ROCK PAPER SCISOR LIZARD SPOCKS (AKASH GARG, 23336943)

UI/UX Design : https://app.uizard.io/p/ba810551

Demo Video :

2023.12.20_2514.mp4
WhatsApp Video 2023-12-15 at 8.58.40 PM.mp4

Ideal Distance between Hand and camera : 2ft

Ideal distance between hands of player : 1ft

20/dec

- need to improve on logic for spocks and lizard hand gesture detection , as its currently working only working on specific size of hands. Need to make it more dynamics.
- Will start working on OpenPose variation for comparison .
- Gesture recognition does not work properly when gesture shown from use of back of the hand to camera

---

15/jan

- Added dynamic function to calculate and compare distance between two Points of hands . Used Palm length as base. Point 5 and 17.
- While using MideaPipe and running code on Flask Server, when using website on phone, mis match Frame rate is causing exception (https://github.com/google/mediapipe/issues/1553)
- OpenPose application is very bulky, unable to integrate it with website or app. Too much complicated installation work, dependency on other software.
- When Creating CNN using SqueezeNet, public data set was not available , after creating data set manually , accuracy achieve was approx 70% , which is not better than MediaPipe
- Fixed back hand gesture recognition issue.

Testing:

- Mediapipe work farely well
- Able to detect gesture for different hand size and from various distance.
- Noticed some mis-hand tracking , if the background contains some edge patterns.

Research paper read : (https://www.researchrabbitapp.com/collection/public/OZYD37D3ZY)

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9622242

https://ieeexplore-ieee-org.elib.tcd.ie/stamp/stamp.jsp?tp=&arnumber=10215427

https://ieeexplore-ieee-org.elib.tcd.ie/stamp/stamp.jsp?tp=&arnumber=9984411

https://ieeexplore.ieee.org/document/8765346

https://arxiv.org/pdf/1704.07809.pdf

https://arxiv.org/abs/2006.10214

App Development Tutorials:

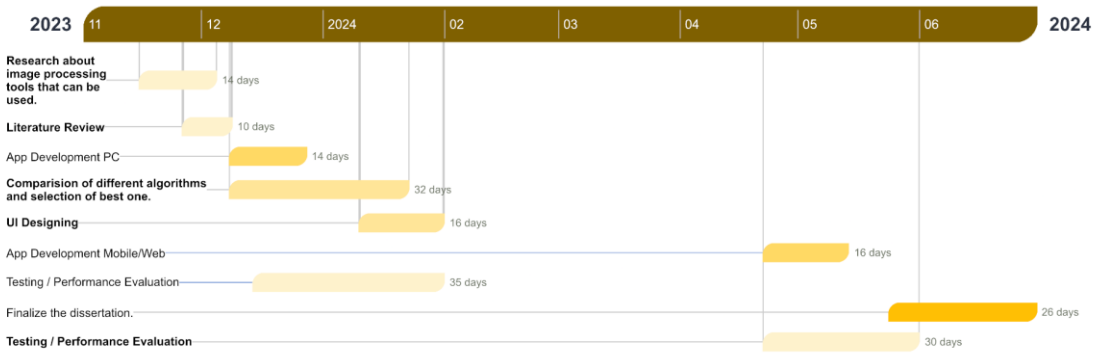
https://youtu.be/dFBqjoexXgs?si=l5NxBNRD4wDRf_fz (flutter)

https://www.youtube.com/watch?v=6gNpSuE01qE (Kivy)

https://youtu.be/6Tj8_iKqh_k?si=PIaXgRUc9ra1QA4t (Flet) no camera support for mobile device

Dissertation TimeLine:

https://online.officetimeline.com/app/#/file/31087bcf-5eb2-410e-8317-2c52c38bfe99/gantt-view



Install openpose on mac

https://www.thomasvanhoey.com/post/installing-openpose-on-mac-october-2020-version/

https://maelfabien.github.io/tutorials/open-pose/#run-openpose

OPENPOSE VS MEDIAPIPE

Algorithm Name	Downsides vs. OpenPose	Upsides vs. OpenPose	Best Fitted Use Cases	License Type and Cost
MediaPipe	Lower accuracy than OpenPose	Lightweight and real-time performance	Augmented reality, gaming, sports analysis	Open-source, free for commercial use

