CRITTIUM

PRODUCTIONS

CRITTER CLASH

Team 2: CRITTIUM

Nick Parkman, Curtis Melton, Alysha McCullough, Bryan Snyder, Clark Rabe

Formally: Erik Games

MEET THE CRITTIUM TEAM!

PRODUCTIONS

Erik Powell (Client)

Curtis Melton

Nick Parkman

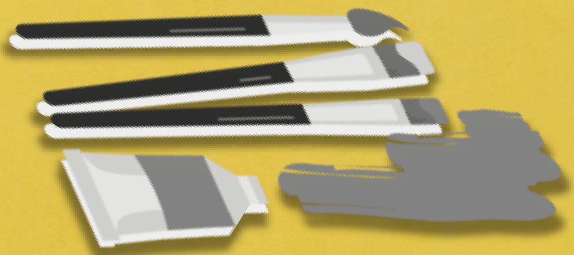
Bryan Snyder

Clark Rabe

Alysha McCullough

WHAT IS CRITTER CLASH?

A fun, interactive, and engaging solution that aims to make argumentative writing more entertaining for students and teachers alike. Critter Clash is an assortment of games that each have their own specific purpose in introducing and expanding on essential argumentative writing skills while maintaining a fun and open environment within the classroom.



REQUIREMENTS LIST

R1. There will be a feature that allows the host to disqualify answers. ✓

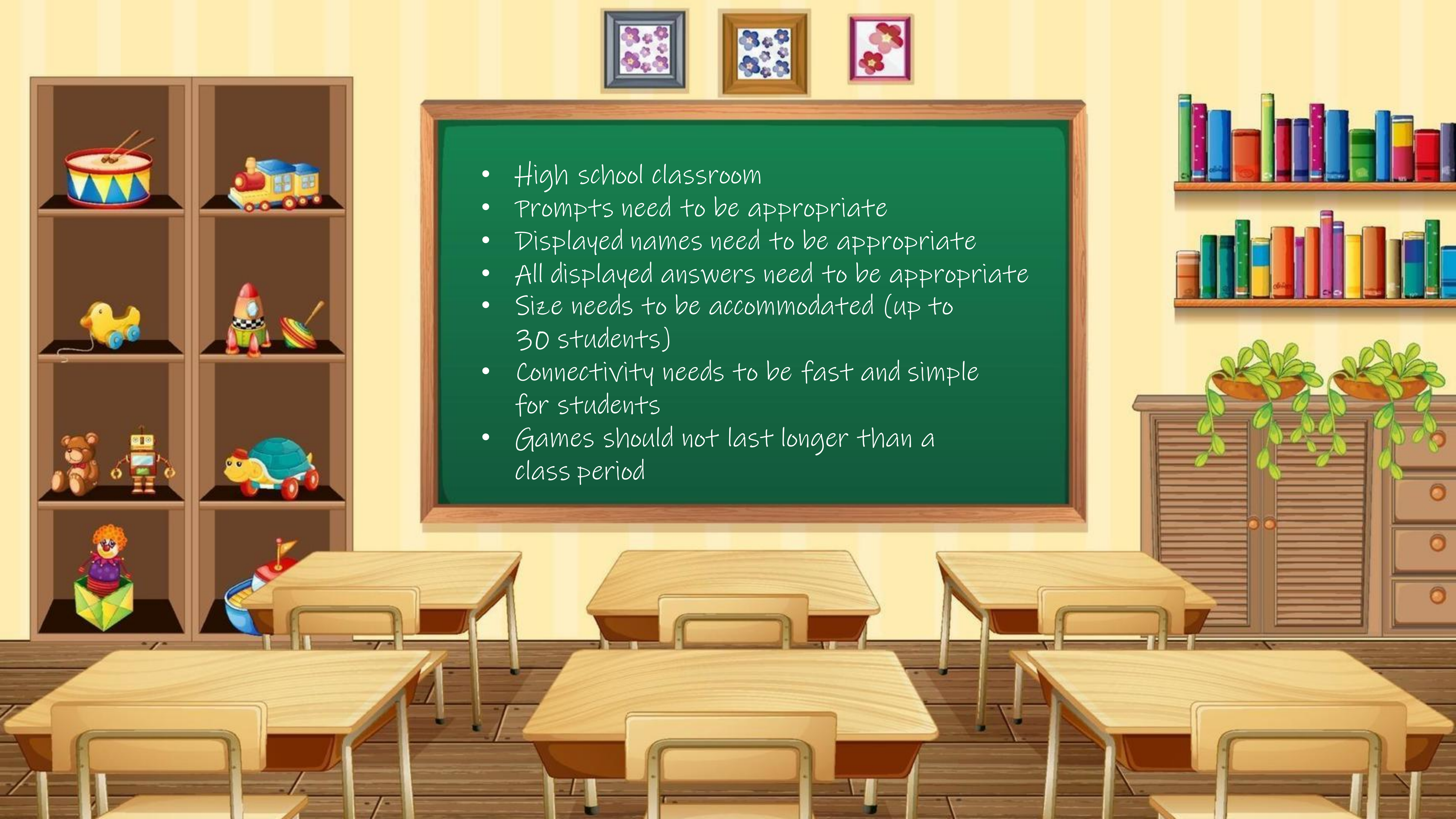
R2. Players will be able to choose their usernames. ✓

R3. There will be a filter on usernames that doesn't allow inappropriate language. ✓

R4. Players will be able to connect to the host's game through their phone. ✓

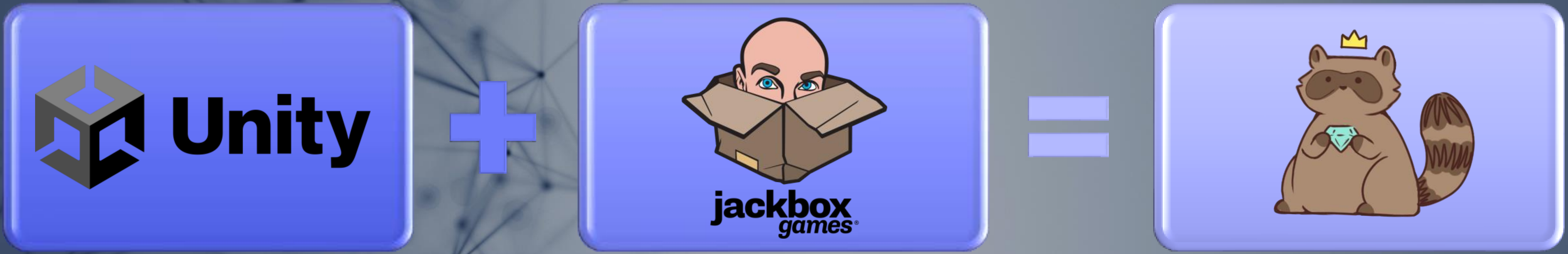
R5. There will be 3-5 minigames that the host can choose from. ✓

R6. There will be a scoreboard that is displayed, which will keep track of which player is winning. ✓



