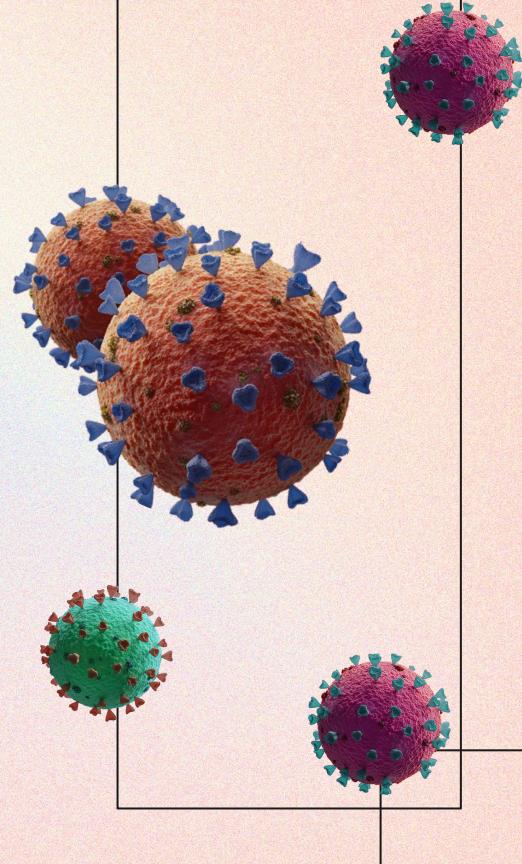


Pandemic

Group: git-help

Kylie Alfaro, Tony Nguyen, Aiden Tabrah, Izzy Tilles





Project Overview

Game: Pandemic Board Game



Overview, Cont.

Limitations

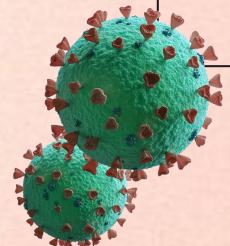
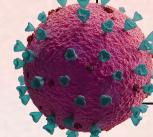
- Resizing
- No event cards
- Less cities

Features

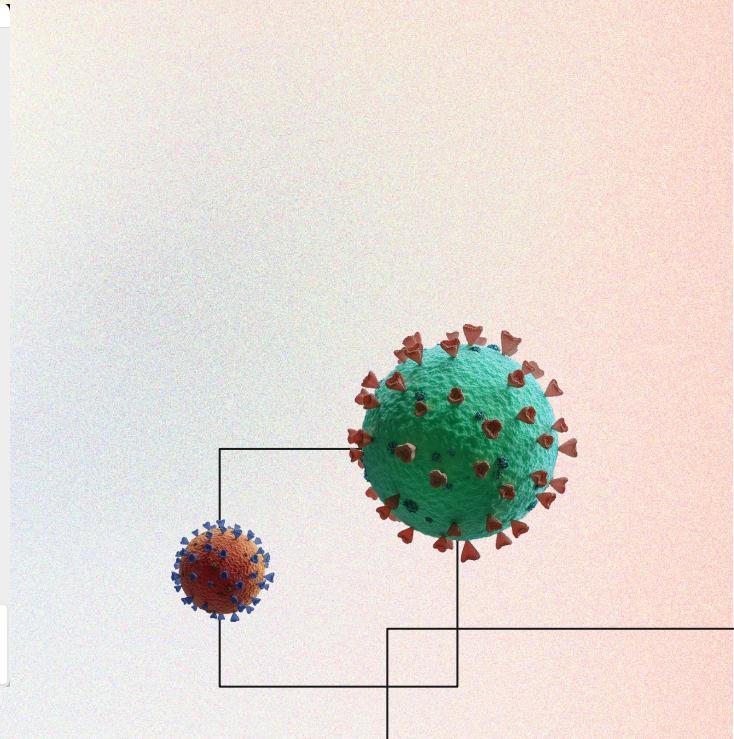
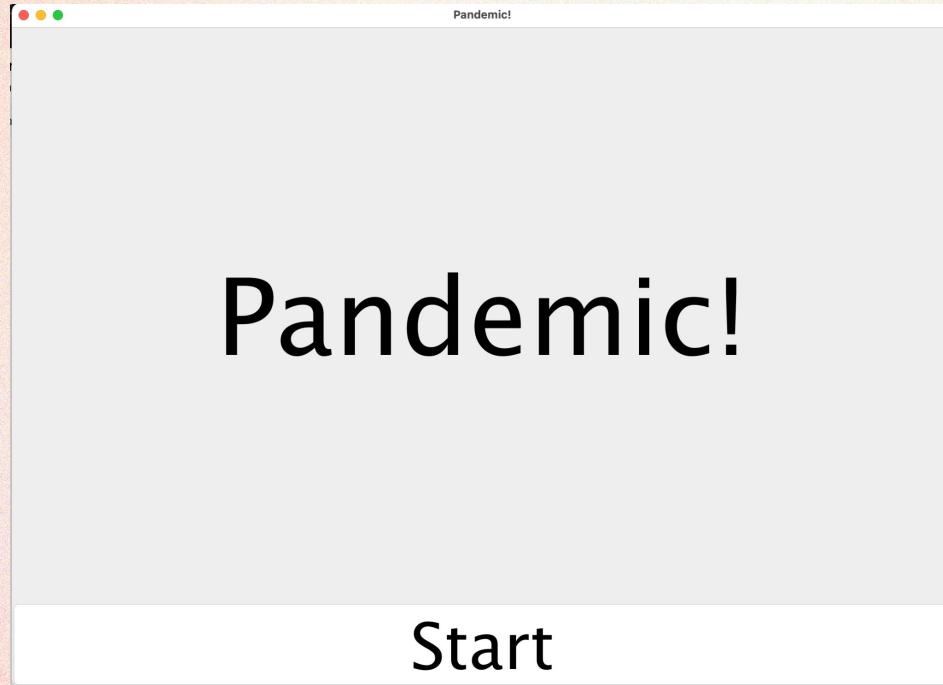
- City cards
- Epidemic cards
- Player hand
- Turn dashboard
- Gameboard
- Cities

