

SCORE
00

HIGH-SCORE
4000



VR-CHESSES

MILESTONE 2

MENU

SIGN UP

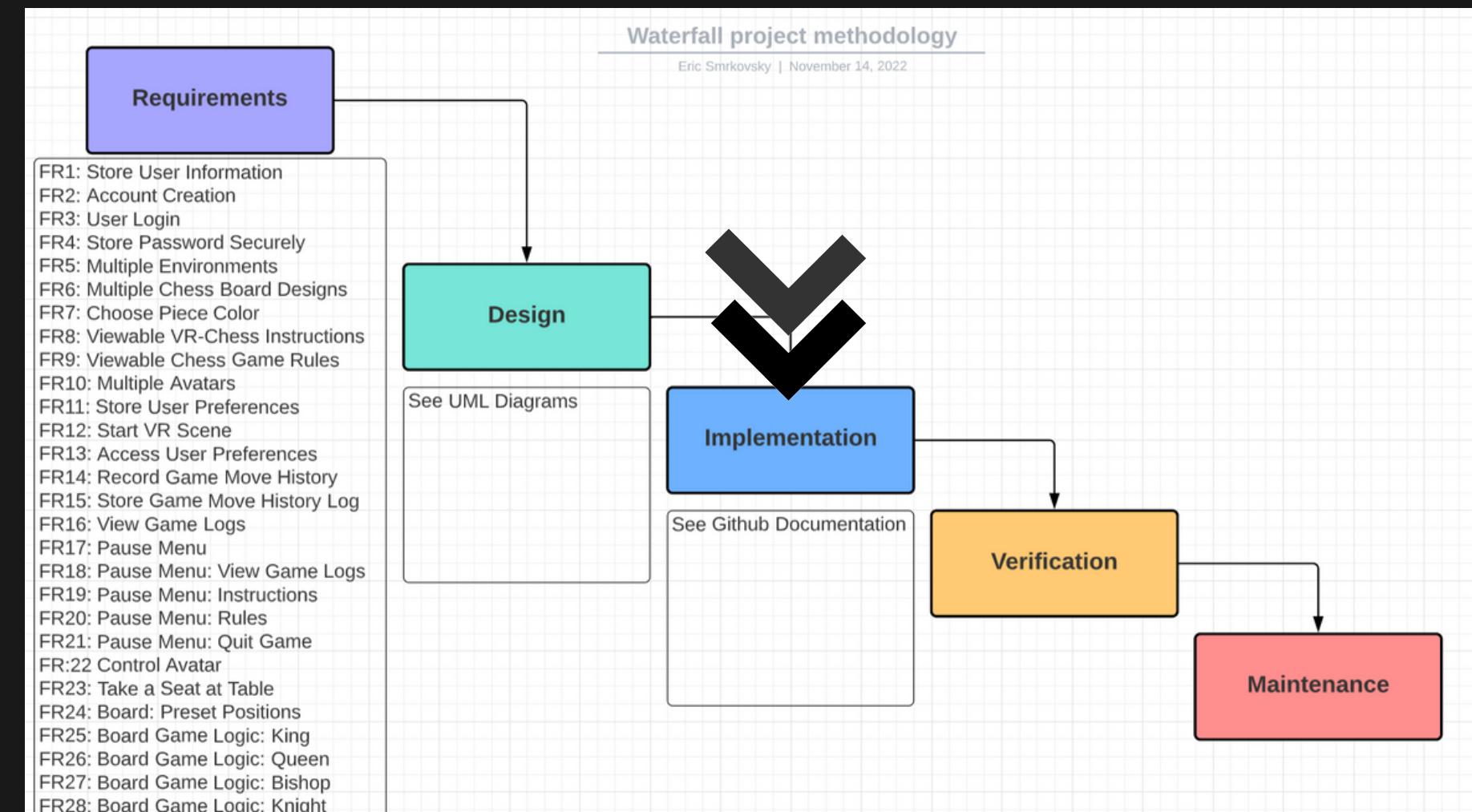


Introduction

VR-Chess Team Members

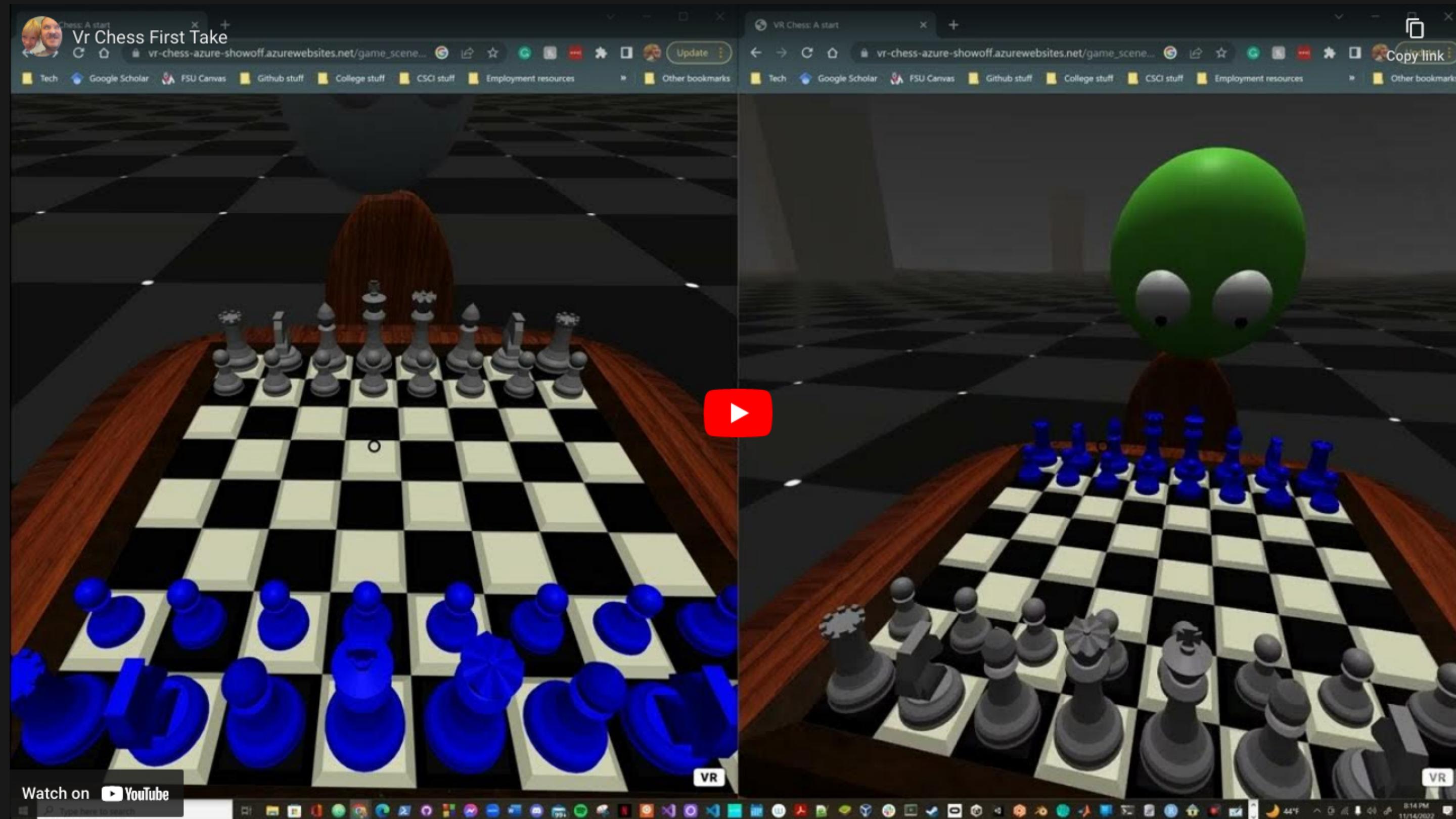
Team Member	Role
Eric Smrkovsky	Project Manager
Christian Leon	Back End Developer
Brett Harris	Graphics Design and Visuals
Jose Fernando Jimenez Chavez	Lead Technical Designer
Jacob Miller	Front End Developer

Milestone Two was all about the VR-Chess team working on the implementation phase of the waterfall approach to software development.



