The background is a dark blue-grey color, decorated with various geometric shapes and patterns in teal and white. There are circles of different sizes, some with dotted interiors, hexagons, and triangles. Some shapes are solid teal, while others are white outlines or dotted patterns. The overall aesthetic is modern and technical.

Picture This! (C6)

Joseph Ayala, Anthony Meza, Sophia Zhang

Use Cases

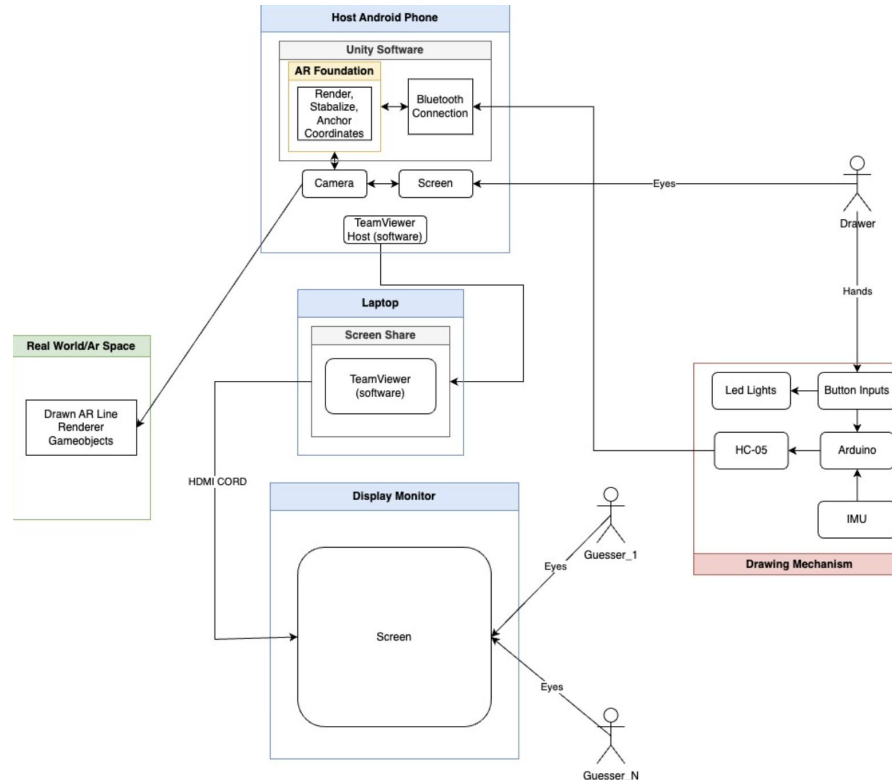
Problem – Most games do not promote social connection nor active movement of the body