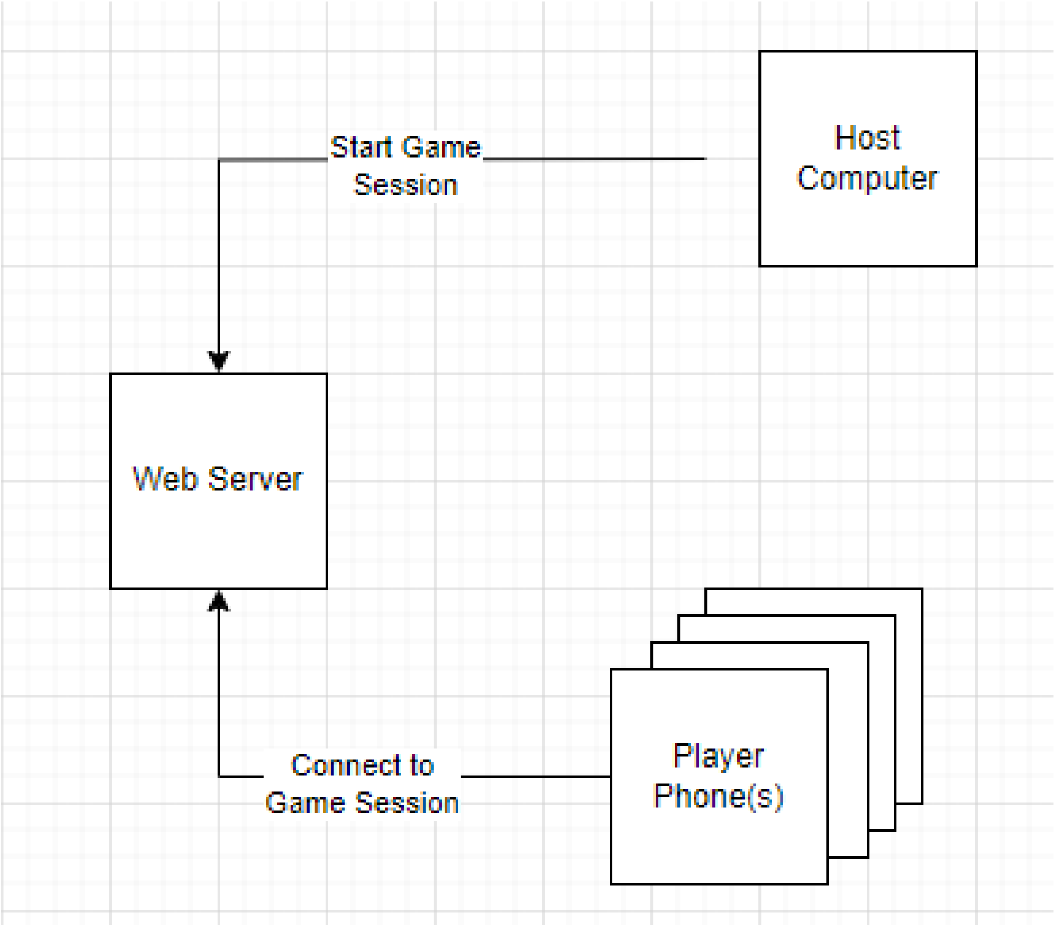
- High school classroom
- Prompts need to be appropriate
- Displayed names need to be appropriate
- All displayed answers need to be appropriate
- Size needs to be accommodated (up to 30 students)
- Connectivity needs to be fast and simple for students
- Games should not last longer than a class period

PROJECT SOLUTION APPROACH

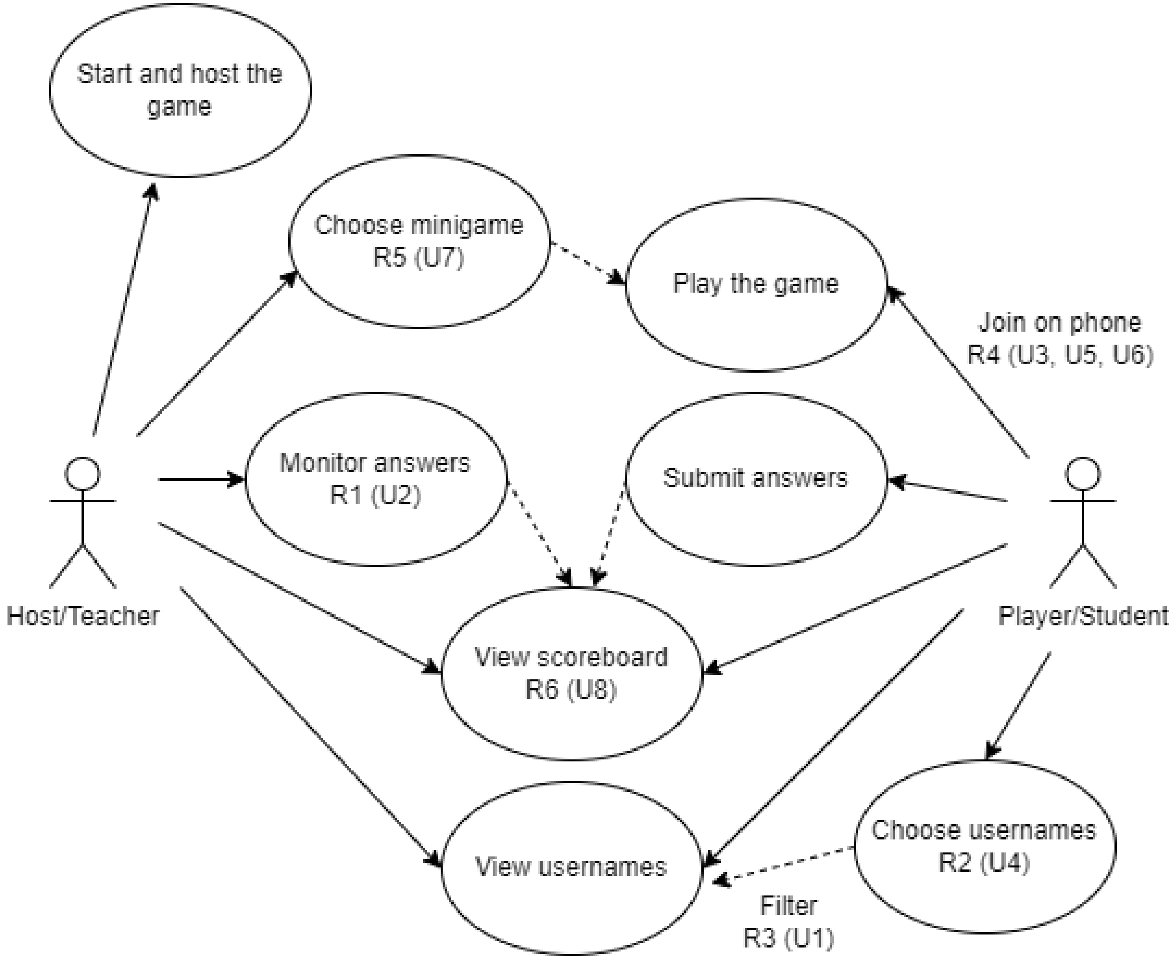


- Start by using Unity's Game Engine to develop the game itself.
- After a lot of head scratching, we decided to base our game on the popular "Jackbox Games"
- We want to keep **argumentative** writing as the main theme and implement those fundamentals into each game.

