The Resistance:

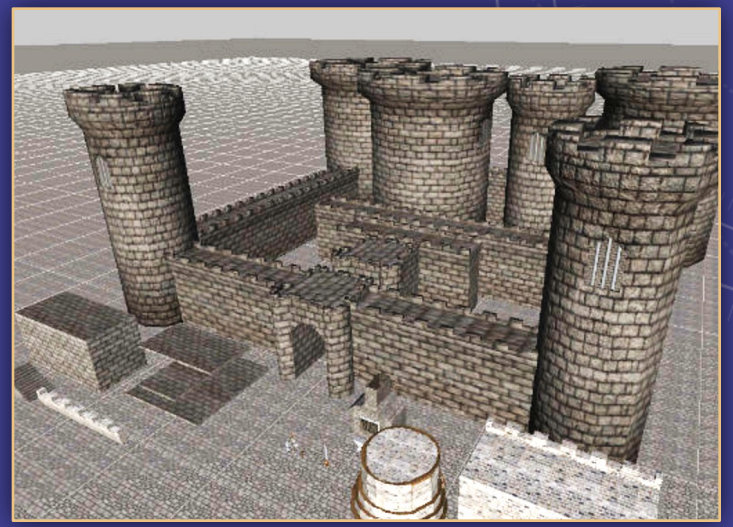
AVALON

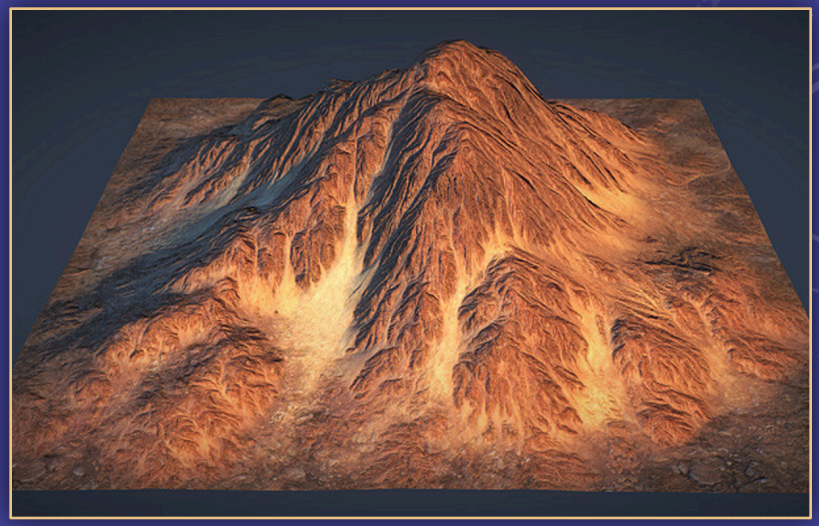
Final Project Demo Presented by SUPER POLLO

- Daniel Ni (danielni@uchicago.edu)
- Amy Sitwala (asitwala@uchicago.edu)

RECAP (MULTI-PLAYER & AUGMENTED REALITY)

- With adventure-type board games (Lost Cities, Avalon), the board is relatively static
- Most mobile app games are single player; multiplayer mobile app games are often not played in real time/in the same room





GAMEPLAY

Overview for 4-player version:

- 1 "Bad" and 3 "Good" Players
- 4 quests
- "Bad" player needs 2 quest "Fails" to win
- "Good" players need 3 quest "Successes" to win
- **Players can only see their colors and cannot see anyone else's colors or decisions**

GAMEPLAY (STAGE 1)

1. Players are randomly assigned “Good” or “Bad” on game start:

Good (Green)