Assumptions

- Exactly 4 players
- Have basic understanding of the rules of the game before playing



Start Screen

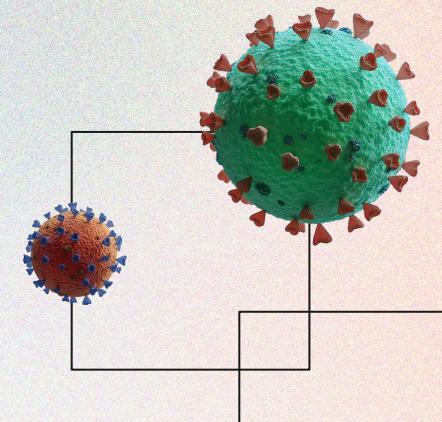


Difficulty & Insert Player Names

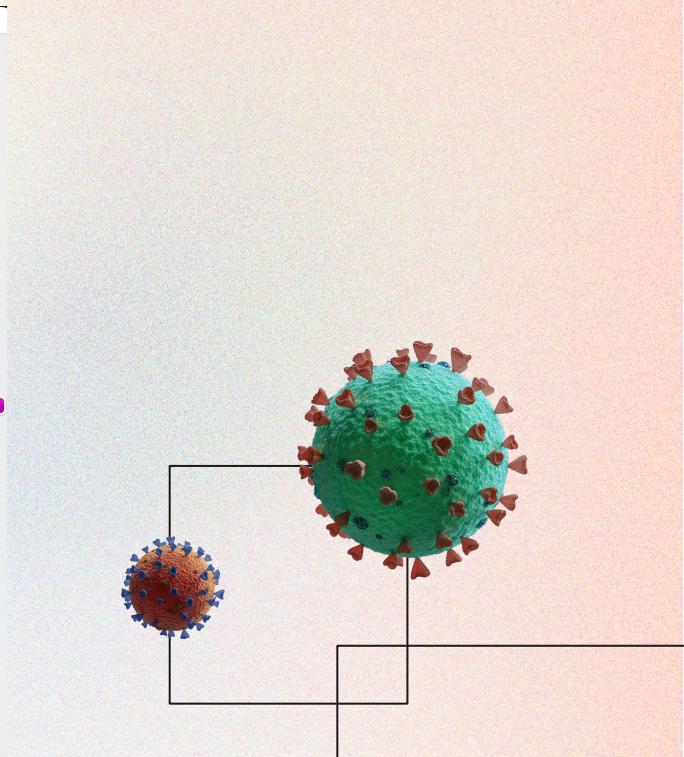
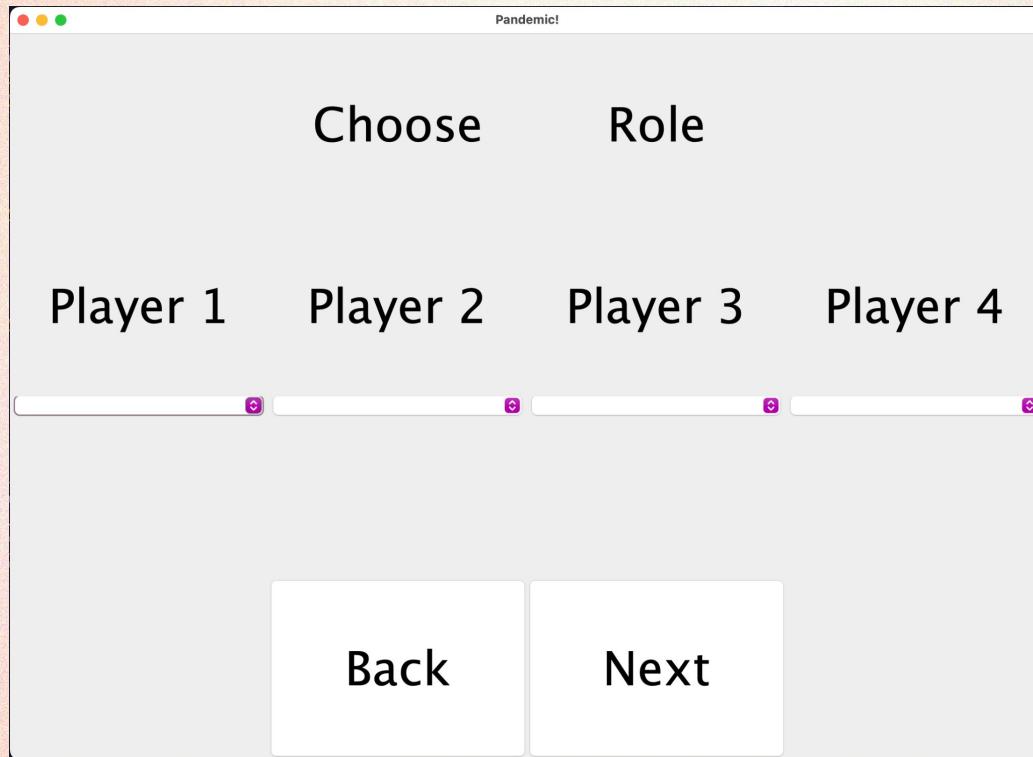
Pandemic!

