Pins:** This pin of the Arduino board is used to produce the External interrupt and it is done by pin numbers 2 and 3.
10. **PWM Pins:** These pins of the board are used to convert the digital signal into an analog by varying the width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.
11. **SPI Pins:** This is the Serial Peripheral Interface pin, it is used to maintain SPI communication with the help of the SPI library. SPI pins include:
12. **SS:** Pin number 10 is used as a Slave Select
13. **MOSI:** Pin number 11 is used as a Master Out Slave In
14. **MISO:** Pin number 12 is used as a Master in Slave Out
15. **SCK:** Pin number 13 is used as a Serial Clock
16. **LED Pin:** The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.
17. **AREF Pin:** This is an analog reference pin of the Arduino board. It is used to provide a reference voltage from an external power supply.

3.2.4 Features of Arduino UNO

1. In the Arduino UNO board, there is a chip placed that is directly plugged to the USB port and acts as a virtual type serial port for the computer system.
2. The microcontroller used in the Arduino UNO board ATMega328 is easy to available and can be used easily.

3. The board contains other components like PWM pins, timers, external interrupts or internal interrupts, and other types of sleep modes.
4. The board is provided as an open-source tool that has its own advantage as a large number of users use this board and help to troubleshoot other problems related to the board.

5. The pins used in the board act as an oscillator that has a frequency of around 16 MHz that is beneficial for most of the applications.

6. The board can be provided power supply directly without using external power and USB port can be used for this purpose.

3.2.5 Advantages Of Arduino:

1. **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
2. **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
3. **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well..
4. **Open source and extensible software -** The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.
5. **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their version of the module, extending it.

3.3 Flex Sensor

A flex sensor is a device that detects the amount of bending or deflection. It's also referred to as a bend sensor. The carbon surface is set on a plastic strip, which can be turned away to modify the sensor's resistance. Because the resistance is proportional to the amount of bend, it is referred to as a goniometer or flexible potentiometer.This flex sensor is used to detect and analyses finger movement in patients who have had a stroke.

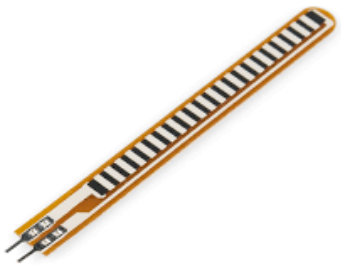


Figure 5: Flex sensor

3.3.1 Types Of Flex Sensor:

These sensors are classified into two types based on its size one is 2.2″ (5.588cm) long and another is 4.5″ (11.43cm) long.

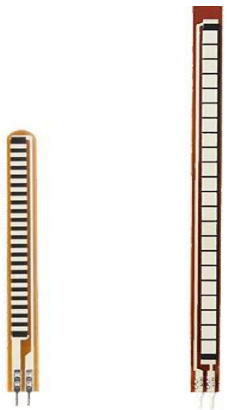


Figure 6: Types of Flex Sensor

3.3.2 Pin Configuration of Flex Sensor

Flex sensor is a tow pin or two terminal device such as p1 and p2. It does not have any polarized terminal like capacitor or diode, means there is no any positive or negative terminal. For power on this flex sensor 3.3V to 5V dc voltage are applied on its terminals

PIN- NO	CONFIGURATION
P1	It is first one pin and is usually connected to positive terminal of power source.
P2	It is second one pin and is usually connected to ground pin of power source.

Table 2- Pin configuration of flex sensor

3.3.3 Specifications & Features:

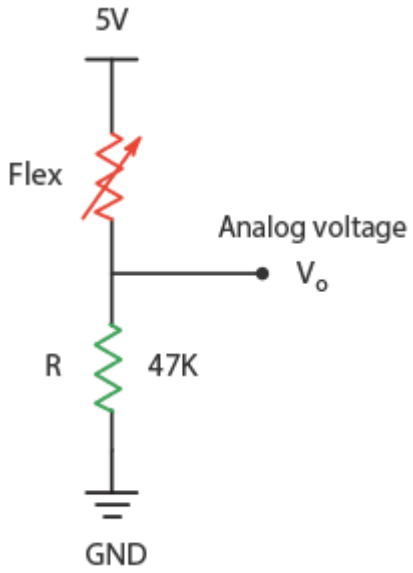
The specifications and features of this sensor include the following:

1. Operating voltage of this sensor ranges from 0V to 5V
2. It can function on low-voltages.
3. Power rating is 1 Watt for peak & 0.5Watt for continuous.
4. Operating temperature ranges from -45ºC to +80ºC
5. Flat resistance is 25K Ω
6. The tolerance of resistance will be ±30%
7. The range of bend resistance will range from45K -125K Ohms

3.3.4 Reading A Flex Sensor:

The flex sensor may be read by connecting it to a fixed value resistor (typically 47k) to form a voltage divider. Connect one end of the sensor to Power and the other to a pull-down resistor to do this.

The location between the fixed value pull-down resistor and the flex sensor is then linked to an Arduino's ADC input. This method, you may generate a variable voltage output that an Arduino's ADC input can read.



It is important to note that the output voltage you measure is the voltage drop across the pull-down resistor, not the voltage drop across the flex sensor. The voltage divider configuration's output is given by the equation:

$$V_O = V_{CC} \frac{R}{R + R_{Flex}}$$

In the shown setup, the output voltage lowers as the bend radius increases.

When the sensor is flat (0°), for example, with a 5V supply and a 47K pull-down resistor, the resistance is reasonably low (about 25k). As a consequence, the output voltage is as follows:

$$V_O = 5V \frac{47k\Omega}{47k\Omega + 25K\Omega}$$

$$= 3.26V$$

When flexed all the way (90°), the resistance rises to 100KΩ. This results in the following output voltage:

$$V_O = 5V \frac{47k\Omega}{47k\Omega + 100K\Omega}$$

$$= 1.59V$$

3.3.5 Circuit diagram:

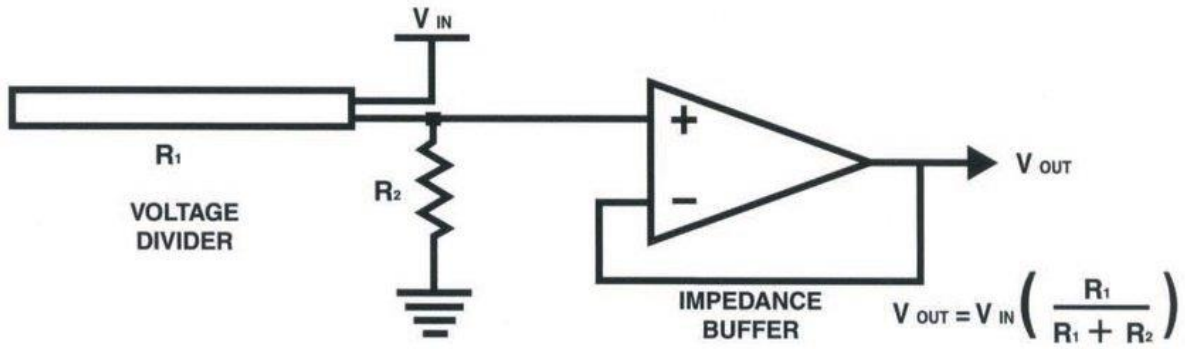
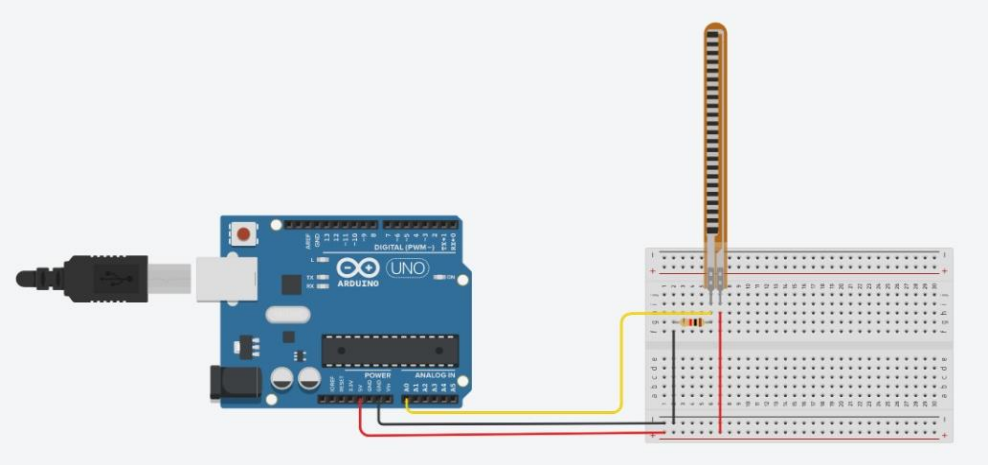


Figure 7: Circuit Diagram Of Flex Sensor

3.3.6 Wiring Flex Sensor To Arduino:

Connecting a flex sensor to an Arduino is simple.

To make a voltage divider circuit, connect a 1k pull-down resistor in series with the flex sensor. The location between the pull-down resistor and the FSR is then linked to an Arduino's A0 ADC input.