Solution – Game of pictiography with drawings visible in virtual space

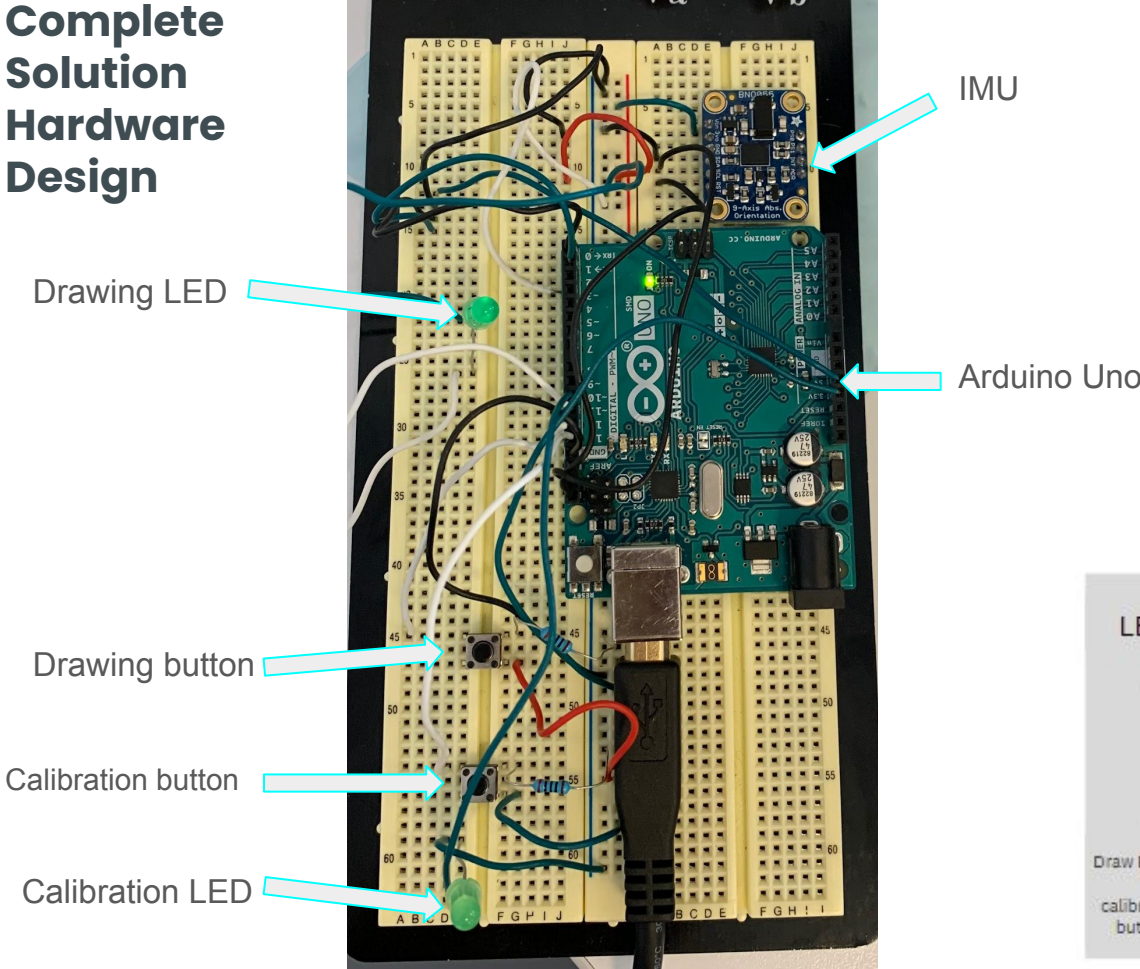
- Encourage Social Interaction
 - 3 Players participate minimum
 - “Real time” (<1 second delay) gameplay where devices will interact
- Hardware is responsive and portable
 - Total latency of < 150 ms
 - Less than 1lb
- Drawn lines are relatively straight
 - Line pixel deviation < ½ inch both vertically and horizontally

Solution Approach

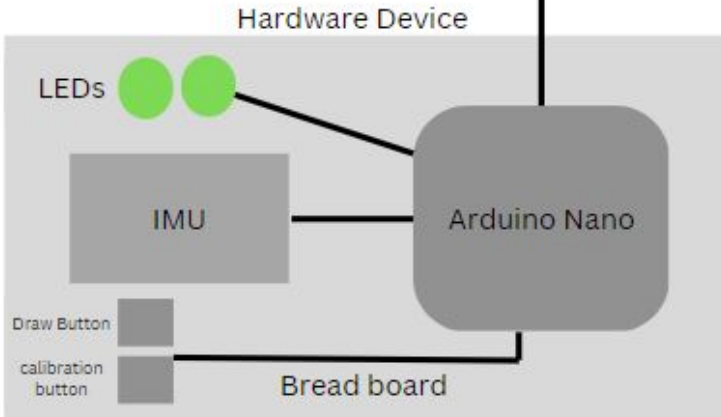
Will be using Unity for game development and for packages it offers
Hardware Pen Component will be used to communicate for image generation