Difficulty	Players
• Easy	
• Medium	
• Hard	
• COVID-19	

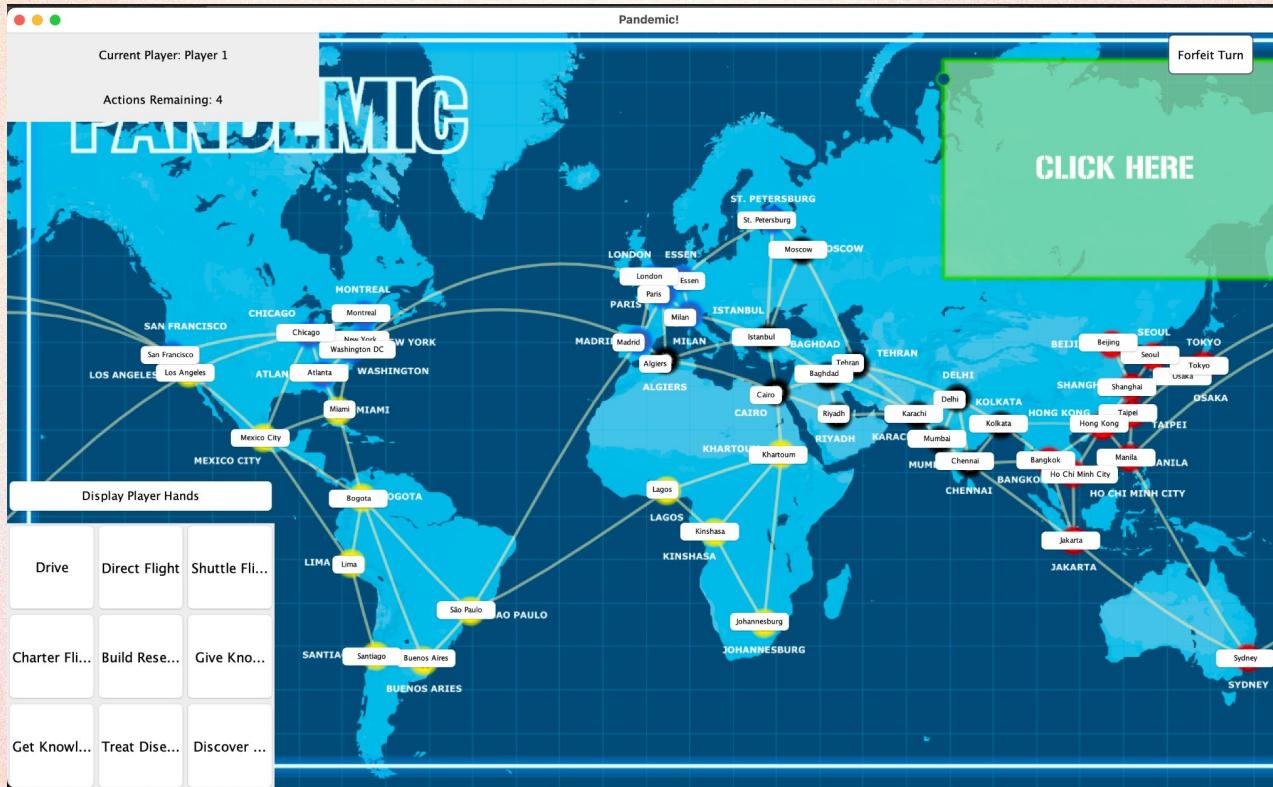
Back Next



Choose Role



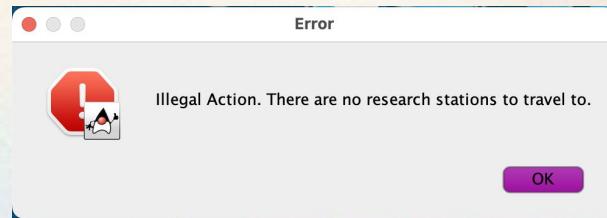
Main Game Board



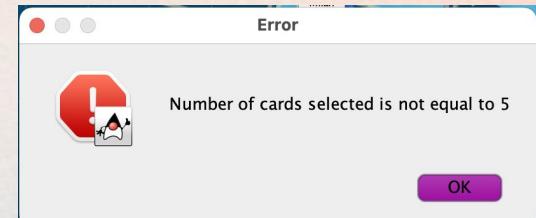
Player Action Screens



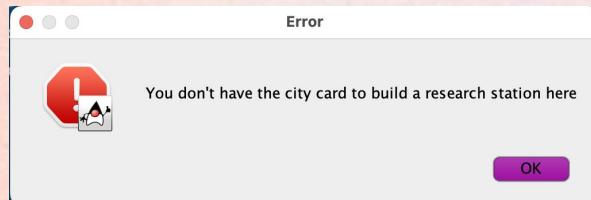