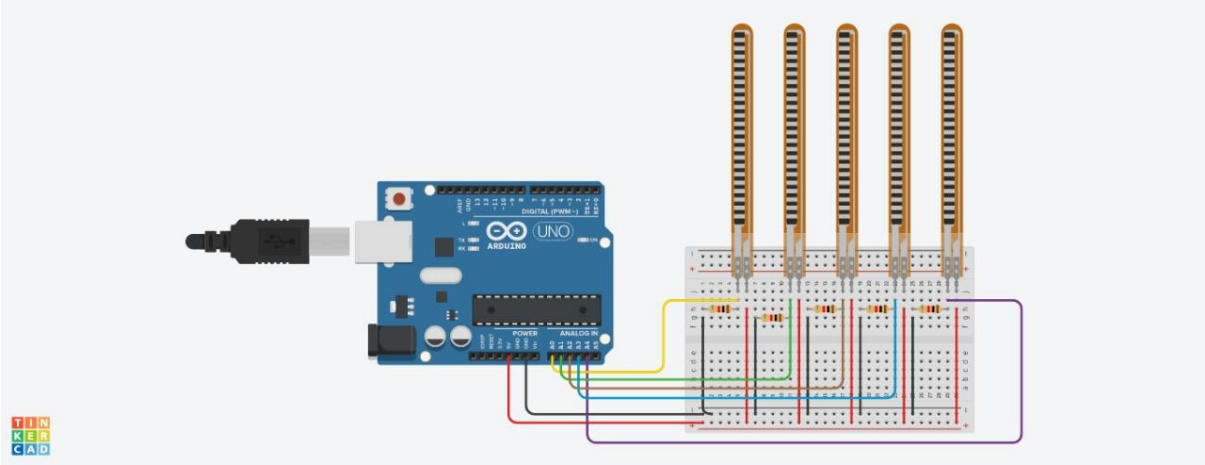


Figure 8: Connection Of Flex Sensor To Arduino

3.3.7 Advantages Of Flex Sensor

a. Medical Instruments
b. Peripherals of Computer
c. Robotics
d. Physical Therapy
e. Virtual Motion (Gaming)
f. Musical Instruments.

3.3.8 USB Cable:

The USB Cable for Arduino is Type A to Type B on both ends. Typically used to link an Arduino board to a PC or laptop. It contains two sets of wires, one for power supply and the other for data transport. The same cable may be used in printers as well. It provides convenient access to operate Arduino boards since we don't need to make any additional power preparations by connecting with a laptop when testing.

Figure 9: Usb Type A To Type B

3.3.10 Breadboard:

A breadboard, often known as a protoboard, is a building platform for electrical prototyping. The solderless breadboard is reusable since it does not require soldering. This makes it ideal for making quick prototypes and experimenting with circuit design.

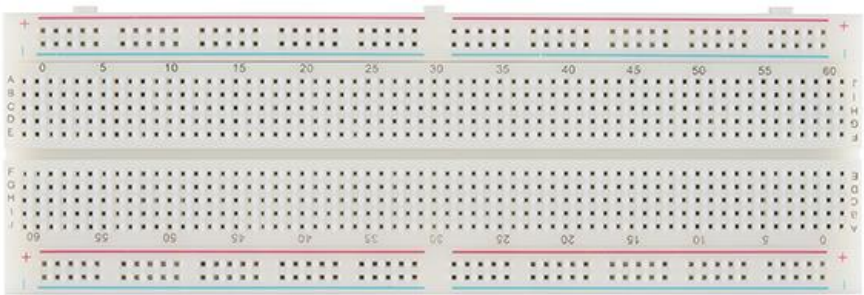


Figure 10:Bread Board

3.3.8 Gloves:

We have developed a movable actuated glove Orthosis equipped with a flex sensor that detects the flexion ability of the fingers. By use of this interactive rehabilitation using game which improves the upper extremity control and task performance.

Figure 11 : Connection Of Flex Sensor To Glove

Each finger of the glove is fitted with a flex sensor. The flex sensor operates on the bending of movement concept. So, when we bend a flex sensor at a specific range, it presses a key on the keyboard or launches the game.

3.4 Software

3.4.1 Unity:

Unity Technologies first produced a cross-platform gaming engine called Unity. Unity's primary focus is the creation of 2D and 3D games as well as interactive multimedia. Unity currently supports over 20 distinct target platforms for deployment, with the PC, Android, and iOS systems being the most popular. Unity has a comprehensive toolset for planning and producing games, including interfaces for graphics, audio, and level-building tools, requiring little to no extra software to work on projects.

C# is the programming language used in Unity (pronounced C-sharp). All of the languages used by Unity are object-oriented scripting languages. Scripting languages, like any other language, include syntax, or parts of speech, with the key components being variables, functions, and classes.

Figure 12: Logo of Unity

3.4.2 Scripting:

Scripting instructs our Game Objects on how to behave; the scripts and components associated to the Game Objects, as well as how they interact with one another, generate your gameplay. Scripting in Unity is now distinct from pure programming.

If you've done any pure programming, such as creating a running app, you should be aware that you don't need to write the code that runs the application in Unity because Unity executes it for you. Instead, you concentrate on gameplay in your scripts. Unity is stuck in a loop. It reads all of the data included within a gaming scene. For example, it reads the lights, meshes, and behaviors and processes all of this information for you.

3.4.3 Unity Interface:

The primary editing window is comprised of tabbed windows, which may be reorganized, detached, grouped, and docked. As a result, we can claim that the editor appears different from one project to the next, and from one developer to the next, based on personal choice and the sort of work you perform. When you create a new project and launch Unity, the following window:

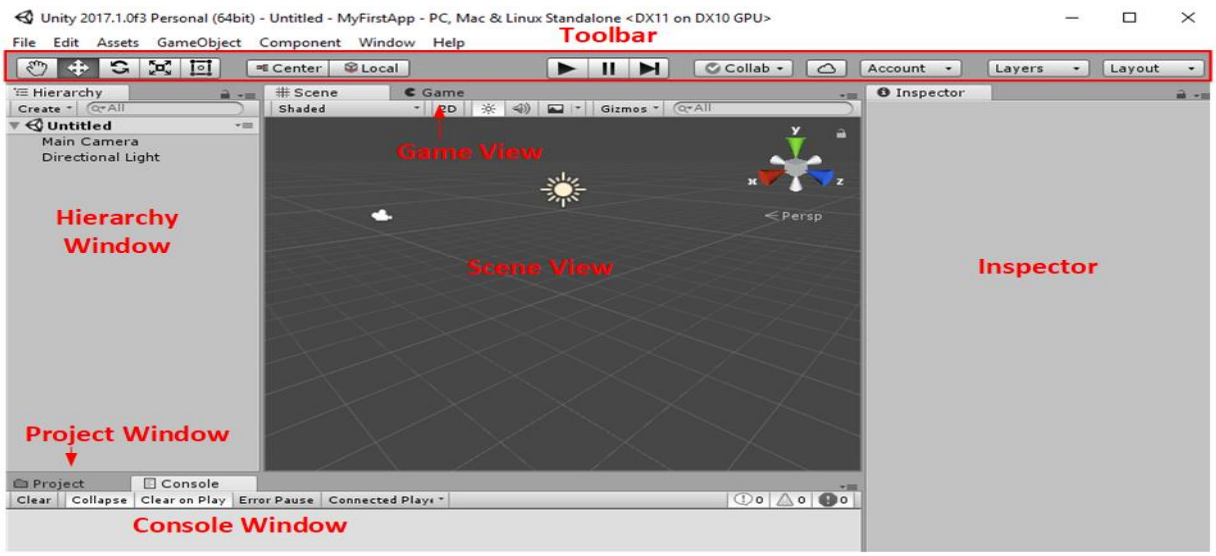


Figure 13: Overview Of Unity Editor

1. Hierarchy Window

The hierarchy window is seen here. Every object in the scene is represented hierarchically by text. It displays a list of all the objects in your newly opened scene, together with their parent-child hierarchy.

Each item in the scene has an entry in the hierarchy, so the two windows are linked. The structure of how items are connected to one another is defined by the hierarchy.

By default, the Hierarchy pane displays Game Objects in the order in which they were generated, with the most recently produced Game Objects at the bottom. We may rearrange the Game Objects by dragging them up and down, or by creating parent and child Game Objects.