We have made significant progress on the project!

Introduction



Milestone two: User interface for choosing environment, board and pieces. Chess board is fully functional with logic for each piece. Networking is available for online multiplayer.

Features for Milestone Two

- Multiplayer is functional over the web
- Login form completed
- Database structure
- Piece movement logic
- Some sound implemented
- Different environments available

Working Features

As shown in the video we currently have the following functional requirements developed.

FR5: Multiple Environments

FR12: Start VR Scene

FR:22 Control Avatar in Virtual Environment

FR23: Take a Seat at Table

FR24: Board: Preset Positions of Pieces

FR25: Board Game Logic: King

FR26: Board Game Logic: Queen

FR27: Board Game Logic: Bishop

FR28: Board Game Logic: Knight

FR29: Board Game Logic: Pawn

FR30: Board Game Logic: Rook

FR31: Board: Piece Movement

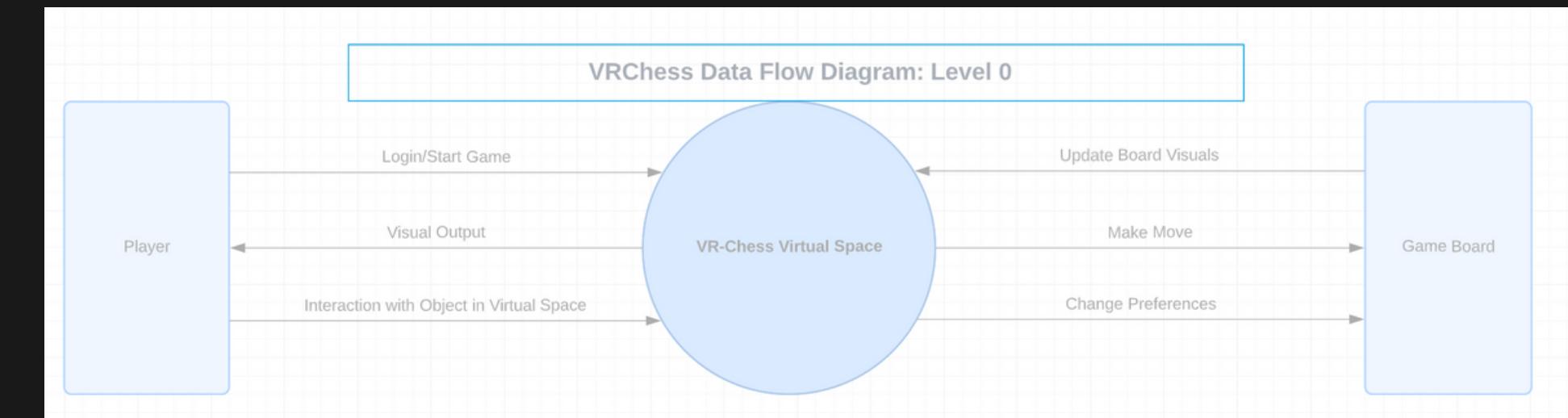
FR32: Board Game Logic: Kill

FR33: Board: Store Killed Pieces

FR34: Board: Red Square Highlight

FR35: Board: Green Square Highlight