Drive



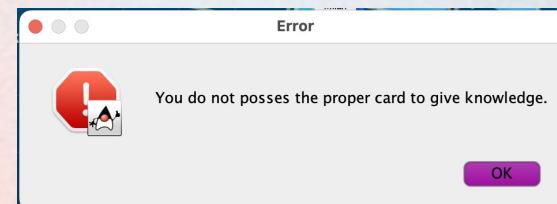
Invalid Shuttle Flight



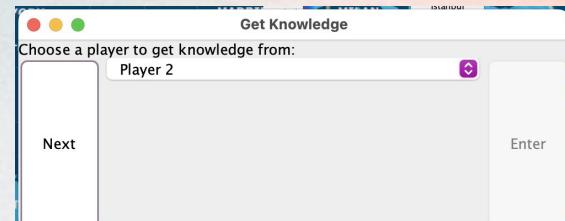
Invalid Cure Disease



Invalid Build Research Station

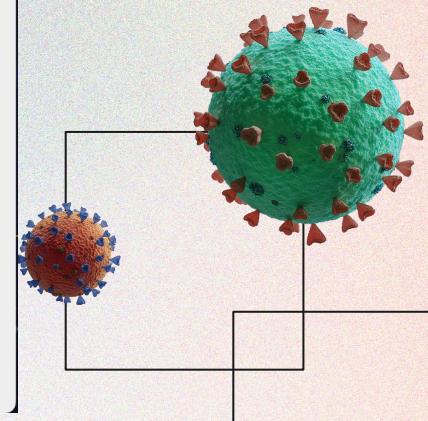
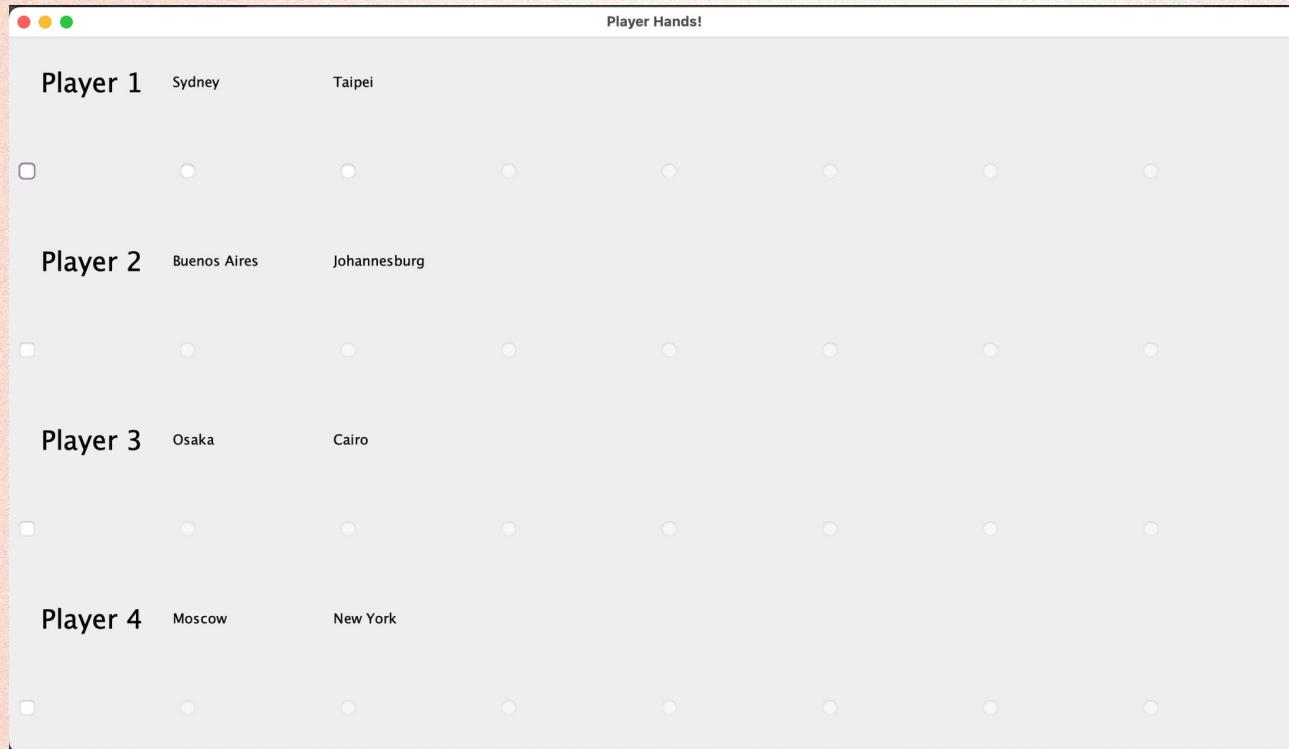