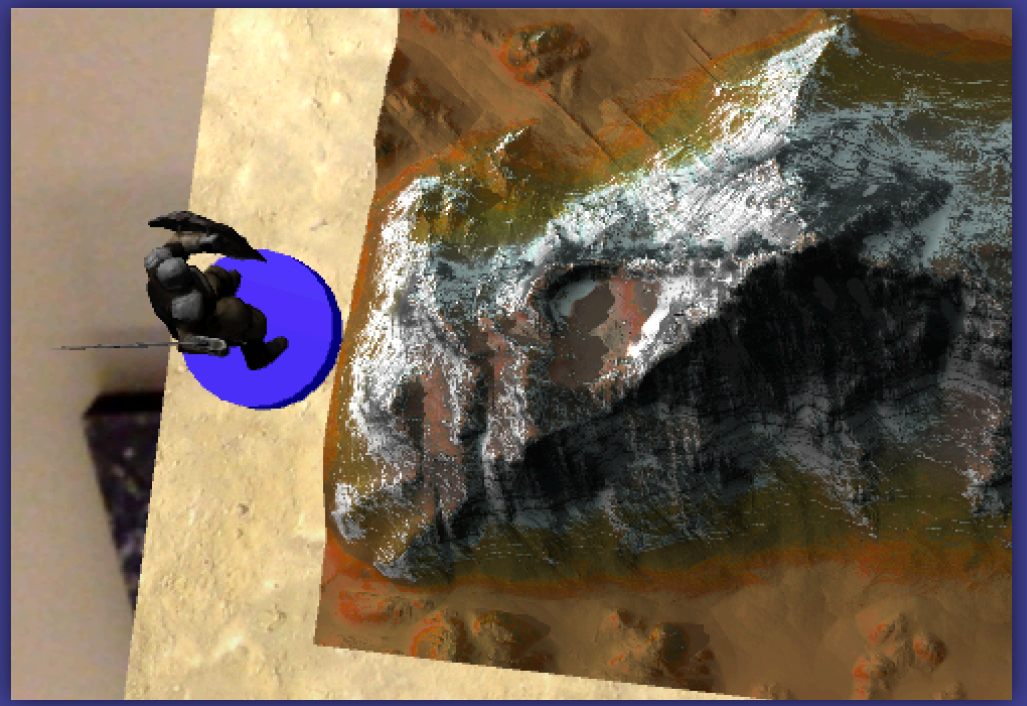
Bad (Red)

GAMEPLAY (STAGE 2)

1. First player to connect is the team leader (is denoted by “leader stand”):

2. “Leader stand” moves to next player on each round (either when there is more than 1 “Reject” when approving players to go on quest or when the quest decision has been made)

Blue token represents who is the leader

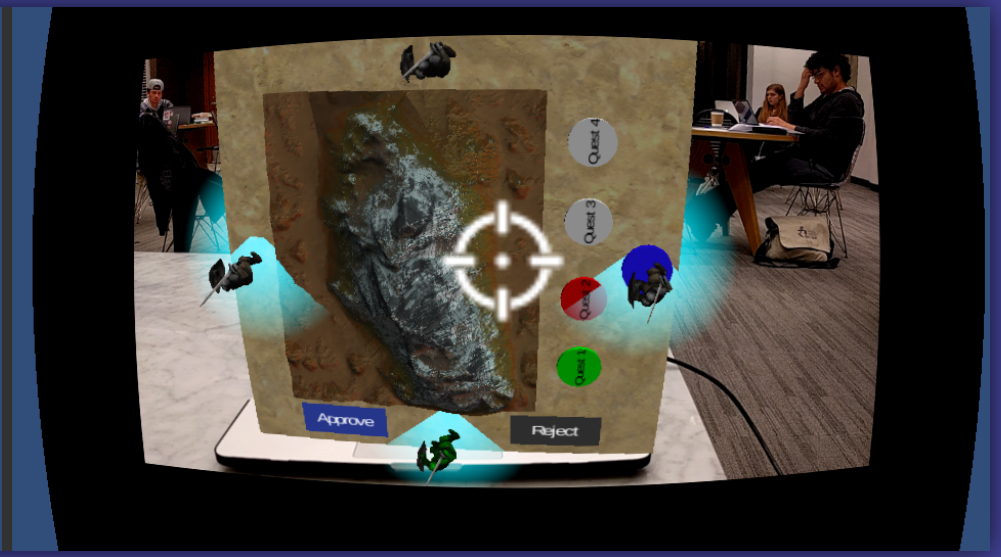


GAMEPLAY (STAGE 3)

1. Leader player chooses which players go on the quest (2 players for first quest, 3 for all other quests). Leader can also pick himself/herself to go on quest.