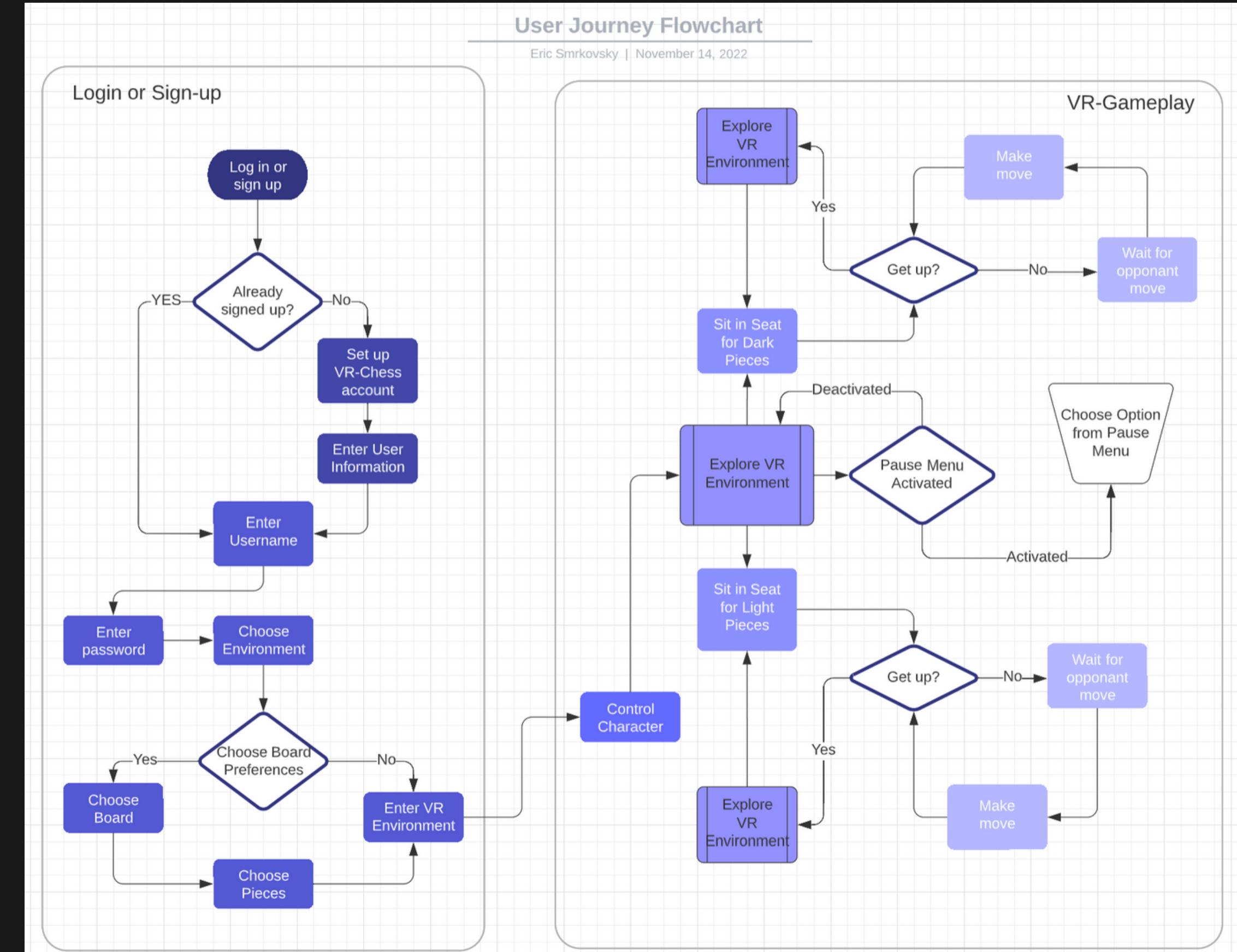
Highest level flow diagram



Working Features

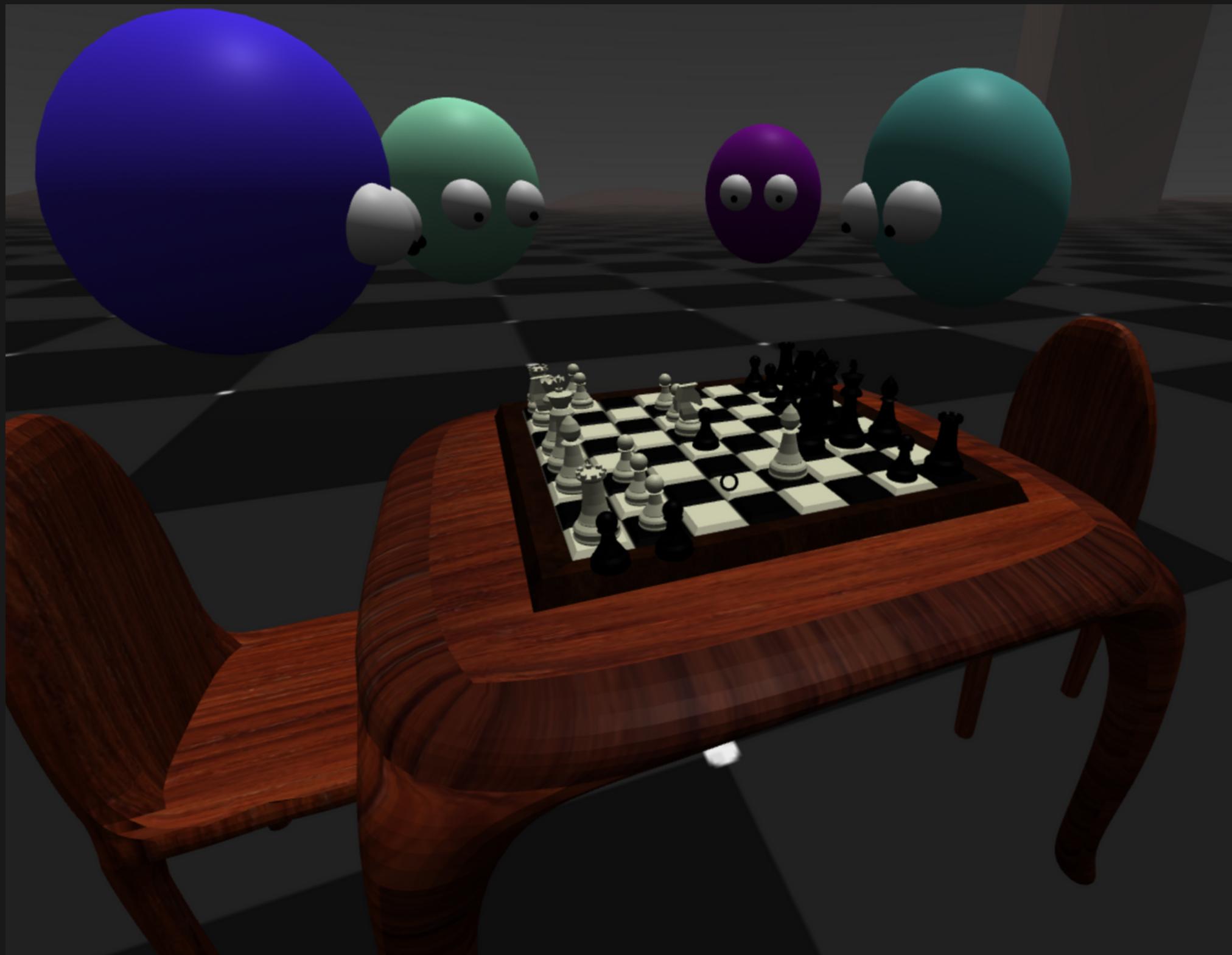
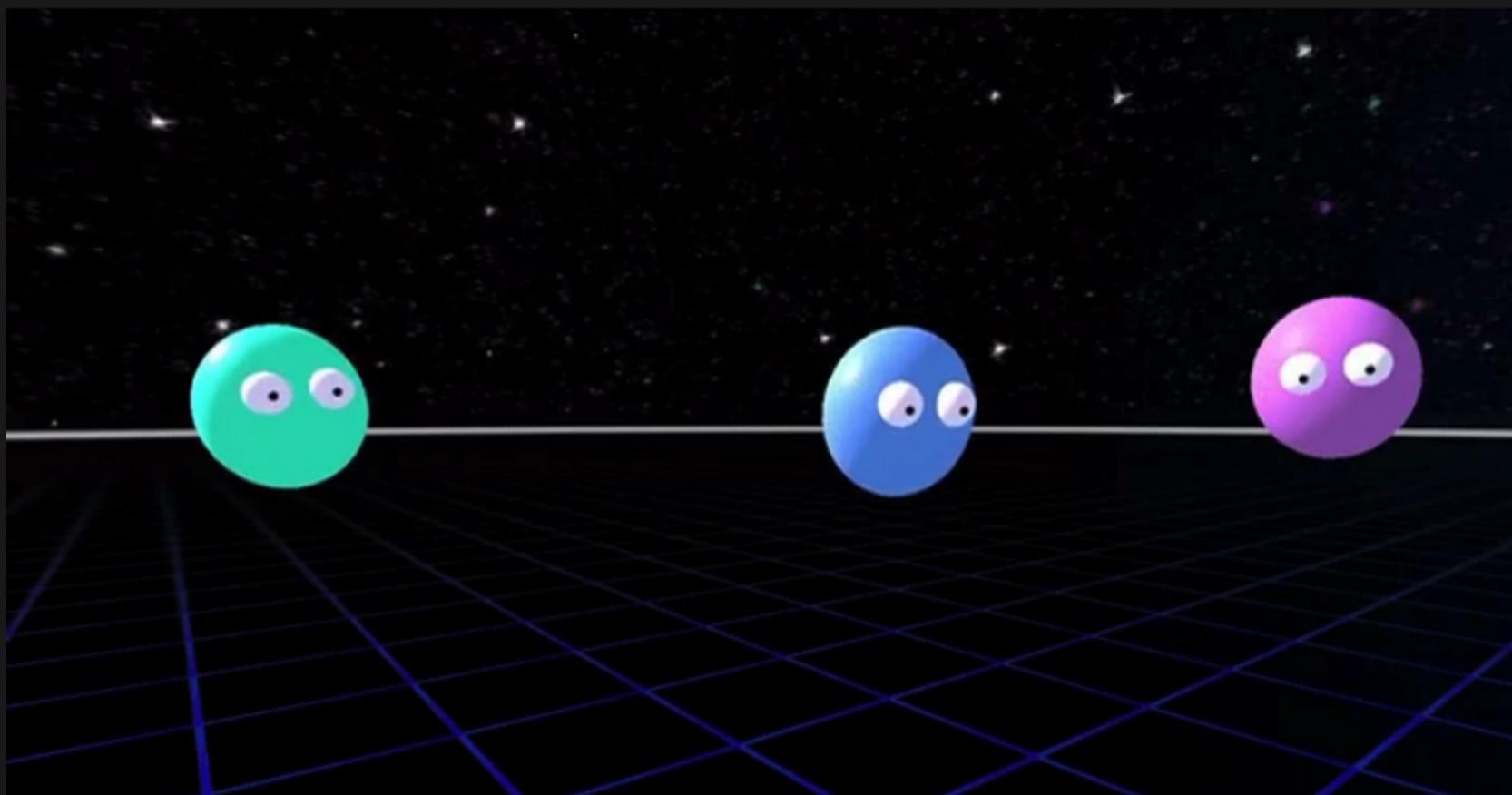
This matches the current user journey flowchart.

- FR5: Multiple Environments
- FR12: Start VR Scene
- FR22 Control Avatar in Virtual Environment
- FR23: Take a Seat at Table
- FR24: Board: Preset Positions of Pieces
 - FR25: Board Game Logic: King
 - FR26: Board Game Logic: Queen
 - FR27: Board Game Logic: Bishop
 - FR28: Board Game Logic: Knight
 - FR29: Board Game Logic: Pawn
 - FR30: Board Game Logic: Rook
 - FR31: Board: Piece Movement
 - FR32: Board Game Logic: Kill
 - FR33: Board: Store Killed Pieces
 - FR34: Board: Red Square Highlight
 - FR35: Board: Green Square Highlight



Features: Multiplayer

Two players (or more) can connect to a room and begin playing VR Chess together. This multi-user virtual reality experience uses the Networked A-Frame web framework to make any A-Frame experience multiplayer with minimal code changes. Currently: All chess piece movements synchronize, ownership is implemented, and dynamic rooms are available.



