A CASE STUDY ON GAME THAT USES TWO HAND PADDLES TO HIT THE BALL BACK AND FORTH(PONG)

M.Sruthi[1], Dr.R.Yadagiri Rao[2], T.Naresh kumar[3], V.Shashikanth[4], SK.Shaheena[5], R.Santhosh[6]

[1]Assistant professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[2]Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[3,4,5,6] IV[th]Btech Student, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

Abstract

This paper describes a command interface for games based on hand gestures defined by postures, movement and location. The large variety of gestures thus possible increases usability by allowing a better match between gesture and action. The system uses computer vision requiring no sensors or markers on the user or background. The analysis of requirements for games, the architecture and implementation are discussed, as well as the results of several tests to evaluate how well each requirement is met. Play the famous Pong Game using hand gestures. You don't have to touch anything with your hands! We use vision recognition and machine learning to detect the activity on your webcam. Digital video games can be enjoyed more naturally and conveniently using hand gestures and machine learning in-corporate. The game can be controlled using specific hand gestures. This project aims to create a unique application that connects to the game. Input and interaction devices are an important part of every game for human interaction but in recent times it has been developed with various technologies to control computer- based systems. Interaction between humans comes from different modes like gestures, speech, text etc. All gesture interaction is best for interacting with games.

1. Introduction

The possibility of relaying commands to a computer system using one's own hands and gestures has interested researches and users for a long time and was one of the first topics in user interface research, partly because it uses well-developed, everyday skills [Bowman 2005]. With the computational capacity available today and widespread use of image capture devices, even in domestic systems it is possible to implement this sort of interaction using computer vision. This brings the benefit of leaving the user's hands free of any gloves, cables or sensors. Gestures2Go, the system described here, provides this functionality and its implementation (in C++, illustrated in figure 1) is focused on electronic games.

Games are an ideal platform to test and popularize new user interface systems, for several reasons, such as an increased user willingness to explore in this medium [Starner et al. 2004]. There are many examples of academic research developing and studying new interfaces with games, particularly incorporating Augmented Reality [Bernardes et al., 2008].

The game industry has also introduced new (or of previously restricted use) interfaces and devices to the public.

From the joystick to increasingly complex gamepads and controllers shaped as musical instruments, from datagloves to "pistols" that function as pointing devices and even haptic devices [Novint 2009], many are such examples, to the point that, today, some professionals are encouraged to play games to improve job-related skills [Dobnik 2004]. On the other hand, both the industry and academia acknowledge that new, more natural (and hopefully fun) interfaces are one way to attract new consumers to this economically important but still restricted market [Kane 2005]. And in the past few years, the search for these interfaces has been more widespread, continuous, well-publicized and commercially successful.

After a popular gaming platform introduced motion and tilt detection in a simpler controller as its most innovating feature [AiLive 2007], motion detection was quickly added to other platforms and games and continues to be researched and improved upon. Several portable gaming systems, in particular, are taking advantage of motion and tilt sensing, touchscreens and even microphones in their interface. More recently still a project was unveiled to add interaction based on recognition of full-body motion, speech and faces to a popular platform [Snider 2009].

Despite this ebullience in game interfaces, the use of hand gestures, especially leaving the user's hands free, has seen little academic or commercial research in this area and is usually limited to analyzing only hand movement or a small number of hand postures. One of Gestures2Go's objectives is greater flexibility, to allow the use of a greater variety of gestures (currently defined by hand postures, movement or location and using both hands).

Another important goal is that it must be easy to use for both players and developers. Gestures2Go should also be usable with existing games (designed for traditional interfaces) and allow multimodal interaction. These and other requirements arose, during system design, from an analysis focusing specifically on gestures and on game applications. Many of the same requirements exist in other applications as well, such as education or

virtual and augmented reality, and the authors believe this system may be well suited for these applications, but will leave this discussion outside the scope of this paper.

