VR Ping Pong Team Final Presentation

Logan Herman, Henry Yi, William Wang

Use Case

Problem: Long distance, virtual interactions are not as personal as physical interactions. This makes digital ***social interactions less interesting and engaging*** to everyone involved.

Solution: Creating a virtual reality ping pong game to play against other people around the world in real-time. The VR aspect creates a pseudo-presence that will make the interactions more fun.

Use Case Requirements: User Experience

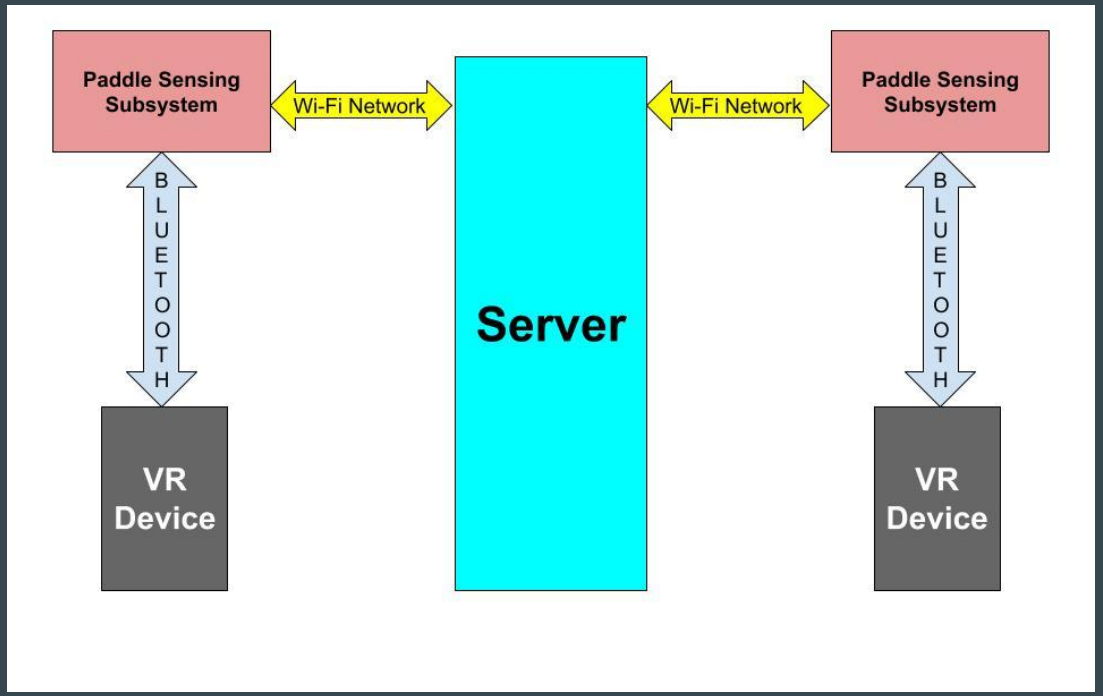
- Latencies
  - Ball movement latency - < 50ms to calculate ball trajectory
  - Paddle movement latency - < 100 ms
- Smooth Frame Rate
  - 30 FPS is acceptable, allows for 15 frames to show ball flight path of a professional-speed rally
- Moderate resolution (~360p)
- Paddle power lifespan
  - Allow for 1 hour of continuous, wireless gameplay
- Accurate Paddle Motion Tracking
  - +/- 3 inches in terms of position
  - +/- 7.5 degrees in terms of orientation

Solution: Virtual Reality Ping Pong Game

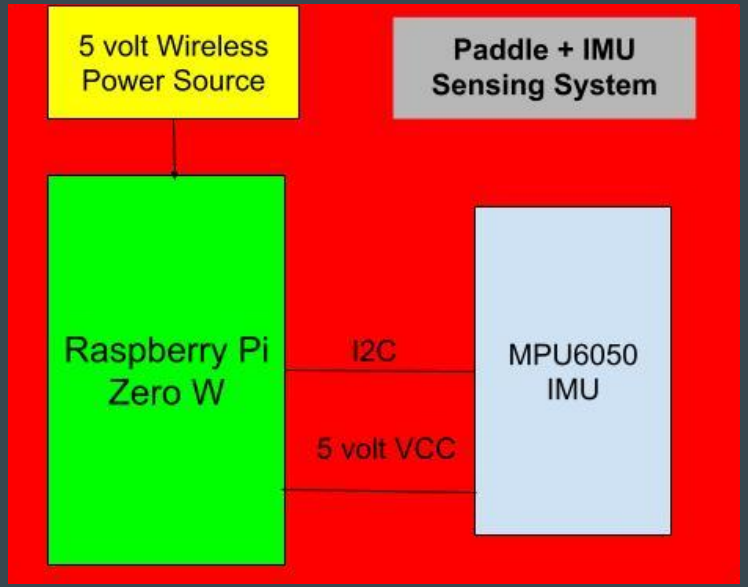
- Create a multiplayer VR ping pong game where two players will be able to play against each other. The game will be developed using the Unity game engine and uses OpenCV and data from an IMU to determine the paddle state. The game will be played on a Google Cardboard VR headset.