Features: Database

For our database we chose to use mySQL. This database allows us to store important information that unlocks more user features for us to be able to implement. We currently have 3 tables to store information about a users login credentials, game preferences, and statistics about their wins/losses.

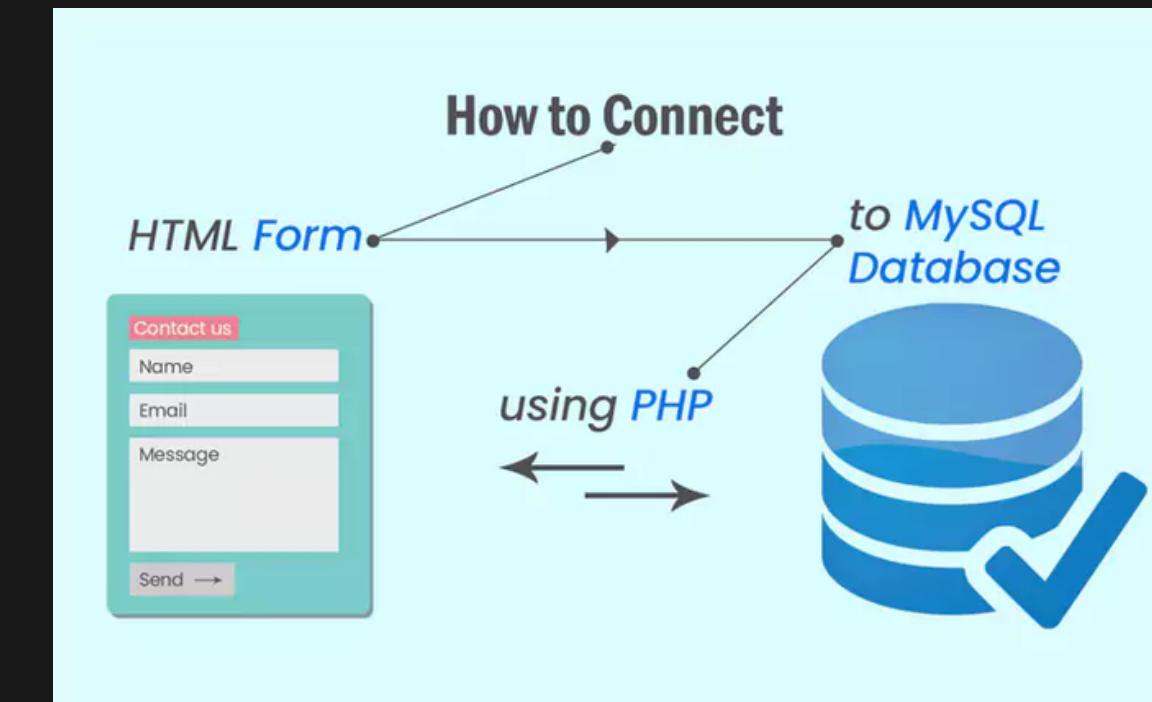
username	wins	losses
brad	0	1
mike	3	2
stacey	2	2
NULL	NULL	NULL

username	password
brad	12345
mike	abc123
stacey	44738b
NULL	NULL

username	boardtype	piececolor	envtype
brad	classic	blue	forest
mike	old	orange	ocean
stacey	old	pink	dark
NULL	NULL	NULL	NULL

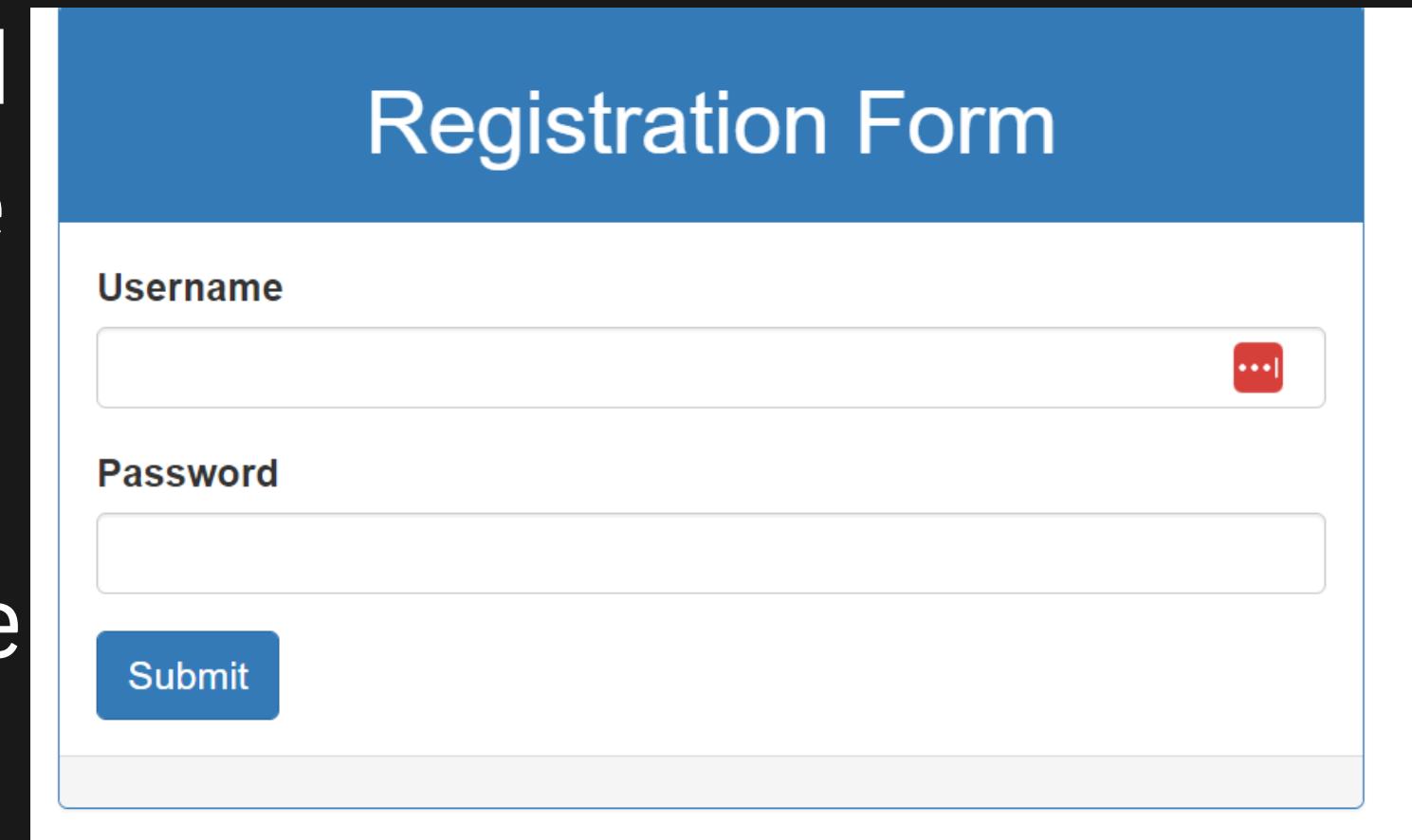
Features: Database

- Hosted by Heroku using clearDB:
- Communicated with using PHP:
- Using mySQL Workbench for management:
- Information to be extracted in .html files:



Features: Login Form

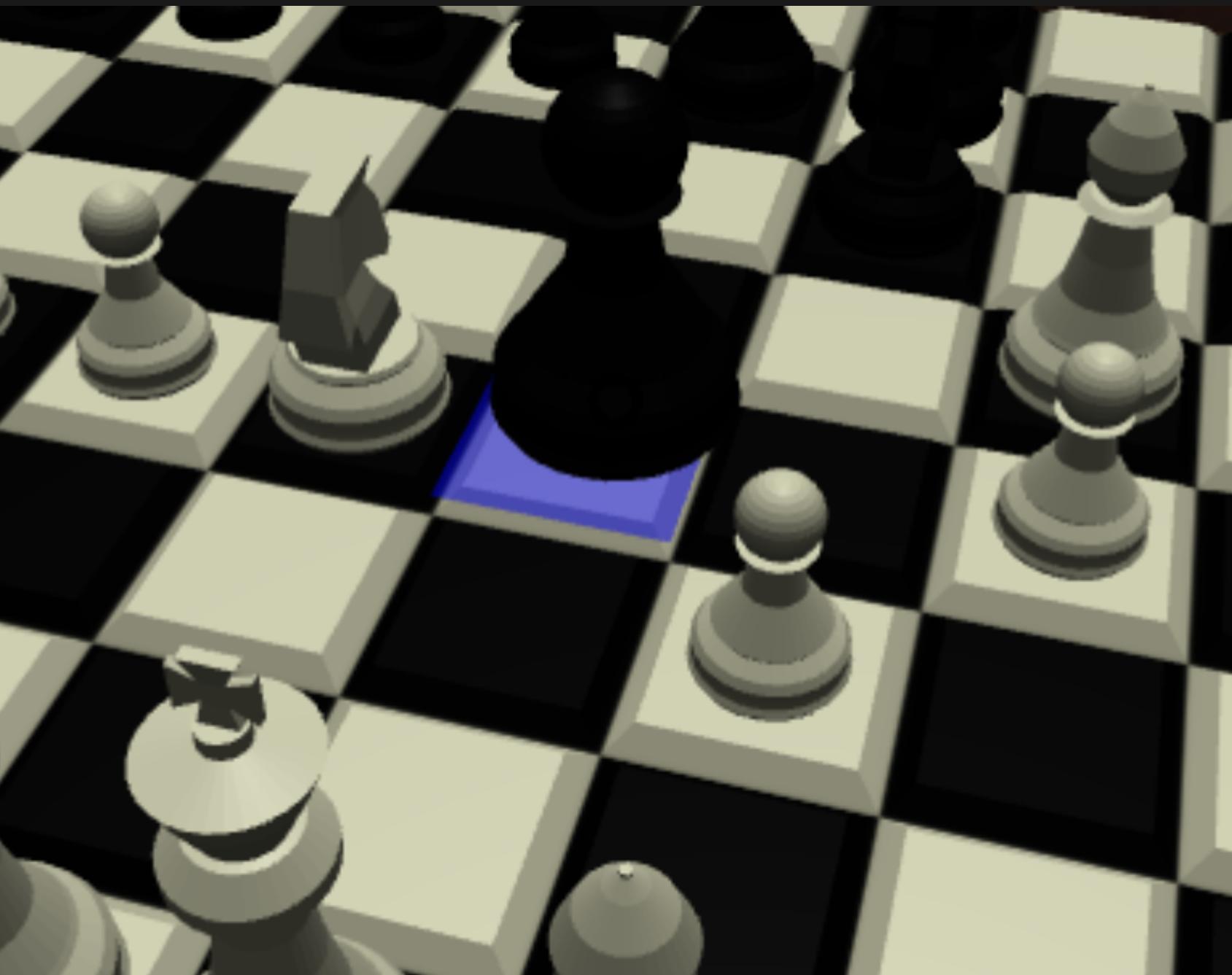
With the help of our database, we created a registration form for our users to create an account. This will be the initial step when a new player comes to our website. After creating an account they will be able to play chess and customize their account.



The image shows a registration form titled "Registration Form". It features two input fields: "Username" and "Password", each with a red "..." button to the right. Below the password field is a "Submit" button. The entire form is set against a light gray background with a blue header bar.

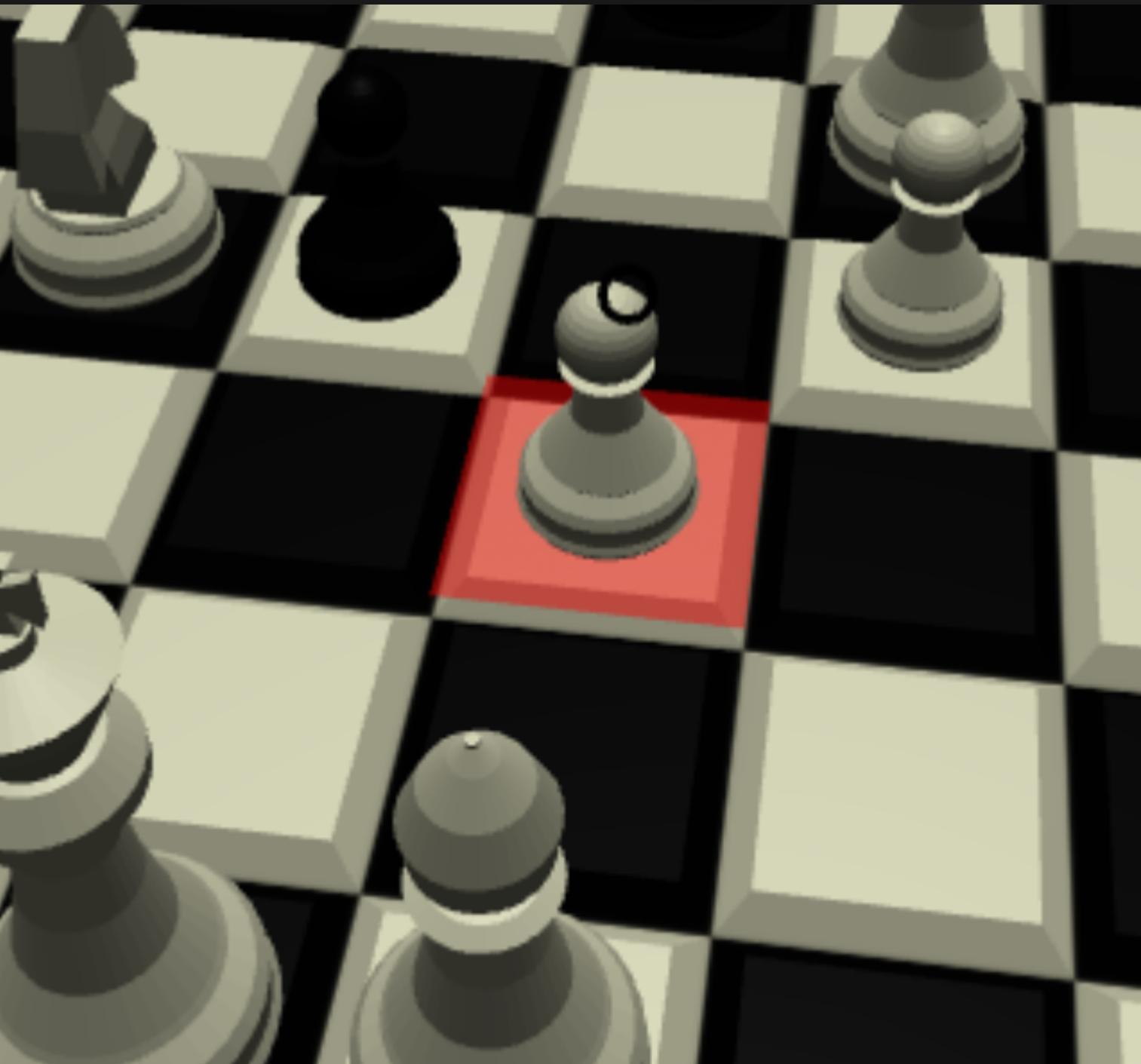
Features: Chess Logic

Logic for each piece was implemented for this milestone. Each chess piece has it's own specified possible moves on the board. When A player tries to move a piece during their turn, the square under the piece turns blue.