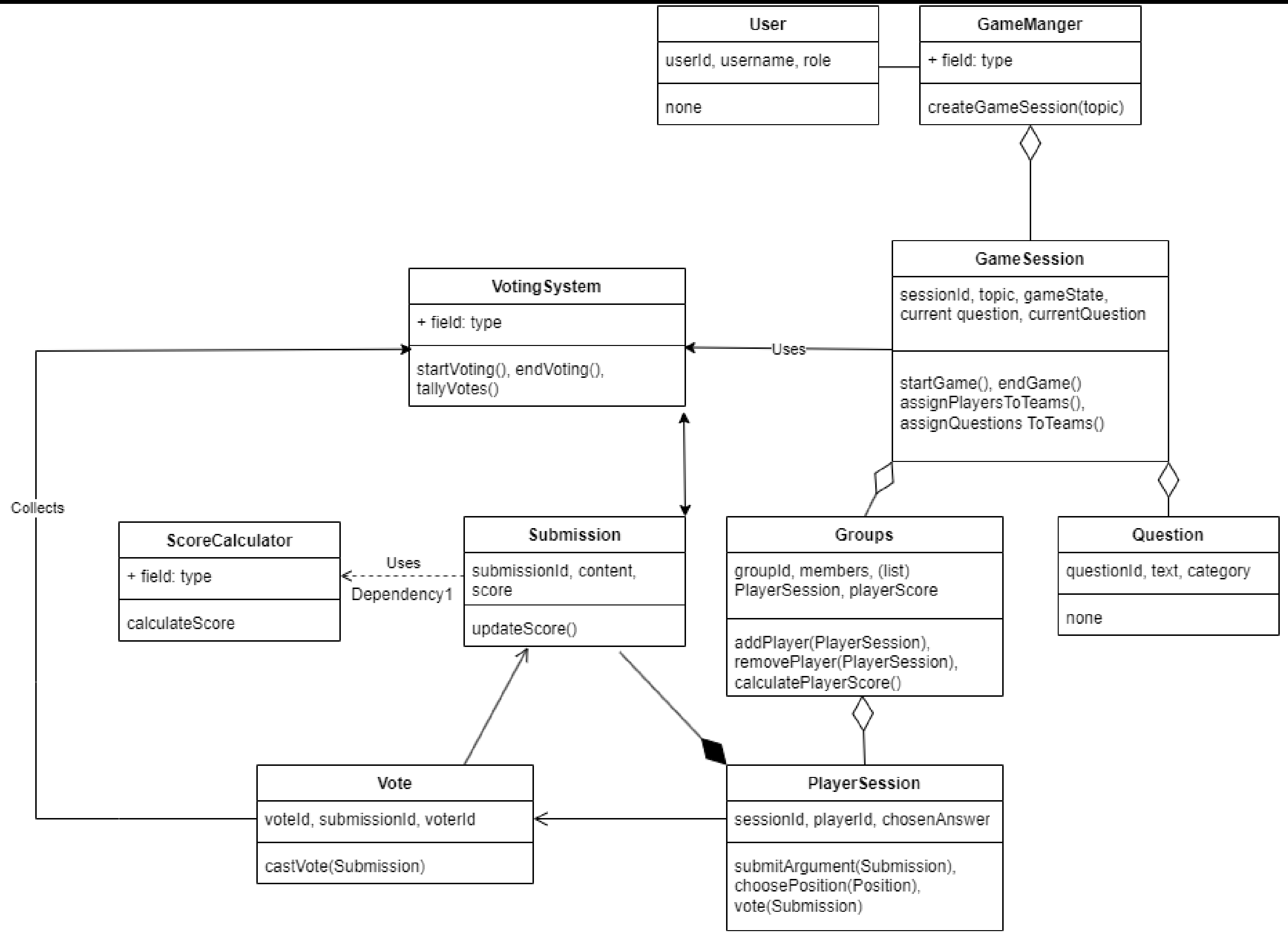
SYSTEM ARCHITECTURE DIAGRAM



USE CASE MODELING



STRUCTURAL MODELING: CLASS DIAGRAM



User
userId, username, role
none

GameManger
+ field: type
createGameSession(topic)

GameSession
sessionId, topic, gameState, current question, currentQuestion
startGame(), endGame() assignPlayersToTeams(), assignQuestions ToTeams()

VotingSystem
+ field: type
startVoting(), endVoting(), tallyVotes()

Uses

Collects

ScoreCalculator
+ field: type
calculateScore

Uses

Dependency1

Submission
submissionId, content, score
updateScore()

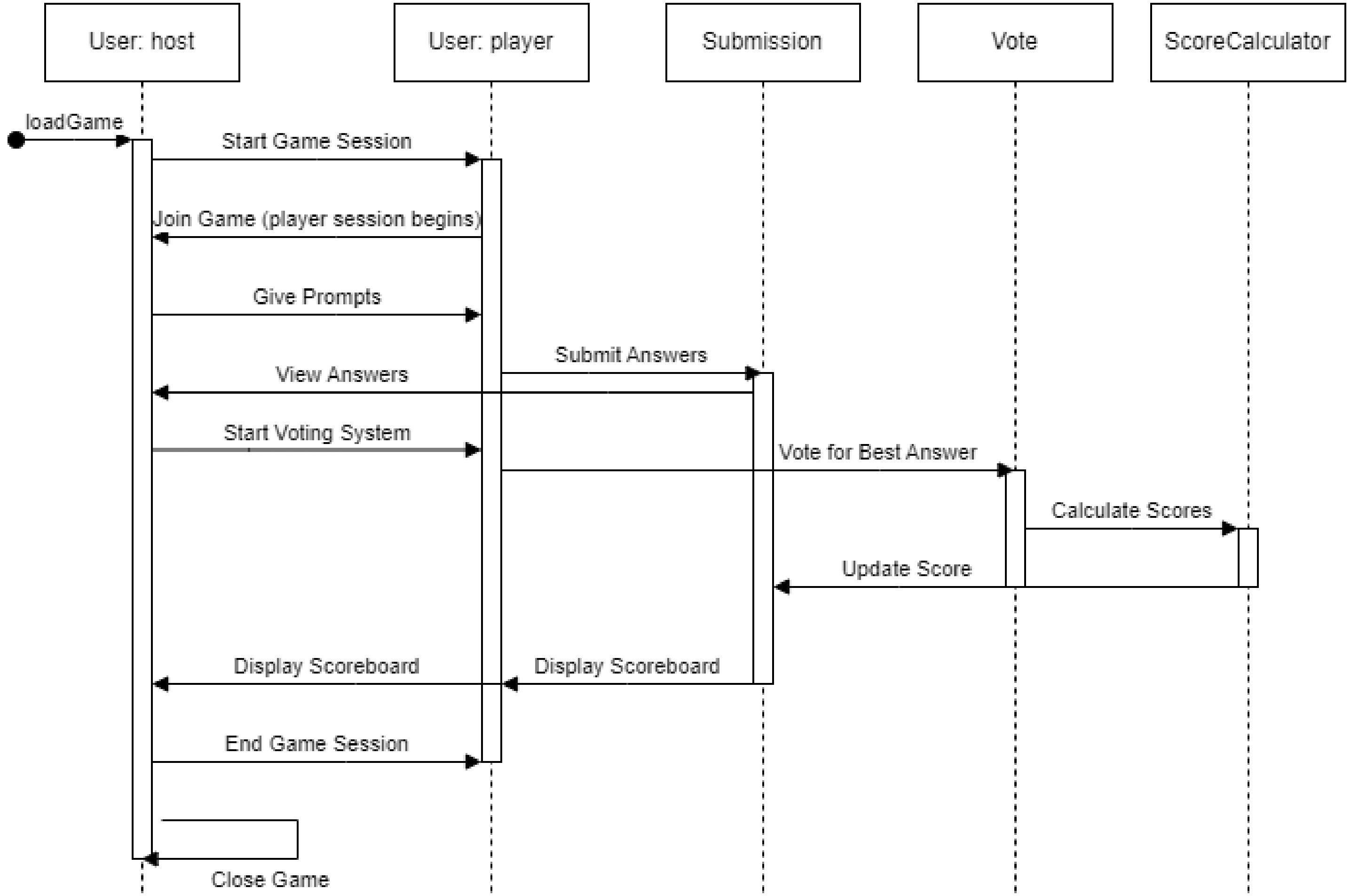
Groups
groupId, members, (list) PlayerSession, playerScore
addPlayer(PlayerSession), removePlayer(PlayerSession), calculatePlayerScore()