2. All players must "Approve" or "Reject" the quest player choice (if more than 1 "Reject", leader stand moves over and new round begins)

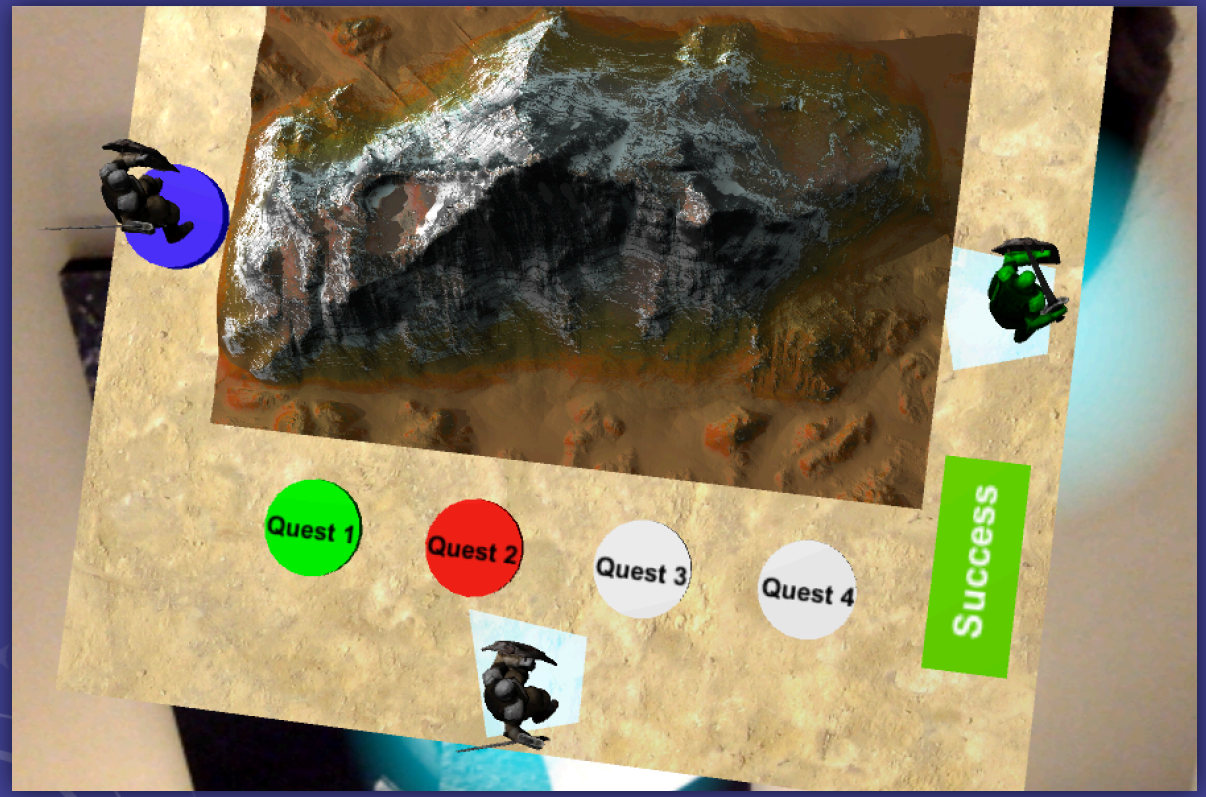
Glowing halo represents who is chosen to go on quest



Players can "Approve" or "Reject"

GAMEPLAY (STAGE 4)

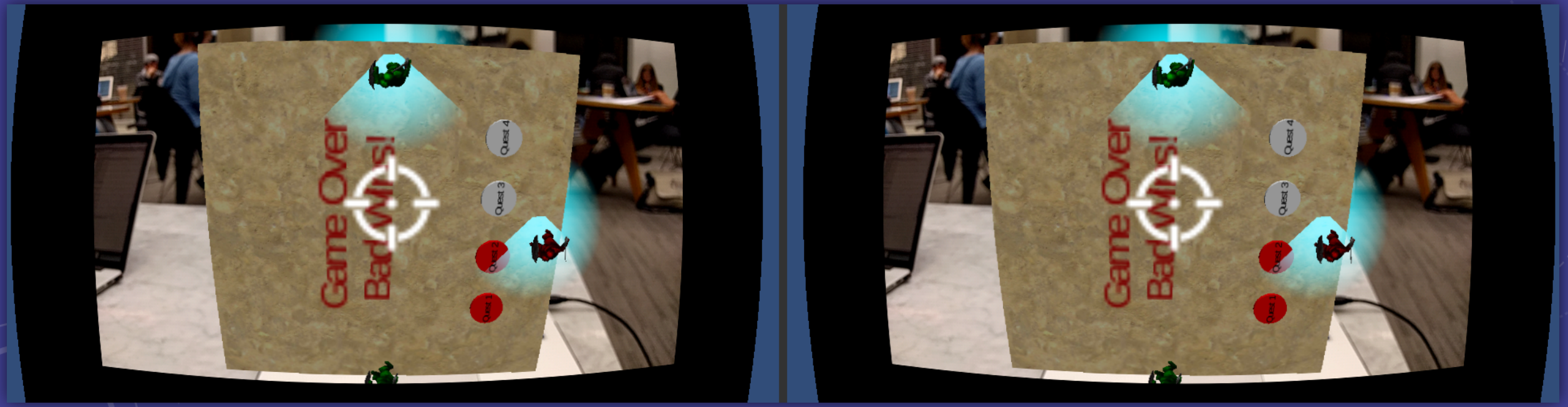
1. If majority approves the quest, then those selected to go on quest choose whether the quest "Succeeds" or "Fails"