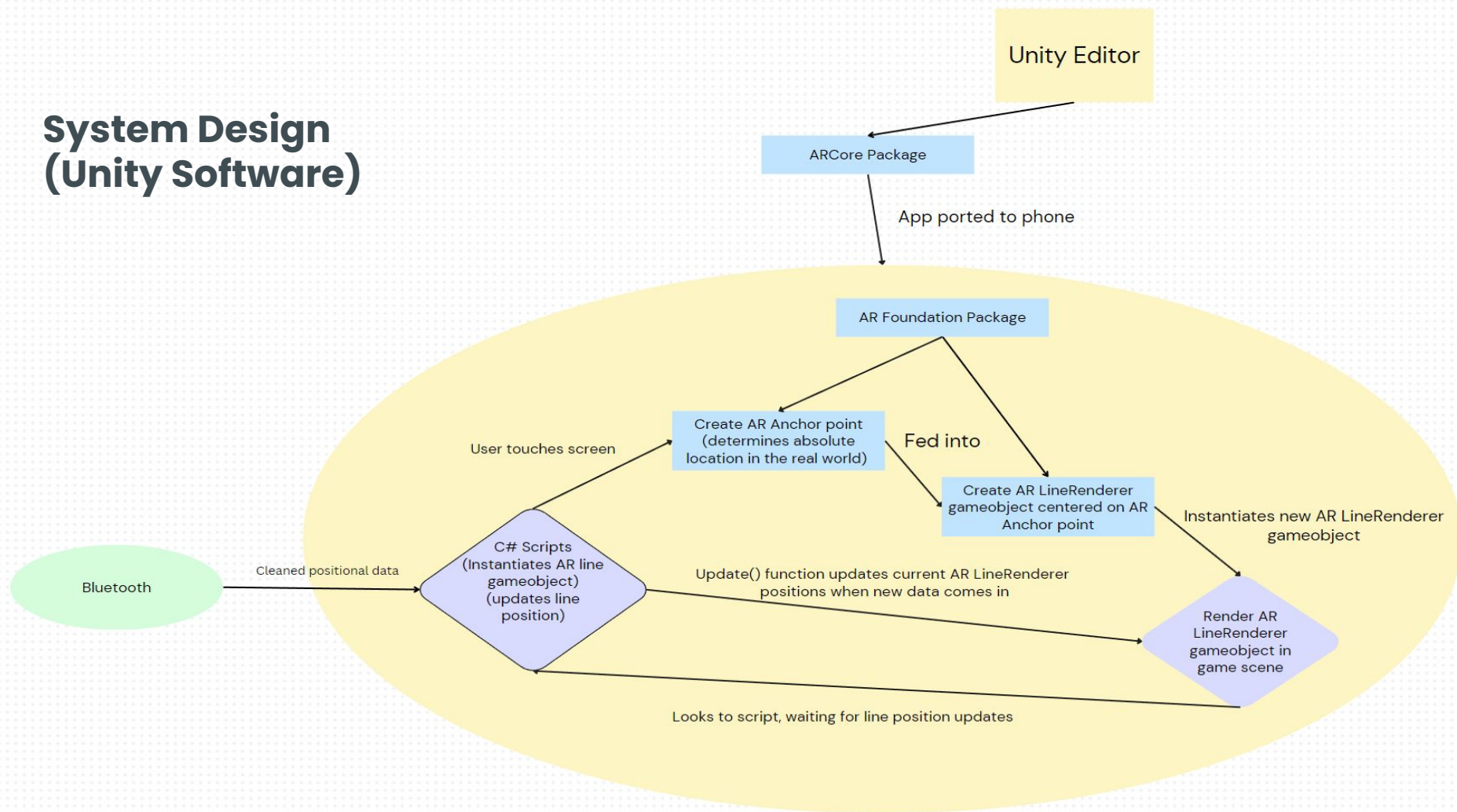
Complete Solution Hardware Design



IMU data
Bluetooth Channel

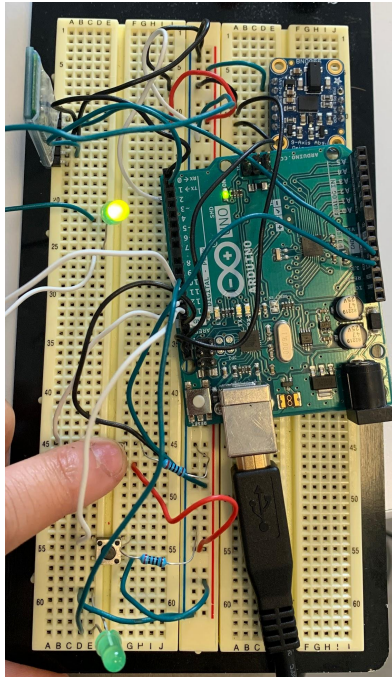


System Design (Unity Software)