2. Literature survey

2.1 Existing methods

Computer vision-based hand gesture recognition: This method involves using computer vision techniques to detect and recognize hand gestures in real-time. A camera can be used to capture the hand gestures of the player, and then the gestures can be classified and mapped to the movements of the Pong paddle.

Wearable sensor-based hand gesture recognition: This method involves using wearable sensors, such as accelerometers and gyroscopes, to capture the movements of the player's hand. The sensor data can then be processed and mapped to the movements of the Pong paddle.Depth camera-based hand gesture recognition: This method involves using a depth camera, such as the Microsoft Kinect, to capture the 3D movements of the player's hand. The depth data can then be processed and mapped to the movements of the Pong paddle.

2.1.1 Disadvantages

Limitations of gesture recognition: The accuracy of the hand gesture recognition system can be affected by factors such as lighting conditions, hand positions, and occlusion, which can lead to errors in gameplay.Learning curve: Players may need some time to get used to the hand gesture controls, which can be a disadvantage for beginners.

Limited range of motion: The use of hand gestures for gameplay can be limited by the range of motion of the hands, which can make some game maneuvers difficult to execute.Equipment requirement: This project requires specialized equipment such as a camera or sensor to capture hand gestures, which can be a barrier to entry for some players.

2.2 Proposed methods

Machine learning-based hand gesture recognition: This method involves training a machine learning model, such as a neural network, to recognize hand gestures from image or sensor data. The model can be trained on a dataset of hand gestures and their corresponding movements in the Pong game. Once trained, the model can be used to recognize the hand gestures of the player and map them to the movements of the Pong paddle.

Regardless of the method used, the Pong game using hand gestures project requires careful consideration of the design, implementation, and testing phases to ensure an enjoyable and accurate gaming experience for the user. This Pong Game Project system has a single user and multi-user option (maximum two users). Each level has multiple blocks to create obstructions and multiple balls to take the difficulty to the next level. One can change the number of blocks and the number of balls to be present in each level. The game has intriguing background music. This game is different from the usual pong games that are available online and is much more interesting to play. Moreover, it is

freeware and can be played by anyone and everyone alike.

2.2.1 Advantages

Interactive gameplay: The use of hand gestures to control the paddle in the Pong game offers an interactive gameplay experience that can be engaging and fun for players.Unique user interface: This project offers a unique user interface that can attract more players than the traditional keyboard or joystick control.

Physical activity: Playing Pong with hand gestures requires some physical activity, which is beneficial for health and can encourage people to exercise more.Technology demonstration: This project can demonstrate the potential of hand gesture recognition technology, which can have a wide range of applications beyond gaming.

3. Modules

Pong module:– This module complies and runs the game as a single element. This is the main running module.

Brick module:– This module controls the placement positions of the bricks and the number of bricks.

Ball module:– This module controls the movement of the balls as well as the number of balls.

4. Implementation

4.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

1. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
2. Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
3. Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
4. Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.1.1 Features

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
Easy-to-read − Python code is more clearly defined and visible to the eyes.
Easy-to-maintain − Python's source code is fairly easy-to-maintain.
A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
Extendable − You can add low-level modules to the Python interpreter. These

modules enable programmers to add to or customize their tools to be more efficient.
Databases − Python provides interfaces to all major commercial databases.
GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
Scalable − Python provides a better structure and support for large programs than shell scripting.

4.2 Django

Django is a Web framework written in Python. A Web framework is a software that supports the development of dynamic Web sites, applications, and services. It provides a set of tools and functionalities that solves many common problems associated with Web development, such as security features, database access, sessions, template processing, URL routing, internationalization, localization, and much more.

Using a Web framework, such as Django, enables us to develop secure and reliable Web applications very quickly in a standardized way.The development of Django is supported by the Django Software Foundation, and it's sponsored by companies like JetBrains and Instagram.

