Fall 2023

Final Report

Beast Brawl: Furry Fury

Team Ryan: Ryan Schneider, Ryan Rizzo, Paolo Rances, Jonathon Eichenberg

1. Introduction

Opening Paragraph

− What is the project?

We created a pc video game in the style of a card based auto-battler such as “Hearthstone Battlegrounds.” The game's name is *Beast Brawl: Furry Fury.* This game implements many features such as a large card pool, 3 difficulty-level AI to play against, an intuitive graphical interface, a help page, and more!

− What was the motivation for the project?

We wanted to make something this semester that was going to be challenging to implement, but also fun and creative. Because of this, we knew that we had to make some sort of game. This led us to discover we all had both interest in this genre of game, and understanding of how we could develop it.

− What is novel about it?

What makes our game novel is that unlike most of these games, you are able to play it singleplayer. In addition, our game uses a unique theme that is both cute and family friendly.

Challenges

− Briefly describe the main challenges for the team.

The main challenge for our team was early in the development of game. We knew what we wanted to build, however we were not sure what tools we should use. We originally decided to use Unity engine, however we quickly learned that we didn't know how to collaborate in Unity.

− How did you address them?

After realizing that Unity was not going to be an, we switched to Java. However this meant that we needed to start all over again. Because of this, our team had to work even harder to try and catch back up to the phase of our classmates.

− Was the technology known or new to the team?

As previously mentioned, we started off with Unity. The team had some knowledge, but generally little. However, all of us have had extensive experience with Java, as well as adequate experience with JavaFX.

Highlights

− Highlight what was accomplished.

Some main features of the game that are working are first the shop phase where you can buy cards by clicking on them and we have a drag and drop feature of the cards so you can organize them more fluidly however the user wishes. In the shop phase there is also the reroll button which refreshes the shop to give new cards and the upgrade button which will make the cards better whenever the shop refreshes. Then we have an end turn button which starts the attack phase where we have attack animations of the cards. Then there is obviously the logic of health and money distribution after every round. Then we have win or lose screens depending on if the user lost or won against the AI.

There was also a help page to help users who have not played before get familiar with the game.

Significant Project Decisions

− Summarize the top four decisions, events, or changes during the project.

1. We decided to switch from Unity to Java for the code base.
2. We decided not to have a deck builder in the game.
3. We decided to change the button layout by user request.
4. We decided to change how we showed who was attacking and defending to highlights.

− With each of the top four, include the date, motivation, and consequences.

5. *Date:* 10/12/23
   Motivation: A struggle to collaborate and efficiently develop in Unity.
   Consequences: Having to start all over again in Java.

6. *Date:* 10/15/23
   Motivation: This mechanism broke the rules of an “auto-battler”
   Consequences: Some code was written which was later discarded.

7. *Date:* 11/17/23
   Motivation: User feedback called our current layout “unintuitive.”
   Consequences: The GUI had to be reimplemented

8. *Date:* 11/24/23
   Motivation: User feedback said it was hard to tell who was attacking
   Consequences: The GUI had to be reimplemented

2. Requirements

User Involvement

− Who was the primary customer/user?

Players of autobattler card games. In specific friends of ours who play those games.

− Any other external stakeholders?

CSC436 students and teachers/tas.

− Summarize interactions with customers during the project.

Ryan R met with them regularly, in the second half there wasn't much to share at the start, and let them test the game and give feedback on what they did and didn't like.

User Needs

− What were the stakeholders' wants?

Because the stakeholders were friends of ours who were interested in auto-battler style games, they wanted us to make a game that both reminded them of the genre, as well as felt fresh and challenging.

− What was their motivation for the wants?

They love playing these types of games, however it is still a relatively new genre. Because of this, there is a lack of playable games on the market currently.

− Include the top 5 user stories.

1. As a student, I want a game I can play on my laptop between classes, so that my down time passes quicker.

2. As a competitor, I want a game with challenging opponents, so that I can challenge myself.

3. As a beginner gamer, I want a game with helpful instructions, so that I can learn the game easily.

4. As a strategist, I want to be able to build my own decks, so that the game does not get stale after one or two matches.

5. As a busy individual, I want a game that I can pause, so that I can decide when and when not to play.

Problem Definition

− What benefit did you choose to provide with your project?

