BLOKUS



Team A1: Blokus Nadine Bao, Jonathan Nee, Aria Zhang

Use Case

- Goal of Blokus
 - Get rid of all your pieces
 - Prevent others from doing so
 - Play pieces in turn-based order
 - Pieces can only be placed corner to corner
- Socially-distanced Blokus board game
 - Playing on separate physical boards with other players in their own locations
- Area
 - Software + Hardware (Circuits)



Game Requirements

- Support up to 4 players per game
- Game play mechanics:
 - Place pieces down on your own board
 - See opponents' board light up based on your move via LEDs
- Only allow valid moves
- Ability to resume game at a later time

Technical Requirements (LEDs)

- 3 possible approaches
 - LEDs with corresponding colors to show where pieces are placed
 - AR/MR
 - Display in software and users have to place the piece themselves
- Each blokus board will have custom LEDs
- LEDs light up orange if moves are invalid
- When a piece is placed on 1 board, the corresponding LEDs light up on opponents' boards
- Sufficient power for 400 LEDs (~4-15 Watts/m of strip LED)
- RPi GPIO latency negligible

Technical Requirements (CV)

- 4 possible approaches
 - Camera with CV to track board state
 - Sensors (Hall effect/Light sensors)
 - RFID tagging of pieces
 - PCB
- Can detect changes in board state when new pieces are placed
- Can accurately detect when there is a hand in the image
- Can process given images within 100 ms
- CV color detection algorithm for a board is highly parallelizable

Technical Requirements (Software)

- 2 possible approaches
 - <u>Server-Client</u>
 - Peer-to-Peer
- Lower latency (40 ms for client \rightarrow server \rightarrow client)
- Easier to implement (solid architecture)
- Scalable
- Persist board state to DB

System Overview



Solution Approach

- Web Server:
 - Deploy to AWS EC2
 - MongoDB Atlas for persisting game state
- Computer Vision:
 - **OpenCV** to process live video from **Logitech/Raspberry Pi camera**
- LEDs:
 - **RPi** + long strip of 400 **LEDs** + external power supply

Testing, Verification, and Metrics

Requirement	Testing Strategy	Metrics
Functional Blokus game	Software testbench	Valid/invalid moves, player turns
Working LEDs	MCU testing	Ability to control specific LEDs to change color
CV detection	Software + Visual	Ability to correctly identify tiles with pieces
Total latency	Software timer	CV + Software + Web latency + MCU < 150ms

Risks/Challenges

- Biggest challenge: low latency
 - Potential bottlenecks
 - CV processing
 - Client-server communication
- Accuracy of detected pieces
 - \circ Lighting of the room could affect CV
 - Need to avoid detecting hands/external objects as pieces
 - Resampling if this is the case
- Timeliness of LED circuit construction
 - Necessary for integration/testing

Tasks

- LED & Blokus Board Design (Nadine)
 - Pi program
 - Integrate LEDs into Blokus board
- Computer Vision (Jonathan)
 - Recognize pieces placed on board
 - Avoid other objects (e.g. hands placing the piece)
- Game Logic & Web Server (Aria)
 - Game mechanisms
 - Multiplayer
 - Game lobbies

Gantt Chart

Nadine Aria Jonathan Everyone

asks	9/21- 9/23	9/24- 9/26	9/27- 9/29	9/30- 10/2	10/3- 10/6	10/7- 10/10	10/11- 10/13	10/14- 10/16	10/17- 10/19	10/20- 10/23	10/23- 10/26	10/27- 10/30	10/31- 11/2	11/3- 11/5	11/6- 11/12 Demo	11/13- 11/16	11/17- 11/20	11/21- 11/26	11/27- 11/30 Final
Research: Computer Vision																			
Research: AWS																			
Research: LED Circuit/Pi	1																		
Computer Vision:																			
Code for Image Processing																			
Get Camera																			
Testing on Stock Image																			
Camera Setup with Computer																			
Camera itegration with Code																			
Testing with Camera Images																			
Avoid detection of external objects																			
Optimization for Speed																			
ED Gameboard:																			
Raspberry Pi Purchase																			
Blokus Purchase																			
LED Research and Purchase																			
Raspberry Pi Coding																			
Circuit Design																			
LED Strip Testing w/ RPi																			
Circuit/Gameboard Construction																			
RPi and LED testing									ſ										
Same Software											1								
Processing Board State																			
Turn Based Detection																			
Keeping Track of Available Pieces																			
Handling Multiplayer																			
Detection of Invalid Moves																			
Optimization for Speed																			
Veb Communication:											1								
Webserver Lobby Code																_			
Webserver Deployment																			
Database Setup																			
Board State Persistence																			
Latency Optimization																			
ntegration																			
Software Integration with CV																			
Software Integration with RPi																			
Software Integration with Web Comm																			
Metric Testing																			
Speed testing of CV detection	-										-								
Speed testing of software																			
Speed testing of web server																			
Speed testing of move communication																			
User testing of game																			
Report and Presentations																			
Design Presentation																			
Midpoint Demo																			
Final Report																			
Final Presentation + Demo																			
Ireak																			