Invalid Give Knowledge

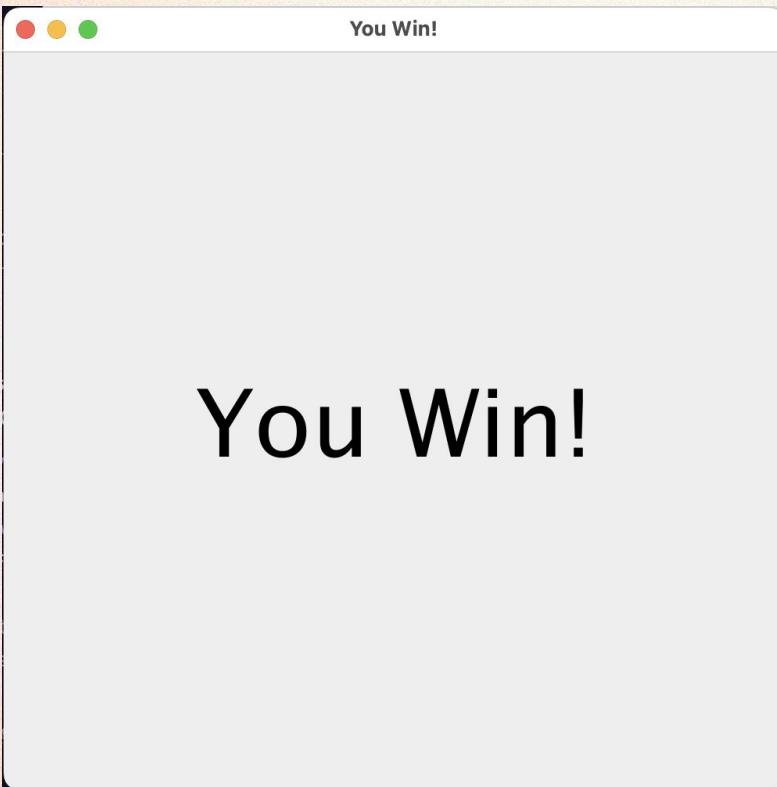


Get Knowledge

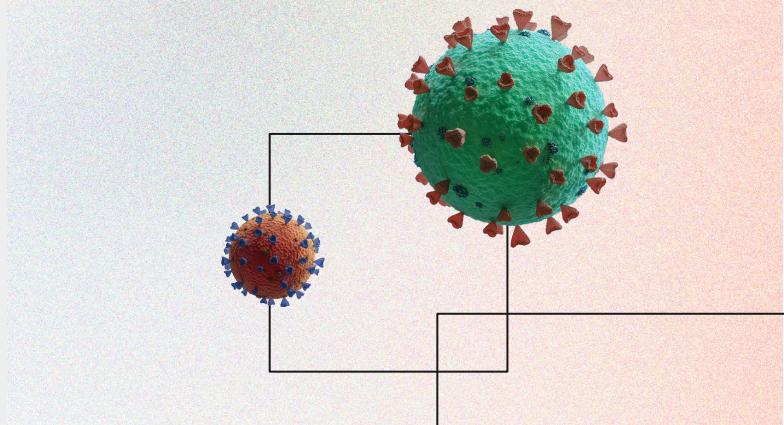
Player Hands Display



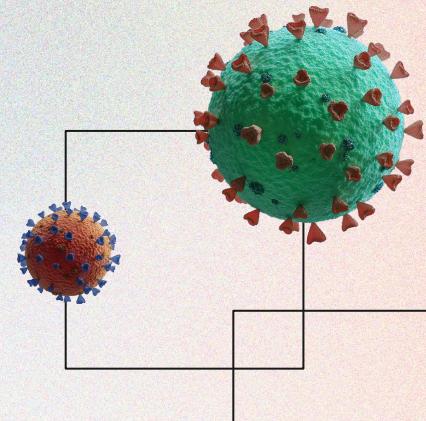
Winning



You Win!



Losing



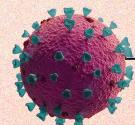
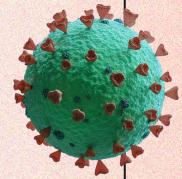
Functional/Non-Functional Requirements

Functional

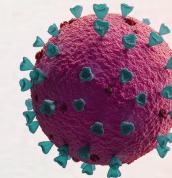
- Track & display the players in a city
- Allow users to set names and roles
- Store status of a city (outbreaks, infections, research station)

Non-Functional

- Show players each others' hands
- Display actions
- Show status of city when selected



Solution Approach



Logic

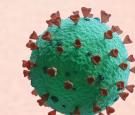
- Moving players
- Keeping track of moves and turns
- Recognizing winning and losing conditions
- Updating player hands