Figure 14: Hierarchy window

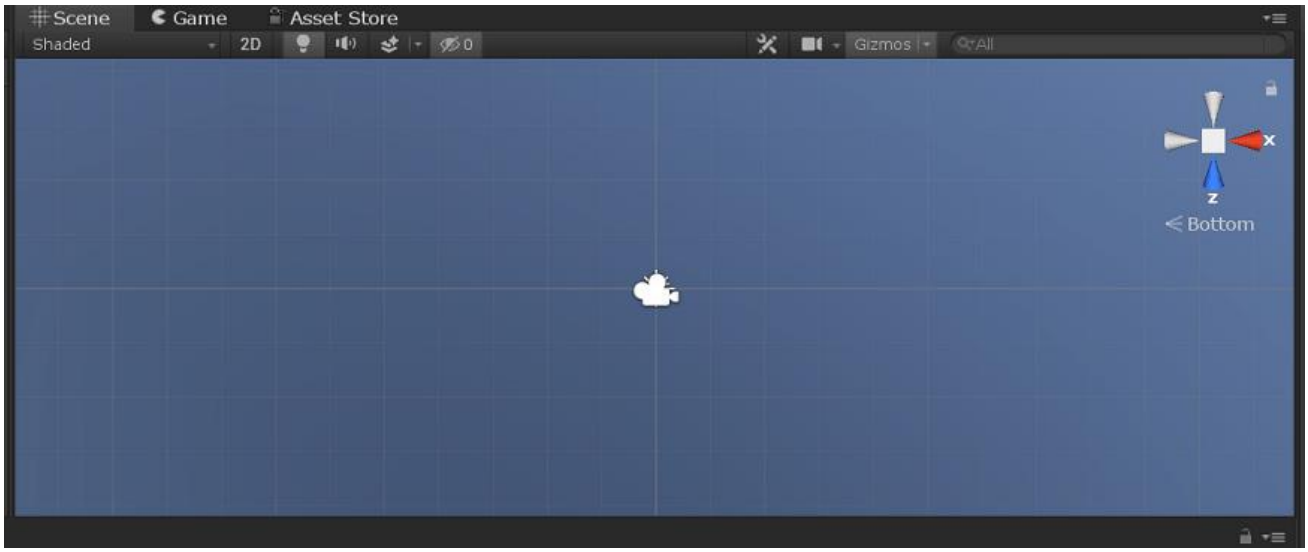
2. Scene View

This is the window in which we will design our sceneries. This view allows you to visually browse and alter your scene.

Depending on the type of project, the scene view can display a 2D or 3D viewpoint.

The scene view is used to choose and place scenery, cameras, people, lighting, and other Game Objects.

Some of the most fundamental abilities you must acquire to operate with Unity include the ability to select, manipulate, and alter objects in the scene view.



.

Figure 15:Scence View

3. Inspector Window

The Inspector window displays and edits all of the attributes of the currently selected item. Because various types of objects have distinct sets of properties, the inspector window's layout and contents will alter.

The inspector window provides complete information about the currently chosen Game object, including all associated components and their characteristics, and allows you to change the behavior of Game objects in your environment.

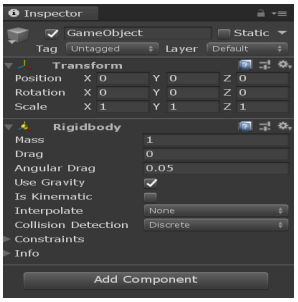


Figure 16: Inspector

4. PROJECT WINDOW:

This pane shows the files that are being utilized in the game. By clicking create in the project box, you may create scripts, folders, and so on.

You may access and manage the assets associated with your project from this page.

All assets in your project are saved and retained in this location. External assets such as textures, fonts, and sound files are also saved here before being utilized in a scene. Above the project structure list is a favorites section. Where you may keep commonly used objects for quick access.

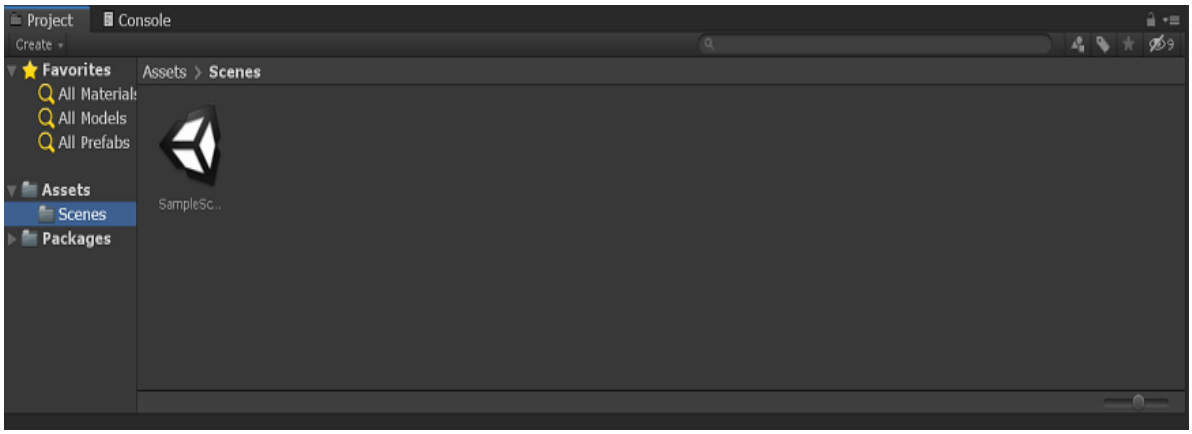


Figure 17- Project Window

5.GameWindow

This window shows the view that the main camera sees when the game is playing. Means here, you can see a preview window of how the game looks like to the player. It is representative of your final game. You will have to use one or more cameras to control what the player actually sees when they are playing your game.

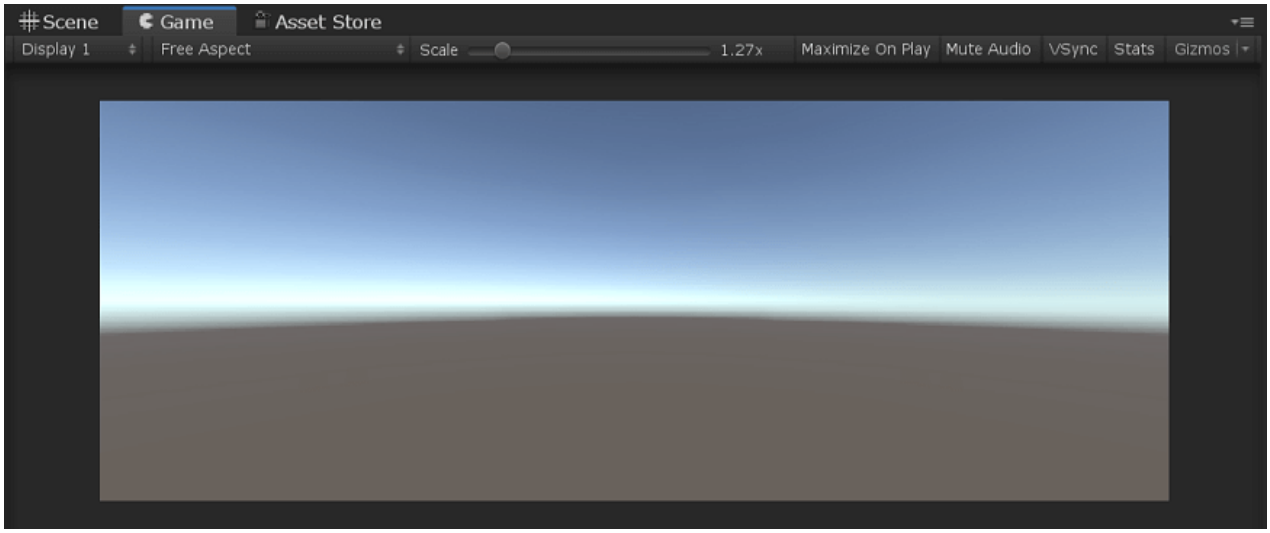


Figure 18: Game Window

6. ASSET STORE

The Unity Asset Store is a growing library of free and commercial Assets provided by Unity Technologies as well as community users.

There is a vast range of Assets accessible, ranging from Models, Textures, and Animations to whole Project examples, tutorials, and Editor Extensions.

The components are downloaded and imported straight into your project using a simple interface provided in the Unity Editor.

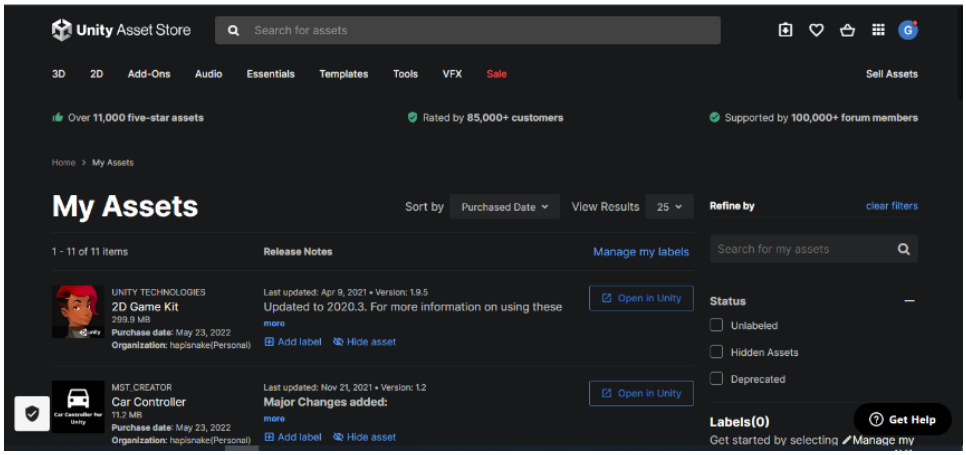


Figure 19- asset store

7. Console Windows

If you're experienced with programming, you'll see that all output messages, errors, warnings, and debug messages are displayed here. It is identical for Unity, except that output messages are handled differently than you may imagine. The Unity console window displays errors, warnings, and other information created by Unity.