2. Good players can only "Succeed" the quest
3. Bad player can either "Succeed" or "Fail" the quest
4. If there is 1 "Fail" in the number of decisions, the entire quest "Fails" -> turns Red
5. If all "Successes", then quest "Succeeds" -> turns Green

GAMEPLAY (STAGE 5)

1. If there are 2 failed quests at any time, the game ends and the "Bad" player wins
2. If there are 3 successful quests at any time, the game ends and the "Good" players win
3. All players' colors are revealed at end of gameplay



TECHNICAL APPROACH

- Game Engine to produce Google Cardboard VR app
- Organize scenes, environment, and players
- Add scripts to game objects to allow for dynamic gameplay

- Unity 3D extension that supports AR
- Allows images to act as surfaces upon which models crafted in Unity can be placed
- These "image targets" are registered in a database
- Board is essentially an image target

- Photon Unity Networking builds on Unity's native networking API
- Players join a common room on a "master server"
- Create wrapper "networked" functions (using "[PunRPC]" – similar to Unity's "[syncvar]" and "[command]") around functions dealing with game state

SUCCESSES (WHAT WORKED WELL)

1. Image target set-up and recognition (not too difficult to connect to Unity Project through Vuforia Developer Portal)

    a) Had to add image target to database
    b) Download database into Unity project along with Vuforia package
    c) Download a specific app Key through Vuforia and attach to "ARCamera" prefab

2. Real-time changes through Photon Networking

    1. More intuitive set-up and higher level than native Unity Networking API
    2. Helped us focus less on networking and more on getting the game logic to work properly
    3. Real-time changes through Photon Networking

3. Game board designed with free, beautiful assets from Unity Asset Store

CHALLENGES

1. Connecting AR version of the game to Cardboard (ran into major issues with event handling)
   a) For example, preset events from "Event Trigger" -- "PointerDown" and "PointerClick" would not work for us
   b) Had to define our own "PointerClick" function using "PointerEnter" and "PointerExit"

2. Keeping track of game state for all players and quests across multiple files

3. Making sure all changes were properly networked and were consistent on all devices

4. Testing an instance of the game with 4 devices
   a) Could take up to 10 minutes to build and run and also iterate through the game checking for edge cases (players are assigned proper colors, quests change colors, "leader stand" given to the next player, etc.)

LIMITATIONS

1. 4 – player game with no AI (no scripted, "phantom" players)
2. Players must be on the same network to play (makes game ideal to play in one room but maybe not across long distances due to differences in available networks)
3. Certain image targets are easier to recognize:

VS.

➤ QR code wins:

Recognized from greater distance and in different lighting

POSSIBLE EXPANSIONS

1. Add more players and change the shape of the board accordingly (5 player board would be a pentagon, 6 player game would be a hexagon, etc.)
2. Adding an AI (have different levels -- Easy, Medium, Hard -- for single-player mode/when there are fewer than 4 players present)
3. Customized player roles (Merlin, Assassin, Mordred, etc.)
4. More dramatic animations (for example, when quest succeeds, enable particle system)
5. More environments (Mountain, Castle, Forest, Island, etc.)
6. Create a waiting room (where players connect before the actual start of the game and where players can select an environment of their choice)