Interaction with virtual game through hand gestures based on computer vision

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CSE4015 - Human Computer Interaction

19BDS0084 - MD Ismail

19BCE0656 - Yaswanth M

19BCE0210 - Gokul Nath

Under the Guidance of

Prof. Swarnalatha P

Vellore Institute of Technology

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

AIM:

For creating a Human computer interaction video game which we can play with the assist of hand gestures. Gesture recognition primarily based on interactions, offer a greater sensible and immersive interaction in comparison to today’s peripherals.

SCOPE:

A platform to make a personality's computer interaction game which might be played without using the standard peripherals through hand gestures were the user are going to be able to experience a more real-like and captivating interaction than ever. The game is designed with the notion that anyone with no prior knowledge may play it. Gamers who are having impaired vision are excluded. The users must have their background plain within the webcam view so as detect the object/palm clearly, have their background plain in the webcam view in order detect the object/palm clearly.

1. **Project Justification:**

   A platform to create a human computer interaction video game which can be played without using the traditional peripherals through hand gestures where the user will be able to experience a more real-like and captivating interaction than ever.

2. **Assumptions:**

   The game is implemented with an assumption that anyone can play the game without prior knowledge.

3. **Project exclusions:**

   Gamers who are having impaired vision are excluded.

4. **Constraints:**

   The users must have their background plain in the webcam view in orderdetect the object/palm clearly.

Abstract:

Hand gestures are a simple way for people to communicate, and they may be addressed promptly in human-computer interaction (HCI). This way of user interaction can be used efficiently in the user system interface for a user-friendly interaction. Hand gesture interaction has been the trending era for human computer interaction (HCI). Frequently some of research work are accomplished on this area to expedite and contrive interaction with computers. In this venture, we design a real-time human computer interface system for interaction using two different approaches - hand pointing and clenched fist gesture. This project entails the design and implementation of a HCI the use of webcam or the front cam of laptop. It analyses hand gestures in actual time 2D space of hand using segmentation, thresholding, contouring and convex hulling is used to identify hand gestures.

Objectives:

The Primary objective of the game is to provide fun and interesting way of playing a computer game. The game controls are rather more natural and adaptable than the same old keyboard controls. Using hand gestures also promotes physical activity which improves health and general well-being as against traditional keyboard control.

Introduction:

Hand gesture recognition systems provides an improved interaction experience for the user because it combines a virtual and real-world item. Gesture popularity primarily based totally interactions, offer an extra connection that is both in comparison to normal peripherals, it is more realistic and immersive. The gesture-based interaction interface showcased here are often applied towards many applications like video game, communication techniques and Games. The main focus of our project is on games because the application domain for this interaction method. Gestures, particularly hand gestures are faster and possibly may well be more accurate than using the keyboard–mouse combination of peripherals. The non-touch system might be a cutting-edge form of computer interface technology that reshapes human-computer interaction.

MOTIVATION:

The motivation for the implementation of the project is a concept to build an interface as a goal to provide the gaming world real-time user experience which is easy to use by the gamers as the gaming world in modern days is emerging into one of the largest growing markets around the globe with more new people developing the interest in this field.

Individual Contribution:

Name	Registration No	Contribution
Ismail	19BDS0084	Review Study, Algorithm Analysis, Documentation, Project analysis using HCI techniques.
Yaswanth	19BCE0656	Review Study, Implementation of Code (Python), Interface.
Gokul Nath	19BCE0210	Algorithm Analysis, Implementation of Code (Python), Project analysis using HCI techniques.

Our Contribution:

The proposed project has a promising future of the forth coming pro- digital era as currently the game focuses on breaking the norms of current gaming industries by hand gesture recognition system. The method chosen i.e., hand gesture recognition would provide better indulging gaming due to better physical enhancement and a more realistic interaction between the user and the interface. The project requires no large space for playing and proves its adaptability and is highly user-friendly nature. Thus, the game can even be played in a virtual space by the player. Due to the involvement of physical activities the user will have better health and will not lose his physique even if the user is playing in the long run.

Proposed Methodology:

The player would play the game with the help of his/her fist or palm. When the palm is gestured up, the character moves up and vice versa. The computer’s front camera is the sensor which detects motion.

Expected Outcome:

Designing a game where random obstacles would be created for the player who must dodge them in order to achieve a high score. There will be a timer when the game starts and it stops when the player collides with any of the obstacles. This game will be different from others due to the interesting and easy method of controlling the character in the game.

The game is made to be user friendly, so that users from all sections of society will be able to easily play the game without any prior practice or learning of controls as the controls are the natural human movement achieve a high score. There will be a timer when the game starts and it stops when the player collides with any of the obstacles. This game will be different from others due to the interesting and easy method of controlling the character in the game. The game is made to be user friendly, so that users from all sections of society will be able to easily play the game without any prior practice or learning of controls as the controls are the natural human movements.

Project Novelty:

Our hand gestures are easy to use and learn for first time users. The hand gesture recognition system can be used in any game which has the same key binds. No need to alter or have different implementation for these games. It also provides a more fun and exciting way to play games as well as help the disabled people who have issues with utilizing the traditional methods of input. This form of human – computer interaction definitely is improving every day and has a lot of potential in the future for development and usage in everyday life. Key advantage here is that games can be played in a virtual space through the same movements in the real world, without installation of special controllers.

Proposed Work:

The main domain of the project lies in developing a game with the application of gesture recognition system. The usage of hand gestures to promote virtual activity as one does in real world, results in the main advantage that the game can be played in a virtual space with enhanced interaction much better than conventional peripherals.

Method and approaches:

When the user runs the program the game in command prompt window opens up and accesses the webcam of the user. Once the game starts, the player needs only to move his hand to make his game character move and dodge the obstacles and score points.

Hardware Requirements:

1.Processor - Processor frequency of minimum of 1.9 gigahertz (GHz) needed

2.Architecture - x86- or x64-, dual-core processor

3.Instruction Set - SSE2

4.Memory - min 4 GB RAM

5.Graphic Card Required 2 GB.

Software Requirements:

1.Operating system: Windows 7, Windows 8, Windows, Linux, and Mac compatible.

2.PyCharm python IDE, Visual Studios, Python libraries.

Other Requirements:

Web drivers

Touchless user interface.

REQUIREMENT SPECIFICATION:

1. Touchless Interface:

Touchless user interface is a new way for motion control. TUI interface talks about your PC through body movements and motions without touching the console, mouse, or screen. In addition to gesture control, Touchless less interfaces are also increased because they provide the ability to interact with devices without physically touching them.

2. Input Device:

A) Gesture Based Controllers:

Gesture controllers act as an extension of the body which allow users to interact with their surroundings without using controllers, when gestures are working, some of their motions are conveniently captured by the software.

For example, by taking gesture-based motion capture is skeletal hand tracking, which it is developed for virtual reality and augmented reality applications.

B) Standard web camera:

A standard web camera present Infront of PC's can be used for gesture recognition where it would be convenient for other forms of image- based recognition.

C) PyCharm python IDE:

This IDE is chosen over the rest because it is cross-platform and for accessing smart built in developer tools, scientific tools and customizability.

D) Webcam driver:

It is used to recognize the webcam associated with that device and for proper functioning of the webcam.

E) Tensor flow:

Tensor Flow is a Python library for fast numerical calculations. It is a basic library that can be used to create deep learning models or use wrapper libraries to simplify the process of building on Tensor Flow.

F) OpenCV:

This library is used to access the webcam when we run the code for gesture recognition.

G) Multiprocessing:

It is a package that requires an API similar to the threading module to support process spawning. This package also provides 13 local and remote concurrencies, effectively by-passing global interpreter blocking by using threads instead of threads.

H) Cv2:

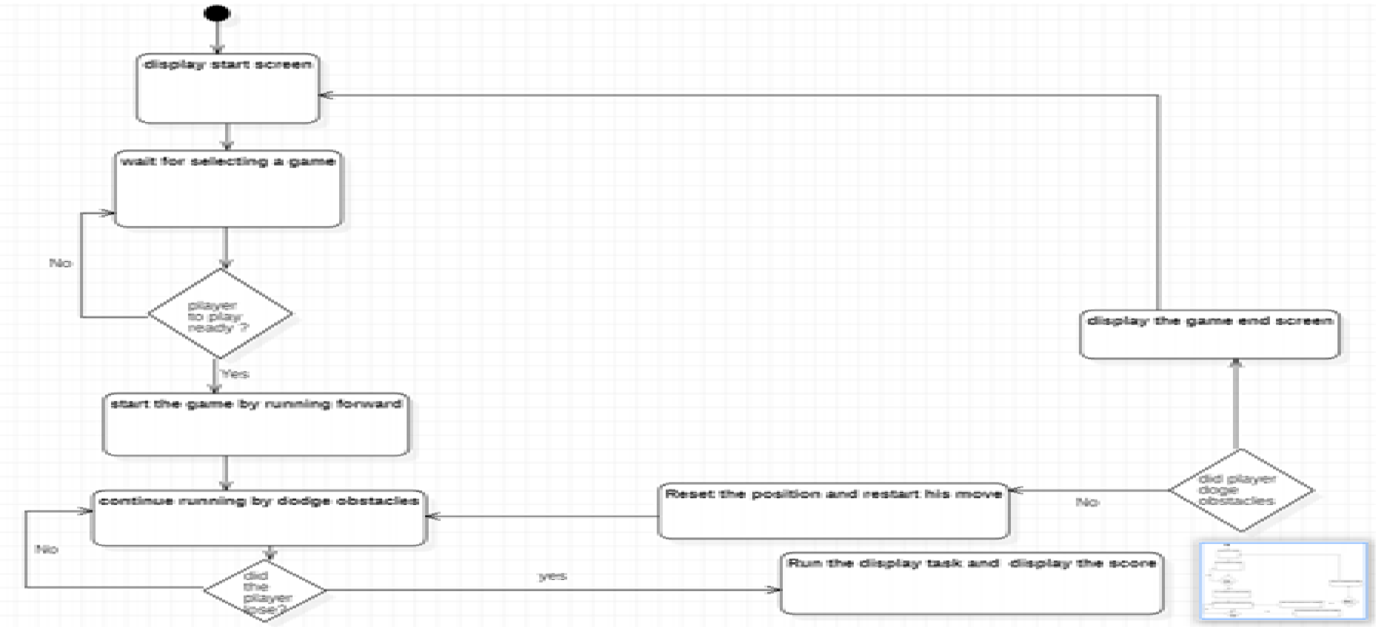
OpenCV is a cross-platform library, which is used to develop real-time computer vision applications. Mainly we use this package for image processing, video capture and analysis, including features such as face detection and object detection.

I) Nespy:

nes-py is an NES emulator and OpenAI Gym interface based on the SimpleNES emulator, suitable for MacOS, Linux and Windows.

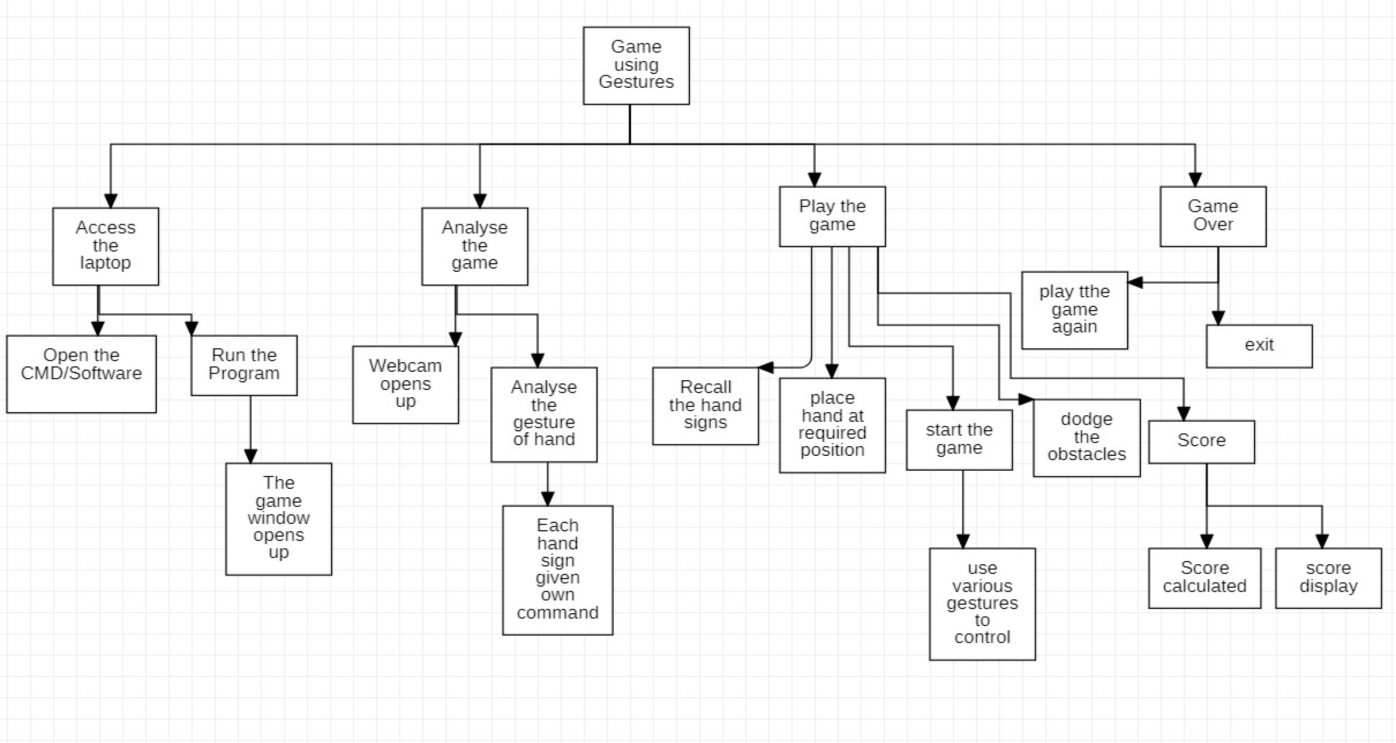
PROCESS MODEL:

The game is implemented using well-ordered set of activities to give an abstract description based on the player's needs utilizing specifying, testing, validating, and evaluating the system based on the user needs.


display start screen
wait for selecting a game
No
player to play ready ?
Yes
start the game by running forward
continue running by dodge obstacles
No
did the player lose?
yes
Run the display task and display the score
Reset the position and restart his move
No
did player doge obstacles
display the game end screen

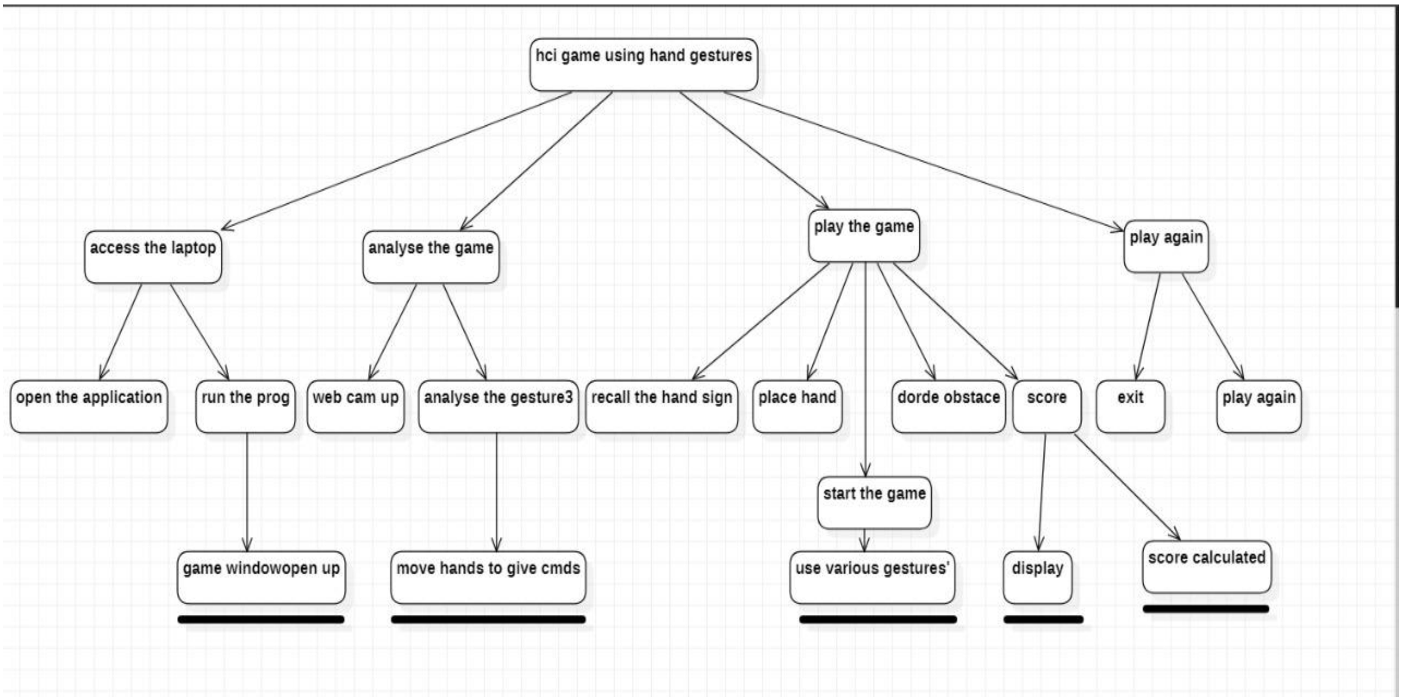
BLOCK DIAGRAM:

Here we describe the fundamental processes and how the system works through the interconnection of blocks, where each block depicts the working mechanism of a particular component for the user-game interface design.



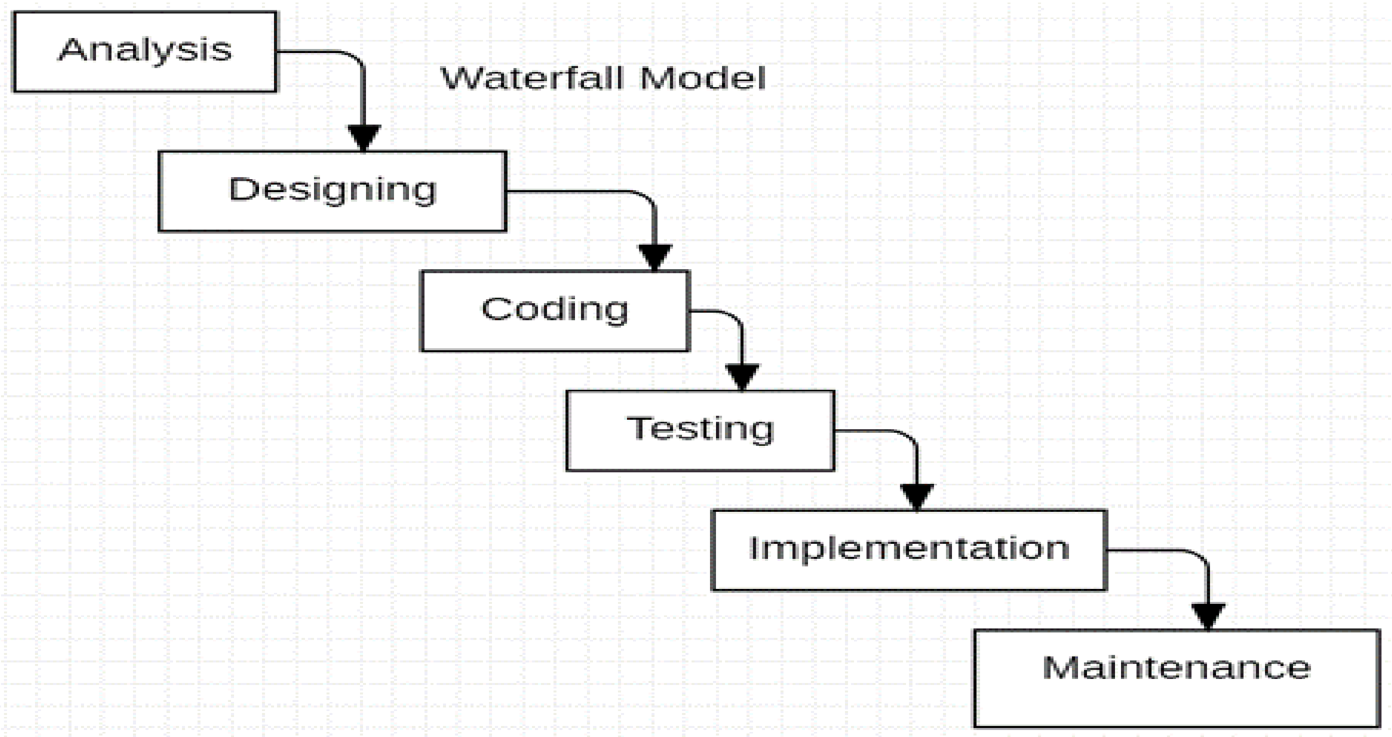
HTA:

Hierarchical Task Analysis is one of the most effectively way of deciding what tools must the Game developers must use in order to design the games as this approach helps them to either easily modify the existing model with the better alternative solution or when creating a new design in order to achieve required goal.



INTERACTION DESIGN (4 STEPS):

This design helps in improving the interaction experience of the gamers with the product at the time of interaction. It is also used for testing and specification for interface designers and game developers.



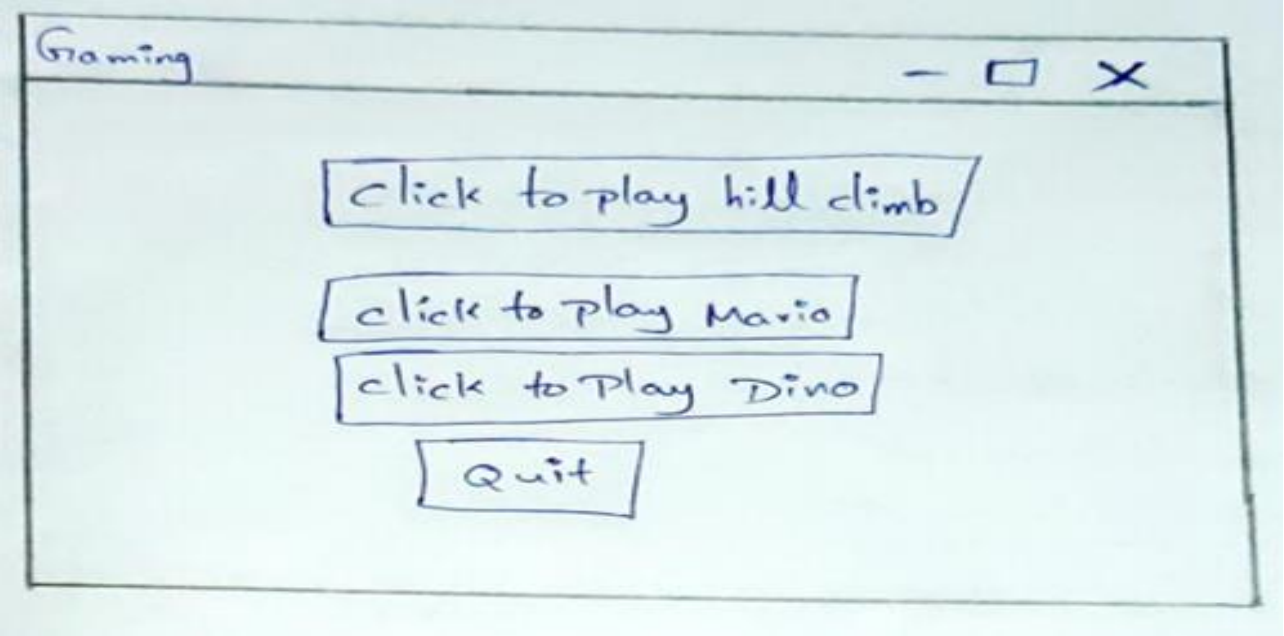
STAKE HOLDERS IDENTIFICATION:

1. **User/Customer:** With respect to our project one of the important stakeholders are the players/gamers who come under the description of users who wish to purchase the game.
2. **Project Member:** Team members who want to showcase their skills in the game development, image processing, python developer and related technical aspects of this project.
3. **Management team:** they manage the important decision making from the project managers by listening to their opinions before making a decision in order to work as a team in coordination to reach the goal. Therefore, management team plays an important role as a primary stakeholder.
4. **3rd Party companies:** There might be various problems from the complexities with integration to the inabilities to use the third-party solutions. They also come under the primary stakeholder's class.