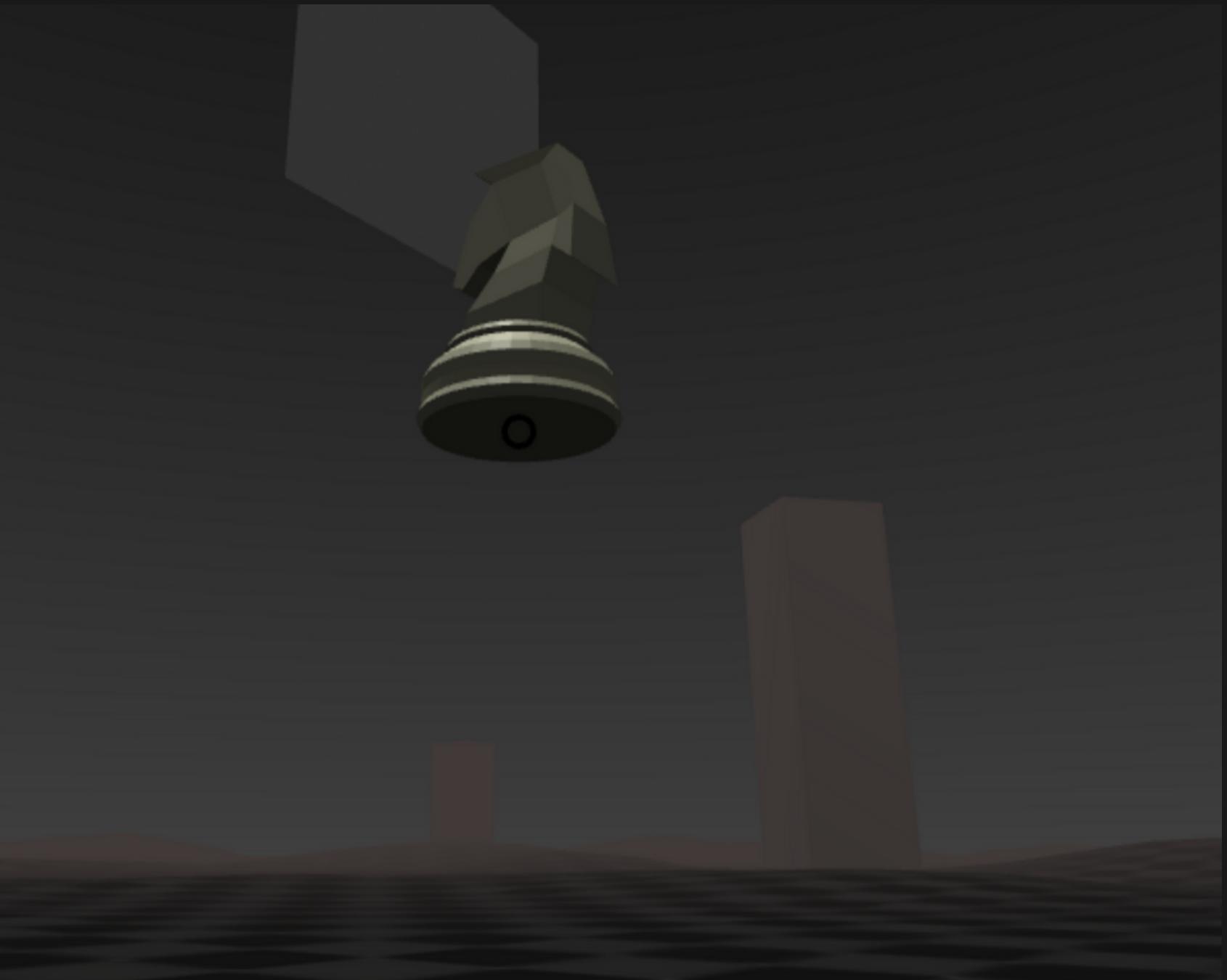
Features: Chess Logic

Logic for each piece was implemented for this milestone. Each chess piece has it's own specified possible moves on the board. When A player tries to move a piece during their turn, the square under the piece turns blue. When the piece is moved the square turns red.



Features: Chess Logic

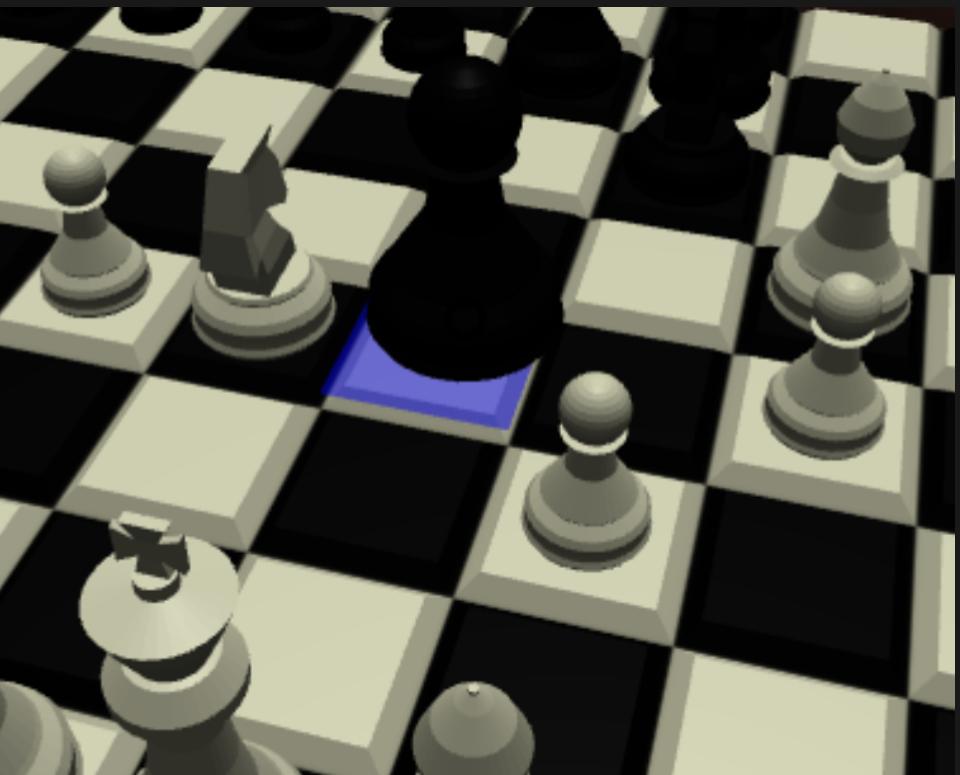
Logic for each piece was implemented for this milestone. Each chess piece has it's own specified possible moves on the board. When A player tries to move a piece during their turn, the square under the piece turns blue. When the piece is moved the square turns red. You can also grab a piece and hold it up to admire it's structure.



Features: Chess Logic

Logic for each piece was implemented for this milestone. Each chess piece has it's own specified possible moves on the board. When A player tries to move a piece during their turn, the square under the piece turns blue. When the piece is moved the square turns red. You can also grab a piece and hold it up to admire it's structure.