A fun game to pass time with.

− How does the benefit contribute to the customer's desired end-to-end experience?

They had fun playing, which was the goal.

− Include the top 3 use cases.

1. As a beginner gamer, I want a game with helpful instructions, so that I can learn the game easily.
2. As a gamer, I don't like overly repetitive content, so I want the AI to act differently across different matches.
3. As a gamer, I want the game to be free of freezes, so that I don't have to task manager the game when an error occurs**.**

− For each use case, provide only the primary actor, the user goal, and the basic flow.

User goals built are in use cases.

1. Help page test flow
   - {User clicks on instructions button}
   - Load help page
   - User reads instructions
   - User clicks X to leave
   - Return to step 2

3. Multi AI test flow
   - Start game
   - Load menu page
   - {user clicks one of three difficulties}
   - {User clicks on Start game button}
   - Game plays out with correct AI opponent

3. Error test flow
   - Game encounters an error
   - Game displays Error message
   - User clicks ok button
   - Game closes

3. Significant Project Design Decisions

System Overview

− In a paragraph, the system and the main challenges.

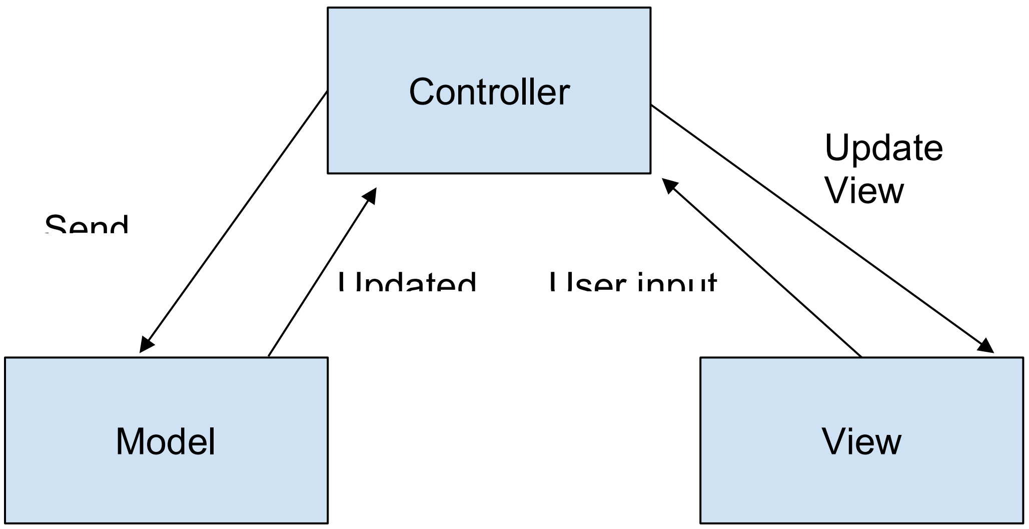
We do have a MVC architecture of code where all of the data of the game is stored in the model and then the model communicates to the view through the controller. One of the main challenges we encountered was trying to get the battle animations working in the attack phase. This issue was mainly due to threading issues between the model and GUI which was eventually resolved. Another main challenge was getting some of the menus in the GUI working properly which was mainly due to javafx being difficult.

− What were the significant decisions that shaped the system? Include the motivation for the decision and the resulting consequences/constraints.

Firstly, one significant design change in the beginning was the switch from unity to Java. The reason for this decision was that the team felt more comfortable coding and had more experience in Java rather than unity. The consequences/constraints of that was we were very limited in how to make the gui and had to deal with Javafx. Since we were coding in Java we decided to use MVC architecture. There were no consequences/constraints from this at this time.

Context Diagram

− Draw a context diagram to show how the system interacts with external services, databases, and Software.