Question
questionId, text, category
none

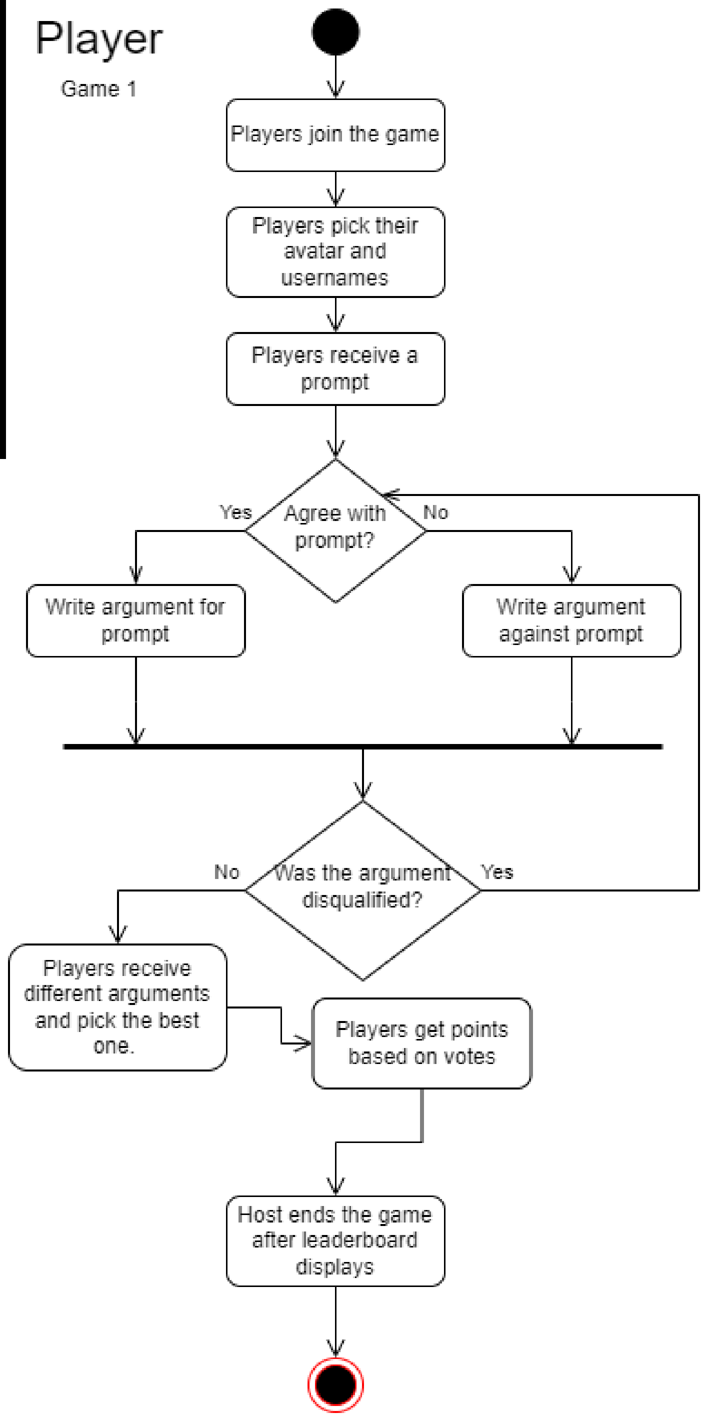
Vote
voteId, submissionId, voterId
castVote(Submission)

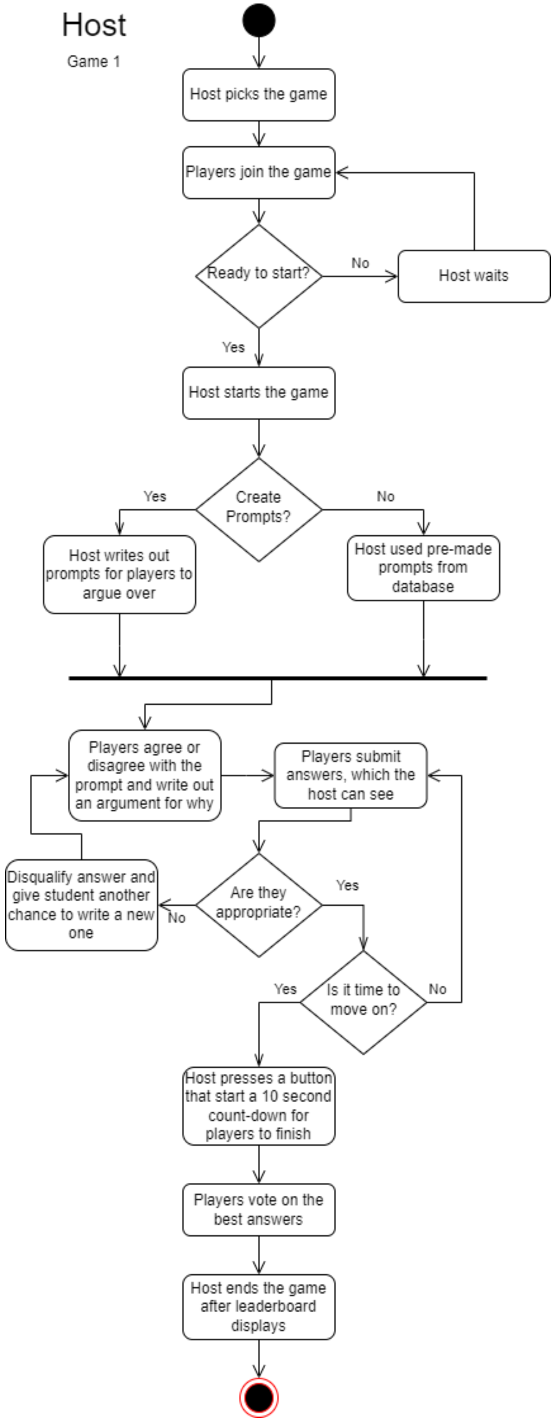
PlayerSession
sessionId, playerId, chosenAnswer
submitArgument(Submission), choosePosition(Position), vote(Submission)

BEHAVIORAL MODELING: SEQUENCE DIAGRAM

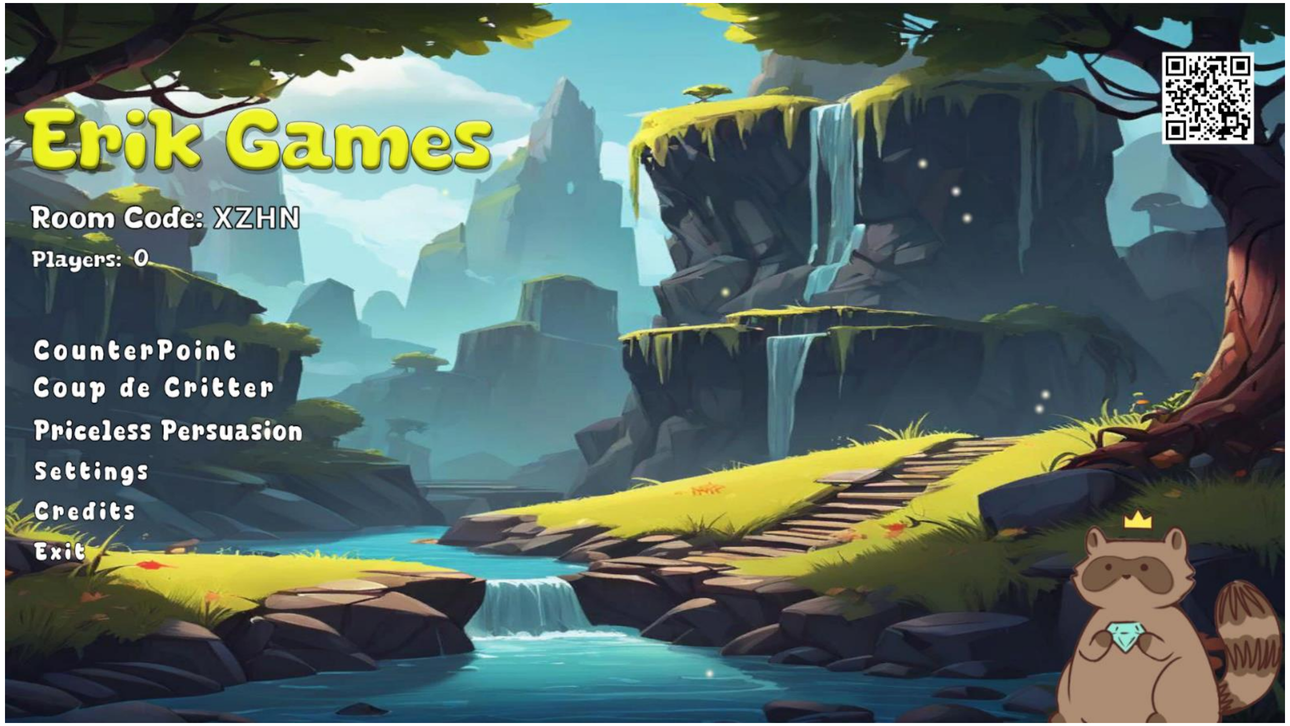


BEHAVIORAL MODELING: ACTIVITY DIAGRAM


Player
Game 1
Players join the game
Players pick their avatar and usernames
Players receive a prompt
Agree with prompt?
Yes
No
Write argument for prompt
Write argument against prompt
Was the argument disqualified?
No
Yes
Players receive different arguments and pick the best one.
Players get points based on votes
Host ends the game after leaderboard displays