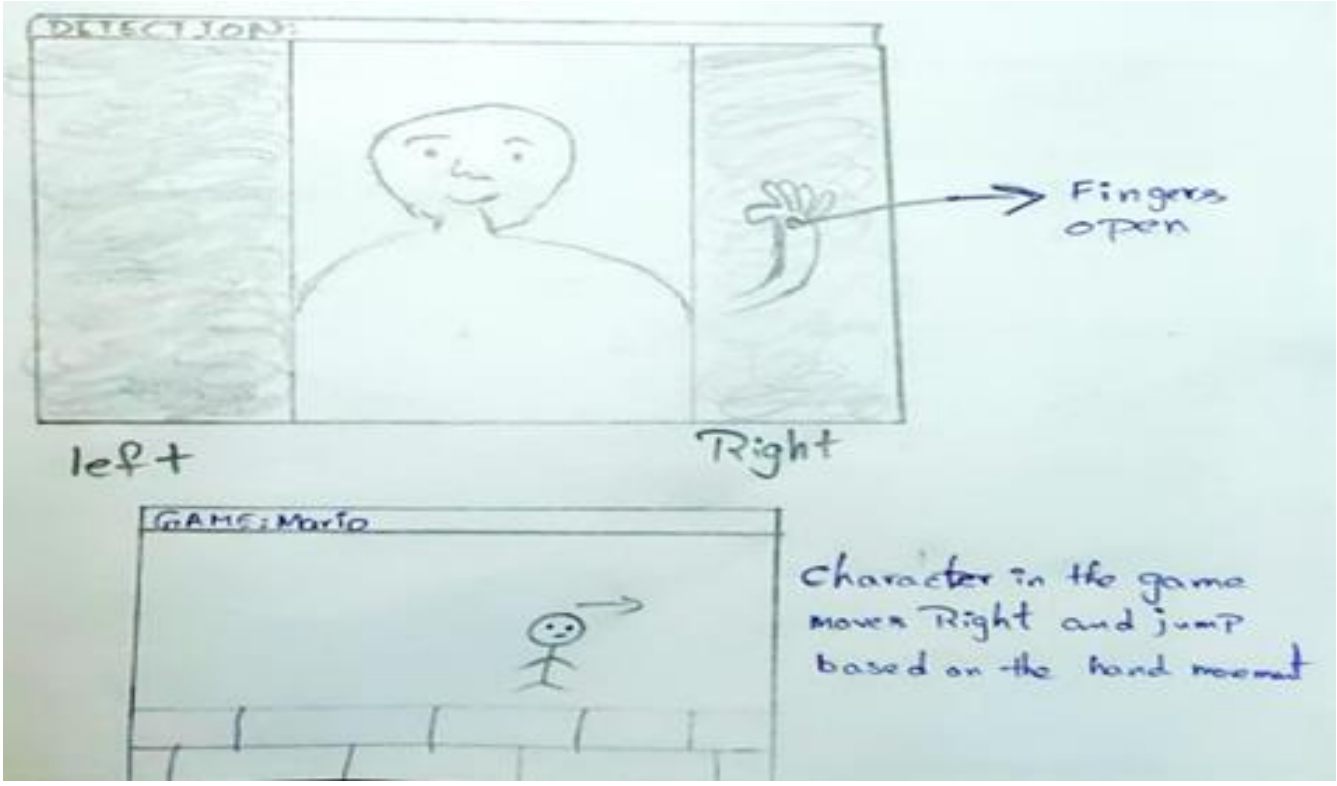
STORY BOARDING:

To make the interface more interactive and user-friendly we make use of Storyboarding as this process uses illustrations or images in a sequential manner to illustrate the important steps of the user experience. This process works effectively as the producer and engineer can modify it based on the user needs before actually releasing the product into the market.

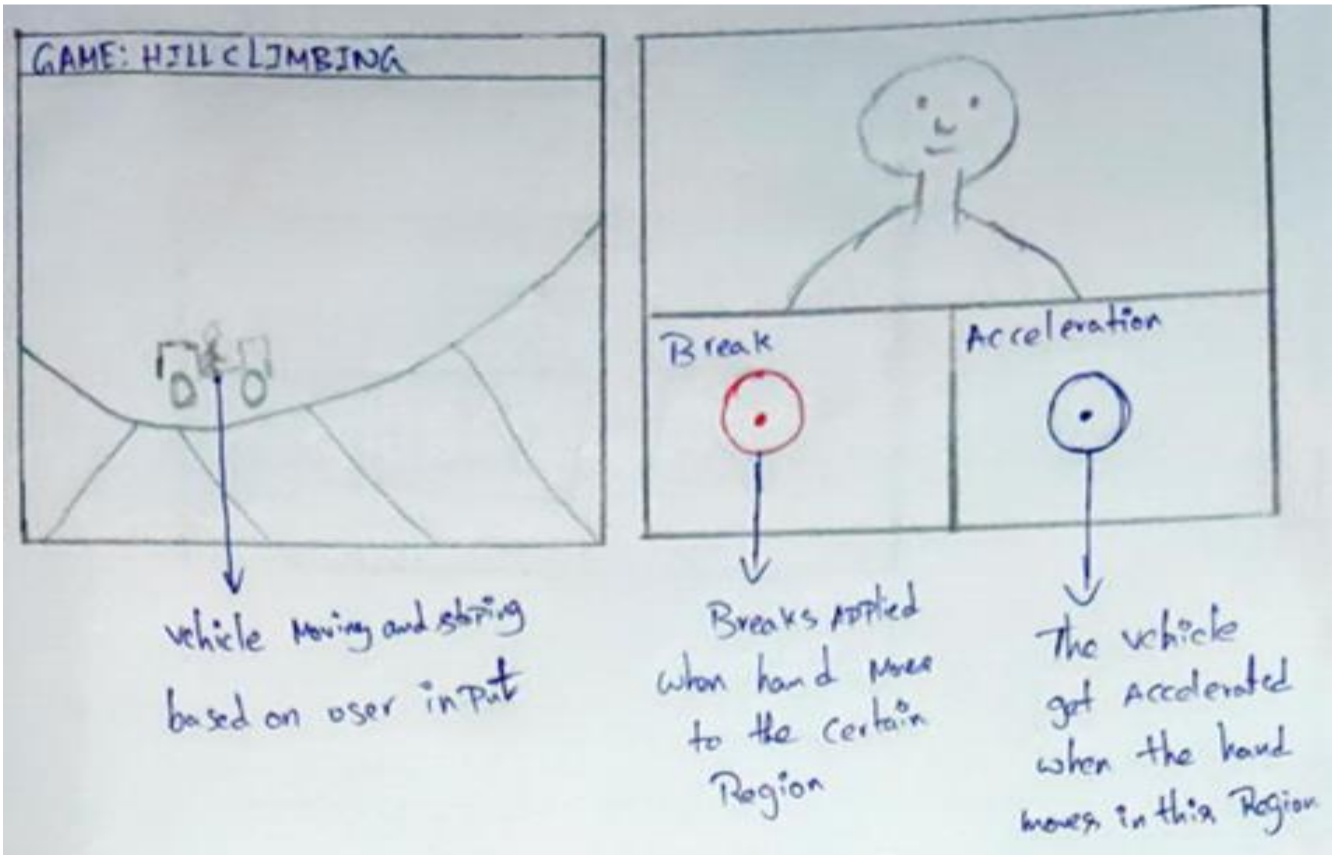
1) **Home page:** This is the home page where user can select a specific game.



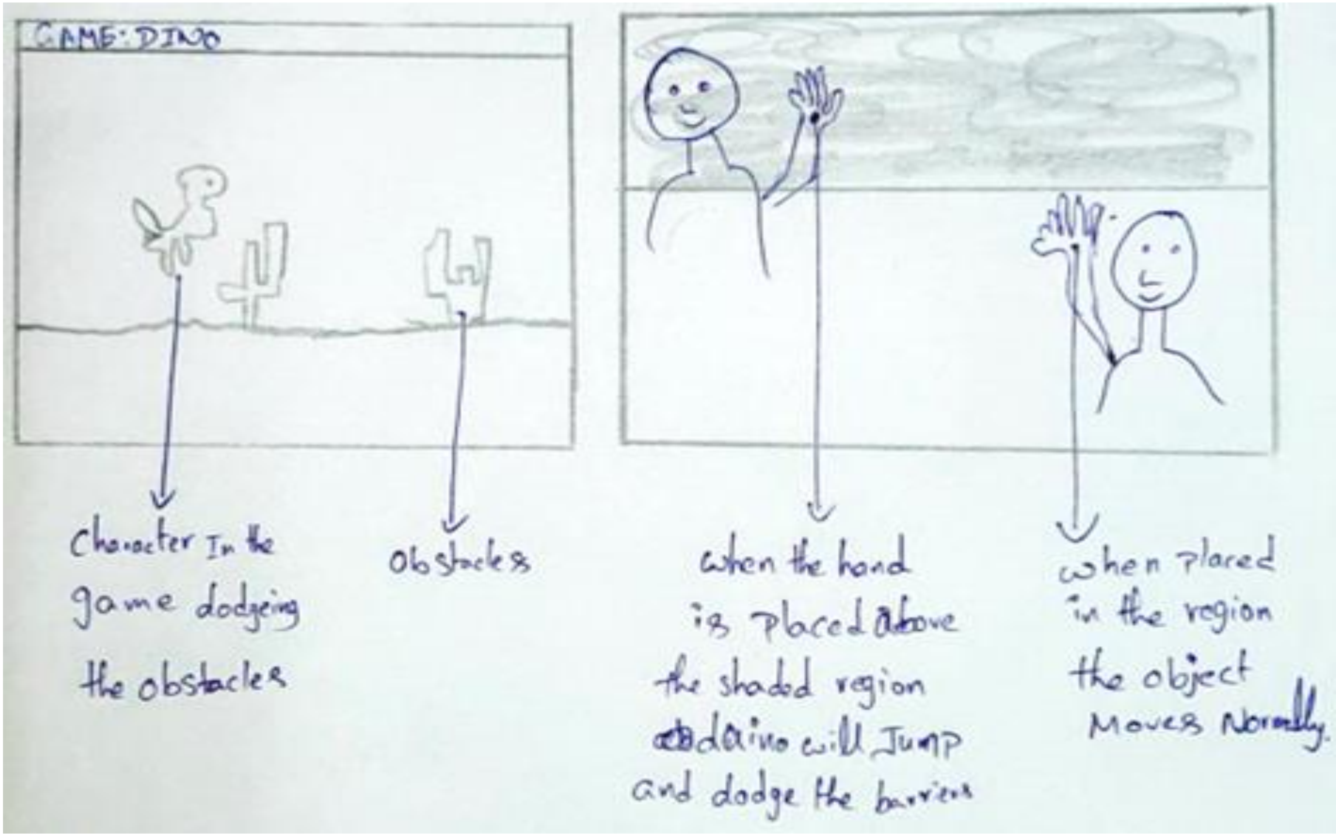
2) **Mario Game approach from user:** Here the user can control the character in the game by giving instructions through hand based on palm-recognition algorithm.


DETECTION:
Fingers open
left
Right
GAME: Mario
Character in the game moves Right and jump based on the hand moement

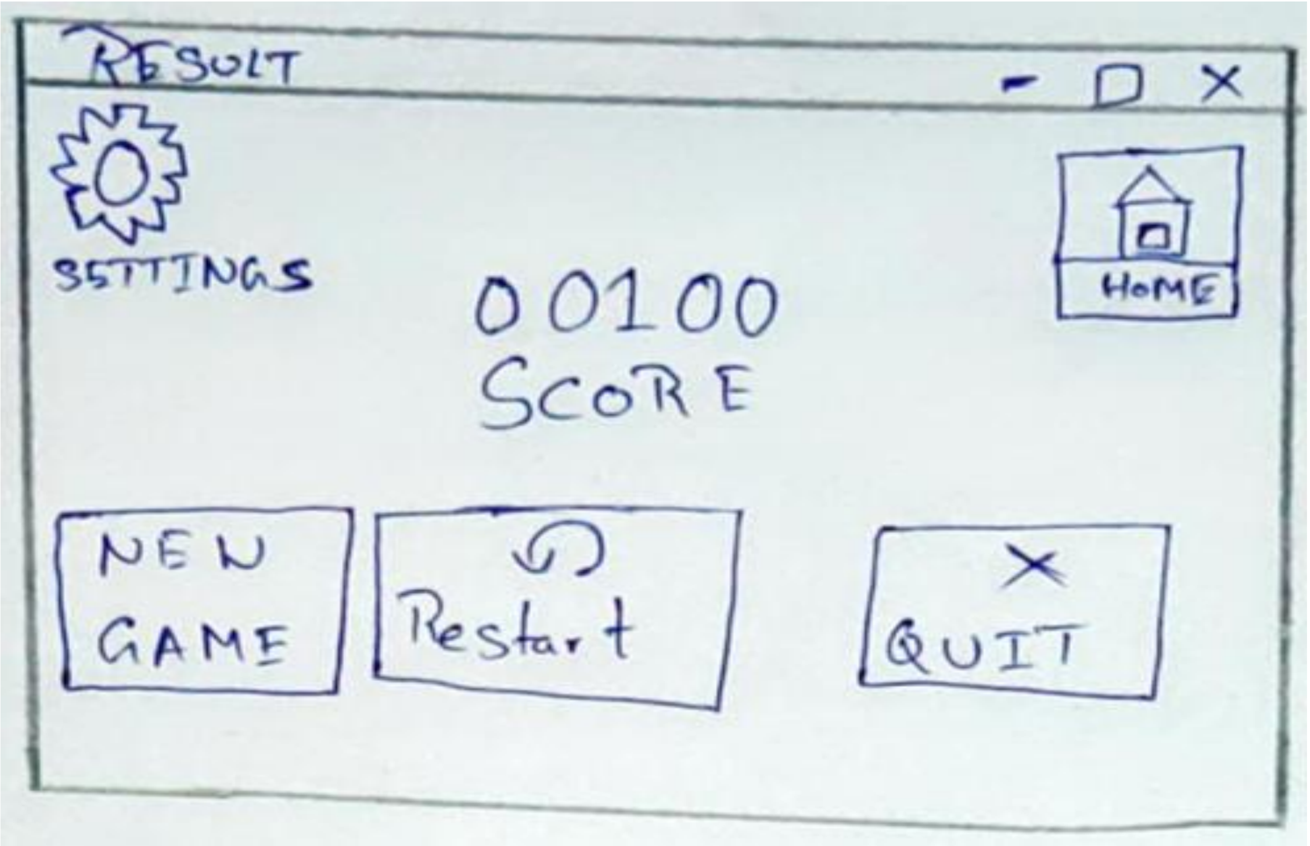
3) **Hill Climbing Game approach from user:** The user may operate the vehicle in game by issuing commands with their hands using a blue color Detection Algorithm.