User Interface

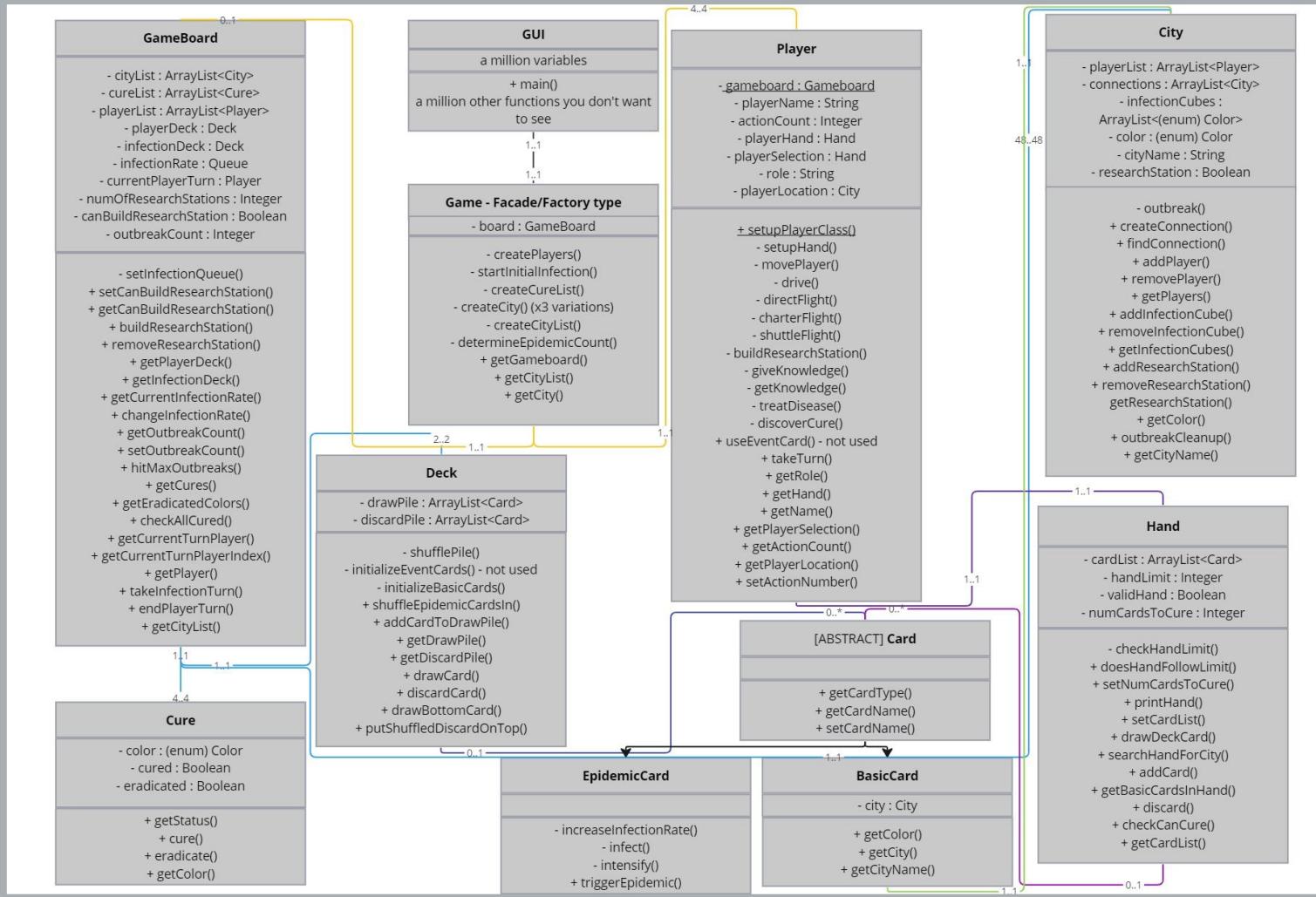
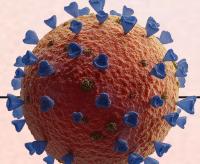
- Visual game board with city buttons
- Player move dashboard
- Hand display
- Text display of what player(s) are in a city
- Text display of research station status

Additional Features

- 4 Difficulty levels