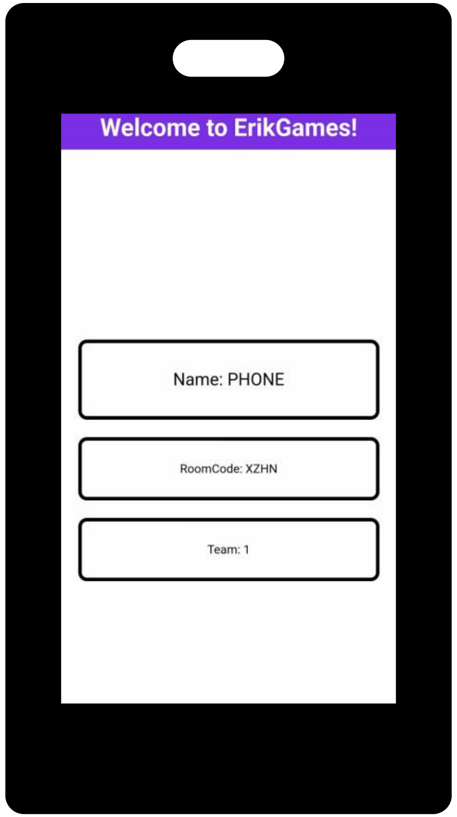

Host
Game 1
Host picks the game
Players join the game
Ready to start?
No
Host waits
Yes
Host starts the game
Create Prompts?
Yes
No
Host writes out prompts for players to argue over
Host used pre-made prompts from database
Players agree or disagree with the prompt and write out an argument for why
Players submit answers, which the host can see
Disqualify answer and give student another chance to write a new one
No
Are they appropriate?
Yes
Yes
Is it time to move on?
No
Host presses a button that start a 10 second count-down for players to finish
Players vote on the best answers
Host ends the game after leaderboard displays

SCREENSHOTS (FRONTEND)



Room Code: XZHN

Players: 0



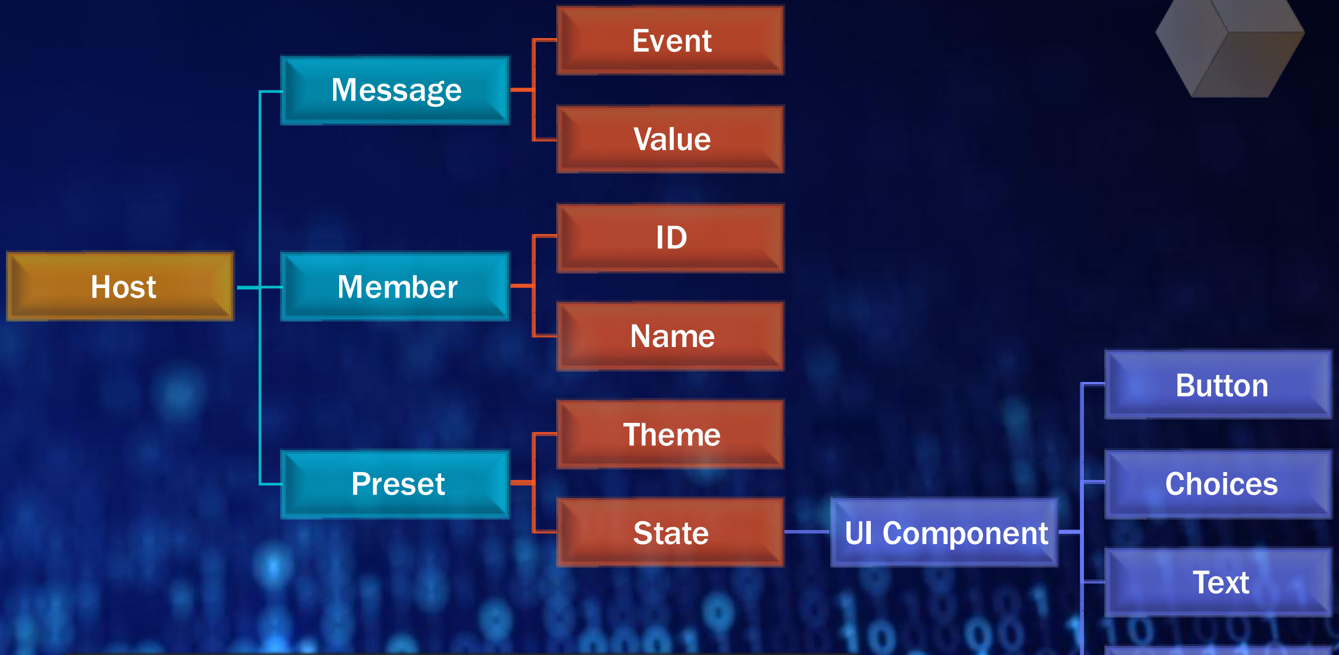
Name: PHONE

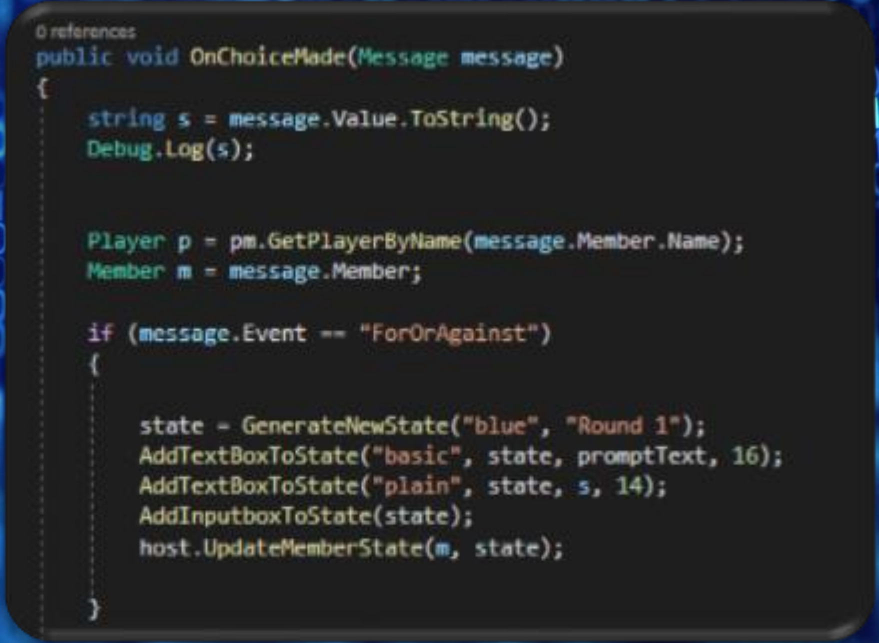
RoomCode: XZHN

Team: 1

[HACKBOX]