4.2.1 Installation

Now that we have the venv activated, run the following command to install Django: pip install Django

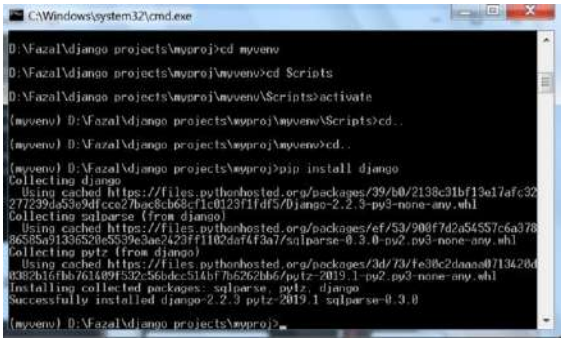

Fig:Django installation through commands

To start a new Django project, run the command below:

django-admin startproject myproject
The command-line utility django-admin is automatically installed with Django.After we run the command above, it will generate the base folder structure for a Django project.Our initial project structure is composed of five files:

manage.py: a shortcut to use the django-admin command-line utility. It's used to run management commands related to our project.We will use it to run the development server, run tests, create migrations and much more. init .py: this empty file tells Python that this folder is a Python package.

Settings.py: this file contains all the project's configuration. urls.py: this file is responsible for mapping the routes and paths in our project. For example, if you want to show something in the URL /about/, you have to map it here first.wsgi.py: this file is a simple gateway interface used for deployment.

You don't have to bother about it. Just let it be for now.Django comes with a simple web server installed. It's very convenient during the development, so we

don’t have to install anything else to run the project locally.

We can test it by executing the command:

`python manage.py runserver`For now, you can ignore the migration errors; we will get to that later.

Now open the following URL in a Web browser: http://127.0.0.1:8000 and you should see the following page:

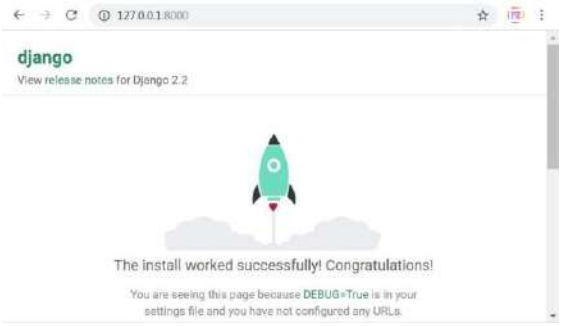

Fig: Django installation

5. Analysis

5.1 Introduction

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

5.2 Requriments Specification

5.2.1 User Requirements

User requirements are statements in natural languag**e** along with corresponding diagrams (tables, forms, intuitive diagrams) detailing the services provided by the system and operational constraints it must comply with. Additionally, it’s worth noting that user requirements primarily focus on the user’s needs. Thus, these user requirements cater to the customer.

5.2.1.1 Functional Requirements

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet. Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified. The functional specification describes what the system must do, how the system does it is described in the design specification. If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements

Graphical User interface with the User.

5.2.1.2 Non-Functional Requirements

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e., how fast the system reacts to user commands.) or accuracy (i.e., how precise are the systems numerical answers.)

Non-functional requirements are

1. Reliability

2.Usability

3.Time Constraints

4.Security

5.Privacy

6.Portability

7.Scalability

8.Performance

5.2.2 Software Requirements

For developing the application the following are the Software Requirements:

1.Windows 10 OS

2.Python

3.Django

4.MySQL

5.2.3 Hardware Requirements

For developing the application the following are the Hardware Requirements:

1.Processor: Intel i3

2.RAM: 1 GB

3.Space on Hard Disk: minimum 120 GB

6. Testing

6.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 Testing Methodologies

6.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by

successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of the data flows; data fields, predefined processes, and successive processes must be considered for testing.

Integration testing will focuses mainly on the interfaces & flow of data/information between the modules**.** Here priority is to be given for the integrating links rather than the unit functions which are already tested. Integration Testing focuses on checking data communication amongst these modules

6.2.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.4 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

6.2.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

7. Conclusion

7.1 Conclusion

The proposed system uses a shape-based approach including several steps for hand gesture recognition, palm recognition, finger counts, etc. It is an attempt to generalize all attempts to use gesture recognition for playing games or manipulating or controlling computer systems. However, the system, at present, tends to achieve this in the domain of playing games. Compared to previously used procedures, the success rate has increased by about 30%. To make it more robust, many of the barriers in previous approaches are removed.