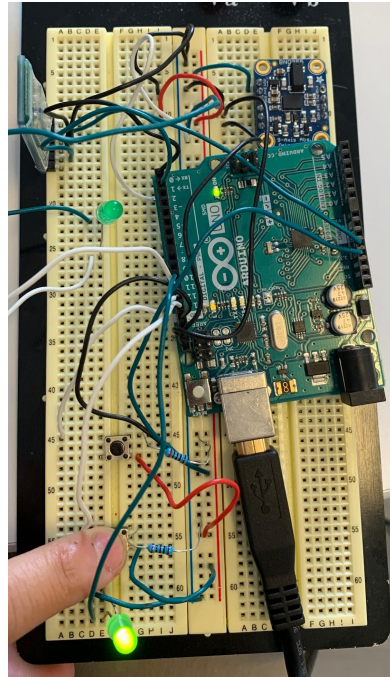


Solution

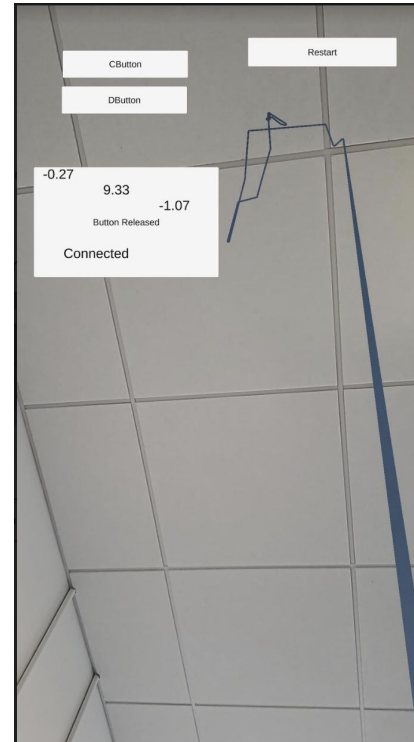
Drawing Mode



Calibration

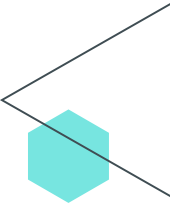
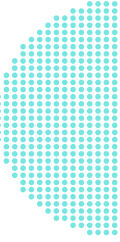


Drawing



Testing, Verification, and Metrics

- Pen
 - Collection of data during specified intervals
 - Measurement on latency of sending data
- Visual verification of drawn lines
 - User testing - 4 out of 5 users are satisfied with the generated drawing
 - Achieve 90% accuracy between AR line and real line in overlapping line test
 - E.g. a virtual red line drawn over a real black line, 90% of line pixels should be red
- Game Functionality
 - 3 players are able to view drawing on devices



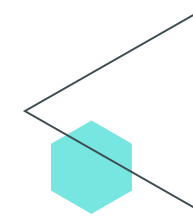
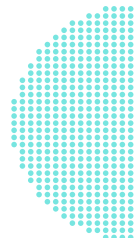
Testing, Verification, and Metrics

Metric	Target	Actual	Met
Pen Size	6"x9"x3", < 1lb	2.2"x7"x1", ~6 oz	✓
Pen latency	< 150ms	< 8ms	✓
End-to-end latency	< 1s	~ 10ms	✓
Line accuracy	90%	0%	✗
User testing	4 out of 5 users	0 out of 5 users	✗
Game functionality	>= 3 players	1 drawer player/personal device, 1 monitor	N/A

Design tradeoffs

Hardware-software communication

- Python netsocket
 - Too much latency in communication
 - Requires external server to help receive/send data
- Uduino Wifi Package
 - Original method described in design review
 - Wifi communication between Unity and Arduino
 - Not compatible with Android devices
- Unity bluetooth plugin
 - Not compatible with original NodeMCU + ESP8266
 - Switch to Arduino Uno
 - 5V power supply to stabilize bluetooth connection



Design tradeoffs

IMU calibration methods

- Sensor fusion with multiple IMUs
- Using a better IMU
- Trying a different algorithm
 - Kalman filter + calibration phase + quaternion rotation
 - Other ways of tracking position (ie. gps, UWB, ...)
 - Change the way we draw
 - Similar to a mouse or joystick
 - Computer vision to track pen

Multiple Apps

- Networking between 3 different apps + ARCloud Anchoring
 - Functional ARCloud Anchoring on ONE device
 - Networking, device separated game state code, and full integration with line algorithm software, hardware pen, and multiple devices unfeasible given remaining timeframe
- Screen sharing
 - TeamViewer Host run on android device running game, TeamViewer run on separate laptop to connect to and see screen of host android device
 - Laptop hooked up to large monitor through HDMI, visible by all users

Adjustments and To Do

Hardware:

- Calibrate IMU further to get more meaningful data
- Solder Arduino and components to shield
 - More compact and neat

Software:

- Refine data received from hardware
- Test and decide best/most user friendly way of creating line anchor points
- Refine game UI