You may also use the Debug.Log, Debug.logError, and Debug.LogWarning functions to display your own messages on the console.

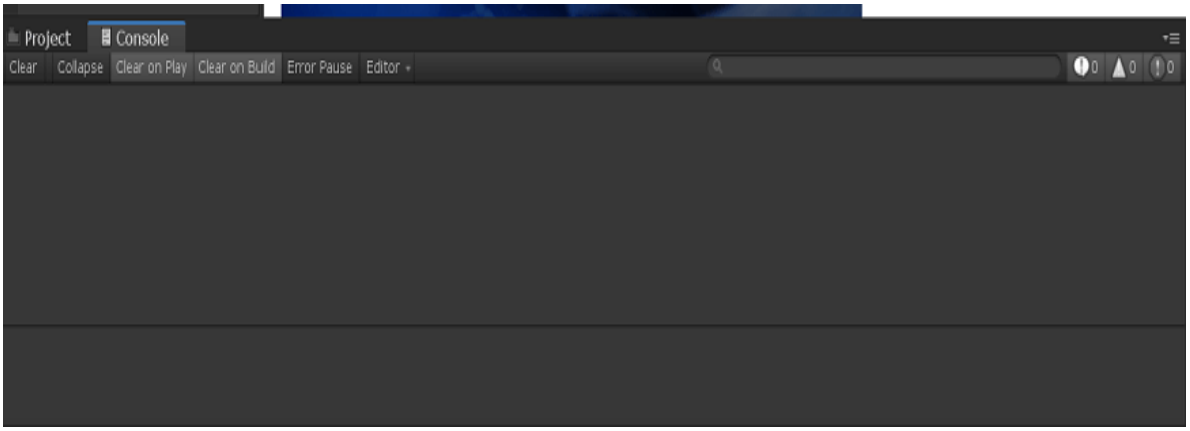


Figure 20: Console Window

8. TOOLBAR

The toolbar is split into seven components. Each toolbar component corresponds to a separate section of the Unity editor.

- Transform Tools: Use transform tools to change the shape of a game item. Modify the location, rotation, and size of the selected game.
- Toggle the transform gizmo to check the centre and pivot of the selected game item.
- Play/Pause/Step: Use these buttons to play, pause, and advance to the next step.

- Cloud: Access to unified services.
- Account: This is where you manage your Unity account.
- Layer: Modify the layer of game items. This determines which game item is displayed in the view.
- Layout: Modify the layout of the Unity editor.

Figure 21: Toolbar

3.4.4 Arduino IDE:

The Arduino Integrated Development Environment (IDE), sometimes known as the Arduino Software (IDE), has a code editor, a message area, a text terminal, a toolbar with buttons for basic operations, and a series of menus. It communicates with and uploads programmes to the Arduino hardware.

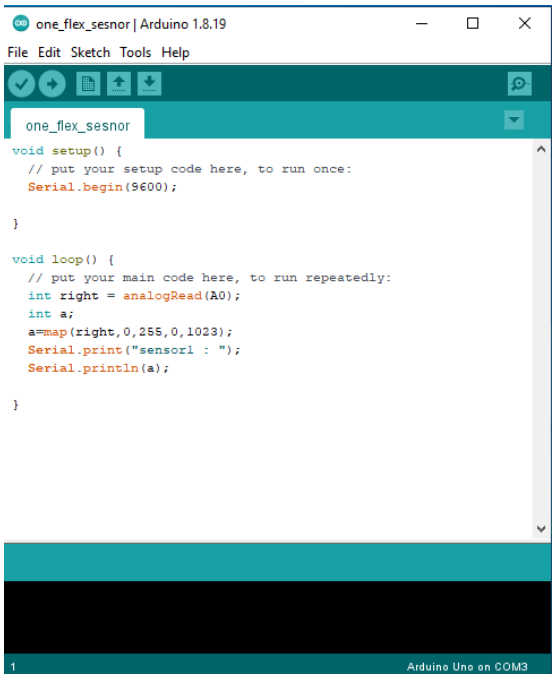


Figure 22: Arduino Ide

Sketches are programmes created with the Arduino Software (IDE). These drawings are created in a text editor and saved with the.ino file extension. The editor has functions for cutting/pasting and searching/replacing text. The message section indicates faults and provides feedback while storing and exporting. The terminal shows text output from of the Arduino Software (IDE), including error warnings and other data.

The IDE environment is divided into three components.

- Output Pane
- Text Editor
- Menu Bar

The top bar is called the Menu Bar, and it has five distinct options, as seen below.

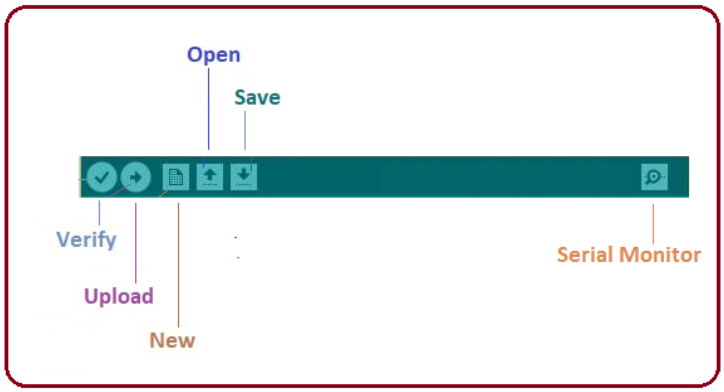


Figure 23- Menu Bar Of Arduino

1. File – You can open a new window for writing the code or open an existing one.
2. Edit – Used for copying and pasting the code with further modification for font
3. Sketch – For compiling and programming
4. Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.

5. Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

- The checkmark on the circular button is used to validate the code. Once you've finished writing your code, click here.
- The arrow key is used to upload and transmit the necessary code to the Arduino board.
- To make a new file, use the dotted paper.
- The upward arrow is for launching an existing Arduino project.
- The downward arrow saves the currently running code.
- The button in the upper right corner is a Serial Monitor — a distinct pop-up window that functions as an independent terminal and is essential for sending and receiving Serial Data.
- You may also access it by going to the Tools panel and selecting Serial Monitor, or by hitting Ctrl+Shift+M all at once.
- The Serial Monitor will really assist you in debugging the written Sketches by allowing you to see how your application is running. To use the Serial Monitor, connect your Arduino Module to your computer using a USB connection.

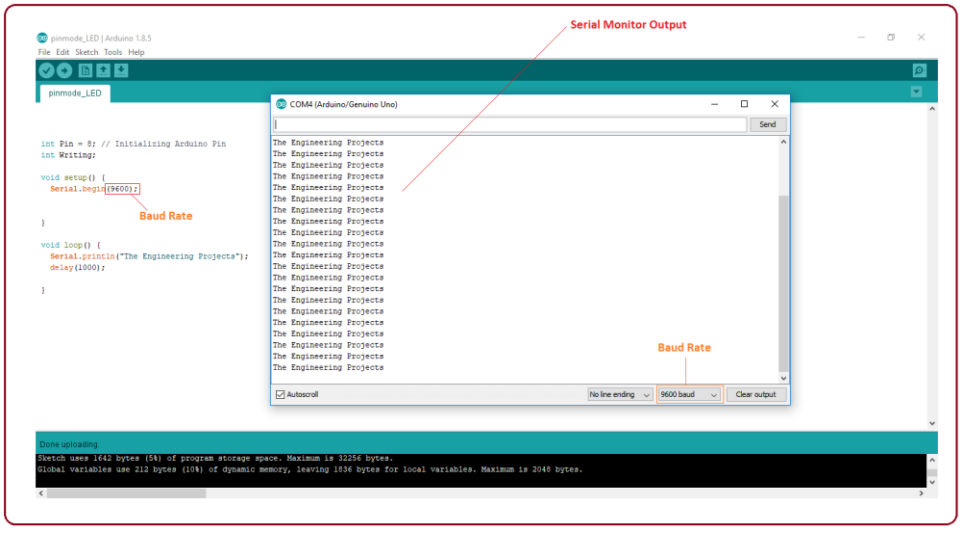


Figure 24: Reading The Analog Vale In Serial Monitor

- The Baud Rate for my Arduino Uno is 9600, thus if you enter the following code and then click the Serial Monitor, the output will look like the image below.

3.4.4.1 Selection And Uploading Of Code To Board:

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system.