- Players choose their role



UML Design



Collaboration

Communication

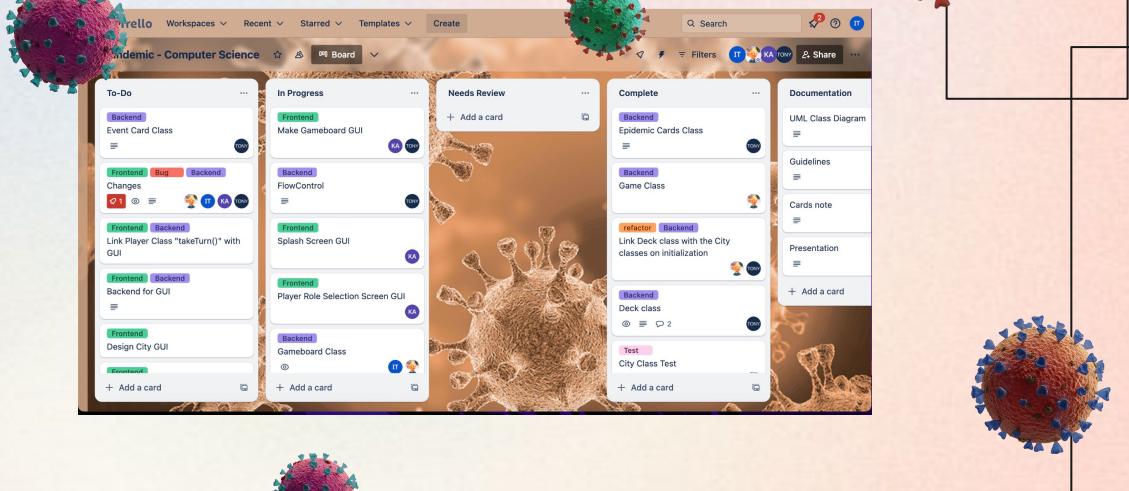
- Texting

GitHub Stuff

- Branching
- Code reviews

Lessons

- Merge often
- To-Do's make it easier
 - Trello board inset
- More group hacking
- Game is more complicated than we thought