Motivation Statement

Motivated by a deep passion for innovation and a commitment to revolutionize traditional gaming experiences, this project endeavors to craft an advanced Rock-Paper-Scissors-Spock-Lizard (RPSSL) mobile application, integrating sophisticated computer vision technology. The aim is to elevate the timeless RPS game by introducing real-time hand gesture recognition, enabling a dynamic and immersive multiplayer experience with the unique inclusion of Spock and Lizard gestures. Going beyond technical sophistication, the project seeks to seamlessly bridge the physical and digital realms, allowing users to intuitively interact with the game through hand gestures. A distinctive feature of this application is its groundbreaking support for real-time multiplayer on a single device, fostering a communal and socially engaging gameplay environment. Additionally, the application boasts a challenging VS AI mode, adding a strategic layer to the gaming experience. Emphasizing data privacy and security, the project is dedicated to ensuring ethical handling of personal information. This undertaking is not just about delivering an advanced gaming application; it's about cultivating joy, connectivity, and innovation in the ever-evolving landscape of digital entertainment.

Week 1-4: Project Kick-off and Background Research (Research and Literature Review)

Objective: Establish a solid foundation for the project and delve into relevant background information.

Tasks:

Define the scope and objectives of the Rock-Paper-Scissors-Lizard-Spock (RPSSL) mobile application.

Conduct a thorough literature review on computer vision, hand gesture recognition, and existing RPS applications.

Identify key challenges and opportunities in the field.

Week 4-8: Selecting Technology Stack and Frameworks

Objective: Choose the appropriate tools and frameworks for the application development.

Tasks:

Evaluate and select the computer vision libraries and frameworks suitable for hand gesture recognition.

Decide on the mobile development platform (iOS, Android) and the corresponding development frameworks.

Create a technology roadmap outlining the chosen stack.

Week 9-12: Data Collection and Annotation

Objective: Gather a diverse dataset for training and testing the gesture recognition model.

Tasks:

Collect a comprehensive dataset of hand gestures, including all RPSSL symbols.

Annotate the dataset with correct labels for each gesture.

Ensure diversity in terms of hand poses, lighting conditions, and individuals.

Week 13-16: Model Development and Training

Objective: Develop and train the hand gesture recognition model.

Tasks:

Design the architecture for the gesture recognition model, considering CNNs or transfer learning.

Train the model on the annotated dataset.

Implement real-time processing to ensure swift recognition during gameplay.

Week 17-20: User Interface Design and Prototyping

Objective: Create an intuitive and visually appealing user interface for the mobile application.

Tasks:

Design the user interface, incorporating countdown timers, score displays, and feedback mechanisms.

Develop a prototype to visualize the flow and interactions within the application.

Week 21-24: Multiplayer and AI Integration

Objective: Implement multiplayer functionality and integrate AI for single-player mode.

Tasks:

Develop the networking component for real-time multiplayer support.

Implement an AI module for VS AI mode, offering different difficulty levels.

Test and refine the multiplayer and AI functionalities.

Week 25-28: Testing and Iteration

Objective: Conduct thorough testing and gather user feedback for iterative improvements.

Tasks:

Perform functional, usability, and performance testing on the application.

Iterate on the application based on the testing results.

Week 28-40: Documentation and Finalization

Objective: Document the development process and finalize the mobile application.

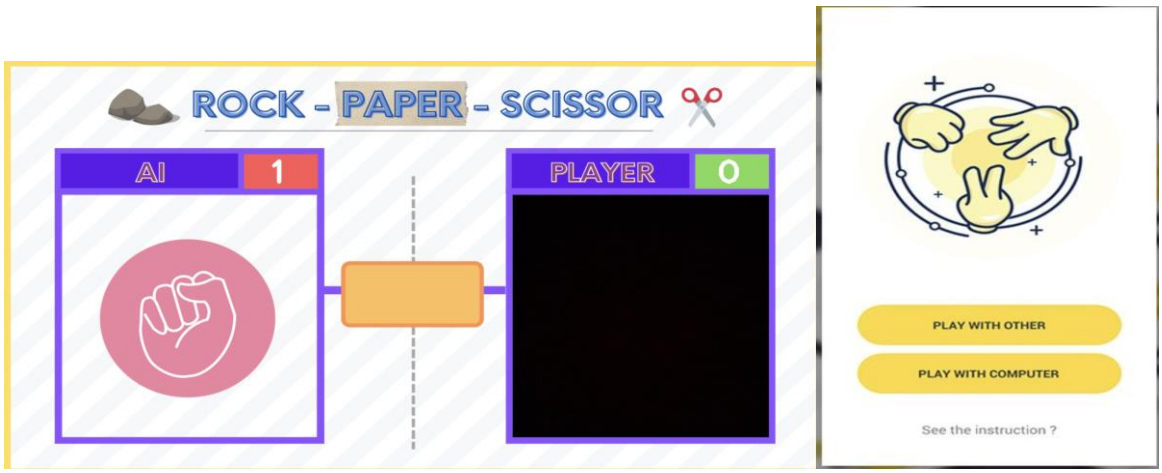
Tasks:

Prepare detailed documentation, including design decisions, codebase overview, and user guides.

Make final adjustments based on feedback and testing.

Submit the completed application and thesis.

Sample UI design :

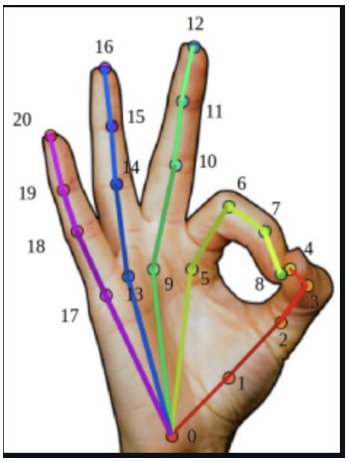


OpenPose :

Hand right =4

Hand left= 7

21 hand keypoint estimation



https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_advanced_standalone_face_or_hand_keypoint_detector.html

https://viso.ai/deep-learning/openpose/

https://github.com/CMU-Perceptual-Computing-Lab/openpose

https://www.youtube.com/watch?v=IlsXQPOF9IE&list=PL_Nji0JOuXg24bHB60SB2TwF0PpwhJkCF&index=3

Physical Aspect of App:

The camera will be placed on the side of the players, front camera facing them on phone stands. Both users' hands should be visible Display would be split into two parts running hand classifiers on it. The back of hand should be directed toward the camera, and the palm away from the camera (except in case of Lizard, in which the palm should be faced downward to recognize Lizard)

Type of testing planning to perform:

1. Functional Testing/ Gesture Recognition Testing: Test the accuracy and responsiveness of the gesture recognition component by having users play the game with various hand gestures. Test using various backgrounds.
2. Usability Testing: Ensure that the application's UI is intuitive, visually appealing, and user-friendly. Gather feedback from users on their overall experience while playing the game.
3. Performance Testing: Assess the application's real-time performance, ensuring that it can process gestures and update game status without lag or delays.
4. Resource Usage: Monitor the application's resource consumption, such as CPU and memory usage.
5. Error Handling and Edge Case Testing: Identify and test error-handling scenarios, such as when the camera feed is interrupted, when players make unusual gestures, or when the application encounters unexpected conditions.
6. User Feedback and Beta Testing: Involve real users in beta testing to gather feedback on their experiences and to identify any unanticipated issues.
7. Evaluate it using the confusion matrix

Test Data Collection:

Gather a dataset of hand gesture images for each game symbol (rock, paper, scissors, lizard, Spock). Ensure diversity in hand poses, skin tones, lighting conditions, and backgrounds to simulate real-world scenarios.

Manual Testing:

a. Testing Environment:

Create a controlled testing environment with consistent lighting to minimize external variables. Use a mobile device with the application installed.

b. Test Scenarios:

Test each hand gesture individually in isolation.

Test for variations in hand orientation, size, and position.

Test gestures made by different individuals with varying skin tones.

Error Analysis:

Analyse common error patterns, such as when the model misclassifies certain gestures.

Determine if certain gestures or conditions lead to higher recognition errors.