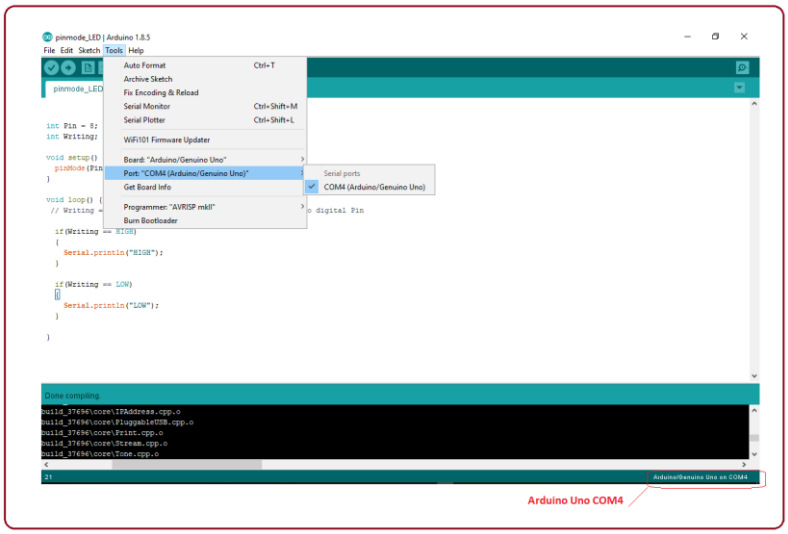


Figure 25: Selecting The Port

As you click the Tools on the menu, it will open like the figure below:

Simply navigate to the "Board" area and choose the board you want to work on. Similarly, the serial and USB boards are assigned COM1, COM2, COM4, COM5, COM7, or higher. The USB serial device may be found in the ports area of the Windows Device Manager.

After selecting both the Board and the Serial Port correctly, click the verify and then upload button in the upper left corner of the six-button section, or go to the Sketch section and hit verify/compile and then upload.

The drawing is created in a text editor and saved with the.**ino** file extension

It is crucial to note that modern Arduino Modules will automatically reset as soon as you build and push the upload button in the IDE software; however, earlier versions may require a physical reset on the board.

When you upload the code, the TX and RX LEDs on the board will flicker, indicating that the desired programme is executing successfully.

Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded.

The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

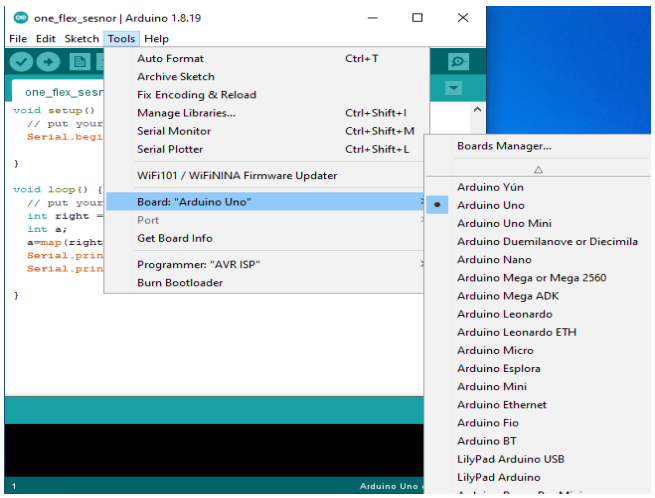


Figure 26: Uploading Of Code To Arudino Uno

3.4.5 Python:

Python is a programming language that is interpreted, interactive, and object-oriented. Modules, exceptions, dynamic typing, and a high degree of dynamic data types and classes are all included.

It supports a variety of programming paradigms in addition to object-oriented programming languages. It combines incredible power with simple syntax. It communicates with several system functions and libraries.

It can also be used as an extension language for programmes that require a programmable interface. It is also portable, since it can operate on a variety of UNIX variations, including Linux and Macs on Windows.

3.4.6 Python Modules

We are using python as a connector between the Arduino and unity game, the python reads the serial output of Arduino and runs the unity game from the movement of fingers.

We are using pynput, keyboard, time, and serial modules. The pyserial module is the basic building block of this project it reads the flex sensor output and moves the player movement. The time module is used to reduce the collapse of the player's movements.

The pynput and keyboard modules are acts as a keyboard for the game. If the player bends the range it sends the data to python the pyserial modules read the data and the keyboard, pynput modules press the respective keys of the keyboard to run the game.

3.4.7 Working on The Modules

In python, we are importing four modules pyserial, time, keyboard, and pynput. The time module consists of the sleep function it needs to delay the press and release of the keys.

The keyboard module consists to detect the press of the key and release of the key by using the function of the read_key(), and pressed_key(). The pynput module can act as a keyboard as well as a mouse. The pyserial module read the serial output of the Arduino Uno based of the finger movements .

3.4.8 Installing And Importing Modules In Python

First, we need to install the required packages in the laptop to run the code without the error. We are installing the pyserial, pynput, and keyboard using the pip method.

Open the command prompt and type these commands line by line:

1. Pip install pyserial

```
C:\Users\GIRIP>pip install pyserial
Collecting pyserial
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Installing collected packages: pyserial
Successfully installed pyserial-3.5
```

Figure 27.:installing pyserial in laptop

2. Pip install pynput

```
C:\Users\GIRIP>pip install pynput
Collecting pynput
  Using cached pynput-1.7.6-py2.py3-none-any.whl (89 kB)
Requirement already satisfied: six in c:\users\girip\app
put) (1.16.0)
Installing collected packages: pynput
Successfully installed pynput-1.7.6
```

Figure 28: Installing Pynput In System

3. Pip install keyboard

```
C:\Users\GIRIP>pip install keyboard
Collecting keyboard
  Using cached keyboard-0.13.5-py3-none-any.whl (58 kB)
Installing collected packages: keyboard
Successfully installed keyboard-0.13.5

C:\Users\GIRIP>
```

Figure 29:Installing Keyboard Package In Python

Time package was preinstalled with the python. So we need to import the time during the exection of the program.

3.4.8.1 Pyserial:

This module is including the serial open to read the data from the Arduino serial monitor. It requires to choose correct port of the usb cable.

Readline & readlines:

1.**READLINE():** reads up to one line, including the \n at the end. Be careful when using readline(). Do specify a timeout when opening the serial port otherwise it could block forever if no newline character is received.

2.**READLINES():** tries to read “all” lines which is not well defined for a serial port that is still open. It raises an exception if the port is not opened correctly. The returned list of lines do not include the \n.

While we need to read the serial data of Arduino in python . first, we close the serial monitor of Arduino ide . if we didn’t close the ide of Arduino it will show error as com was busy are not selected. Second, choose the correct

com of the laptop it will be different from each OS . in windows on the device manager and see which com is using for uploading the code to the Arduino.

Opening serial ports

Open port at “9600,8,N,1”, no timeout:

PYNPUT

```
import serial
ser = serial.Serial('COM5')  # open serial port
print(ser.name)         # check which port was really used
ser.write(b'hello')     # write a string
ser.close()             # close port
ser = serial.Serial()
ser.baudrate = 19200
ser.port = 'COM1'
ser.open()
ser.is_open
True
ser.close()
ser.is_open
False
```

Figure 30: Pynput Modules

Third, after typing the correct com and baud rate in the program. We need to use the binary format only Because data from Arduino are in the binary format so we are using binary format in python also.

3.4.8.2 Pynput:

It allows the user to control the keyboard and mouse using the python. To use these packages in python we are importing this module using import function. We are using the key and controller function only in this module.

It contains subpackages for each type of input device supported:

- **pynput.mouse**

  Contains classes for controlling and monitoring a mouse or trackpad.

- **pynput.keyboard**

  Contains classes for controlling and monitoring the keyboard.

The above images shows what are the function in the pynput module. First we are using only keyboard in pynput . so, we are import the pynput in the python using the import function from pynput.keyboard we are using key and controller modules only.

Figure 31 : Functions In Pynput

For the press of keys in keyboard we are using the press function. The release of the key we use the release function for the special functions like space bar, ctrl, alt so we need to use the key infront of them.

3.4.8.3 Keyboard

This python packages consists of the key press and key read functions this can help us to make the game to run without any delay**.**

- It helps to enter keys, record the keyboard activities and block the keys until a specified key is entered and simulate the keys.
- It captures all keys, even onscreen keyboard events are also captured.
- Using this module we can listen and send keyboard events.
- It works on both windows and linux operating system.

keyboard

```
import keyboard
# press and release
keyboard.press_and_release('shift+s, space')
keyboard.write('The quick brown fox jumps over the lazy dog.')
# Blocks until you press esc.
keyboard.wait('esc')
# Record events until 'esc' is pressed.
recorded = keyboard.record(until='esc')
# Then replay back at three times the speed.
keyboard.play(recorded, speed_factor=3)
# Block forever, like `while True`.
keyboard.wait()
```

Figure 32: Functions In Keyboard

3.4.8.4 Time:

This module uses to delay the key press and key release for to run the program. Python time sleep function is used to add delay in the execution of a program. We can use python sleep function to halt the execution of the program for given time in seconds. Notice that python time sleep function actually stops the execution of current thread only, not the whole program.