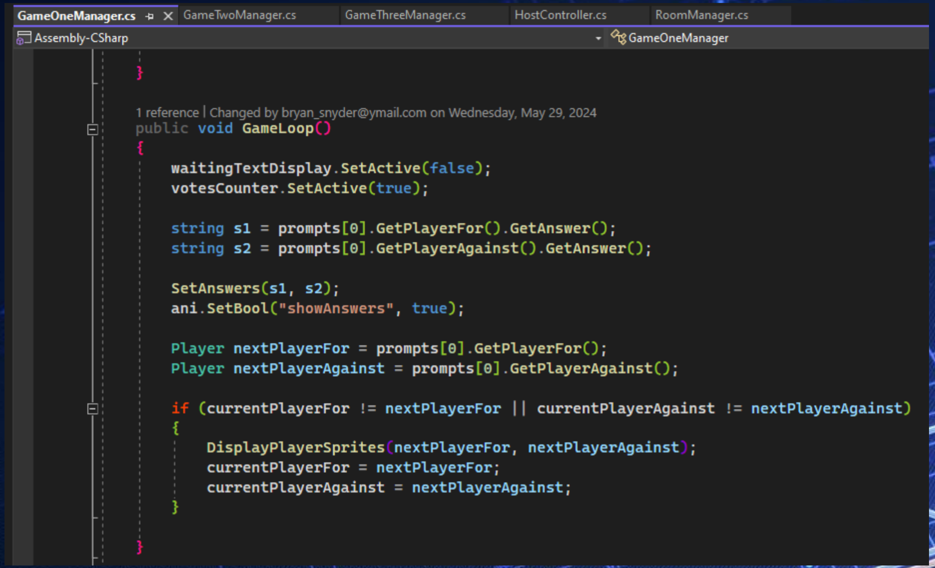
- A Unity Package that allows for its users to implement their own Jackbox-style games in Unity.
- Connects players to Unity projects via Hackbox.ca and allows both parties (host and players) to interact with one another.



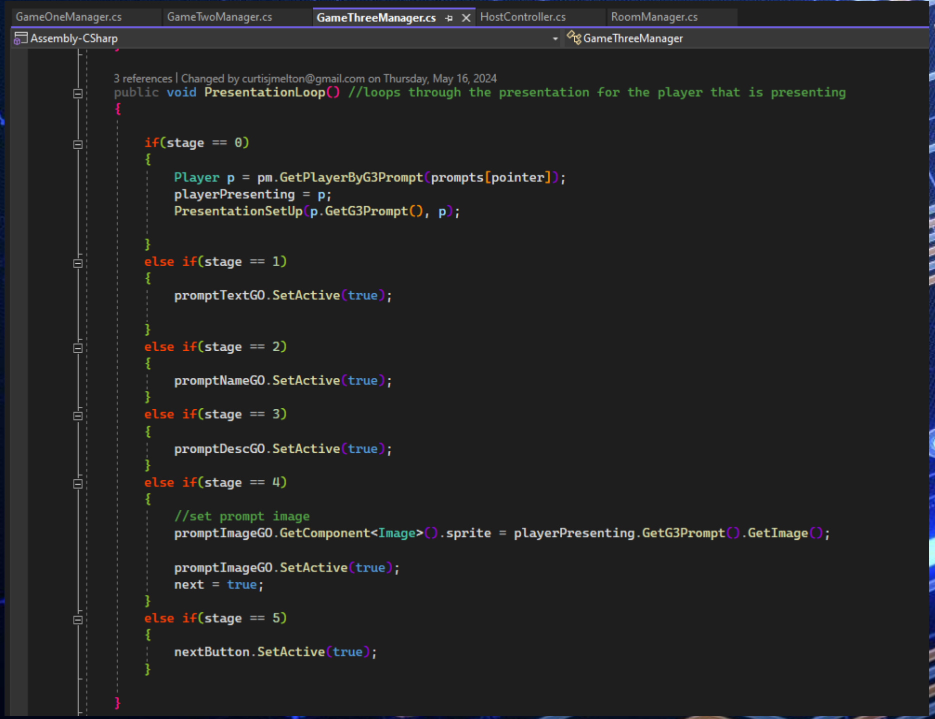


```
0 references
public void OnChoiceMade(Message message)
{
    string s = message.Value.ToString();
    Debug.Log(s);


    Player p = pm.GetPlayerByName(message.Member.Name);
    Member m = message.Member;

    if (message.Event == "ForOrAgainst")
    {

        state = GenerateNewState("blue", "Round 1");
        AddTextBoxToState("basic", state, promptText, 16);
        AddTextBoxToState("plain", state, s, 14);
        AddInputboxToState(state);
        host.UpdateMemberState(m, state);

    }
```

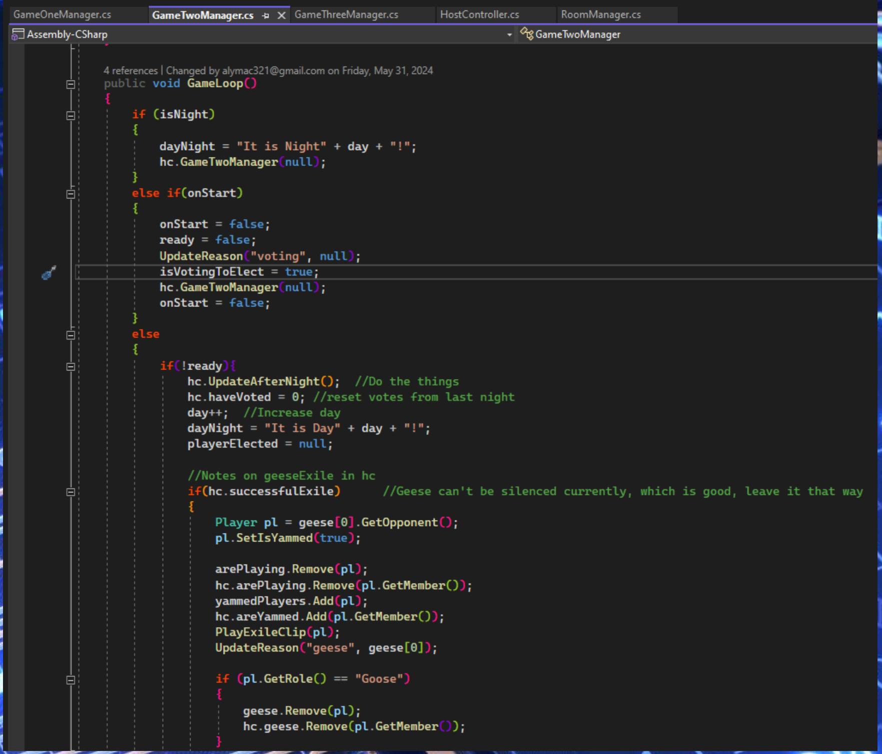
SCREENSHOTS (CODE)



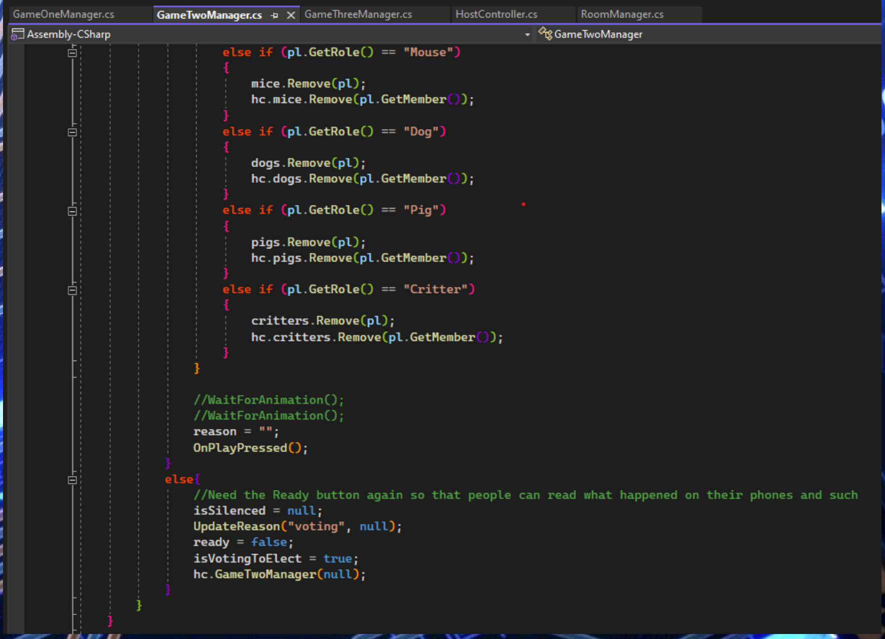
```
    }

    1 reference | Changed by bryan_snyder@ymail.com on Wednesday, May 29, 2024
    public void GameLoop()
    {
        waitingTextDisplay.SetActive(false);
        votesCounter.SetActive(true);

        string s1 = prompts[0].GetPlayerFor().GetAnswer();
        string s2 = prompts[0].GetPlayerAgainst().GetAnswer();

        SetAnswers(s1, s2);
        ani.SetBool("showAnswers", true);

        Player nextPlayerFor = prompts[0].GetPlayerFor();
        Player nextPlayerAgainst = prompts[0].GetPlayerAgainst();

        if (currentPlayerFor != nextPlayerFor || currentPlayerAgainst != nextPlayerAgainst)
        {
            DisplayPlayerSprites(nextPlayerFor, nextPlayerAgainst);
            currentPlayerFor = nextPlayerFor;
            currentPlayerAgainst = nextPlayerAgainst;
        }

    }
```