GAME: HILL CLIMBING
Break
Acceleration
vehicle Moving and stiring based on user input
Breaks applied when hand Moves to the certain Region
The vehicle got Accelerated when the hand moves in this Region

4) **Dino Game approach from user:** As the previous game where the character was controlled by palm-recognition Algorithm here the same approach follows.

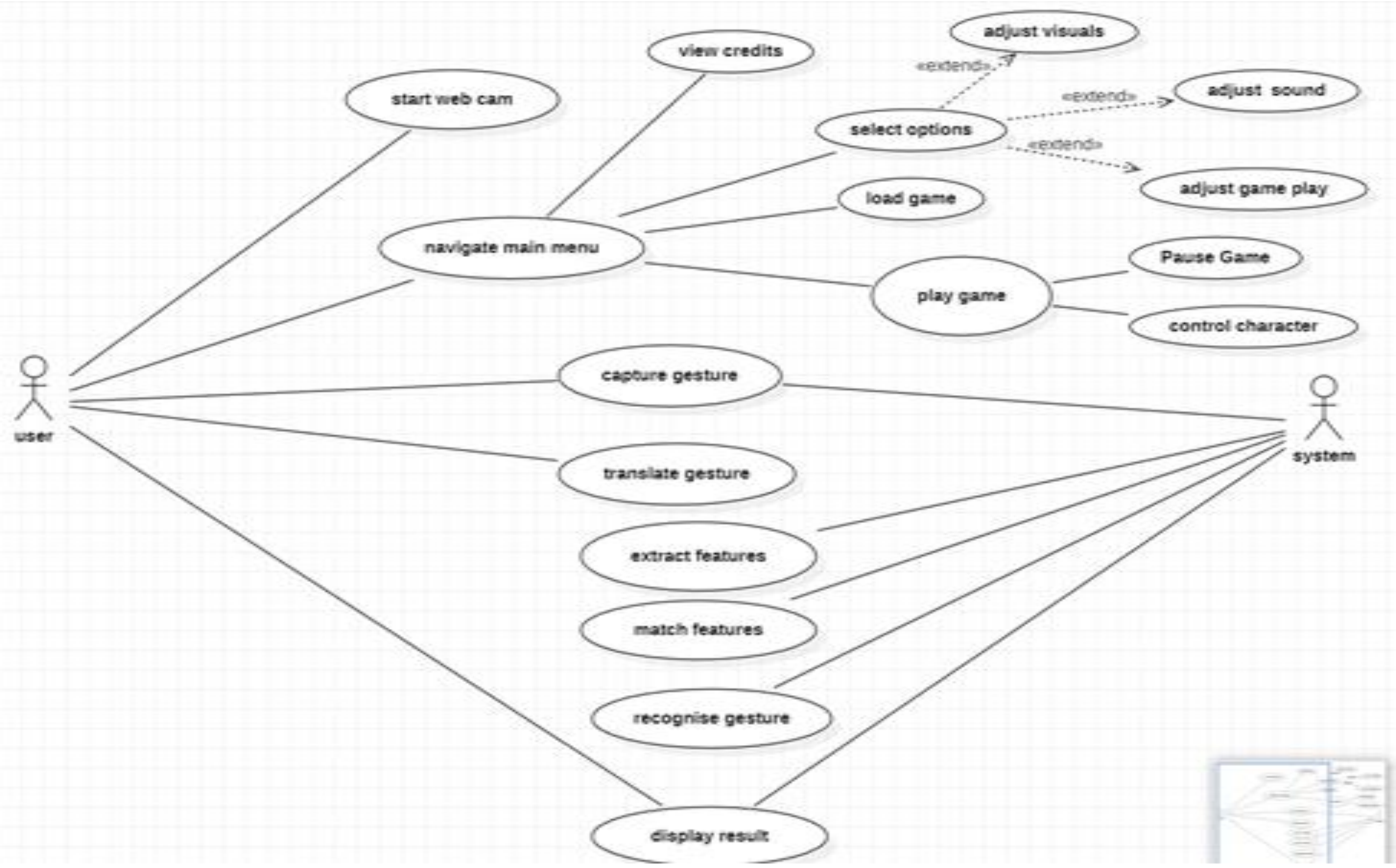

GAME: DINO
Character In the game dodgeing the obstacles
Obstacles
when the hand is placed above the shaded region dino will Jump and dodge the barriers
when placed in the region the object Moves Normally.

5) **Final Score view to user:** This is a final page where the user gets feedback and various options.



USE CASE DIAGRAM:

This is how the process where the user interacting to the system interface to play game.



PERSONAS:

Goals:

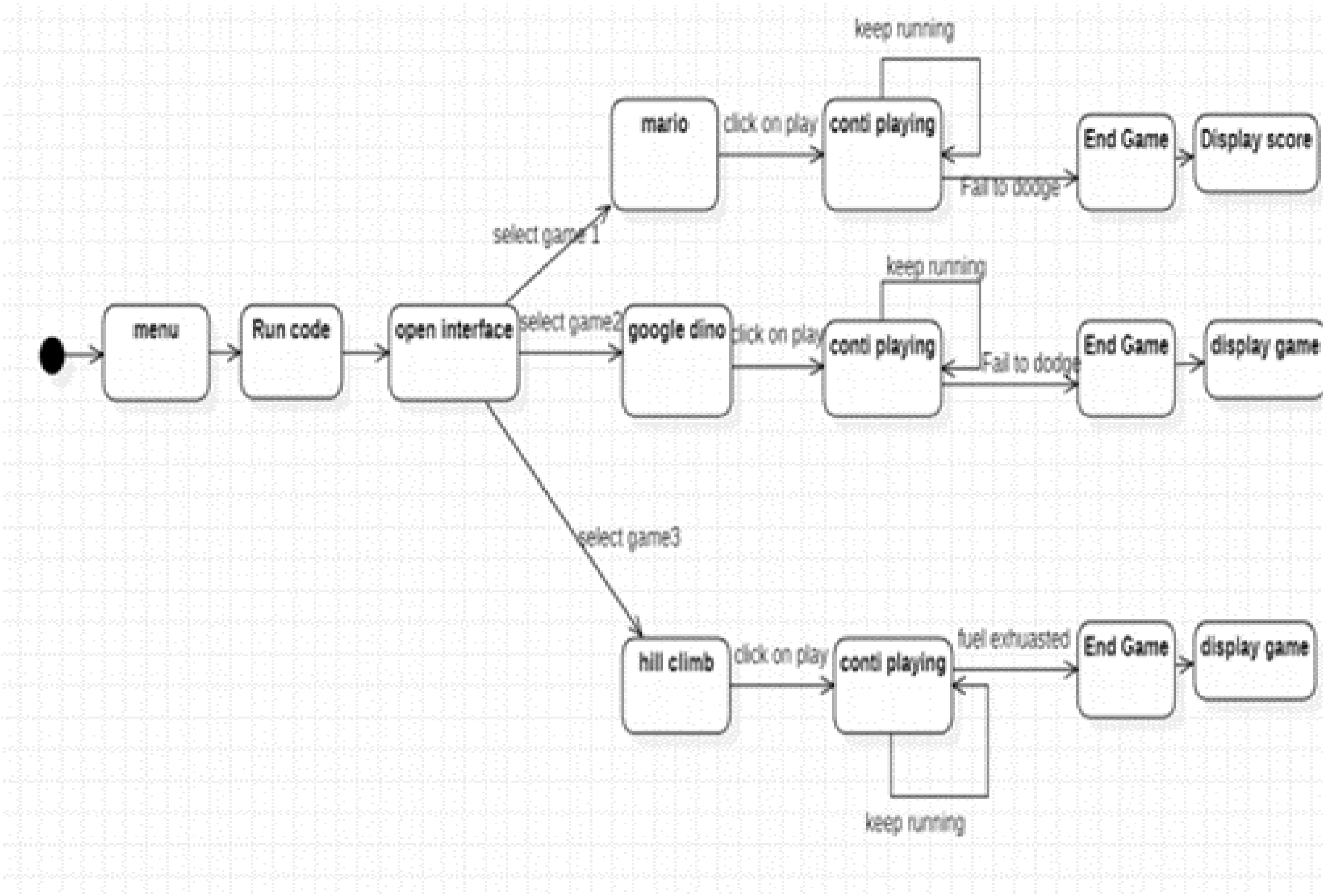
- To improve his mental health.
- To live a calm and peaceful life

1) Primary users: gamers, Designers, Developers and Marketers.

2) secondary users: distributors, media, government, parents.

STATE TRANSITION NETWORK:

While designing a user interface the designer must help the user easily understand the state of the device based on the actions he takes, as the STN which is not easy to be described by the system directly implies that it is difficult for the user to use it. This indirectly explains how the gesture-based interactions are lesser complex than that of the peripheral controlled interaction.



GOLDEN RULES:

1. **Strive for consistency:**

Our interface strives for consistency as in both peripheral and hand gesture modes of gaming, all the places for the respective icons are in the same placeand it makes the users inhabit it quickly.

2. **Enable frequent users to use shortcuts:**

It is enabling users to use shortcuts as we can change the settings as per the convenience like palm moving up/down for dodging and palm movement to left/right for running etc.

3. **Offer informative feedback:**

After putting our palm in the ROI a text appears in a color indicating move left,move right, jump up, dodge down etc., which enable the user to understand what they are selecting while playing game.

4. **Design dialogue to yield closure:**

When the game ends the score will be displayed as a design dialogue on the screen and the player will be asked if he would like to replay the game or endthe game by moving to home screen etc.

5. **Offers simple error handling:**

The player might forget to adjust and confirm the settings or choose a player (Mario or Luigi) or difficulty level. So, it has to be made sure the userknows about the error made for smooth functioning of the game.

6. **Permit easy reversal of actions:**

Although the gaming interface need not have a reversal action the user can save the game using pause option and continue later once, they are back whenever they want.

7. **Support internal locus of control:**

The interface designed makes the gamers get the feel of having total controlover the system and the system responds to their actions.

8. **Reduce short term memory load:**

Here the game and its interface are made in such a way that the user need not remember each and every operation as in the case of the peripheral gaming mode. Here the gamers need less training time to get used to the interface as itseasier and has lower memory load on them.

Jakob Nielsen Principles in accordance with our project:

Visibility of system status:

Players are always informed about their status in the game and the action/control details as they play the game.

User control and freedom:

All the controls are given solely to the user for any particular game. No controls are hidden or not accessible through hand gestures. Also, the exit option is pretty clear.

Consistency and standards:

Consistency is tried to be maintained throughout the project by evenly making use of hand gestures in a related fashion. Same UI is tried to be maintained for each game.

Recognition rather than recall:

We chose games with limited control options, hence there is no need of remembering enormous amount of hand gestures for controlling any Particular game. Also, the UI provides the dialogs and feedback for the actions being performed hence the actions are easily recognizable.

Aesthetic and minimalist design:

The design for this project is kept as minimalistic as possible to rule out the possibility of errors. Appropriate dialogs are also provided in the game in a very concise way.

Help and Documentation:

We are preparing the proper documentation side-by-side depicting the implementation of the project and guide to its proper usage.

Norman’s fundamental design principles in accordance with our project

Discoverability:

The software is designed with well discoverable and self-explanatory architectural design. Anyone can very well understand how the program works with little or no aid.

Feedback:

At each moment feedback is being given depicting the action being performed for each game. Like running, standing still, jumping and moving left or right.

Signifiers:

We have used control signifiers in the project which tells us about a particular action being performed currently.

KLM:

It is a model which gives an idea about the time to take to complete an operation in any website for various users.