Sleep.py

```
# importing time module
import time
t = 2 # 2 seconds
time.sleep(t)
```

Figure 33: Time Module

HARD WARE SIMULATION

3.5 Tinkercad:

It's a fantastic programme for simulating Arduino-based systems. All activities, including your own inventions, may be simulated before being tested on real hardware. It also allows you to programme using blocks. Rather than having to create it from scratch, you may download / copy-paste the produced code later into the Arduino IDE to programme the real Arduino board.

Figure 34: Tinkercad Logo

3.5.1 Exploring Circuits In Tinkercad:

After logging in to Tinkercad, you'll see a dashboard with a list of your most recent creations. This dashboard will display designs created with Tinkercad's 3D editor by default.

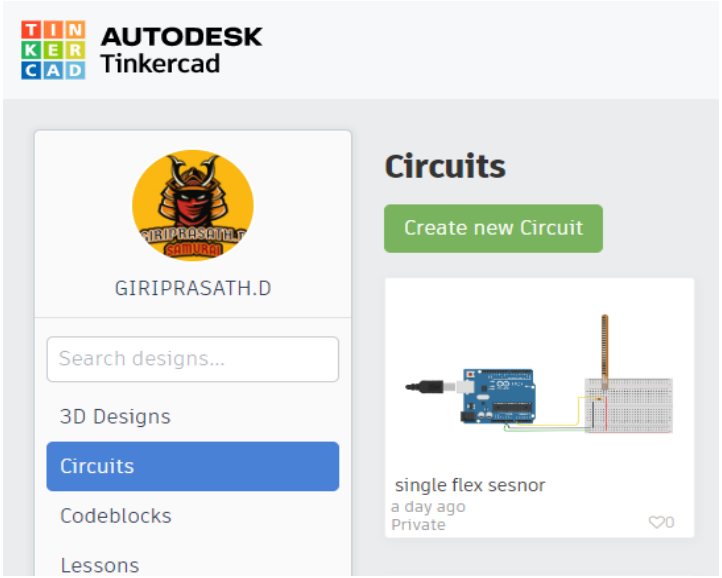


Figure 35: Creating The Circuits In Tinkercad

Tinkercad's Circuits editor is designed similarly to its 3D design editor. On the left, there is a wide window where you may create your design.
On the right side, there is a panel containing components that you can drag and drop onto the workspace to build your circuit.

Circuits' workspace is two-dimensional. You may pan the view around your design by clicking and dragging the empty area around it, or you can move your components around by choosing and dragging them.

You may also use the scroll wheel on your mouse, a two-finger motion on your trackpad, or a key combination of Command + and Command - to zoom in and out of your design.

In the upper left corner of the workspace, there is a "Zoom to fit" button that will centre and zoom your design to fill the window. The letter F on your keyboard serves as a convenient shortcut for this same operation.

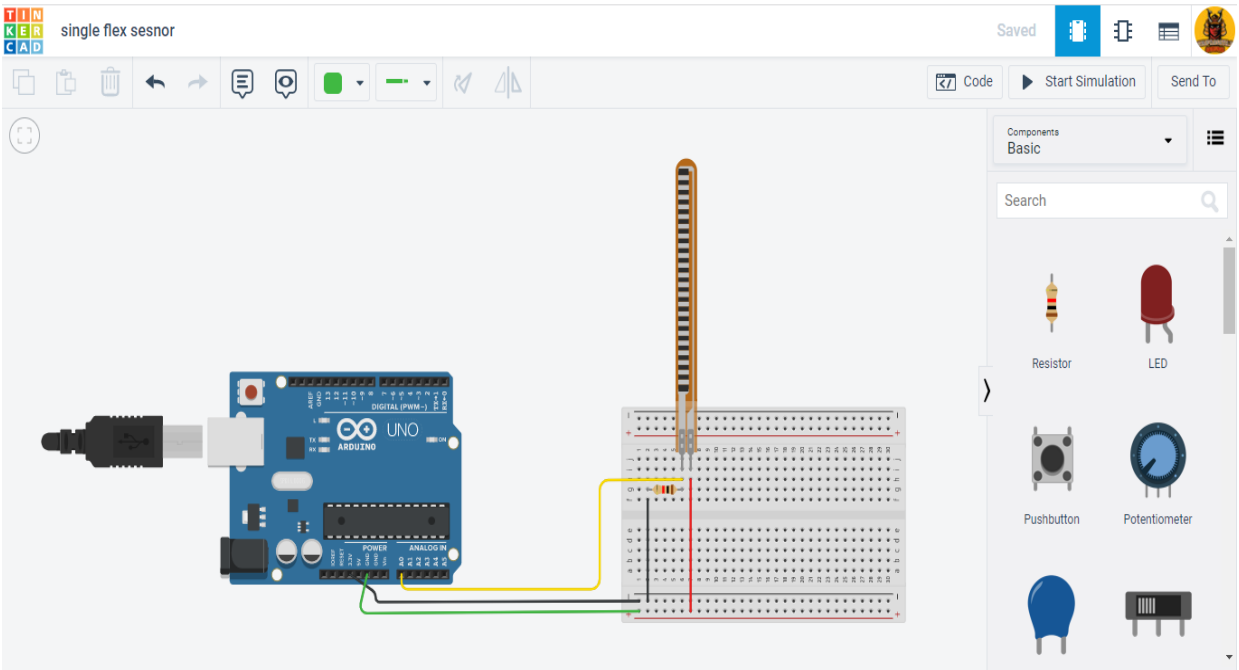


Figure 36: Assembling The Components In Tinkercad

3.5.2 Using Tinkercad

You'll also note that this menu contains more than simply components. You'll find a list of Starters farther down. These are pre-built circuit examples that students may drag into their workspace and simulate, alter, and remix.

These Starters are divided into four groups: Basic, Arduino, Micro: bit, and Circuit Assemblies. When you hit the Start Simulation button, all of our Starters come to life in some fashion

Basic Starters are made from the kind of common electronic components typically used to introduce students to electronics (LEDs, batteries, hobby motors, resistors, and switches). These examples use no microcontrollers, and no code.

3.5.3 Coding Arduino In Tinkercad

Tinkercad allows you to code your Arduino using two different approaches.

Our Blocks code editor offers beginners a visual system of functions that they can drag and rearrange. All of our Arduino Starters, and most of our interactive Arduino lessons will include or refer to Blocks code.

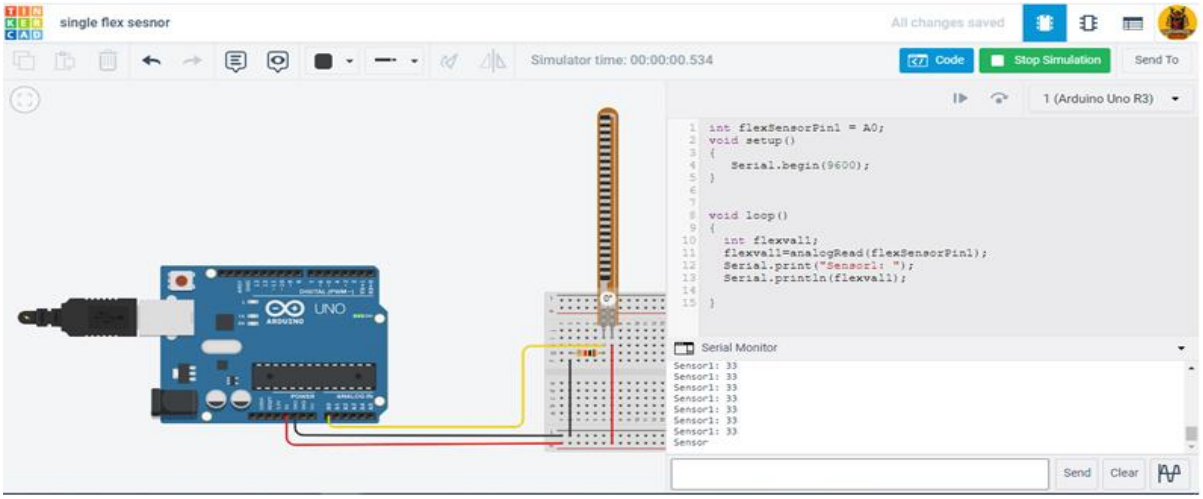


Figure 37: Code Of The Game

Part of the magic of learning to code Arduino in Tinkercad is that our editor will automatically generate text-based code (C++) from students' blocks code. By switching the code view to Blocks + Text, students can see the logic of their blocks code translated to C++ code.

Modifications made to their blocks code will instantly update in the text view, providing insight into the logic and syntax of C++.

Of course, once students are ready to create their code directly in the text editor, they can switch to a pure text view. This view offers an experience similar to programming with Arduino's IDE editor.

Libraries:

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Proposed method:

Stroke is one of the leading causes of physical impairment in the globe. The majority of survivors have mobility disability, sometimes with long-term deficiencies in hand dexterity. Stroke survivors have few treatment choices. Traditional rehabilitation focuses on training that emphasizes task completion rather than reducing motor dysfunction.

We created a game-based rehabilitation programme to help stroke sufferers enhance their unique finger mobility. When compared to other available treatments, this has a quicker recuperation duration. The recovery of existing techniques is projected to take 16-20 weeks. However, our recommended approach takes roughly 8 weeks.