The system aims to allow users to create customized gestures for any

event/action in any domain. The machine vision based keyboard cursor control using hand gesture system is developed in the Python language, using the Open CV library. The system is able to control the movement of a keyboard cursor by tracking the users hand for playing a game. The keyboard cursor functions will perform by using different hand gestures.

The system has the potential of being a viable replacement for the computer keyboard, however due to the constraints encountered; it cannot completely replace the computer keyboard. The accuracy of the hand gesture recognition can be improved, if the template matching hand gesture recognition method is used with a machine learning classifier. Open CV mostly stretches towards real-time vision applications and takes advantage of MMX (Multimedia Extension) and SSE (Streaming SIMD Extensions) instructions when available.

After the implementation and testing of the game, NFS has been developed based on the methodology proposed in this paper. The control using hand gestures has evaluated the game and has increased the game quality. However, in this paper's development, there are some limitations that have been identified. The game itself lacks several visual elements including sounds and interfaces. However, this game was deliberately chosen as it is simple and easy to implement as well as has very few controls.

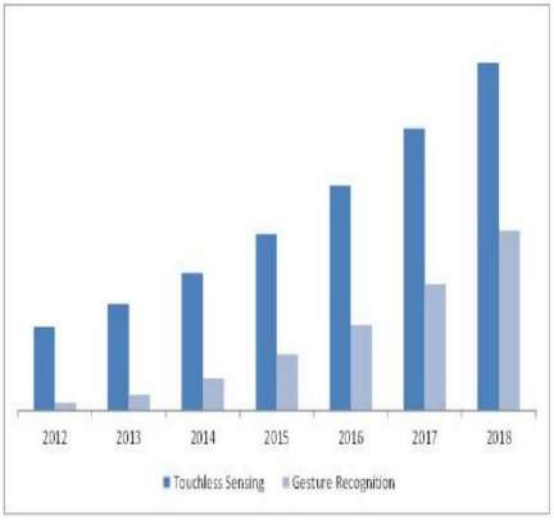

Fig: Statistics of gesture recognition

7.2 Future Enhancement

This is an effective hand gesture recognition system to address the problem of extracting frames from a video and processing it. In the future scope, various hand gestures can be recognized and applied as input to the computer. The hand gestures representing numbers can also be converted into commands to perform related tasks in real time. Enhancing the recognition capability for various lightning conditions, which is encountered as a challenge in this project can be worked upon in future.

This application can be considered as a starter to the HCI application grounds, therefore large upgradations can be done to it. This application can be extended out to the mouse cursor control also using some more HCI concepts & Open CV Algorithms. Much more accuracy can be obtained with the help of Neural Networks based logics. Tracking performance can be improved to ensure better results. The accuracy of the hand gesture recognition can be improved if the template matching

hand gesture recognition method is used with a machine learning classifier. This will take a lot longer to implement, but the accuracy of the gesture recognition will improve.

In this modern world, where technologies is at the peak, there are many facilities available for offering input to any applications running on the computer systems, some of the inputs can be offered using physical touch and some of them without using physical touch (like speech, hand gestures, head gestures etc.). Using hand gestures many users can handle applications from distance without even touching it. But there are many applications which cannot be controlled using hand gestures as an input. This technique can be very helpful for physically challenged people because they can define the gesture according to their need.