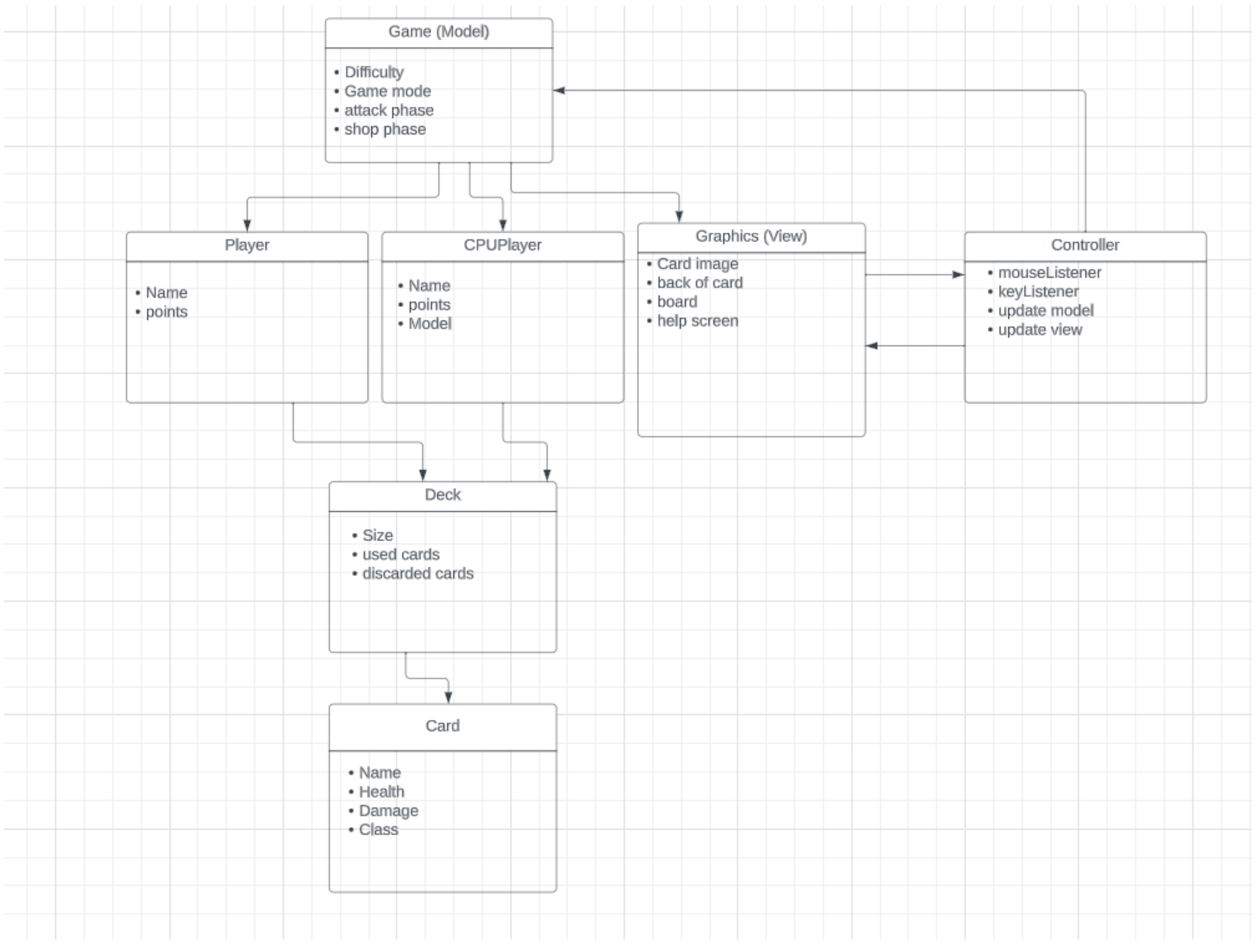
− Clearly mark the system boundaries.

− Highlight message and data transfers across the system boundaries.

Users send data to the view which Passes it to the controller then model to calculate data changes. The model/controller then calls the views UpdateGUI function to change the view the users see based on the new values.

Architectural View

− In the view, show the interactions between the architectural elements.



− Provide a caption to introduce the view.

As in a typical MVC architecture the user would communicate with the view and the view would communicate with the controller which would then communicate with the model. This image also includes the classes that the cards would inherit from and the deck class the player would have that contains all of their card data.

