Custom Card Game

Team 19 - Brenner, Henry, Adrian, Aditya, and Leo

Our 1v1 Custom Card Game

- So our game is a multiplayer card game, similar to Magic: the Gathering or HearthStone. A player will be able to join a quick match as a guest, or create and login into an account to track their stats across games.
- Players choose a deck of cards before their match, that gets mixed in with their opponent's deck at the start of each game. Both players play from this deck, which should make for some unique builds and introduce some variety in each game.
- Each card has its own effects/rules that interact with each other differently, and the goal is to play the right cards to take down your opponent's health before they take down yours.

Card System

- We have categorized our cards under red, green, black, and white:
  - **Red**: aggressive and fast
  - **Green**: drawing and slow but big creatures
  - **Black**: discard and value
  - White: healing and control cards
- Each deck of cards will have 20 cards each, and when we shuffle both decks together it will be 40 cards in total.
- Each person will start with ten cards in their hand, but this may change based on playtesting in later stages.
- We have some rules and the general flow of the game down, but nothing super concrete yet, because we've been more focused on building the game itself and learning Unity so far.

What was the Project Plan’s goal for this?

Our goal for this milestone as a team was to:

- Learn the basics of Unity as a game development engine
- Make a landing page for our game (essentially the main menu)
- Be able to host that game on a web browser
- Finalize the rules which will dictate how the game is to be played by the user

Building a User Interface in Unity

- So Unity has a built in system for UI, allowing us to make buttons, input fields, and different menus rather easily.

- Our main menu is a series of active and inactive images, which are full of buttons and input fields, that can also change to reflect if the user is logged in or not.

- The main menu includes a quick match button that launches a “match”, a login page (that doesn’t currently actually take input, just activates a condition), a tutorial page that will play a video of our tutorial, a settings page for audio and visual settings, and a quit button. If the user is “logged in”, they will be able to see their account settings, and then their account’s stats through their account page.

Unity Exploration

- We chose to use Unity for our game development project, as it is one of the most widely used game development engines that supports both 2D and 3D game development, and is compatible with most of the systems.

- Since none of the team members had any experience with Unity beforehand, we set aside the first two weeks of our project plan for just getting our hands dirty with different tutorials, in order to get familiar to using the Unity Editor.

- We also found several online tutorials using it, which provided a basic direction we could use to start developing our game and to implement the features we wanted.

Using WebGL to run our game in a web browser:

WebGL: graphical API that allows us to render 2D/3D game within web browser.

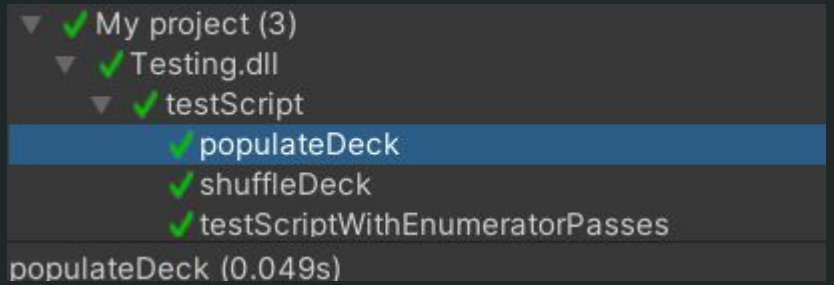
Why WebGL?

- direct integration with Unity
- runs without prior installation
- cross-platform accessibility
- ease of distribution

Live Demo

Test Report

The test report shows the status of the implementation.



Teamwork Strategy

- We meet as a team twice a week, once in class and remotely over Discord.
- We are following the agile development method for our project, as it provides us with enough flexibility to deliver our deliverables early, and have room to change aspects of the project if they lead to improvements.
- Every team member chooses a certain task they will work on for the week, which is then reviewed and pushed on the team GitHub repository.
- We plan to increase the frequency of our meetings so that our project proceeds more smoothly, and each team member is confident in the task they have to complete. This will also serve as a midweek checkpoint to see if any team member is encountering any problems in their respective tasks.

Past Problems

- The first problem we encountered was how to connect our Unity project to GitHub so that all the team members can push their own changes and progress remotely.

- Testing was difficult, since the main portion of the demo and our tutorials didn't cover anything that could be easily Unit Tested (The UI design was almost exclusively done using the unity editor, without much scripting, as an example). Once we have more actual systems built like a login system or password reset system, we will have more things to build tests for.

Potential problems in the future

- The amount of cards to implement in order to keep the game interesting could require more work than expected (card art (even with free assets), and each card’s rules etc.)

- Publishing our unity scenes to WebGL seems to be causing some scaling issues, but it's not super critical to how the game plays (we can still see and use all the features, it’s just a bit small), so we are aiming to fix it after this milestone.

References

Magicthegathering-logo. 5 May 2011. *Wikipedia*, https://en.wikipedia.org/wiki/File:Magicthegathering-logo.svg. Accessed 1 Nov. 2023.

Mrwynd. *HearthStone Beta*, 19 Oct. 2013, *Flickr*, https://flic.kr/p/gNQNMZ Accessed 1 Nov. 2023. (Slightly Cropped)