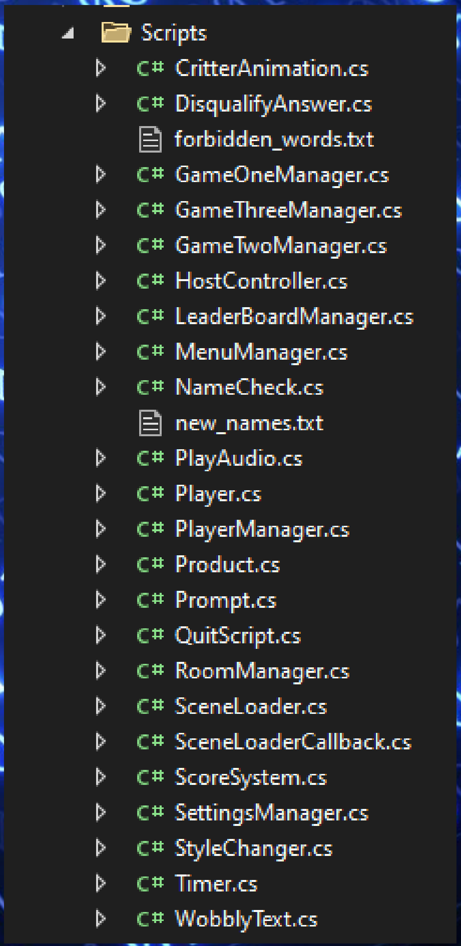
```
    3 references | Changed by curtisjmelton@gmail.com on Thursday, May 16, 2024
    public void PresentationLoop() //loops through the presentation for the player that is presenting
    {

        if(stage == 0)
        {
            Player p = pm.GetPlayerByG3Prompt(prompts[pointer]);
            playerPresenting = p;
            PresentationSetUp(p.GetG3Prompt(), p);

        }
        else if(stage == 1)
        {
            promptTextGO.SetActive(true);

        }
        else if(stage == 2)
        {
            promptNameGO.SetActive(true);
        }
        else if(stage == 3)
        {
            promptDescGO.SetActive(true);
        }
        else if(stage == 4)
        {
            //set prompt image
            promptImageGO.GetComponent<Image>().sprite = playerPresenting.GetG3Prompt().GetImage();

            promptImageGO.SetActive(true);
            next = true;
        }
        else if(stage == 5)
        {
            nextButton.SetActive(true);
        }

    }
```



```
    4 references | Changed by alymac321@gmail.com on Friday, May 31, 2024
    public void GameLoop()
    {
        if (isNight)
        {
            dayNight = "It is Night" + day + "!";
            hc.GameTwoManager(null);
        }
        else if(onStart)
        {
            onStart = false;
            ready = false;
            UpdateReason("voting", null);
            isVotingToElect = true;
            hc.GameTwoManager(null);
            onStart = false;
        }
        else
        {
            if(!ready){
                hc.UpdateAfterNight();  //Do the things
                hc.haveVoted = 0; //reset votes from last night
                day++;  //Increase day
                dayNight = "It is Day" + day + "!";
                playerElected = null;

                //Notes on geeseExile in hc
                if(hc.successfulExile)      //Geese can't be silenced currently, which is good, leave it that way
                {
                    Player pl = geese[0].GetOpponent();
                    pl.SetIsYammed(true);

                    arePlaying.Remove(pl);
                    hc.arePlaying.Remove(pl.GetMember());
                    yammedPlayers.Add(pl);
                    hc.areYammed.Add(pl.GetMember());
                    PlayExileClip(pl);
                    UpdateReason("geese", geese[0]);

                    if (pl.GetRole() == "Goose")
                    {
                        geese.Remove(pl);
                        hc.geese.Remove(pl.GetMember());
                    }
```



```
                    else if (pl.GetRole() == "Mouse")
                    {
                        mice.Remove(pl);
                        hc.mice.Remove(pl.GetMember());
                    }
                    else if (pl.GetRole() == "Dog")
                    {
                        dogs.Remove(pl);
                        hc.dogs.Remove(pl.GetMember());
                    }
                    else if (pl.GetRole() == "Pig")
                    {
                        pigs.Remove(pl);
                        hc.pigs.Remove(pl.GetMember());
                    }
                    else if (pl.GetRole() == "Critter")
                    {
                        critters.Remove(pl);
                        hc.critters.Remove(pl.GetMember());
                    }
                }

                //WaitForAnimation();
                //WaitForAnimation();
                reason = "";
                OnPlayPressed();
            }
            else{
                //Need the Ready button again so that people can read what happened on their phones and such
                isSilenced = null;
                UpdateReason("voting", null);
                ready = false;
                isVotingToElect = true;
                hc.GameTwoManager(null);
            }
        }
    }
```

