CalcMatch

Software Engineering - Spring 2019

Peyton Norder, George Young, Kordell Flores, Jurgen Famula, Elisabeth Bristol

General Outline

Original Project Proposal:

- Supports Calculus Matching Activities
- Show cards that have a color, text, images, or both
- Drag and drop cards into a set and get them off screen
- Submit the answer and get a score/feedback
- Multiplayer functionality to allow for real time drag and drop visibility.

After First Meeting

Top Priorities

- Multiplayer - students must be able to see each other moving cards and interact to play the game
- Card Visibility - Cards must be displayed in a way that students can easily read them
- Match making - Cards must be able to be grouped into matched sets (one of each type of card)

Changes from Original Proposal

- No need to implement right answers
- Not practical for a mobile device - moving to a web application

Picking the Platforms

Game Design

- Considerations - raw JavaScript web code, Android Studio/Swift, TableTop Simulator, Unreal, Unity

Networking

- Considerations - Firebase data storage, raw Websocket code, Unity Networking Client, Photon

Web Hosting

- Considerations - CS Linux Server, itch.io, Canvas

Final Platform Selections

Game Design

- Unity - one of the largest game development programs
  - Easy to create UI
  - C# coding standard for game logic
  - Most of the team had experience on the platform
  - Supports Web GL builds which can be hosted on a server

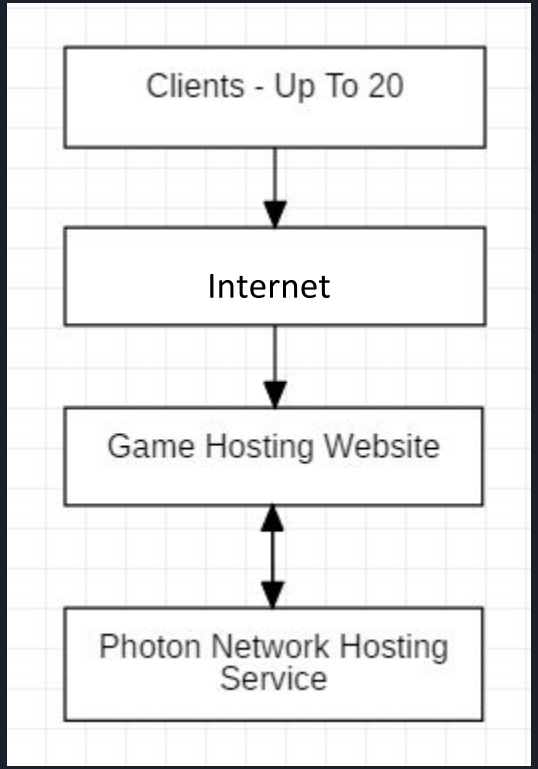
Networking

- Photon - Multiplayer Networking Asset in Unity
  - Not a deprecated standard

Web Hosting

- Itch.io
  - Easy to maintain and can be monitored directly by sponsor

Layered Architecture Model



Networking

- Photon provides a networked backend for multiplayer games.
- Free version we utilized allows for up to 20 clients at once.
- Two main ideas:
  - Ownership - in a multiplayer environment each object has an owner that directly controls it at any given time.
  - State synchronization - updating the positions, ownership and attributes of all networked objects.

Networking

- Photon Views - scripts and components that attach to each object with information that must be sent across the network
- Remote Procedure Call - synchronizes the states of the networked objects across all clients
- Photon - learn it by example, there’s not a lot of documentation

Web Hosting

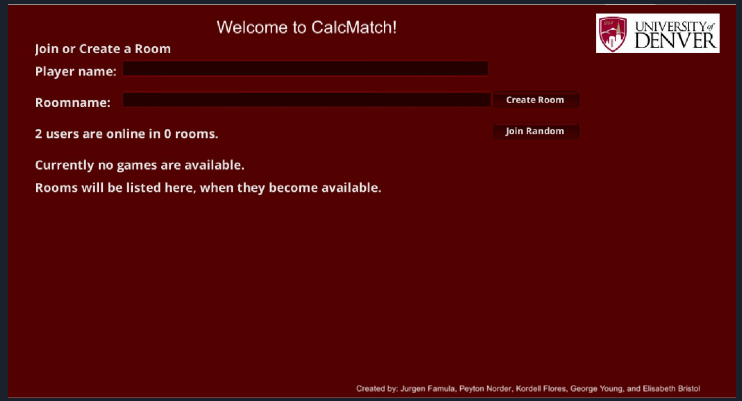
- To host something this complex online can be very difficult to do without a hosting service
- File path directories and permissions on a raw LAMP server are hard to get correct
- Maintainability over the long run
- Itch.io provides resolution controls, detailed display settings, and is easy to use