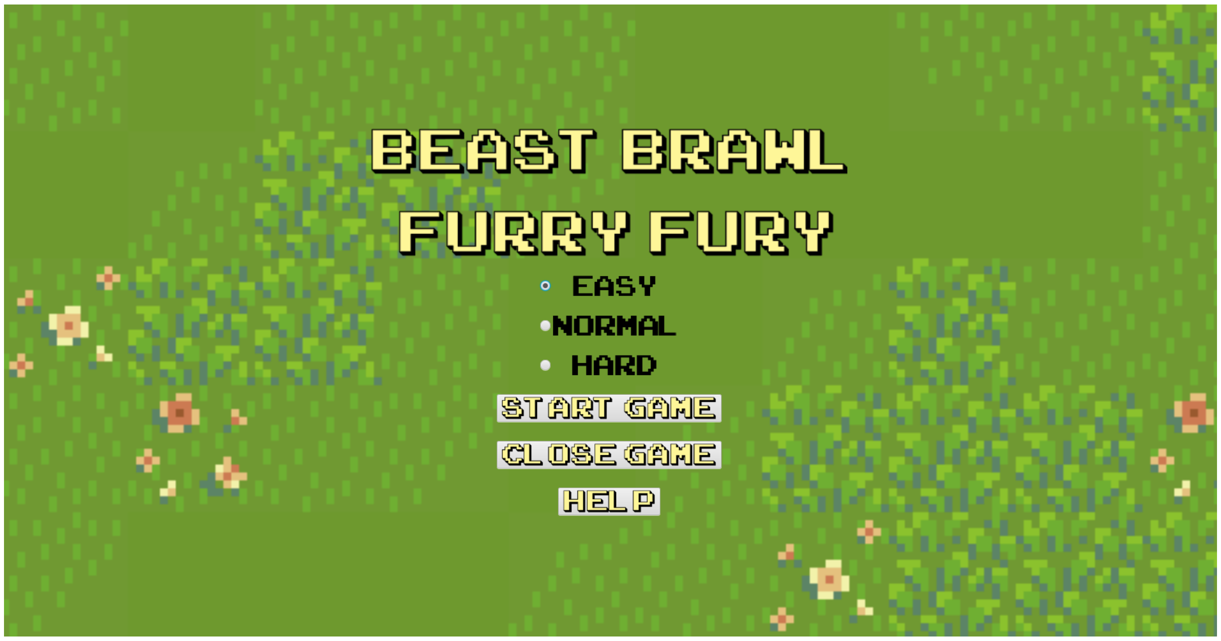
− Provide an element catalog with the roles and responsibilities of the elements in the view.