- The IMU sends data to the VR headset, which is running a client version of the game. The VR headset computes the paddle state which it then sends to a dedicated server and transitively, the opponent. The state of the ball after a player hits it is also transmitted from player->server->opponent.

System Specifications



Components and Communication Protocol

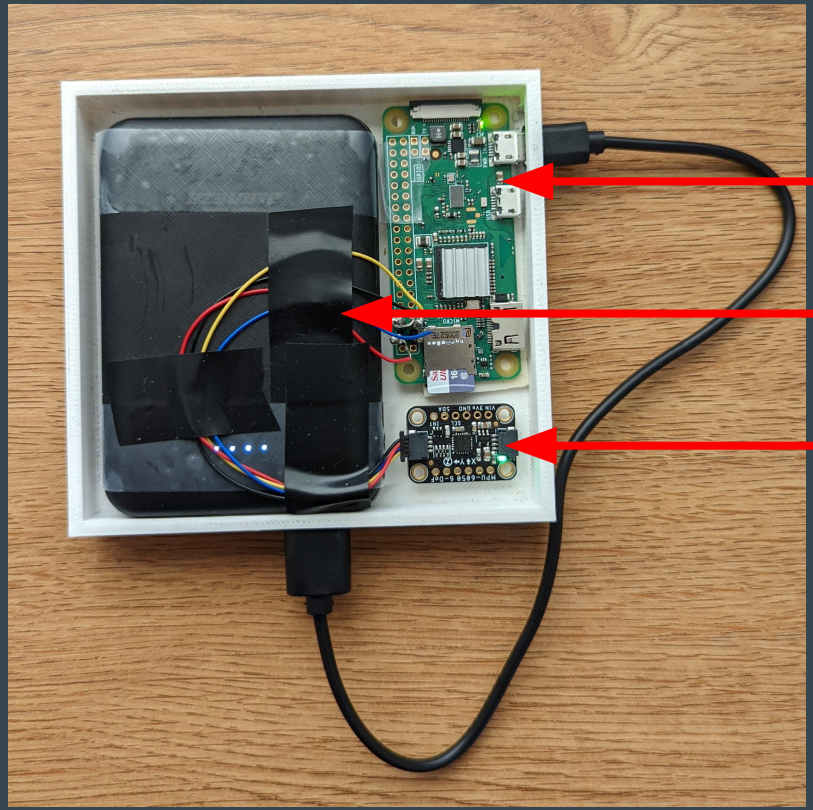


Paddle System Design

Solution: What has changed?

- Instead of having the paddle state be its orientation and position, it will now be a combination of swing type, swing power, and swing direction.
  - Why? Changing the position and orientation of the paddle based on the data the IMU and camera sends it will make the paddle movements really laggy and not smooth. By having this paddle state, we can accurately estimate the motion of a swing and still have really smooth gameplay.
  - How? Using the acceleration and orientation data from the IMU, we can set thresholds that let us classify the swing type, power, and direction. Swing types such as slices, smashes, and lobs can be determined by how the paddle is oriented and how it accelerates. And by integrating the acceleration of the IMU over a period of time, we can determine the swing power.

Complete Solution (Paddle):