Here we are going to analyse the time taken in our platform for a visualization of algorithm.

calculating the time:

1. reach the mouse to open particular software to open **the time taken would be 0.4 sec.**

2. Initially we have to click on our software to run the code here assuming that already having an icon for this software. **The time taken would be 1.1 sec**

3. Then click the "Run button" for opening the interface **The time taken would be 1.1 sec**

4. Thinking/ decision making task to select which game to be open **the time taken would be 1.35sec.**

5. point to correct field to select a game **The time taken would be 1.10**

6. click a game to play. **The time taken would be 0.20**

7. decision making task to dodge the obstacles **The time taken would be 1.35sec**

8. failed to dodge the obstacles then need to think whether to play another game or to quit

The time taken would be 1.35sec

9. if we want to quit the game click quit option **The time taken would be 0.20**

10. After quitting the game need to stop the program **The time taken would be 0.20.**

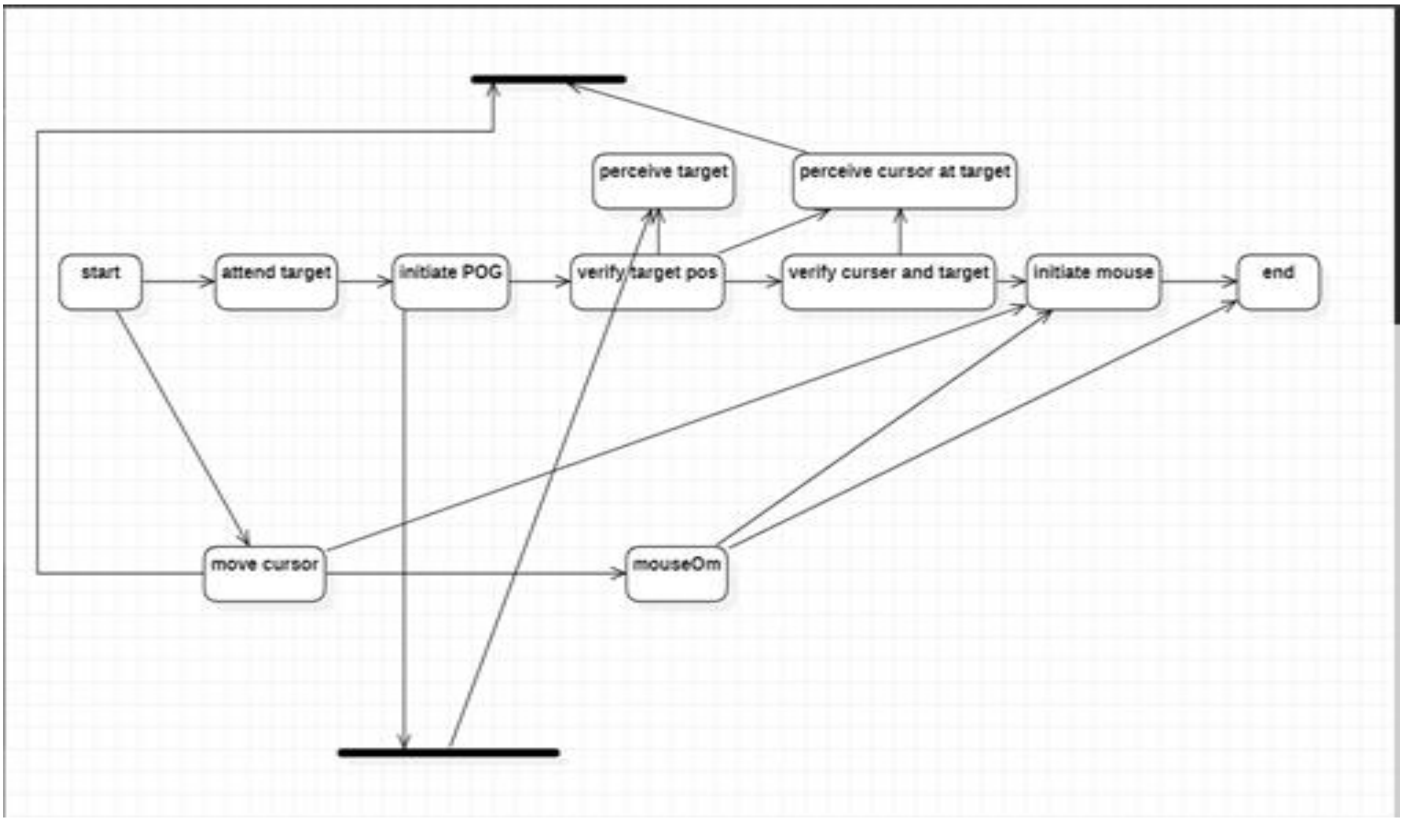
GOMS MODEL:

1.Goals(G): As a task to play the game using hand gestures.

2.Operators(O): As all actions needed to achieve the goal by moving the player swiping either right or left or up and down.

3.Methods(M): As a group of operators as we move the hand in appropriate direction these methods can be implemented through various methods like using image processing or machine learning etc.

4.Selection(S): As a user decision approach among the various potential methods of implementation we will select the image processing Algorithms like blue color detection and palm recognition.



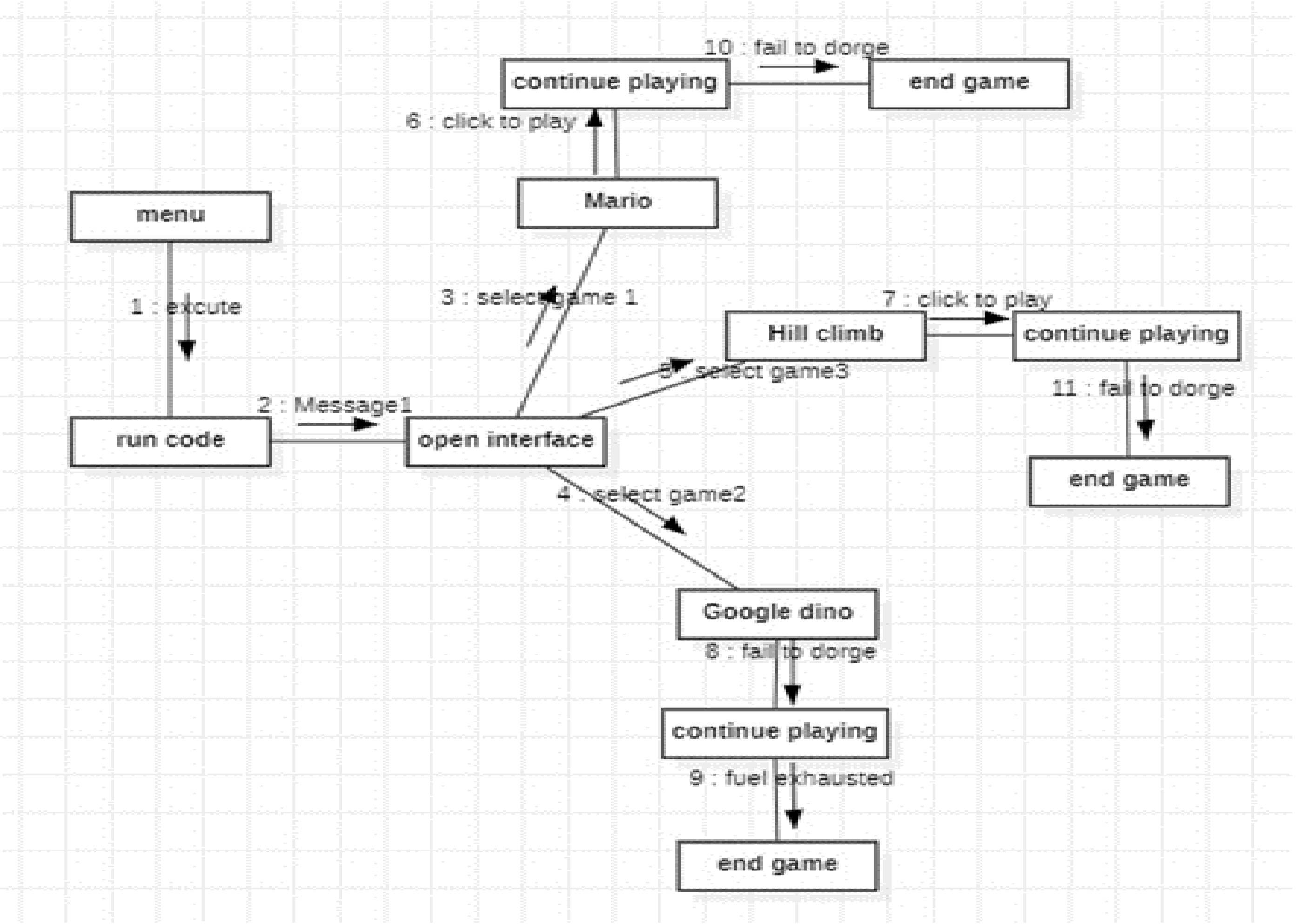
COMMUNICATION, COLLABORATION and Groupware:

1.**Simplify the file sharing process:** Anyone may utilize the interface since the file sharing process is simple for those who have been approved by the vendor.

2.**t desking:** Gaming is no longer like grammar school, where user has to sit in a fixed place near the system in a room to play. The gamers can enjoy the experience of gaming sitting anywhere like in lounge areas, couches, and cafeterias etc...

3.**Don't settle for your current software:** The variety of popular libraries used can be installed by the gamer having no fear of which software they are using.

4.**Reducing inefficiencies in gaming:** a greater number of requirement specifications needed for the game to run might not interest the users because the failure in any of the components may cause inefficiency in user experience hence this project implementation takes care of such possibilities.



Algorithm:

Here we will be using few global variables to set up a background function. So we shall first define some global variables and then afterward we'll set up a function that updates a running average of the background values in a particular region of interest.

This will, later on, allow us to detect new objects such as a hand entering into the region of interest. First of all, we need to have a region of interest. Then we're going to calculate a running average background value for some number of frames of video. Once that average value is found, then we can have the hand enter the region of interest once a hand actually enters, we can detect the change and then apply some thresholding techniques to isolate the hand and isolate that hand segment.

Once we segment the hand region that we obtained through segmentation once we do it we apply the thresholding to the segments in the appropriate manner. Contours as simply a curve help in joining all the continuous points along the boundary, having the same color or same intensity contours are a very useful tool for shape analysis and object detection and recognition.

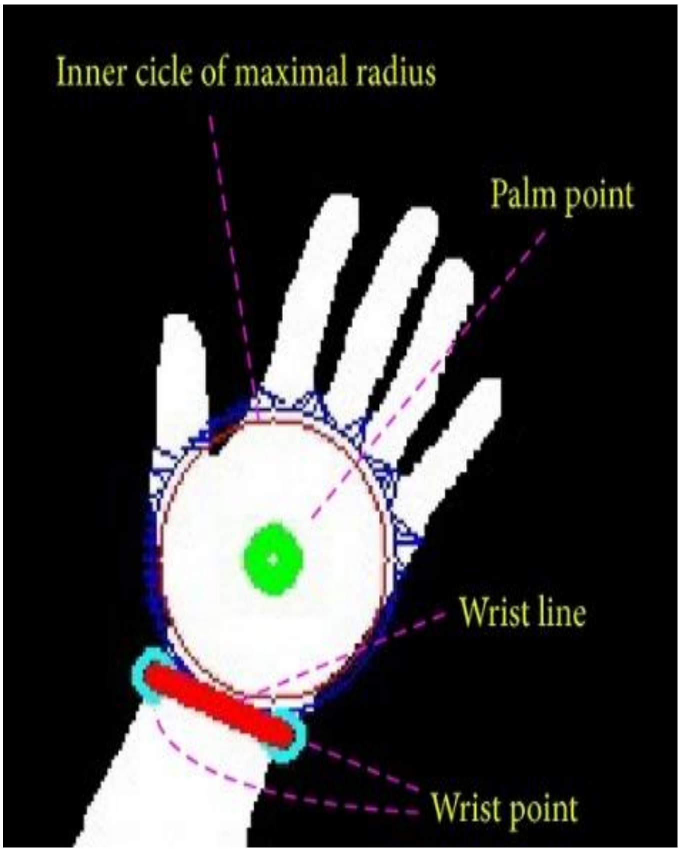
We now then look at Palm Recognition, firstly, we detect the skin precisely by detecting the characteristics of the skin where the skin color gets recognized to act accordingly. Based on this criterion we can efficiently segment the skin portion of image.

We later look at the preprocessing to improve the accuracy of the hand capture by correcting the holes and cracks in the hand segment. In our algorithm, we use the diffusion co-efficient to measure distance and direction between the damaged pixels which will be removed in form of noise. A common operation for image processing is blurring or smoothing an image and smoothing an image can help get rid of the noise or help a computer vision application focus on general details.

We have the Hand segment; the next step is actually to count the fingers being held up we can do this by utilizing a convex hull. A convex hull draws a polygon by connecting points around the most external points in a particular frame or a set of contours passed into that frame by calculating the external dots and then drawing a polygon, connecting them all. In our case, it's going to be the hand that is the input. Depending on whether those points or those fingers are extended, those points are going to be closer or further away from the center of the hand. So, the general shape of the polygon will be extracted this way.