Nearly 76 percent of post-stroke patients with non-functioning hand motions such as flexion and extension have hand-related difficulties, according to research. Proposing a "Game" that entails the use of a portable actuated glove orthosis to enhance finger mobility.

The flex sensor, which is affixed to the patient's gloves, is used to play the game.

With the aid of the Arduino UNO, a flex sensor is utilized to detect any movement in finger motions. The Arduino UNO is a microcontroller that is free to use and modify. It can serve as a link between the game and the glove.

The game was created in the Unity editor to meet the demands of the patient. The player's movement will operate as an input from the flex sensor linked to the glove in the game.

The game begins and continues with various obstacles as a result of this input. This game is played on a regular basis to help the patient's finger mobility.

4.2 Interaction with the game :

It Describes the operation of the proposed system providing evidence of proper working of the project. The user was Playing the game.

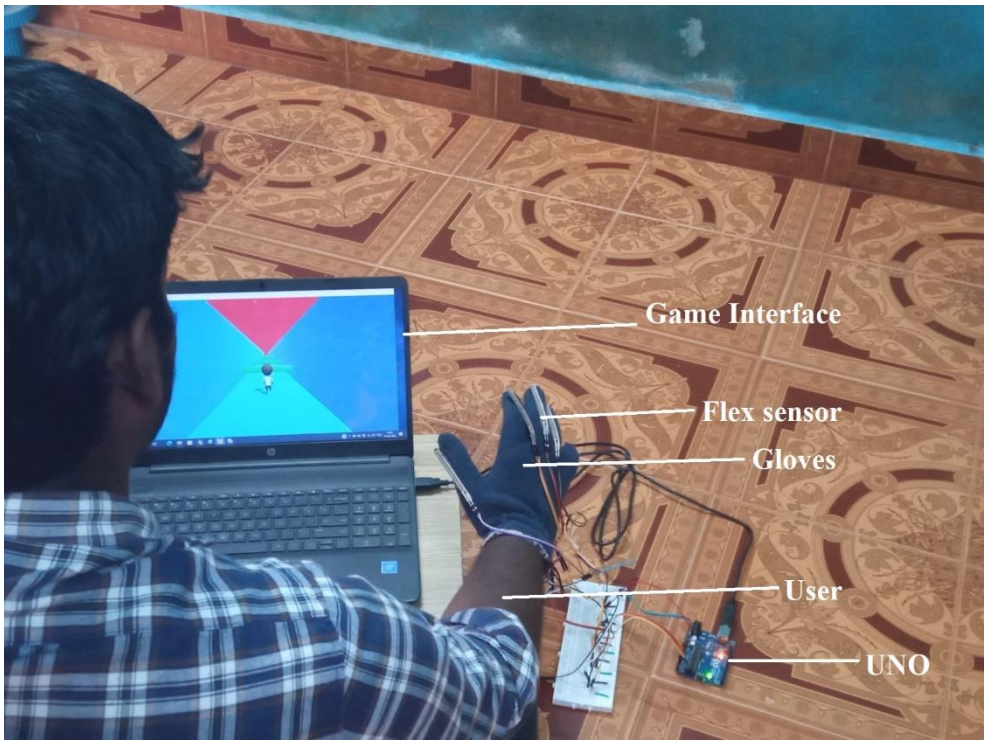


Figure 38: User playing the game

The above image showing that the player was interactive with the game. The movement of the finger helps to run the player of the game

4.3 Game Simulation:

The game was made using the game engine Unity, The assets used in the game consists of the primitive ones and ones imported from the Unity Asset store. The game logic is written using C#, which is an object-oriented programming language built by Microsoft. Unity has a built in support for C#.

The Architecture of the game is as follows: Start scene -> Game Scene -> End Scene.

4.3.1 Main Menu:

This is the scene which is loaded first, when the player opens the game. The main scene consists of start, character, help, credits and quit. This makes the game more visually appealing. There is a start button at the first. Which when clicked will start the game. The looped waving animation and the subsequent idle animation of the character is achieved using the animator controller option in unity. The animation 38 clip of the boy waving is first captured using the animation time frame option. An animation clip when the boy is supposed to be idle is also created. The animation clips are carefully created in such a way that the movements of the boy are intricate and look realistic. The two animations are combined to be synchronized using the animator controller.

The animator controller manages the smooth transition between the two animations using a state machine.

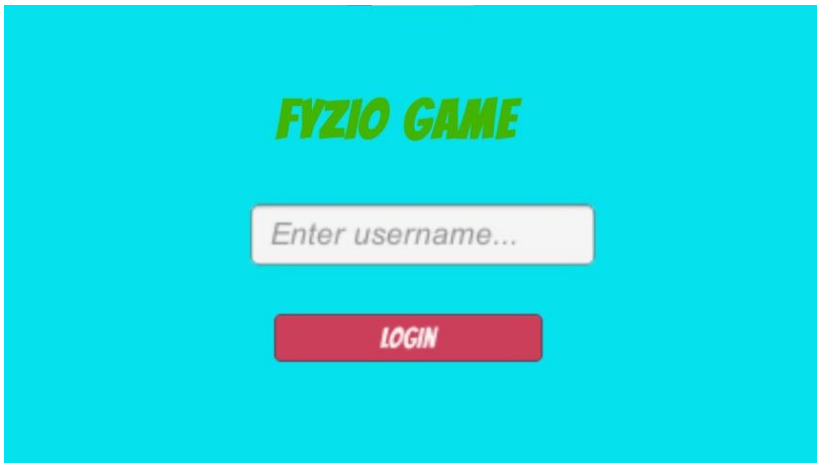


Figure 39: Main Menu Of The Game

4.3.2 The Play Button:

The play button is controlled using a script. A function to load the game Scene when the button is triggered is written. The button is rendered on the screen on a canvas. A canvas is a component where all the UI elements of an app are to be added.

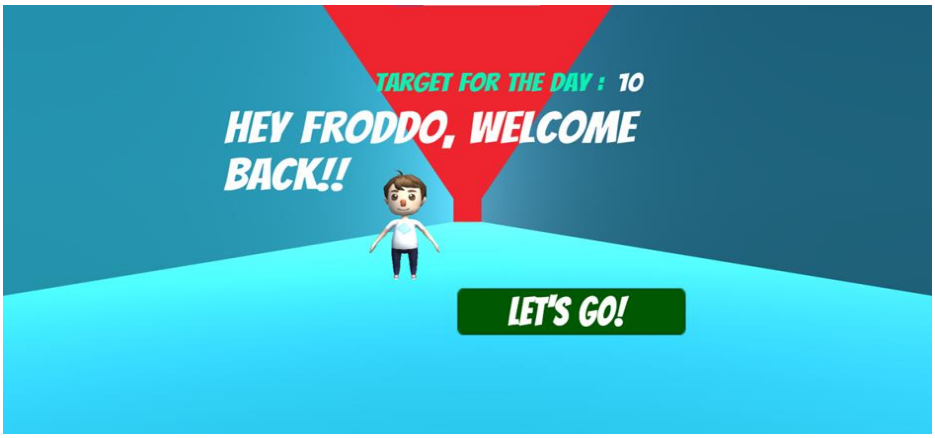


Figure 40: Starting The Game

4.3.3 The Game Scene:

The game starts with the boy standing in the bridge. The character of the boy can be seen running in a bridge. The bridge consists of obstacles. The boy has to dodge the obstacles when he reaches near it by jumping.

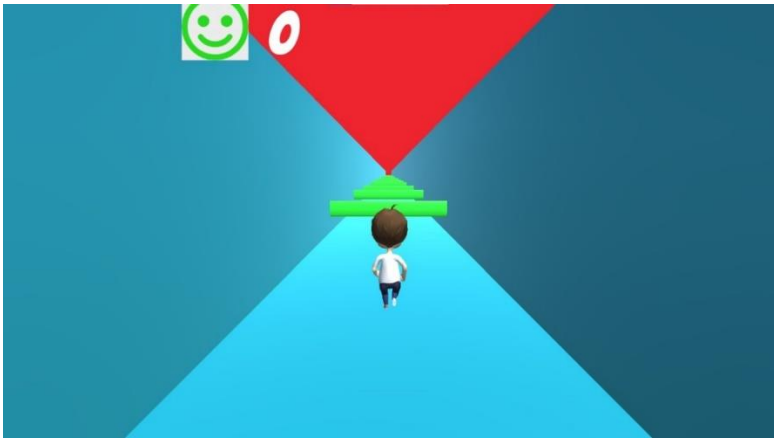


Figure 41: Starting Point Of Game

The score on the canvas will be updated dynamically. If the boy hits the obstacle, the game restart itself. A motivational message is displayed on the screen. The objective of the boy here is to successfully dodge a required number of obstacles irrespective of the number of misses. To make the game more immersive, an energetic music is added in the background of the game.