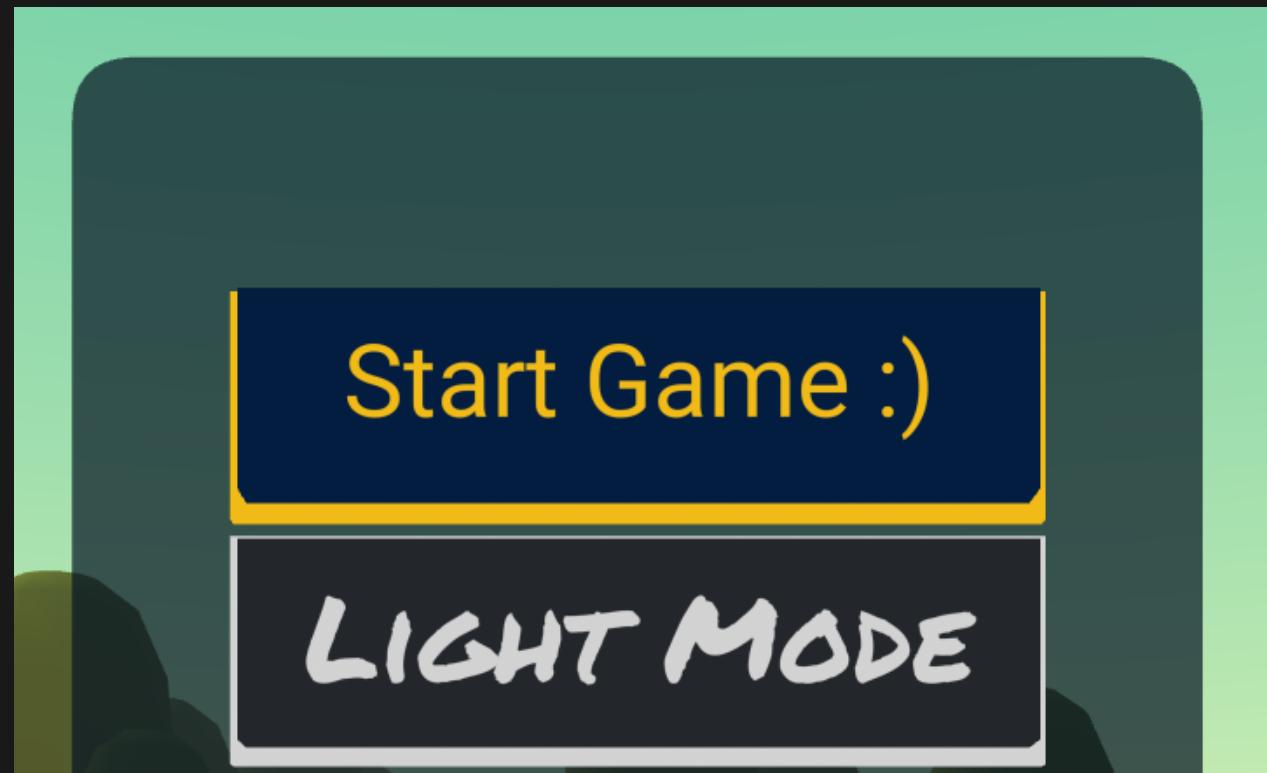
Pieces also make noise then they are moved. The milestone two prototype currently plays lovely music when a piece is taken.



Features: Environments

We currently have two options for environments.

The first we call Light Mode.

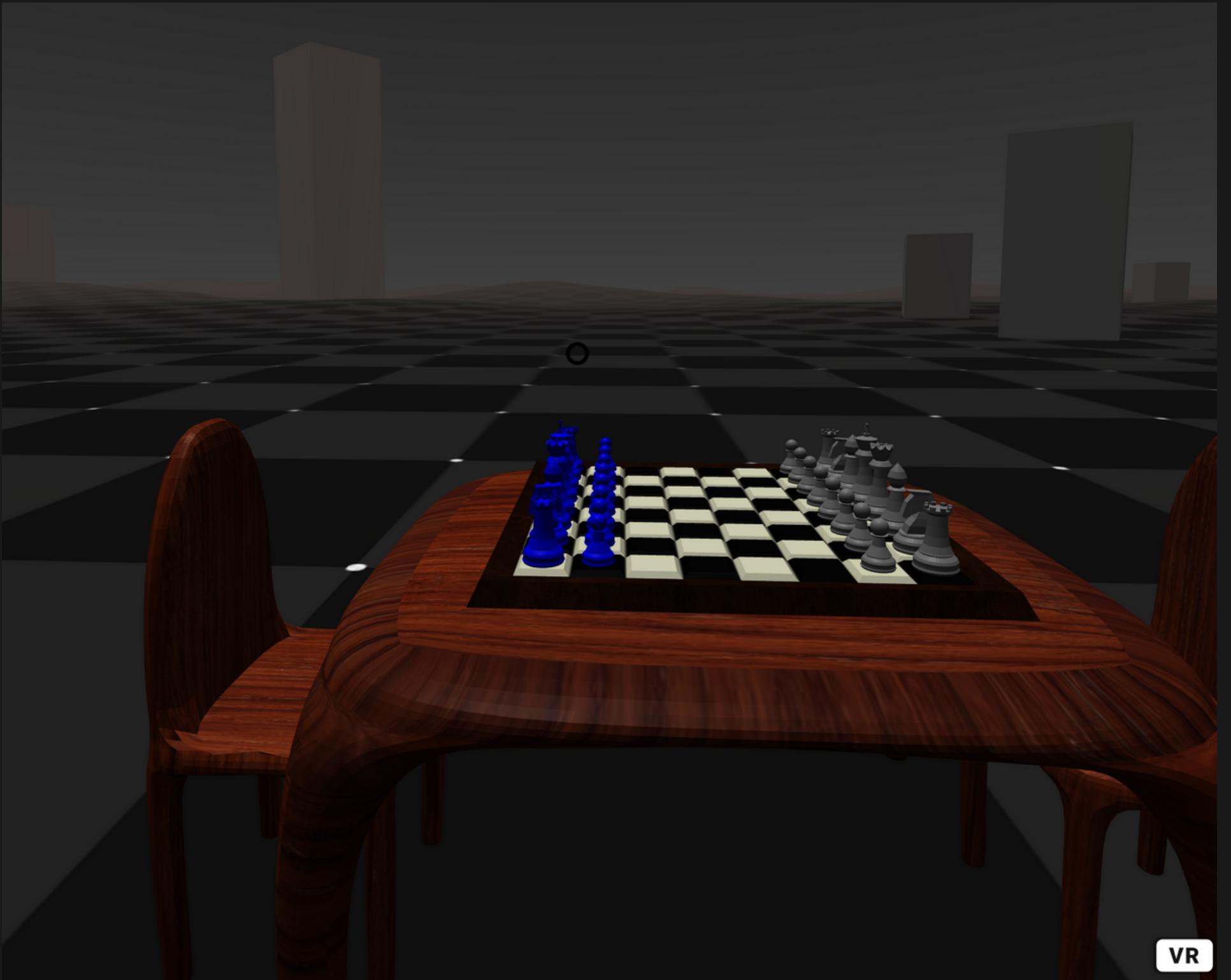


Features: Environments

We currently have two options for environments.

The first we call Light Mode.

The second we call Dark Mode.



Challenges and Contributions: Eric

My main tasks include: managing pull requests, writing requirements, creating UML diagrams, creating presentations, and organizing the workflow the best I can.

My challenges include: trying to write the requirements document and organize the documentation in such a way that will provide a roadmap for the team.

To be honest, the VR-Chess team members have done an amazing job with implementing features. More of my contribution will come later in the form of deliverables and documentation of the steps within the iterative SDLC approach we are using.

Challenges and Contributions: Brett

My main tasks include:

- More visual options
- Rework scenes
- Final graphical polish
- Send scene data between pages

My challenges include:

- Implementing all those tasks at the same time

Challenges and Contributions: Jose

My main tasks include:

- Reorganizing Project and adding dependencies to prepare project to be hosted as a node.js app
- Deployed to heroku, set up CD for automated redepolyment
- Integrated NAF to convert pieces to networked, persistent entities
- Used ownership NAF attribute, and it's transfer, to ensure pieces spawned only once and movements were synced across clients.

My challenges include:

- Mixin's not working correctly with templates.
- Duplication of pieces issue. Fix: Set ownership to non-player entity.
- Issues transferring ownership.
- Using more than 12 networking templates caused new bug

Challenges and Contributions: Chris

My main tasks include:

- Setting up a database.
- Creating tables to store information.
- Extracting information from our html files to send to our database using php.
- Creating a login and registration system.
- Setting this all to work on heroku.