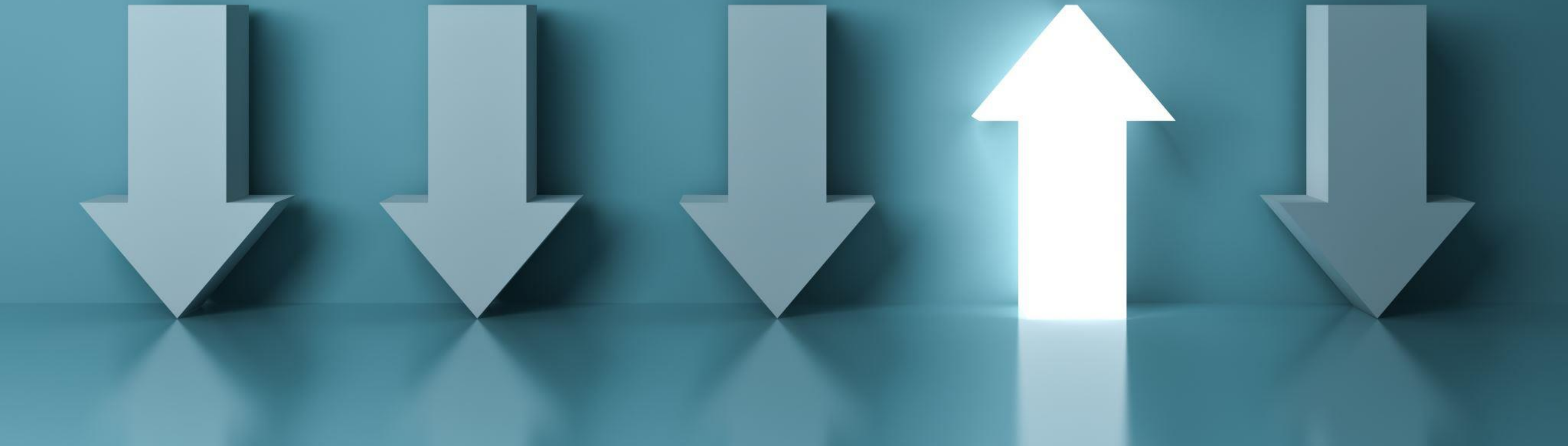


DEMO VIDEOS *(HYPERLINKS)*

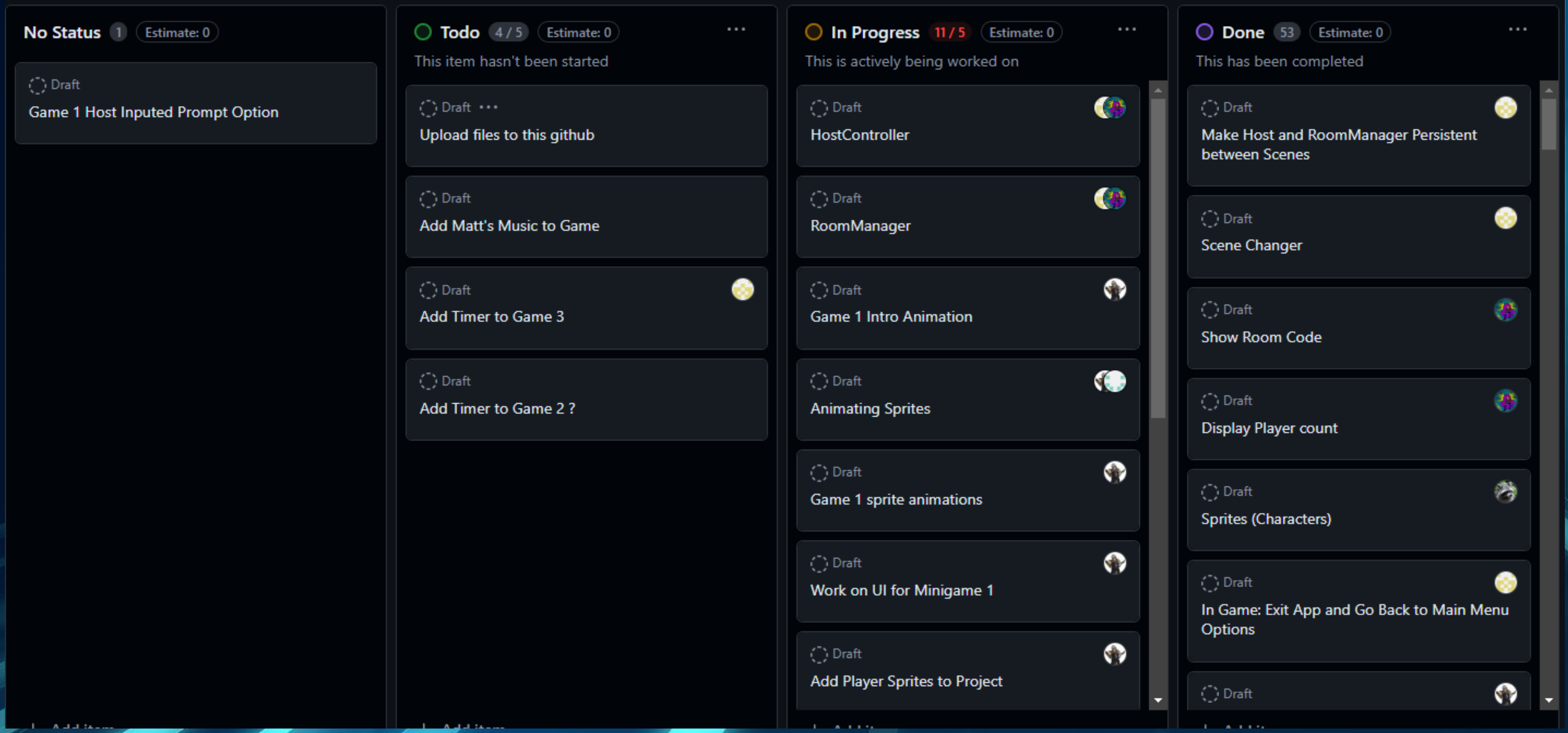
- Counterpoint
- Coup de Critter
- Priceless Persuasion

CONCLUSION

- Gamification is beneficial and a lot of fun!
- Making games is hard.



GITHUB



No Status 1 Estimate: 0

- Draft
  Game 1 Host Inputed Prompt Option

Todo 4 / 5 Estimate: 0

This item hasn't been started

- Draft
  Upload files to this github
- Draft
  Add Matt's Music to Game
- Draft
  Add Timer to Game 3
- Draft
  Add Timer to Game 2 ?

In Progress 11 / 5 Estimate: 0

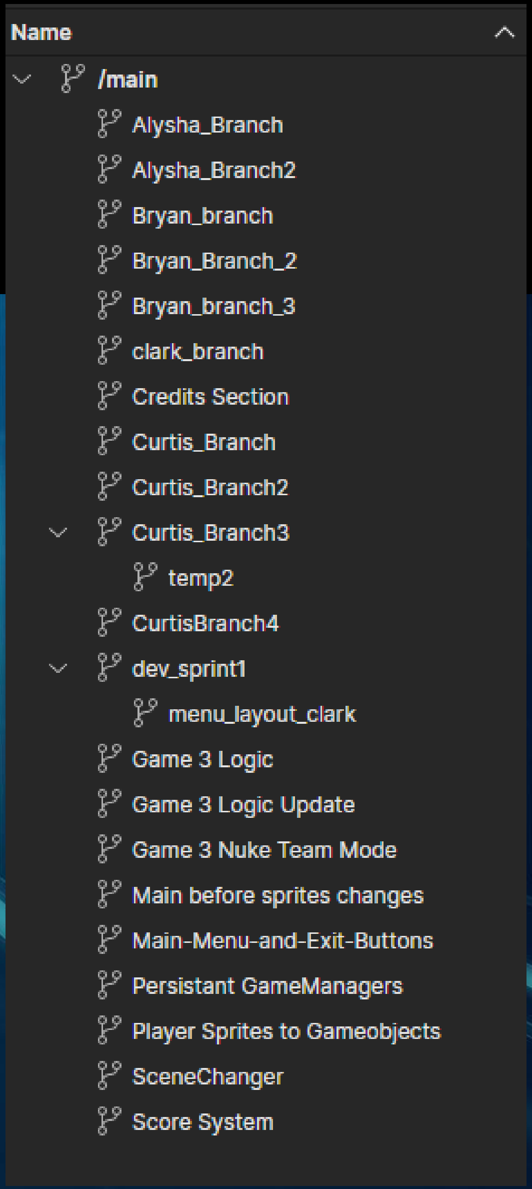
This is actively being worked on

- Draft
  HostController
- Draft
  RoomManager
- Draft
  Game 1 Intro Animation
- Draft
  Animating Sprites
- Draft
  Game 1 sprite animations
- Draft
  Work on UI for Minigame 1
- Draft
  Add Player Sprites to Project

Done 53 Estimate: 0

This has been completed

- Draft
  Make Host and RoomManager Persistent between Scenes
- Draft
  Scene Changer
- Draft
  Show Room Code
- Draft
  Display Player count
- Draft
  Sprites (Characters)
- Draft
  In Game: Exit App and Go Back to Main Menu Options
- Draft



Name

- /main
  - Alysha_Branch
  - Alysha_Branch2
  - Bryan_branch
  - Bryan_Branch_2
  - Bryan_branch_3
  - clark_branch
  - Credits Section
  - Curtis_Branch
  - Curtis_Branch2
  - Curtis_Branch3
    - temp2
  - CurtisBranch4
  - dev_sprint1
    - menu_layout_clark
  - Game 3 Logic
  - Game 3 Logic Update
  - Game 3 Nuke Team Mode
  - Main before sprites changes
  - Main-Menu-and-Exit-Buttons
  - Persistant GameManagers
  - Player Sprites to Gameobjects
  - SceneChanger
  - Score System

https://github.com/CSCD488-Winter2024/project-description-user-stories-and-glossary-ericboxgames

REFERENCES

- Hackbox - server solution
- Unity Assets - Leantween
- ChatGPT (DALL-E) - AI image generation

CRITTIUM

PRODUCTIONS