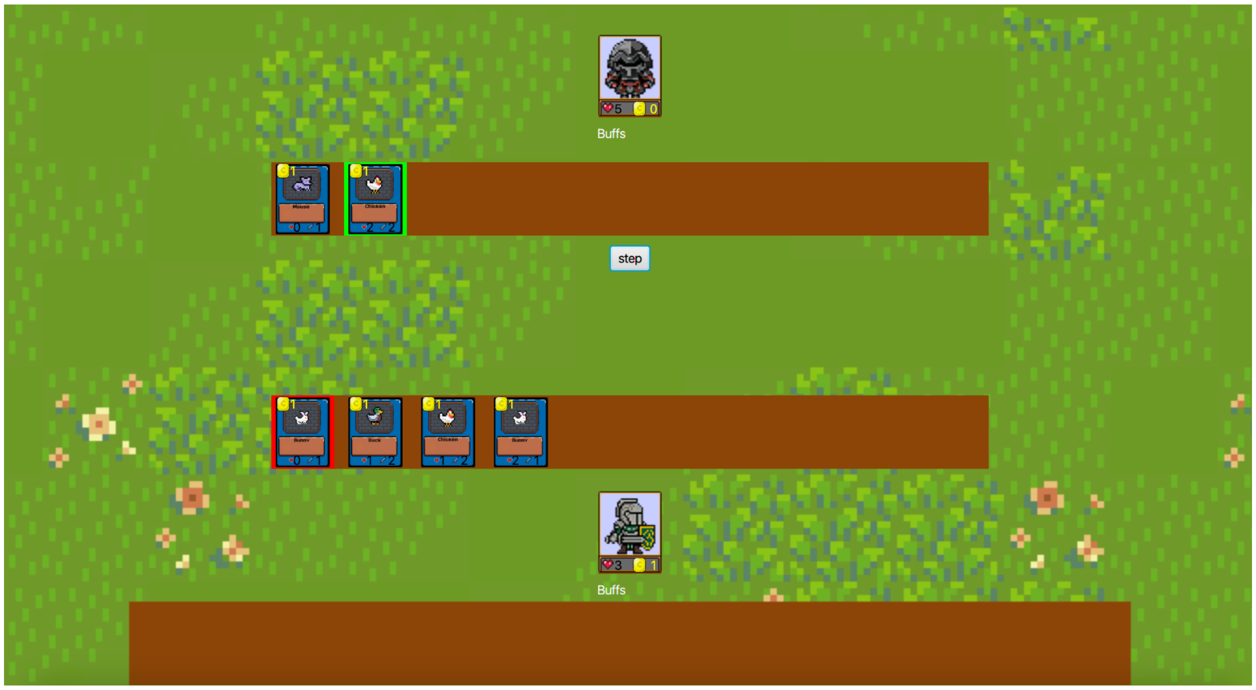
Main Menu- Paolo

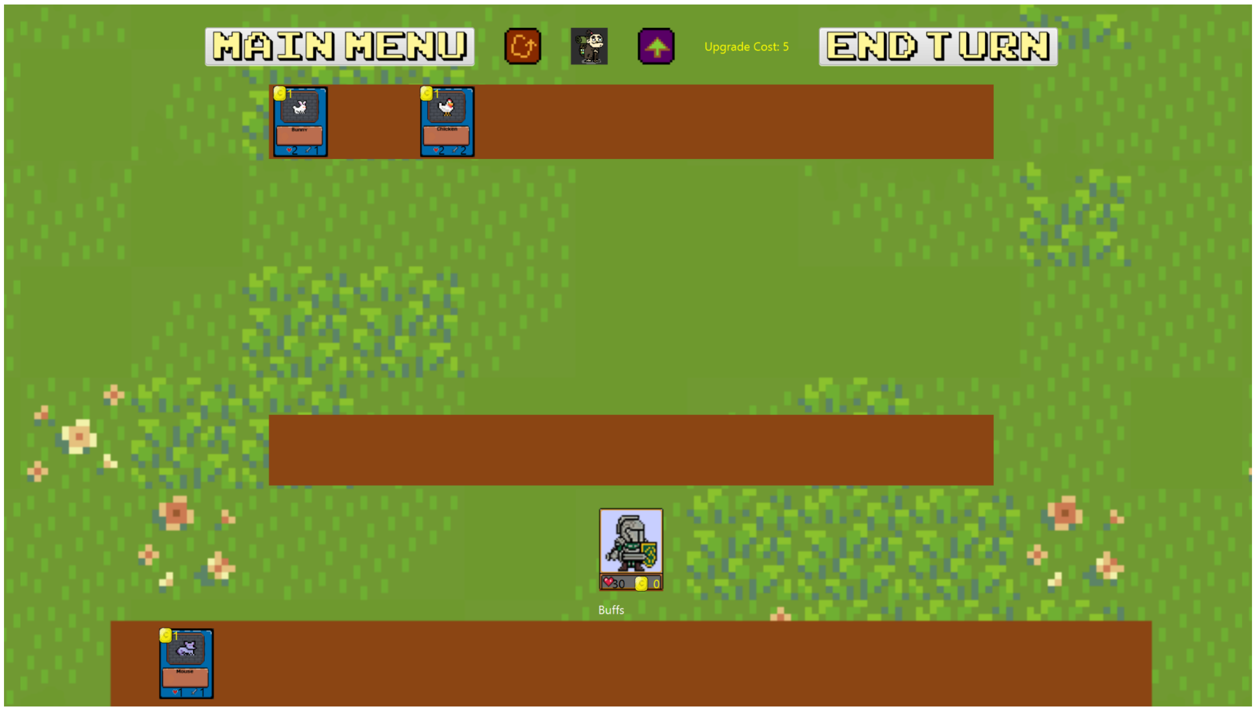
Drag and Drop	Paolo
Win Screen	Ryan R
Lose Screen	Ryan R
Instruction page	Paolo
End Turn button	Ryan S
Step button	Jonathan Eichenberg
Auto step	Ryan S
AI difficulty selector	Ryan R
CPUPlayer	Ryan R
Close game button	Ryan R
Hand	Starter Code
Battlefield	Starter code
Reroll button	Starter code
Card Images	Jonathan Eichenberg
Creatures	Ryan R
Shop	Ryan R