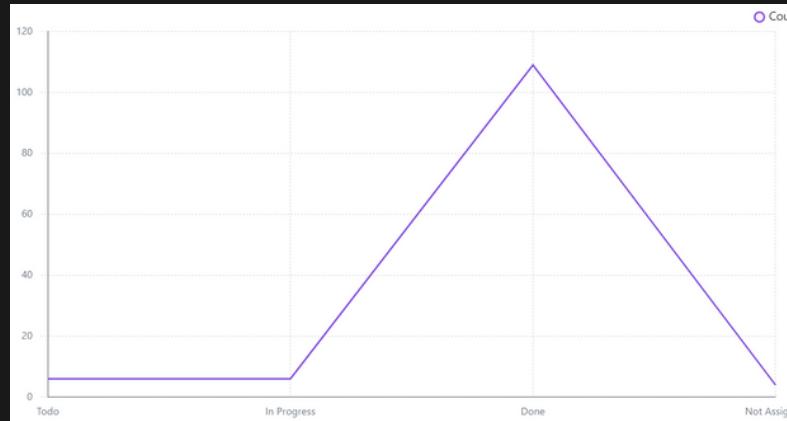
My challenges include:

- Learning how php works to communicate back and forth with our html files and database.
- Understanding how heroku works to deploy our web app.
- Understanding how login credentials work with our DB.

Challenges and Contributions: Jacob

My main tasks include:
Implementing piece logic

My challenges include: Not having enough bug spray for all
the bugs, they're everywhere.



Github Project Board

Currently we have some issues in progress and to do as seen here on our project board.

The team has made over 100 pull requests during the current implementation phase.

Filter by keyword or by field

Category	Count	Issues
Todo	6	<ul style="list-style-type: none">VR-Chess #16 Milestone 2VR-Chess #120 Pause MenuVR-Chess #101 Updating soundsVR-Chess #122 Environment 3VR-Chess #124 Account information in DBVR-Chess #125 User Preferences in DB
In Progress	6	<ul style="list-style-type: none">VR-Chess #66 SRSVR-Chess #67 WikiVR-Chess #68 Nice readmeVR-Chess #106 Adding account creationVR-Chess #97 Piece color changerVR-Chess #121 Dynamically created rooms
Done	109	<ul style="list-style-type: none">Implemented KILL/CAPTURE functionVR-Chess #62 test testVR-Chess #77 Networking for 32 piecesVR-Chess #69 Corrected several scene issues.VR-Chess #70 Fix positions for pieces and cursorVR-Chess #71 Fixed kill function and improved cameraVR-Chess #72 Create Glitch version of milestone 1 for presentationVR-Chess #73 Networked A-Frame ImplementationVR-Chess #76 Fixing the Folder StrutureVR-Chess #78 ... Pawn logic implemented. No highlight function.
Not Assigned	4	<ul style="list-style-type: none">VR-Chess #123 Game Move recorderVR-Chess #126 Create game InstructionsVR-Chess #127 Create Chess rules docVR-Chess #128 Create quit option in pause menu

- FR36: Board Game Logic: King Death
 - Description: The software will stop chess board piece movement when the death of a King piece occurs.
 - Dependency: FR32
- FR37: Board Game Logic: Detect Win
 - Description: The software must notify the user when the game has been won.
 - Dependency: FR36
- FR38: Board Game Logic: Detect Loss
 - Description: The software must notify the user when the game has been lost.
 - Dependency: FR36

- FR13: Access User Preferences
 - Description: The user must be able to choose to load saved preferences.
 - Dependency: FR5, FR6, FR7, FR10, FR11
- FR14: Record Game Move History
 - Description: The software must record all game moves for each game.
 - Dependency: F12, DB
- FR15: Store Game Move History Log
 - Description: The software must store the move log from each game.
 - Dependency: F12, DB
- FR16: View Game Logs
 - Description: The user must be able to view the record of all game logs.
 - Dependency: F12, DB
- FR17: Pause Menu
 - Description: The user must be able to pause the game, which triggers the menu.
 - Dependency: F12
- FR18: Pause Menu: View Game Logs
 - Description: The user must be able to select the option to view game logs from the pause menu.
 - Dependency: FR16, FR17, DB
- FR19: Pause Menu: Instructions
 - Description: The user must be able to select the option to view the VR-Chess Instructions document from the pause menu.
 - Dependency: FR8, FR17
- FR20: Pause Menu: Rules
 - Description: The user must be able to select the option to view the Chess Game Rules document from the pause menu.
 - Dependency: F12

ew the Chess Game Rules document from the pause

uit the game from the pause menu.

THANK YOU!
PLEASE TRY
PROTOTYPE!

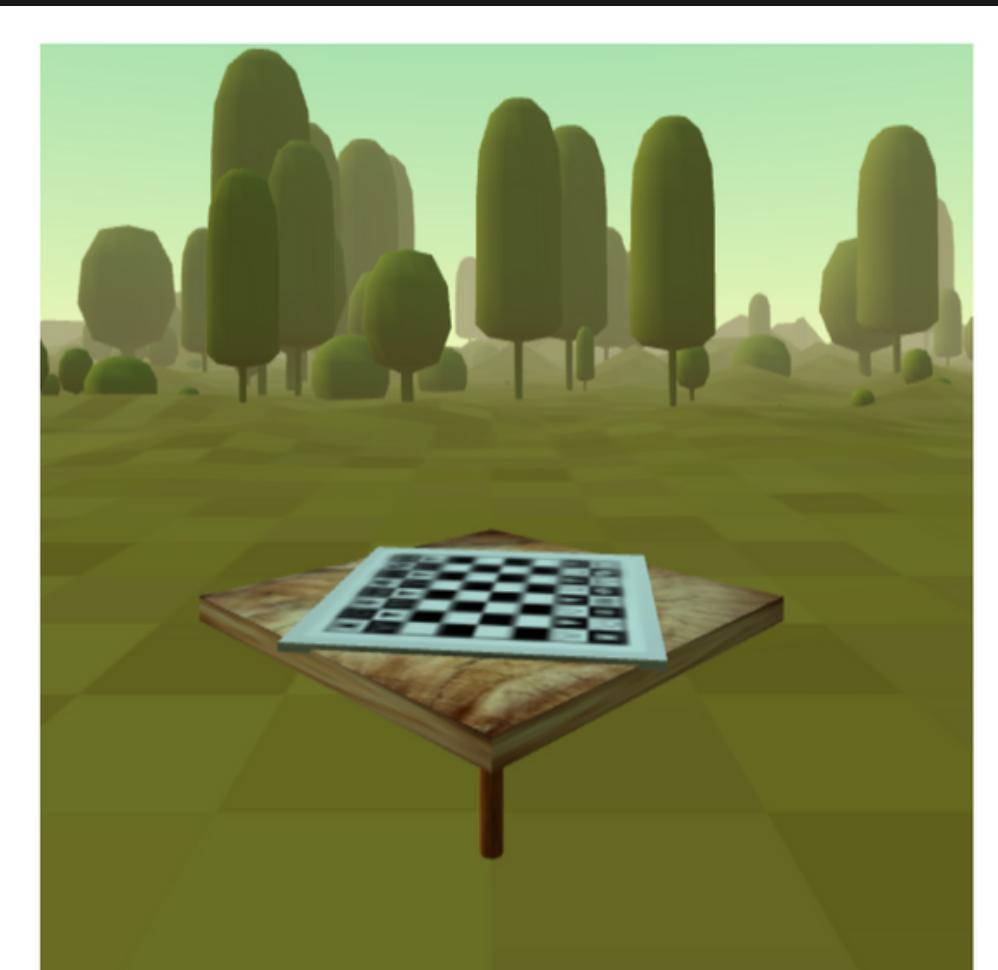
Easy way to access prototype:

Google "vr chess csci 150"

Click:

<https://github.com/Ericsmrk/VR-Chess> ·
Ericsmrk/VR-Chess: Term project for CSCI 150 ... - GitHub ✓

Then click ->



click photo to play with prototype

https://glacial-waters-73707.herokuapp.com/game_scene.html

PLEASE TRY
PROTOTYPE!

NOW AVAILABLE
ON PHONE!



Final Goal of the Project

To finish with the implementing the remaining functional requirements as listed in the Software Requirements Specification Document. This includes implementing a pause menu, adding more environments and finishing with the necessary steps for the database to work with our Heroku deployment. I am currently in the final stages of the SRS/SDD combination and the team is currently working on the issues shown on the project board. The next step in the SDLC will be implementing a testing plan.

<https://github.com/Ericsmrk/VR-Chess>