<u>Summarizing the process:</u>

1. Skin Color Detection
2. Contour Extraction and Hand Region Segmentation
3. Image Preprocessing
4. Gesture Recognition and comparison
5. Recognized Gesture (Open or Close Hand)



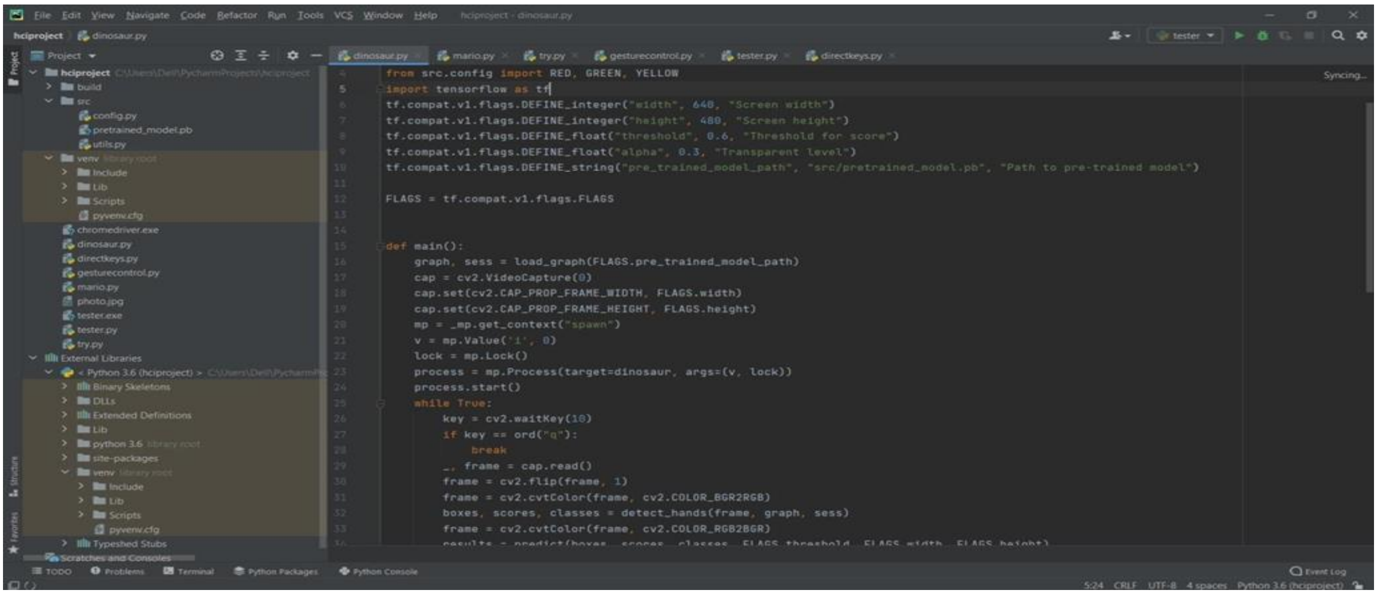
Implementation of Project:

Programming Information:

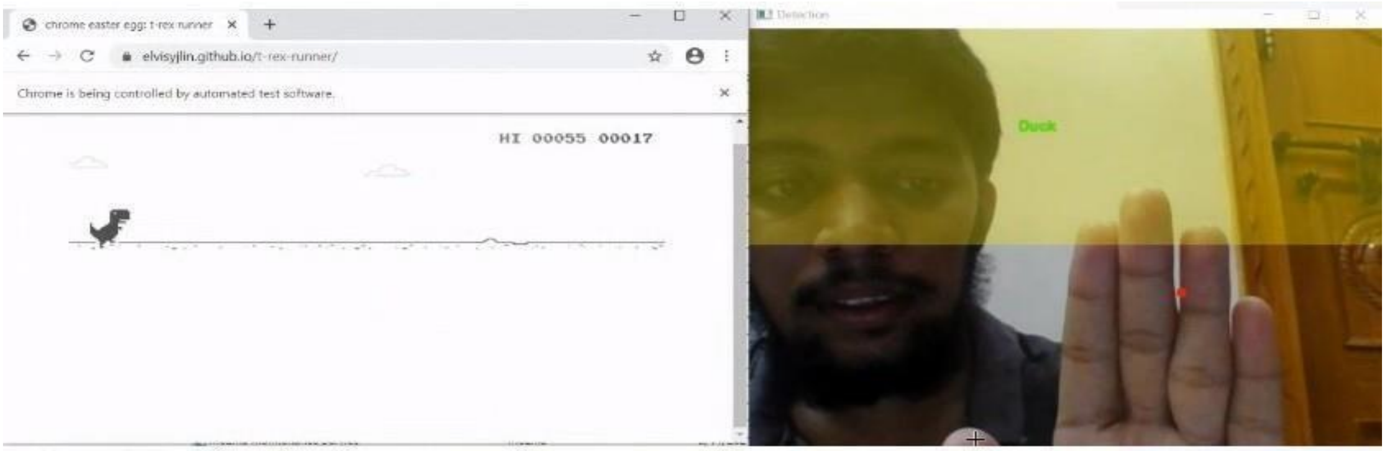
We will be using Python language and Python libraries and some APIs for coding of the project. Python is easy to understand and is best for programming of large projects. Some of the libraries and API's we are going to include in our projects are:

CODEs:

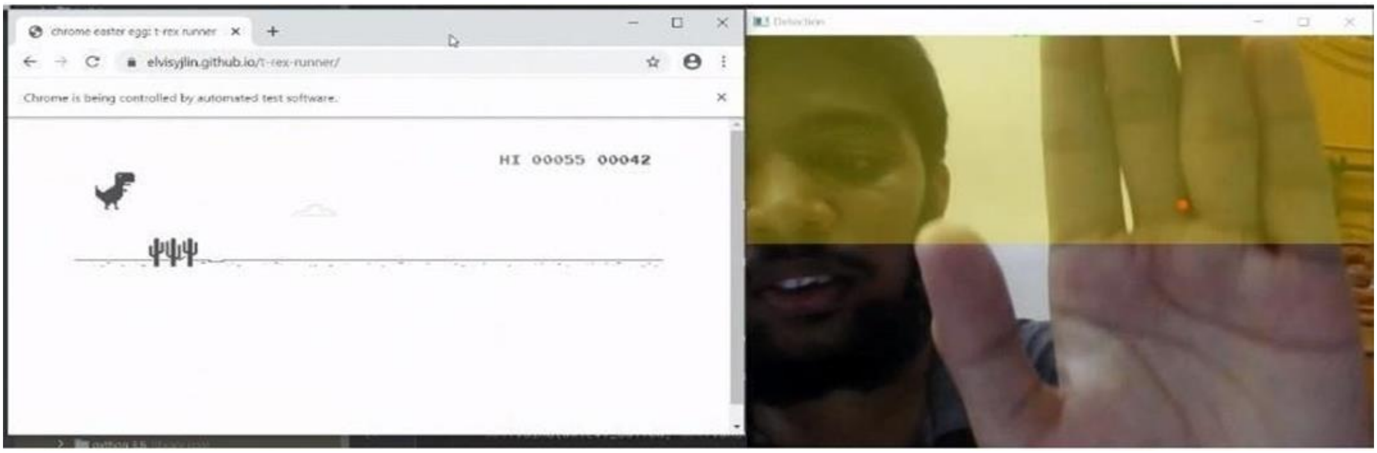
For Dino Game:



```
from src.config import RED, GREEN, YELLOW
import tensorflow as tf
tf.compat.v1.flags.DEFINE_integer("width", 640, "Screen width")
tf.compat.v1.flags.DEFINE_integer("height", 480, "Screen height")
tf.compat.v1.flags.DEFINE_float("threshold", 0.6, "Threshold for score")
tf.compat.v1.flags.DEFINE_float("alpha", 0.3, "Transparent level")
tf.compat.v1.flags.DEFINE_string("pre_trained_model_path", "src/pretrained_model.pb", "Path to pre-trained model")

FLAGS = tf.compat.v1.flags.FLAGS


def main():
    graph, sess = load_graph(FLAGS.pre_trained_model_path)
    cap = cv2.VideoCapture(0)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, FLAGS.width)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, FLAGS.height)
    mp = _mp.get_context("spawn")
    v = mp.Value('i', 0)
    lock = mp.Lock()
    process = mp.Process(target=dinosaur, args=(v, lock))
    process.start()
    while True:
        key = cv2.waitKey(10)
        if key == ord("q"):
            break
        _, frame = cap.read()
        frame = cv2.flip(frame, 1)
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        boxes, scores, classes = detect_hands(frame, graph, sess)
        frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
        results = predict(boxes, scores, classes, FLAGS.threshold, FLAGS.width, FLAGS.height)
```

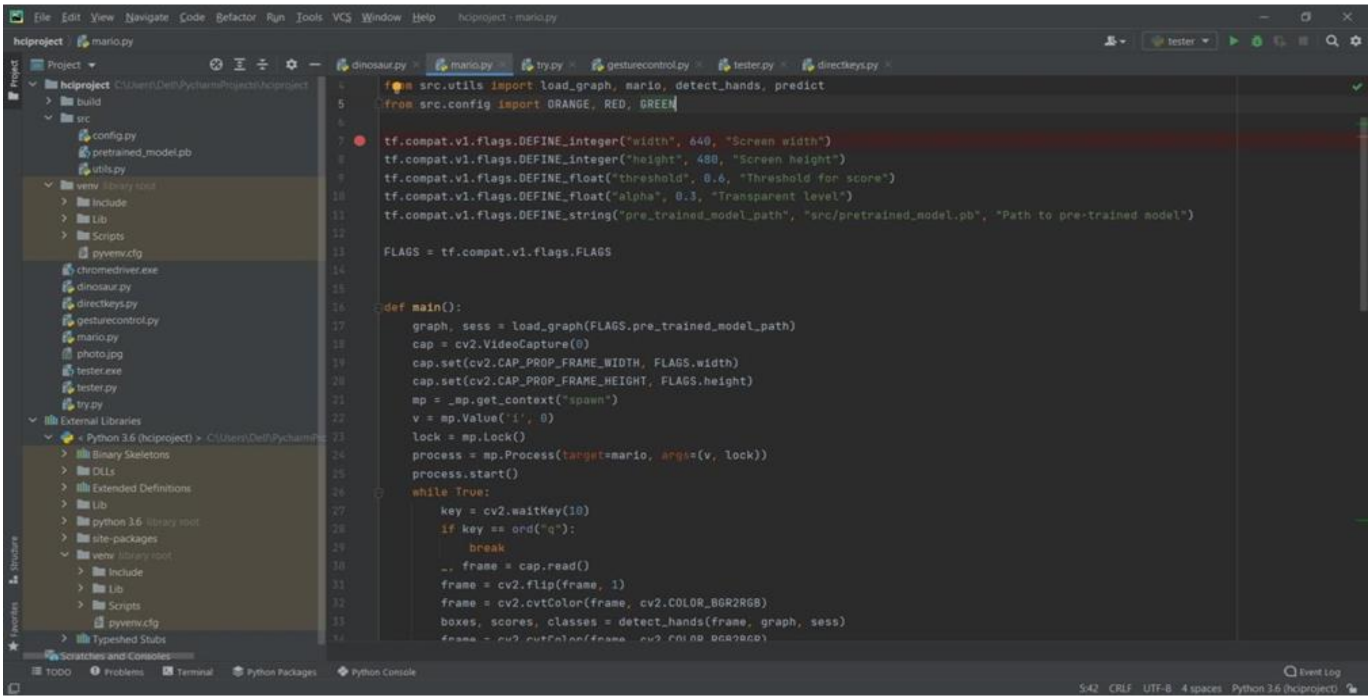

chrome easter egg: t-rex runner
elvisyjlin.github.io/t-rex-runner/
HI 00055 00017
Detection
Duck

Chrome is being controlled by automated test software.

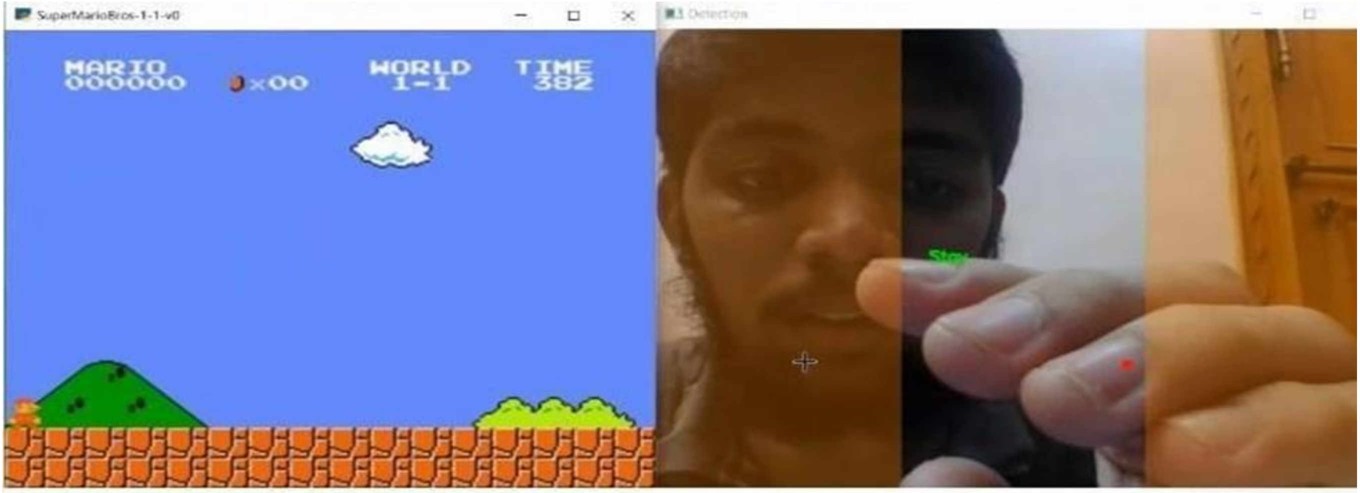

chrome easter egg: t-rex runner
elvisyjlin.github.io/t-rex-runner/
HI 00055 00042
Detection