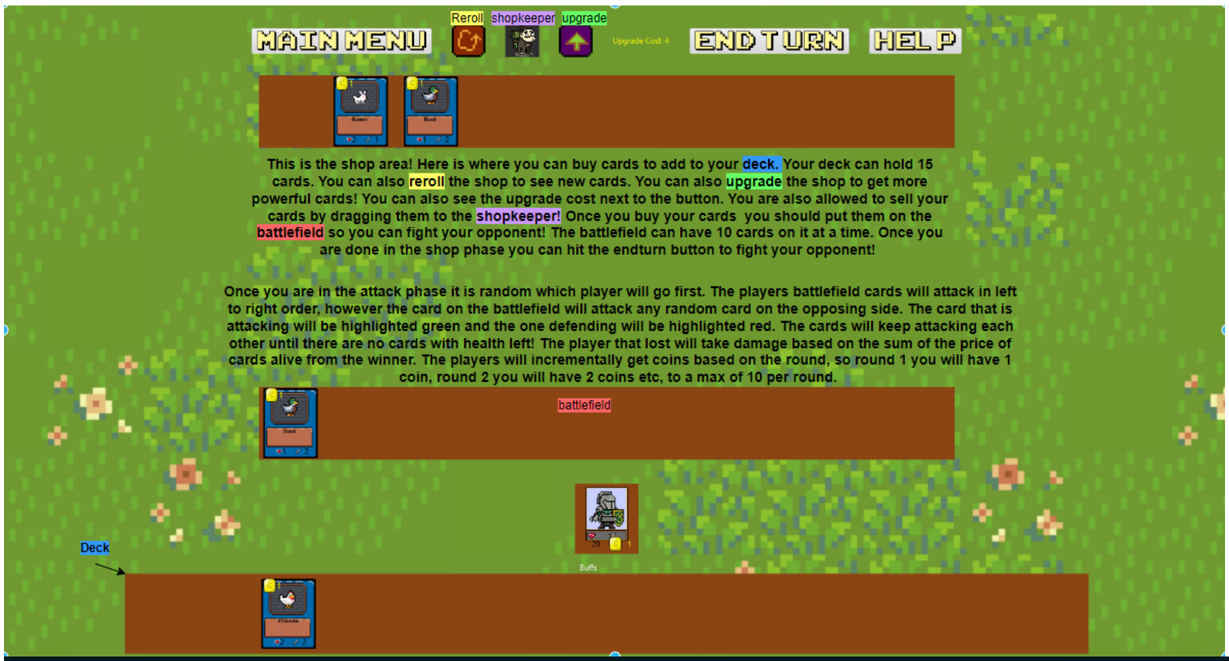
4. Status

– What’s working now? Include screenshots.