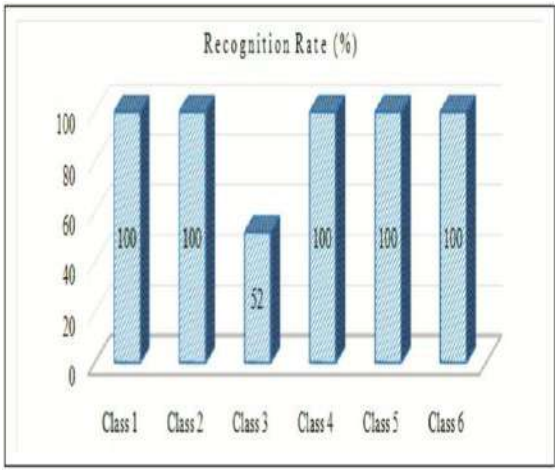

Fig: Recognition rate

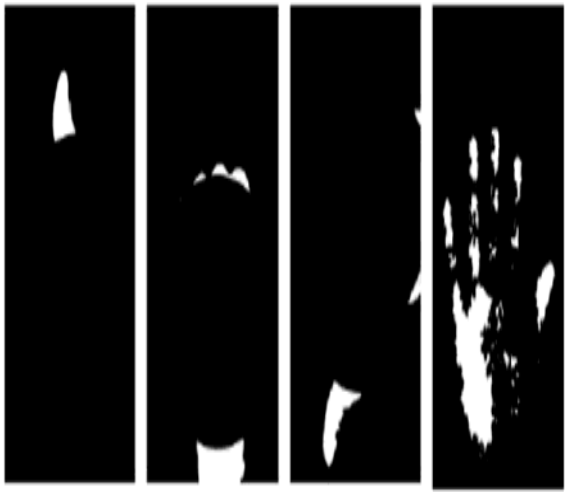

Fig: Hand positions recoginition

The present system which we have implemented although seems to be user friendly as compared to modern device or command based system but it is less robust in detection and recognition as we have seen in the previous step. We need to improve our system and try to build more robust algorithm for both recognition and detection even in

the cluttered background and a normal lighting condition. We also need to extend the system for some more class of gestures as we have implemented it for only 6 classes. However we can use this system to control applications like power point presentation, games, media player, windows picture man

8. Reference

- Khan, R.Z. and Ibraheem, N.A., 2012. Hand gesture recognition: a literature review. International journal of artificial Intelligence & Applications, 3(4), p.161.
- AILIVE, 2007. LiveMove White Paper. Available from: http:

//www.ikuni.com/papers/LiveMoveWhite Paper_en.pdf [Accessed 24 July 2009].

- BOWMAN, 2005. 3D User Interfaces: Theory and Practice. Addison-Wesley.
- BERNARDES, J. ET AL, 2008. Augmented Reality Games In: Extending Experiences: Structure, analysis and design of computer game player experience. Lapland University Press, p. 228-246.
- IMAI, A. ET AL., 2004. 3-D Hand Posture Recognition by Training Contour Variation. In: Proc. Automatic Face and Gesture Recognition 2004, p. 895-900.
- Fang, Y., Wang, K., Cheng, J. and Lu, H., 2007, July. A real-time hand gesture recognition method. In 2007 IEEE International Conference on Multimedia and Expo (pp. 995-998). IEEE.
- Rautaray, S.S. and Agrawal, A., 2011, December. Interaction with virtual game through hand gesture recognition. In 2011 International Conference on Multimedia, Signal Processing and Communication Technologies (pp. 244-247). IEEE.
- Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K. and Yang, J., 2011. A framework for hand gesture recognition based on accelerometer and EMG sensors. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(6), pp.1064-1076.
- Garan, T.A. and Suaib, N.M., Hand Gesture Integration of 3D Virtual Rubik's Cube Using Leap Motion.
- Lewis, J.R., 2006. Usability testing. Handbook of human factors and ergonomics, 12, p.e30.
- Hambling, B. and Van Goethem, P., 2013, May. User acceptance testing: a step-by-step guide. Chippenham: BCS.
- An overview of opencv. Full Scale. (2022, April 12). Retrieved April 15, 2022, from https://fullscale.io/blog/opencv-overview/
- Pygame tutorial - javatpoint. www.javatpoint.com. (n.d.). Retrieved April 15, 2022, from https://www.javatpoint.com/pygame