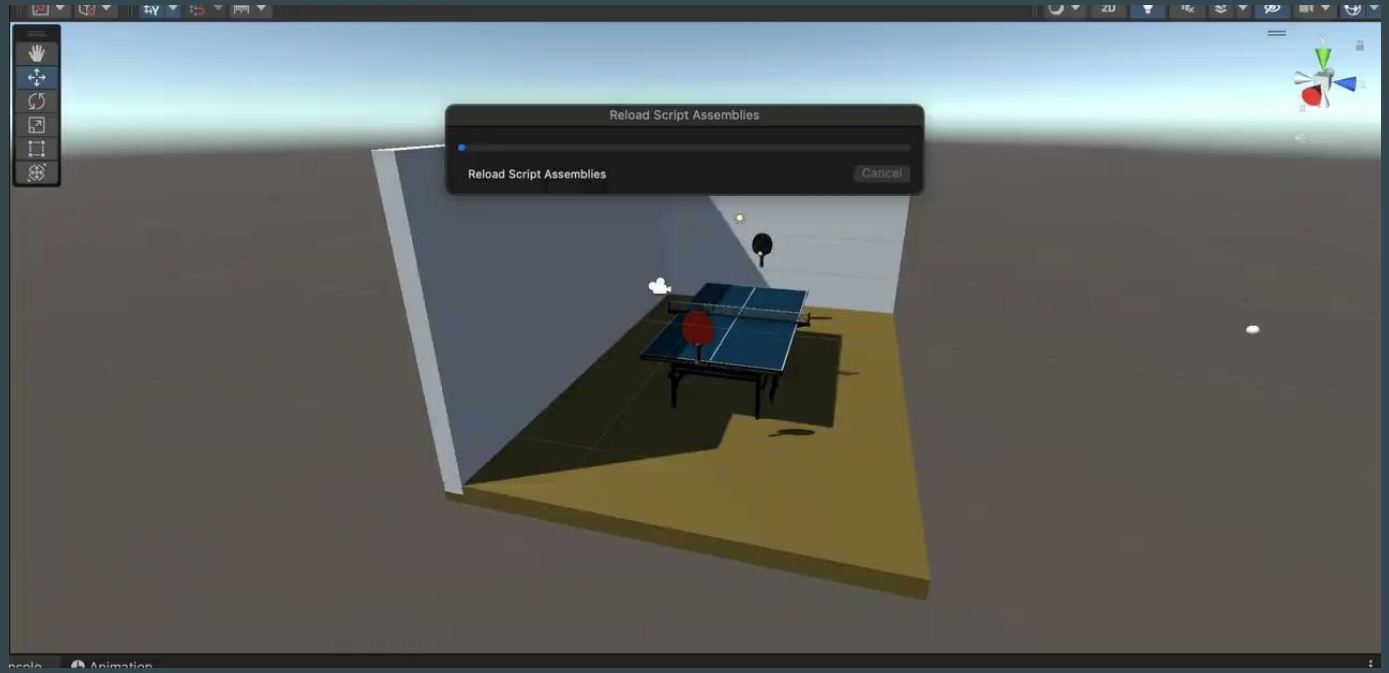
Raspberry Pi Zero

Battery Pack

IMU

*This system will be attached to a ping pong paddle

Complete Solution (Game):



The video should show a few samples of the different swing types and swing powers. We are still working on the ball-paddle and ball-table interactions to make them more realistic

Test, Verification, and Validation

What were we originally planning on testing?

- Latency and Positional and rotational accuracy of the paddle

What are we testing now?

- Latency from paddle to VR headset
- Latency of opponent's movement from player's point of view - tested to prevent scenarios where opponent may seem to swing late, but it is just delayed.
- Paddle state accuracy - how accurately can we correctly identify the swing type, swing power, and swing direction

Test, Verification, and Validation

How are we testing each metric?

- Latency from paddle to VR headset is tested by comparing timestamps from when our data packet is sent to when it is received by the headset. We have only recently integrated the paddle with the VR headset so testing is still to be done.
- Latency of an opponent's movement is tested by moving a paddle from one end of the table to another and measuring how long it takes the paddle to reach the final point from the opponent's point of view. This can be simulated using ParallelSync which gives me the POV and paddle movement control of both players.
- Testing the accuracy of the paddle swing type, power, and direction can be measured by swinging the paddle that has an expected swing type, power, and direction and seeing if our game accurately classifies each characteristic. Testing this is a critical part of helping us tune our thresholds that determine how to classify a player's swing.

Test, Verification, and Validation

Testing Metric:	Tested Values:	Passing Test Value:
Latency (paddle-to-headset)	TBD	0.1 seconds
Latency (opponent-to-player)	0.1 seconds (simulated)	0.3 seconds
Accuracy of Swing Type	70%	90%
Accuracy of Swing Power	TBD	90%
Accuracy of Swing Direction	85%	95%

Remaining Items to be Completed

- Test the latency of data being transferred from the paddle to the VR headset.
- Tune thresholds to get swing accuracies up to our accepted standards
- Adjust the speed and motion swing animations to better match expected swing path of player
- Implement score keeping