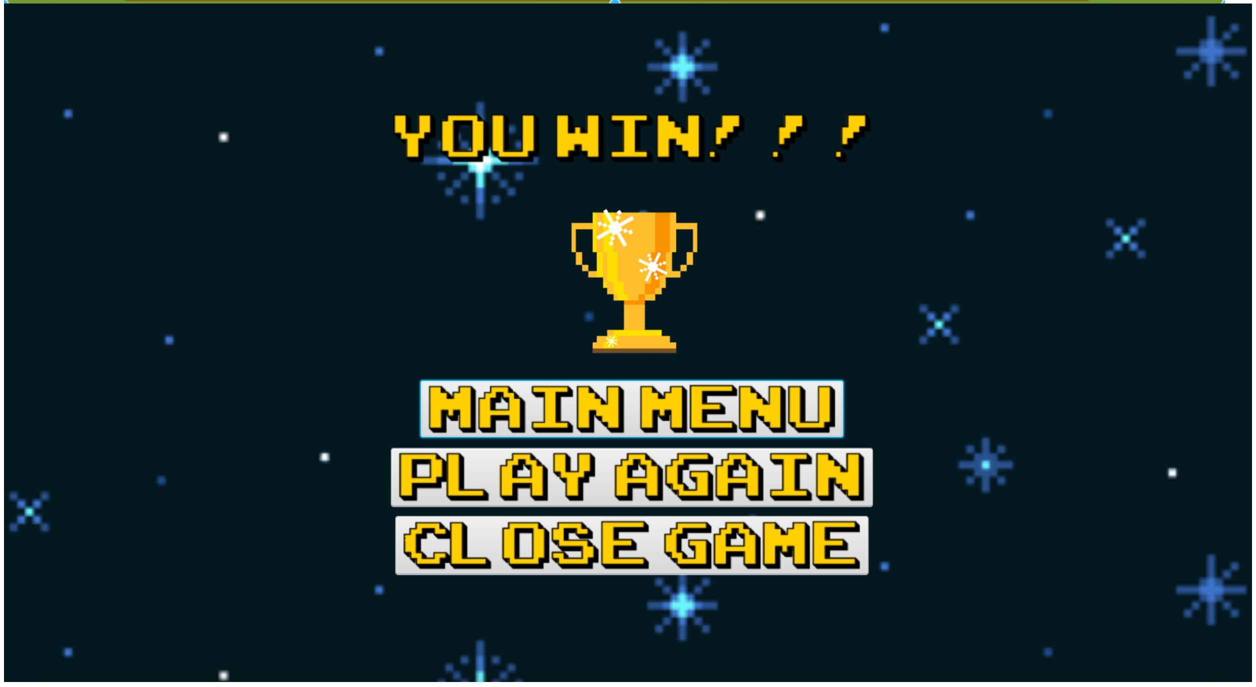


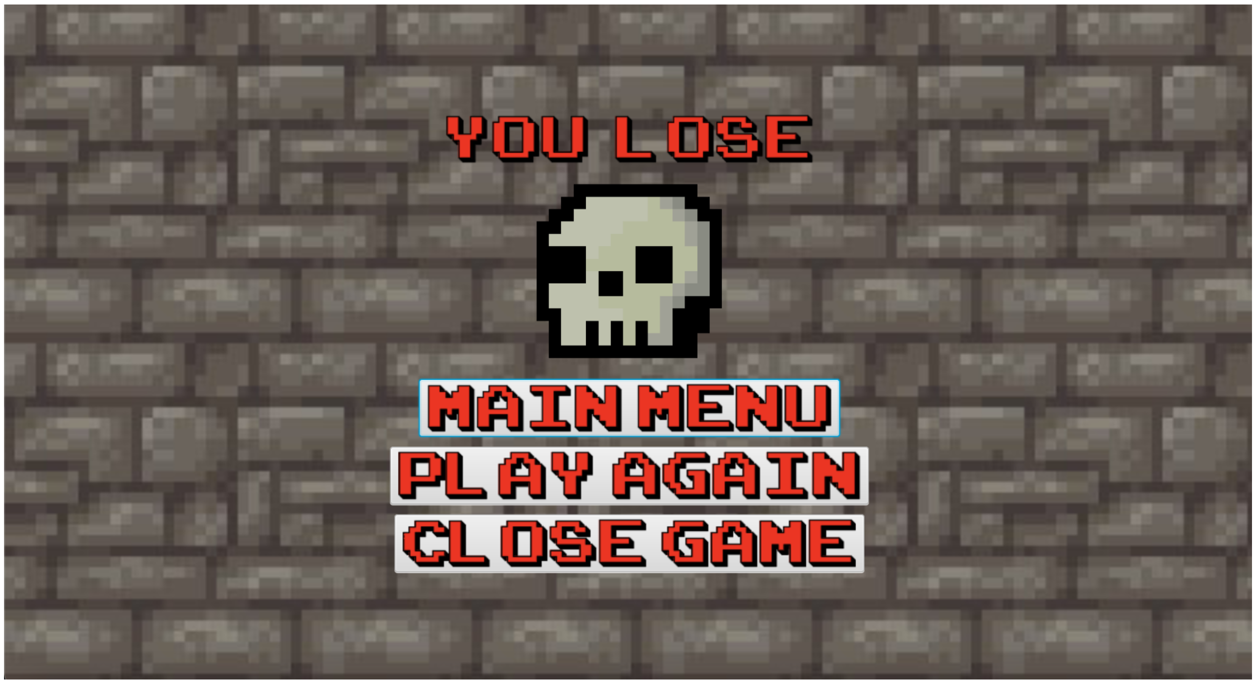

MAIN MENU
Upgrade Cost: 5
END TURN
Bunny
Chicken
30
0
Buffs
Mouse



This is the shop area! Here is where you can buy cards to add to your deck. Your deck can hold 15 cards. You can also reroll the shop to see new cards. You can also upgrade the shop to get more powerful cards! You can also see the upgrade cost next to the button. You are also allowed to sell your cards by dragging them to the shopkeeper! Once you buy your cards you should put them on the battlefield so you can fight your opponent! The battlefield can have 10 cards on it at a time. Once you are done in the shop phase you can hit the endturn button to fight your opponent!

Once you are in the attack phase it is random which player will go first. The players battlefield cards will attack in left to right order, however the card on the battlefield will attack any random card on the opposing side. The card that is attacking will be highlighted green and the one defending will be highlighted red. The cards will keep attacking each other until there are no cards with health left! The player that lost will take damage based on the sum of the price of cards alive from the winner. The players will incrementally get coins based on the round, so round 1 you will have 1 coin, round 2 you will have 2 coins etc, to a max of 10 per round.





− Use the architectural view of the system to explain what's working.

We do have a MVC architecture of code where all of the data of the game is stored in the model and then the model communicates to the view through the controller. Some main features of the game that are working are first the shop phase where you can buy cards by clicking on them and we have a drag and drop feature of the cards so you can organize them more fluidly however the user wishes. In the shop phase there is also the reroll button which refreshes the shop to give new cards and the upgrade button which will make the cards better whenever the shop refreshes. Then we have an end turn button which starts the attack phase where we have attack animations of the cards. Then there is obviously the logic of health and money distribution after every round. Then we have win or lose screens depending on if the user lost or won against the AI. There was also a help page to help users who have not played before get familiar with the game.

− What tests have you run?

We mostly did user testing and had the players try to find bugs in the game or recommend us quality of life changes.