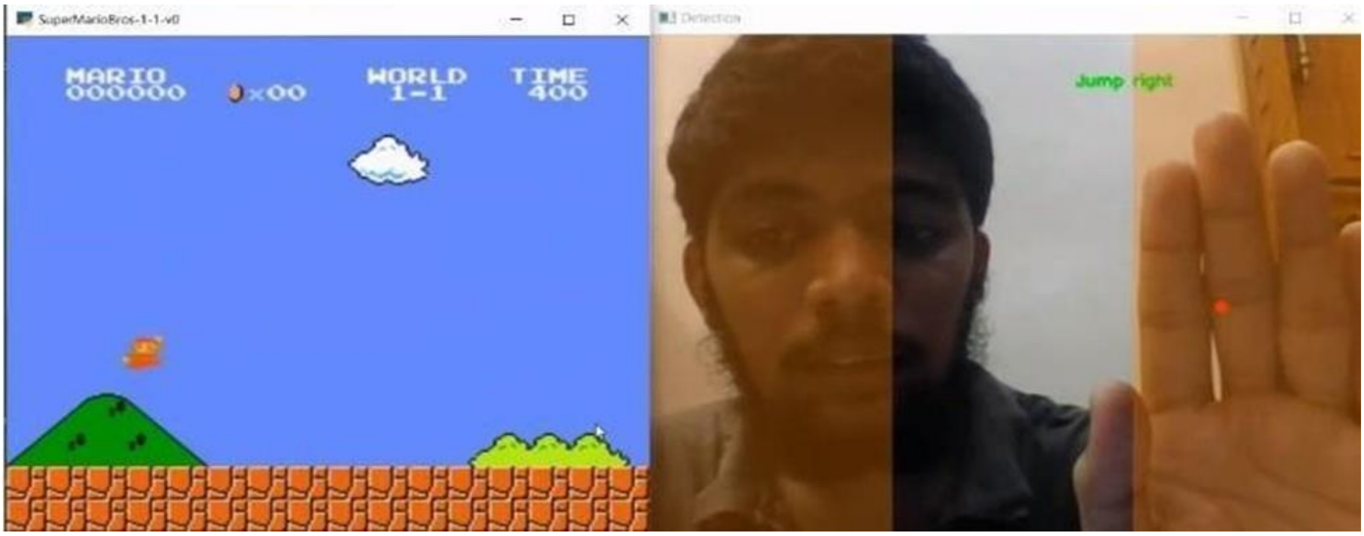
Chrome is being controlled by automated test software.

For Mario Game:

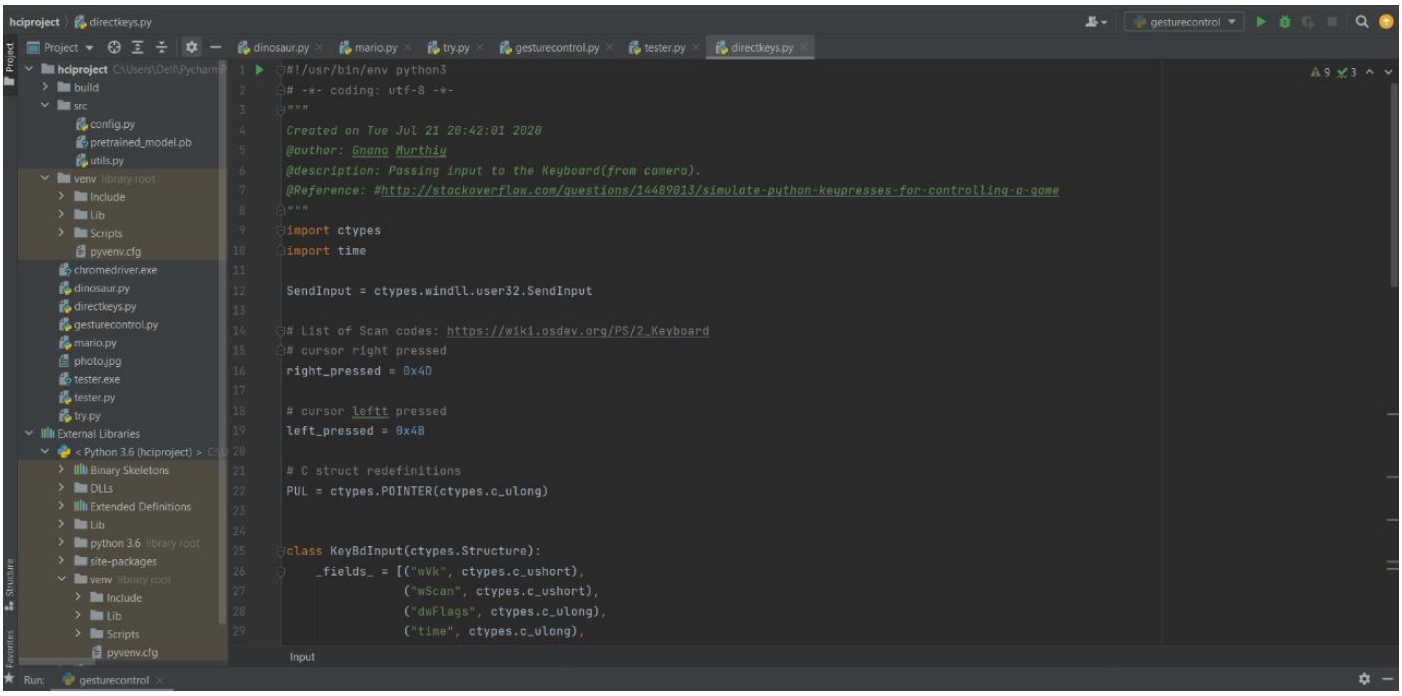


```
from src.utils import load_graph, mario, detect_hands, predict
from src.config import ORANGE, RED, GREEN

tf.compat.v1.flags.DEFINE_integer("width", 640, "Screen width")
tf.compat.v1.flags.DEFINE_integer("height", 480, "Screen height")
tf.compat.v1.flags.DEFINE_float("threshold", 0.6, "Threshold for score")
tf.compat.v1.flags.DEFINE_float("alpha", 0.3, "Transparent level")
tf.compat.v1.flags.DEFINE_string("pre_trained_model_path", "src/pretrained_model.pb", "Path to pre-trained model")

FLAGS = tf.compat.v1.flags.FLAGS


def main():
    graph, sess = load_graph(FLAGS.pre_trained_model_path)
    cap = cv2.VideoCapture(0)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, FLAGS.width)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, FLAGS.height)
    mp = _mp.get_context("spawn")
    v = mp.Value('i', 0)
    lock = mp.Lock()
    process = mp.Process(target=mario, args=(v, lock))
    process.start()
    while True:
        key = cv2.waitKey(10)
        if key == ord("q"):
            break
        _, frame = cap.read()
        frame = cv2.flip(frame, 1)
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        boxes, scores, classes = detect_hands(frame, graph, sess)
        frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
```

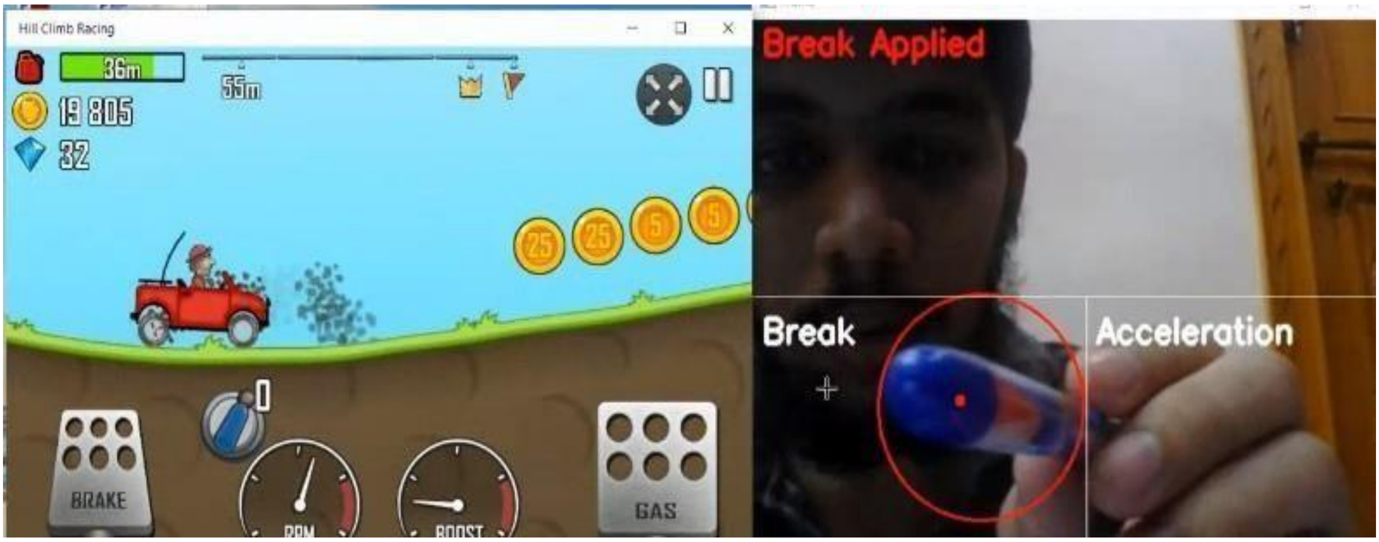


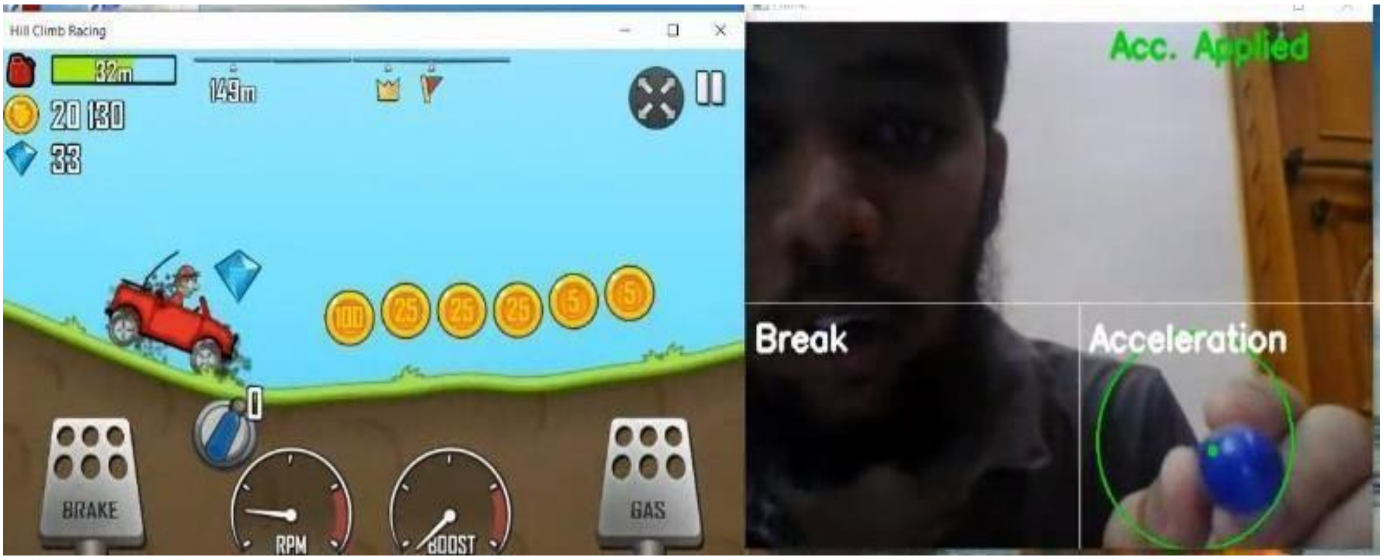


For Hill climbing Game:

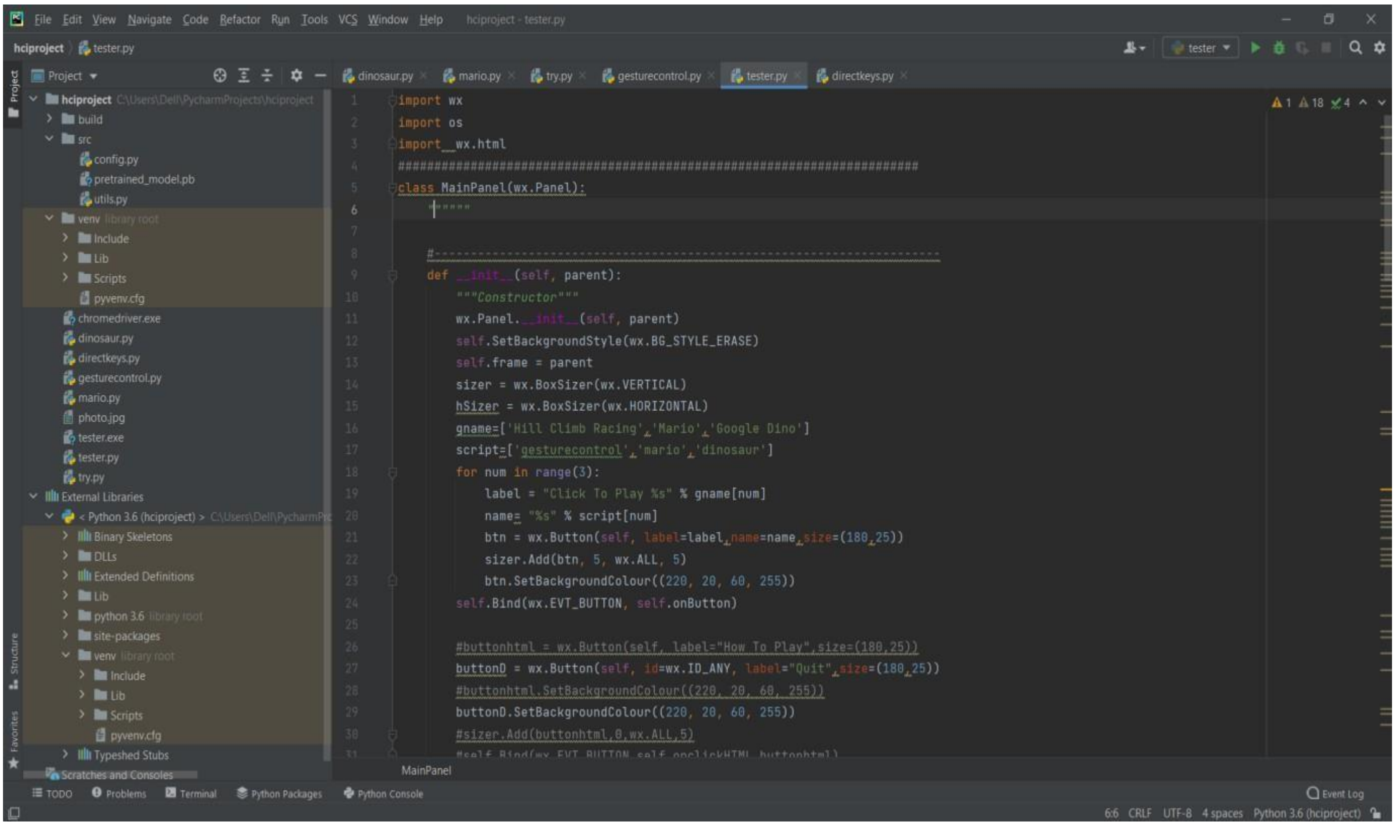


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jul 21 20:42:01 2020
@author: Gnana Murthiy
@description: Passing input to the Keyboard(from camera).
@Reference: #http://stackoverflow.com/questions/14489013/simulate-python-keypresses-for-controlling-a-game
"""
import ctypes
import time

SendInput = ctypes.windll.user32.SendInput

# List of Scan codes: https://wiki.osdev.org/PS/2_Keyboard
# cursor right pressed
right_pressed = 0x4D

# cursor leftt pressed
left_pressed = 0x4B

# C struct redefinitions
PUL = ctypes.POINTER(ctypes.c_ulong)


class KeyBdInput(ctypes.Structure):
    _fields_ = [("wVk", ctypes.c_ushort),
                ("wScan", ctypes.c_ushort),
                ("dwFlags", ctypes.c_ulong),
                ("time", ctypes.c_ulong),
```

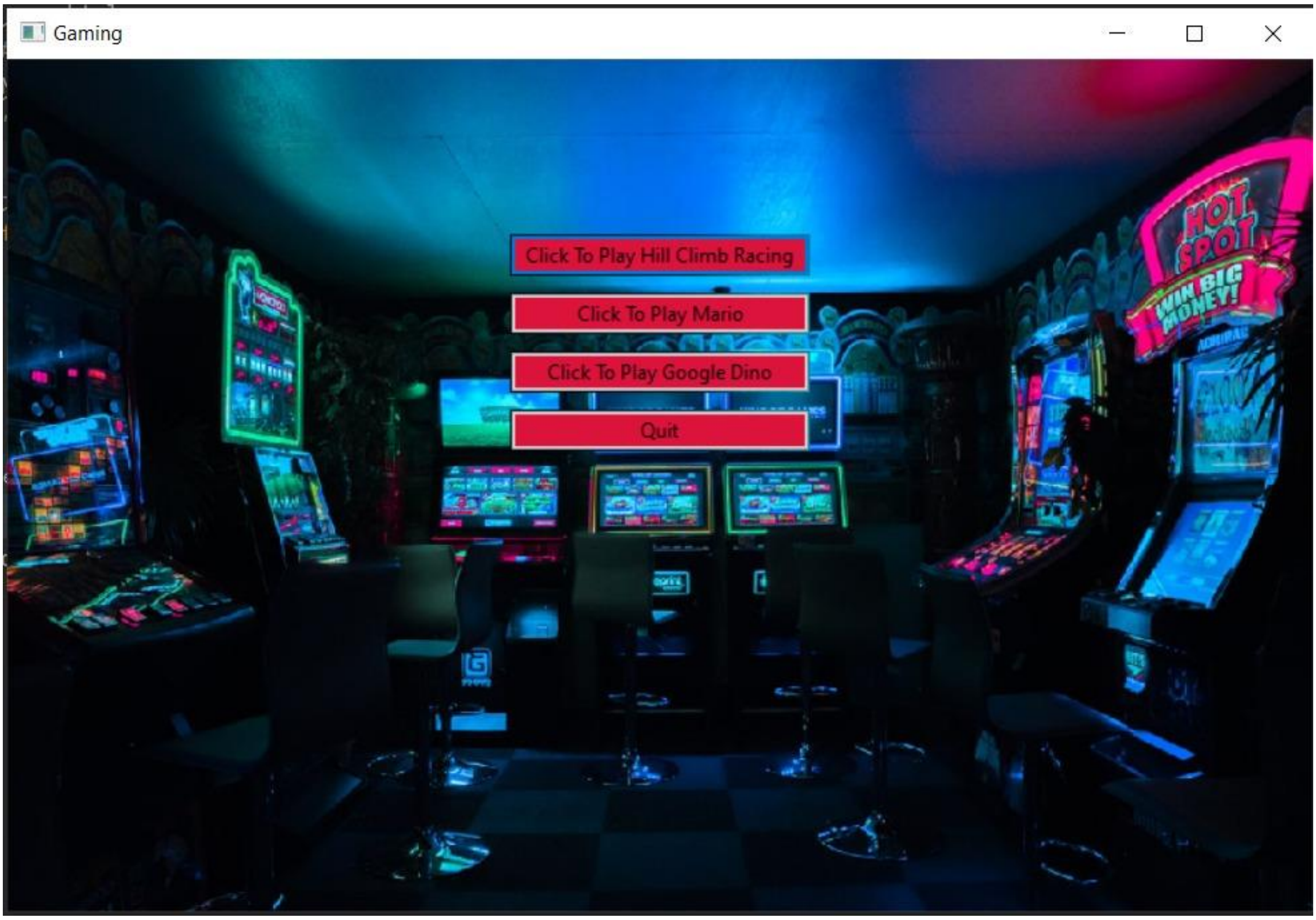




For Interface:



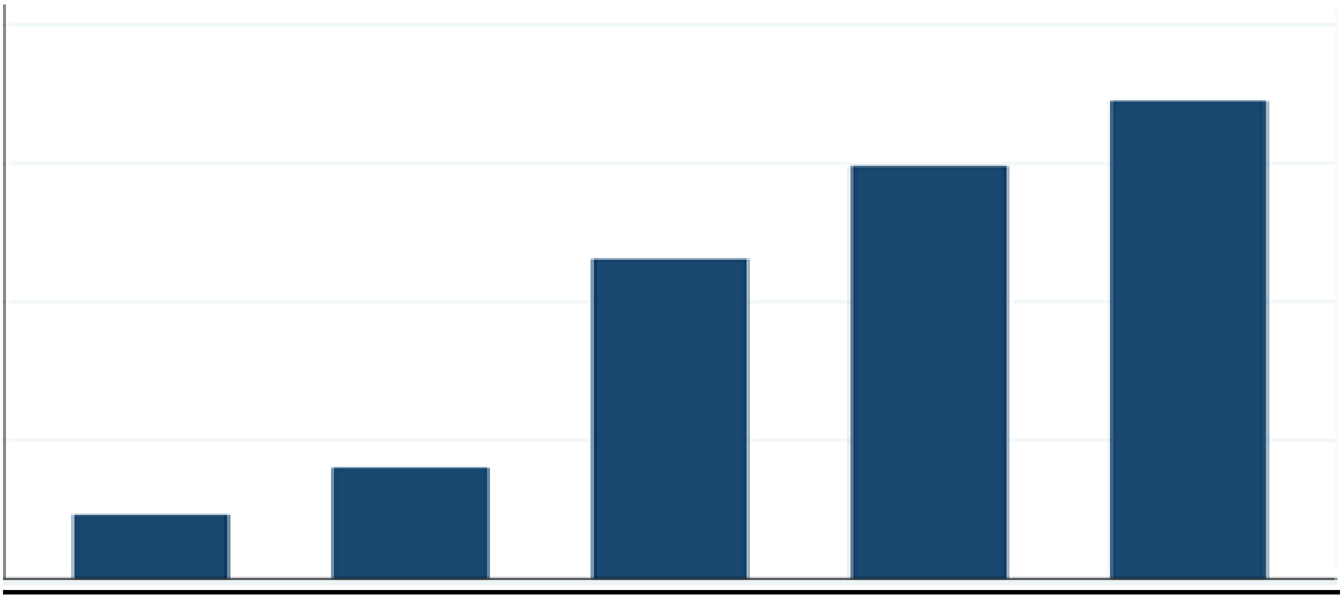
```python
import wx
import os
import  wx.html
##########################################################################
class MainPanel(wx.Panel):
    """"""

    #---------------------------------------------------------------------------
    def __init__(self, parent):
        """Constructor"""
        wx.Panel.__init__(self, parent)
        self.SetBackgroundStyle(wx.BG_STYLE_ERASE)
        self.frame = parent
        sizer = wx.BoxSizer(wx.VERTICAL)
        hSizer = wx.BoxSizer(wx.HORIZONTAL)
        gname=['Hill Climb Racing','Mario','Google Dino']
        script=['gesturecontrol','mario','dinosaur']
        for num in range(3):
            label = "Click To Play %s" % gname[num]
            name= "%s" % script[num]
            btn = wx.Button(self, label=label,name=name,size=(180,25))
            sizer.Add(btn, 5, wx.ALL, 5)
            btn.SetBackgroundColour((220, 20, 60, 255))
        self.Bind(wx.EVT_BUTTON, self.onButton)

        #buttonhtml = wx.Button(self, label="How To Play",size=(180,25))
        buttonD = wx.Button(self, id=wx.ID_ANY, label="Quit",size=(180,25))
        #buttonhtml.SetBackgroundColour((220, 20, 60, 255))
        buttonD.SetBackgroundColour((220, 20, 60, 255))
        #sizer.Add(buttonhtml,0,wx.ALL,5)
        #self.Bind(wx.EVT_BUTTON,self.onclickHTML, buttonhtml)
```



Results and Discussion:

As we can see from the graph that the more frames used the more trained the system becomes in analyzing the image for the better processing along with various algorithmic analysis done in support to the research for better understanding of computational power, space and time complexities.

X-axis: NO. Of Frames; Y-axis: Image Accuracy



Conclusion:

Hand Gestures, a nonverbal collaborative communication-inspired system user interface, included various features which would enhance the gamer's interaction experience is a huge upcoming trend because of its future scope. It is going to add huge profits to the gaming sector and generate great revenue as well as it both customer and producer are going to be benefitted. Also, this implementation will reduce the inefficiencies caused by the peripheral gaming mode. This interface was prototyped and developed with the goal of offering a real-time user experience to the world of gaming by following several concepts of HCI: Interactive design, catering to universal usability, KLM, GOMS, interface testing, and usability testing, heuristic evaluation, HTA, Groupware, communication, and collaboration, Use case modeling, State Transition models, storyboarding, offering informative feedback from the users, selecting appropriate methods and algorithms in order to achieve greater efficiency, aesthetic and minimalistic design. The novelty of our project is the hand gesture interaction with various games, which is integrated into our interface as an aim to satisfy the user with a real-time user experience, because the main reason any user must be well aware of the interfacing as more sophisticated designs causes the ineffective interaction which doesn’t satisfy the user which was the main goal in the project as all these features give the gamer's a positive and holistic experience.

REFERENCES:

[1] Prajakta Dhamanskar, Aniket C Poojari, Harshita S Sarwade, Renita R D’silva. Human Computer Interaction using Hand Gestures and Voice. 2019 International Conference on Advances in Computing, Communication and Control (ICAC3). IEEE Mumbai, India, 2019.

[2] Siddharth S. Rautaray and Anupam Agrawal. Interaction with Virtual Game through Hand Gesture Recognition. 2011 International Conference on Multimedia, Signal Processing and Communication Technologies. Aligarh, India, 2011.

[3] Ricardo Nakamura, Romero Tori and Joao Luiz Bernardes Jr. Design and Implementation of a Flexible Hand Gesture Command Interface for Games Based on Computer Vision. 2009 VIII Brazilian Symposium on Games and Digital Entertainment. Rio de Janeiro, Brazil, 2009.

[4] Yanmin Zhu, Zhibo Yang, Bo Yuan. Vision Based Hand Gesture Recognition. 2013 International Conference on Service Sciences (ICSS). Shenzhen, China, 2013.

[5] K. K. Biswas, Saurav Kumar Basu. Gesture recognition using Microsoft Kinect®. The 5th International Conference on Automation, Robotics and Applications. Wellington, New Zealand, 2011.