4.3.4 Countdown Timer:

There is a countdown from when the game scene loads. The countdown is created using the invoke coroutine in unity. A text area in the canvas is dynamically made to be changed to the numbers 0.1 and 1 with a 1000 millisecond gap in between which is done with the help of the co-routine.

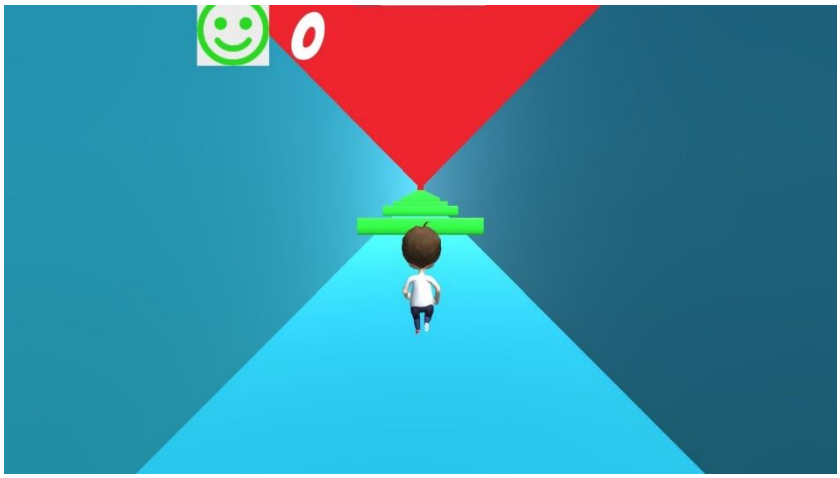


Figure 42: Game Begins

4.3.5 Collision With The Obstacle:

In order to detect the collision of the player with the obstacle. The player and the obstacles are added with the collider component. When two Game Objects containing the collider component clashes, the ―on Collision‖ function is triggered by unity. Hence the necessary operations to be performed after the collision is given inside the on Collision function.

4.3.6 Dynamic Obstacle And Endless Tiles Generation:

The obstacles are generated dynamically in the game scene in the canal with a certain spacing between them. This is done using the Instantiate function in unity. This feature is added so that the number of obstacles and the distance between the obstacles can be adjusted using the level specified by the user. The canal is made to be endless i.e.) The canal is made to behave in such a way that it expands in size, with the movement of the user.

4.3.7 End Scene:

The End scene is loaded when the user successfully dodges the specified number of tiles. The end scene consists of the score of the user displayed in the canvas.

The end scene also contains the character. The running animation of the character is reused here.

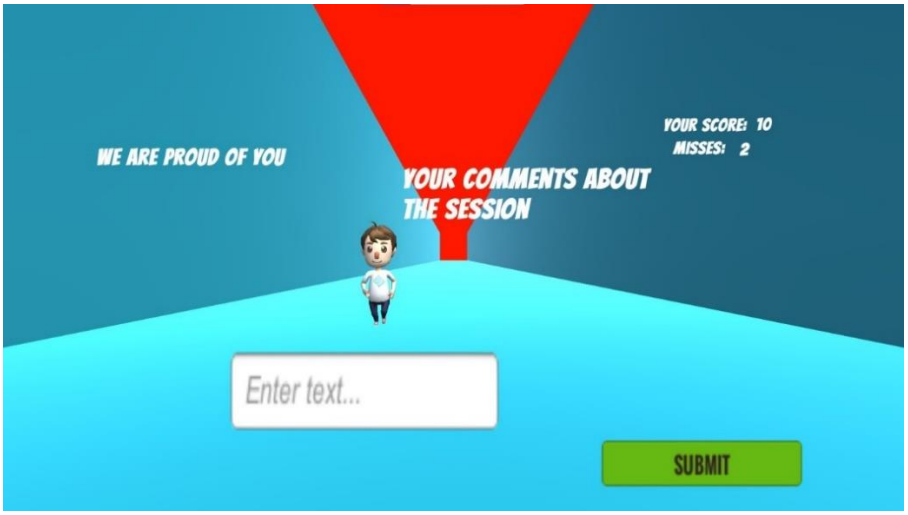


Figure 43:Ending page of game

4.4 Interfacing Flex sensor with Arduino:

The process of connecting the flex sensor to the Arduino is simple. The analogue sensor sends data to the Arduino.

Pins for both interfaces and interrupt pins are accessible in each flex sensor module. Two-wire connections are supported by the sensor.

The board has a number of digital and analogue input/output (I/O) pins that may be used to connect to other expansion boards (shields) and other circuits.

The flex sensor is wired to the breadboard. The sensor's one end is linked to the Arduino's A0 pin, while the other end is connected to the power supply. These wires are linked to the ground via a resistor.

The patient's gloves are equipped with a flex sensor, which works as a key for the player in the game to leap and conquer hurdles.

The flex sensor, as well as the bending range of individual fingers, must be calibrated prior to the commencement of the game. This plays an important part in the patient's finger movement development.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion:

We can conclude that this proposed method, improves the Patient's finger movement. The essence of our game's therapeutic benefit is predicated upon its ability to provide positive reinforcement and the reframing of the therapy experience.

Players found the game enjoyable on all three observed dimensions: graphics, input, and overall enjoyment.

As the required poses had some potential to fatigue a player's muscles, it is possible for exercise-related discomfort. This, however, is a net positive as a player's physical therapy session is expected to cause some amount of discomfort in order to challenge the muscles.

As the goal of the project is to reframe therapy actions as game play actions, if these actions are frustrating then the therapy behavior is unlikely to sustain itself and will soon extinguish. The strongly positive ratings of the game's visual aesthetics are especially encouraging.

The game world should be as enjoyable as possible to play in in order to provide reasons for the player to return there. Further, whatever deficiencies may have existed in the control scheme, it seems they were not sufficiently distracting to detract from player enjoyment of the game world. Snapshots of the output from the sensor, the game and the website have been added below.

5.2 Future Work:

The project's next work will focus on helping post-stroke patients reclaim their normal lives. By increasing the game's difficulty.

Also, each and every finger movement is required for the game to function properly. Patients should be able to enjoy the game while also benefiting from the rehabilitation process.

There's also talk of putting in an accelerometer (ADL335) sensor to help with wrist movement. These sensors can be used to control the game's direction of movement.

CHAPTER 6

REFERENCES

[1] : D. T. Wade, R. Langton-Hewer, V. A. Wood, C. E. Skilbeck, and H. M. Ismail, "The hemiplegic arm after stroke: measurement and recovery," J Neurol Neurosurg Psychiatry, vol. 46, pp. 521-4, Jun 1983.

[2] : Firas Mawase, Kendra Cherry-Allen, Jing Xu, et al,2020, “Hand Motor Imairment Can Be Reduced in Chronic Stroke”, Neurorehabilitation and Neural Repair 2020, Vol. 34(8) 733–745 DOI: 10.1177/1545968320939563

[3] : N.Alberto Borghese, Jacopo Essenziale, Renato Mainetti, et al, Published: 13 December 2019, Hand Rehabilitation and Telemonitoring through Smart Toys.

[4] : Venkatesh N. Dubey, Soumya K. manna,2019, DESIGN OF A GAME-BASED REHABILITATION SYSTEM USING KINECT SENSOR, Proceedings of the 2019 Design of Medical Devices Conference DMD2019 April 15, 16-18, 2019, Minneapolis, MN, USA DMD2019-3237.

[5] : Liu, H., X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C.Zhu (2017). “A glove-based system for studying hand-object manipulation via joint pose and force sensing”. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 6617–6624

[6] : Ates, S.; Haarman, C.J.W.; Stienen, A.H.A. SCRIPT passive orthosis: Design of interactive hand and wrist exoskeleton for rehabilitation at home after a stroke. Auton. Robots 2017, 41, 711–723.

[7] : Arduino -https://www.arduino.cc/ - It Is the Official Page of Arduino Boards from This We Need to Choose Which Board Is Suitable for Our Project

[8] : https://forum.arduino.cc/ - This website helps us to ask questions about how to connect the Arduino board to unity game

[9] : Unity. unity3d.com (2022). Available https://unity3d.com/ - It is a official website of unity to downloaded the unity engine and unity hub to make our game for this project.

[10] : https://assetstore.unity.com/ - From this website we get our project assets for our game. It is the official assets store for unity.

[11] : https://www.python.org/ - It is a official homepage for the python programming. It is also a act a python download pages.

[12] : https://pypi.org/ - It is a python package website . To find, install and publish python packages with the python package index.

[13] : https://pypi.org/project/keyboard/ - It is a page contains of how the python program can controller the laptop's keyboard.

[14] : https://pypi.org/project/pynput/ - It is a page contains of how the python program can read the output of Arduino UNO serial monitor.

[15] : https://pypi.org/project/pyserial/ - It is a page contains of how the connect python program and output of Arduino UNO serial monitor

[16] : https://pyserial.readthedocs.io/en/latest/pyserial.html - It consist of the documentation of the pyserial.

[17] : https://www.tinkercad.com/ - It is a website that use situmlate the arudino uno and flex sensor before it was build.

[18] : https://github.com/ - Our game was uploaded in github for the free of use to other participants