− How many lines of code has the team written, collectively? Do not count third-party code included with the system.

4269ish (a good amount of which is similar code for similar components)

5. Teamwork

− What process improvements did you make during the project?

We got better at coordinating git repo stuff, such as branches and merging in ways that don't cause others problems. We also got better at figuring out how often to hold meetings and at what time.

− Summarize your development process, including meetings or other communications.

We had consistent meetings at 5:00pm on Friday to do a scrum-like meeting.

We have a daily updates group chat where we message each other what we did for the day

Since it was hard to meet up with all of our users at the same time Ryan R met with them and then we used their feedback at the next meeting.

− What were the roles of the team members?

- Ryan R: Product Owner
- Johnny: Scrum Master
- Ryan S: Developer
- Paolo: Developer

− What did each team member contribute?

- Ryan R: Created each AI, both end screens, shop class, the Creature class and its 25 child classes/creatures rules, and asset creation.
- Johnny: A Majority of the model, end of battle and shop phase. Made card images. Made a step button.
- Ryan S: Some shop and battle phase stuff, both View and Model, turn button.

- Paolo: DragNDrop, created and added the main menu and help screens, and made the game reset and go to an end screen at the end of a game.
- All: Various Bug fixes

− Give rough estimates of the percentage contribution to the project by each team member.

- Ryan R: 27
- Johnny: 27
- Ryan S: 19
- Paolo: 27

6. Project Management

− Highlight the top 5 items (decisions, events, changes) from your change log.

10/12/23 We decided to switch from Unity to Java for the code base.
10/15/23 We decided not to have a deck builder in the game.
10/20/23 We decided to add a proper end screen.
11/17/23 We decided to change the button layout by user request.
11/24/23 We decided to change how we showed who was attacking and defending to highlights.

7. Reflection and Continuous Improvement

− What went well?

The Meetings were consistent and productive.

− What didn't go well?

Git and java both caused us numerous problems. Partially because we were using different versions of java.

− What would you do differently?

All pick a version of java and javaFX in the first week. Make sure we all have working installations in that first week.

− What advice do you have for future teams?

Don’t use java and don’t make games. Other groups made websites and that seems more reasonable for this type of project with the same timeframe.