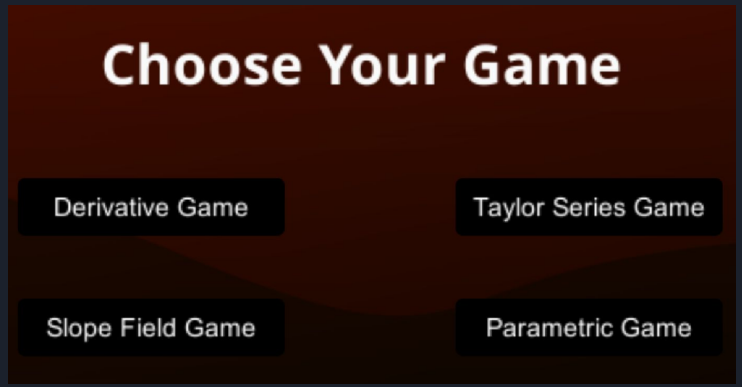
Game Logic

- Ultimate goal - be able to match sets of cards
- Cards are images surrounded by colliders
- Colliders monitor the position of objects to determine if certain behaviors are allowed
- This behavior can be slightly temperamental due to the way colliders interact with an object’s origin in Unity.

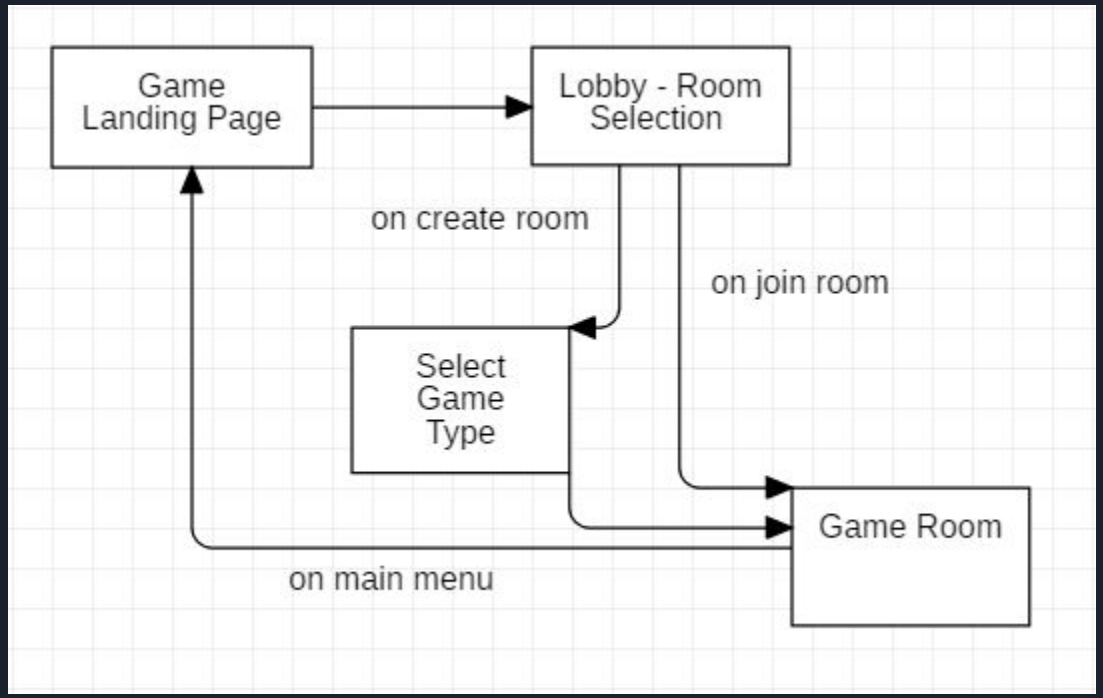
Game Play

- Login Screen - allows users to enter a username, create a room, or join a room from an existing list
- Game Selection Screen - allows the creator of a new room to select which of the four existing games to play





End User Flow Model



In the Game

- Cards are displayed as buttons on the right that can be clicked to take cards on/off the table
- Cards can be matched by dragging and dropping two cards of different colors on top of each other and hitting shift key, and unmatched by right clicking
- A stack of cards can be unfurled by hovering over the stack and hitting space
- Matched stacks can be dragged to the corral on the left to be minimized.