Accuracy :

To Calculate the accuracy of model different metrics can be used such as

a. Confusion Matrix:

b. Accuracy, Precision, and Recall:

c. F1 Score:

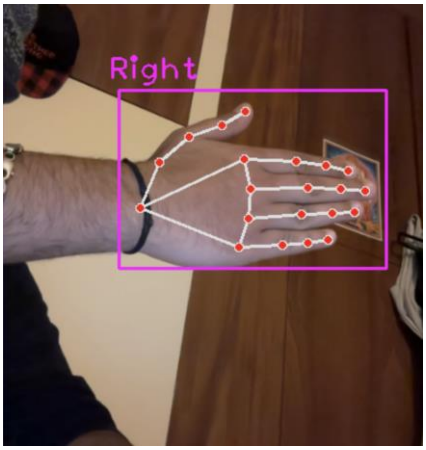
d. Benchmarking Against Baselines

Image detection:

- SqueezeNet:
    - Uses keras_squeezenet , tensorflow, cv2
    - Train and modify the pre trained Squeeze net model, Update its Output layer according to our needs
- HandTrackingModule by cvzone :
    - Uses CV2 and Mediapipe
    - Have feature like to calculate the UpFingers and findDistance to differentiate between gestures.

Suggestion:

- Include advantages of playing games, to help kids develop skills
- Can show AR option Shuffler on Hands
- Can show the hand outline used to classify gesture.



Survey Questions:

1. How would you rate your overall experience with the application?

2. Were you able to easily understand and use the application's features?

3. Did you encounter any difficulties while using the app? If so, please describe them.

4. What did you like most about the application?

5. What aspects of the application do you think need improvement?

7. Were there instances where the app failed to recognize your hand gestures correctly?

10. Did the recognition accuracy vary based on factors like lighting, hand orientation, or skin tone?

11. Did the application respond in real-time to your hand gestures without noticeable delays?

14. What are your thoughts on the user interface's design and layout?

15. Were the countdown timers, scores, and game information clearly presented?

16. Were the UI elements intuitive and easy to use?

Demographic Information (optional):

17. To better understand our users, can you provide information about your age, gender, and location?

Demo Video :

- https://youtu.be/0uSA3xyXlwM?si=D7G2hWhRKFcG_Miz
- https://www.youtube.com/watch?v=k2EahPgl0ho

Library

- https://github.com/cvzone/cvzone/blob/master/cvzone/HandTrackingModule.py

1> Project Goals:

To Develop a Multiplayer RPSLS Application: The primary goal is to create a mobile application that allows multiple players to participate in a game of Rock-Paper-Scissors-Lizard-Spock. using hand gestures. enabling fair and accurate gameplay (with proper count down and Action Time difference). Design a scoring system. Allow support for multiple players.

2> Scope of the Application:

1.Game Variants: The application will support both traditional Rock-Paper-Scissors and the extended Rock-Paper-Scissors-Lizard-Spock variant. With PVP and PvComp option.

2. Computer Vision: The core of the application will involve computer vision and machine learning to recognize hand gestures accurately.

3. Documentation and Reporting: Throughout the project, maintain comprehensive documentation to facilitate the development process and the creation of a final dissertation report.

4. User Testing and Feedback: Gather feedback from users during the development process to enhance the application's usability and address any issues that arise.

3> Literature Review:Research existing computer vision based RPS applications and gesture recognition models. Gather insights and best practices from related projects.

4> Set Up Development Environment: Choose the development tools, libraries, and frameworks you'll use (e.g., Python, OpenCV, TensorFlow, or PyTorch).

5> Data Collection, Train Gesture Recognition Models , Gesture Detection :

Collect a dataset of hand gestures for each game symbol (rock, paper, scissors, lizard, Spock). Develop or fine-tune a machine learning model to recognize hand gestures.

6 > User Interface (UI) Design, Timing Mechanism:

Design a user-friendly mobile application interface that displays a countdown timer and maintains scores. Design a mechanism to ensure fair play by checking the timing of gestures. Implement a scoring system that accurately tracks and updates scores for each player based on game outcomes.

7 > Testing:

Rigorously test the application, especially the gesture recognition component. Involve potential users to playtest the application, gather feedback, and make improvements based on their input. Test on various background with different user of different demographics.

8> Documentation and Reporting:

Maintain detailed documentation of the project, including design decisions, code, and any challenges you faced.

ISSUEs:

- Ethical Considerations: Address privacy and security concerns by obtaining necessary consents for image capture and ensuring that user data is handled securely and ethically.

- Validate/test Efficiency of different models.