Live Demonstration

Things to note:

- Real time movement - fairly smooth across all player’s screens
- Error is displayed if you try and remove a parent card of a matched set from the table.
- Main Menu button removes a player from a room and sends back to login screen
- Player’s can steal cards from each other
- Instructions available on an individual level from the “?” button in the bottom left

Launch CalcMatch

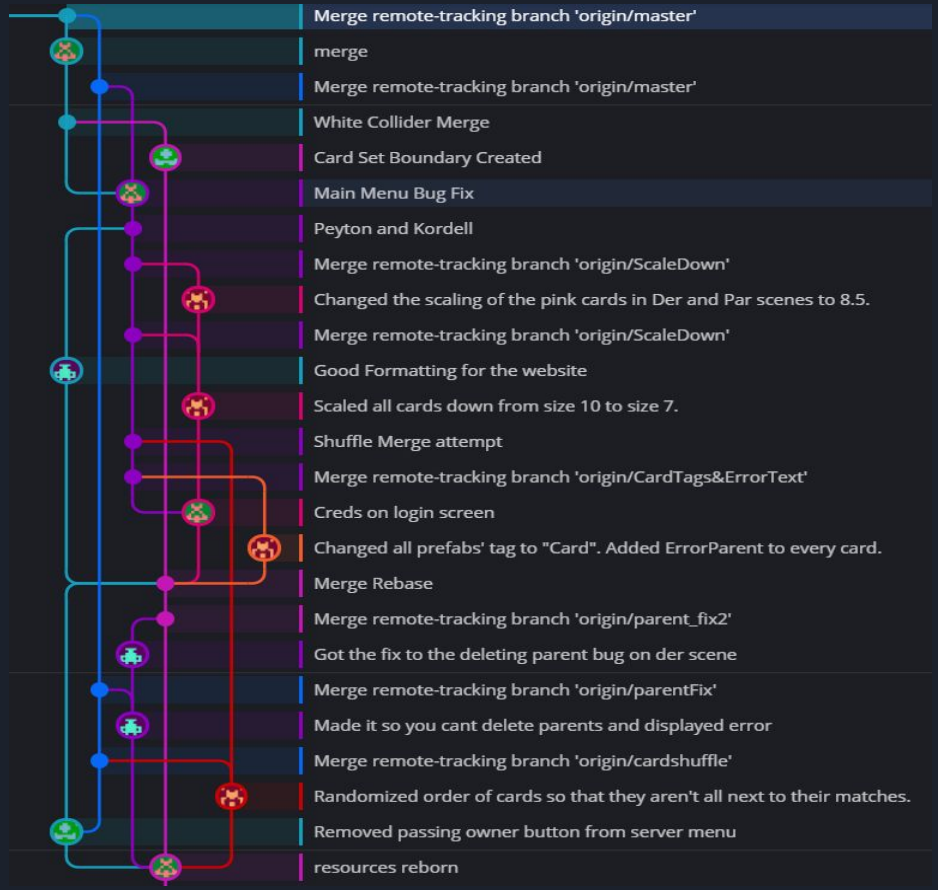
Future Work

- Fixing the temperamental response from hitting the shift key to match cards
- Unfurling cards does not always have the same behavior - make this more consistent
- Occasional Resolution issues for certain monitor sizes
- Implementing a correct answer/scoring system
- On two occasions game has caused complete computer lockup due to a network overflow - very hard to reproduce

Skills and Lessons Learned

- Teamwork - communication, project management, paired programming
- Basic Agile software development practices
- Unity - advanced features, built in assets, C# coding
- Researching and Documentation - picking the right tool for the job
- Networking - how does it even work?

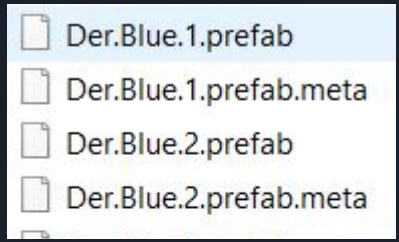
Code Management



- When working with Unity, you cannot forget the git.ignore file
- Communication with who is working on each feature is critical
- Code is highly abstracted by the Unity Editor

Expandability

- Game was designed with expandability in mind
- New “scenes” - game types - are easy to create by duplicating the ones that we have already built
- Inheritance hierarchy for cards prefabs
- Consistent naming convention registered by scripts for function adaptability.



Maintainability

- Game is hosted on itchio.com under a gmail account created for this project
- Documentation provided in thoroughly commented scripts and detailed READ_ME explaining how to add new games and upload new builds
- Git Repo - contains all source code, READ_ME, SCRUM documentation, and this presentation

Source Code and Documentation

- All source code and documentation for this project can be found at the following git repo:

https://git.cs.du.edu/softwareengineering/calcmatch.git

**This link was sent to your emails already

Questions, Conniptions, or Conundrums?