Acceptance Plan and Testing

OK if 4 users can play one round successfully



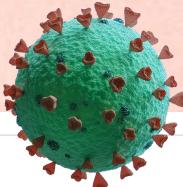
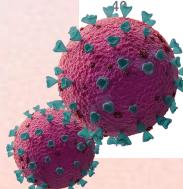
Unit Tests

Very official

Print Statements

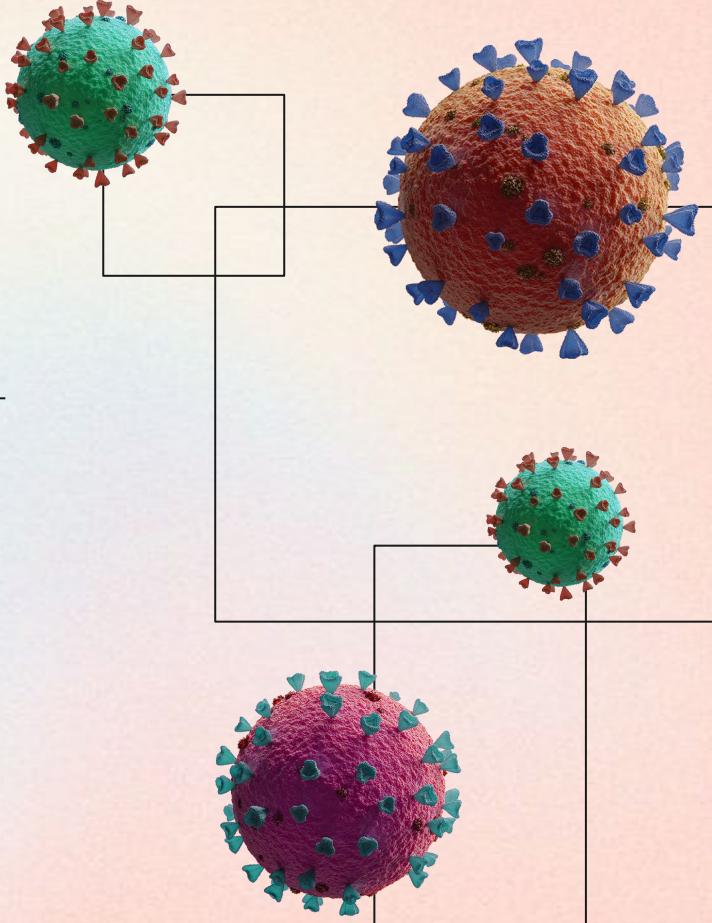
System.out.println()

```
11  public class HandTest {  
14      * @author Izzy T  
15      */  
16      @Test  
17  void testCheckCanCure() {  
18      City newCity1 = new City(Color.YELLOW, "Madrid");  
19      BasicCard cityCard1 = new BasicCard(newCity1);  
20  
21      City newCity2 = new City(Color.YELLOW, "Portugal");  
22      BasicCard cityCard2 = new BasicCard(newCity2);  
23  
24      City newCity3 = new City(Color.YELLOW, "San Francisco");  
25      BasicCard cityCard3 = new BasicCard(newCity3);  
26  
27      City newCity4 = new City(Color.YELLOW, "San Diego");  
28      BasicCard cityCard4 = new BasicCard(newCity4);  
29  
30      City newCity5 = new City(Color.YELLOW, "Spokane");  
31      BasicCard cityCard5 = new BasicCard(newCity5);  
32  
33      ArrayList<Card> cardList = new ArrayList<Card>();  
34      cardList.add(cityCard1);  
35      cardList.add(cityCard2);  
36      cardList.add(cityCard3);  
37      cardList.add(cityCard4);  
38      cardList.add(cityCard5);  
39  
40      Hand newHand = new Hand();  
41      newHand.setCardList(cardList);  
42  
43      boolean canCure = newHand.checkCanCure(Color.YELLOW);  
44  
45      assertEquals(true, canCure);  
46  }
```

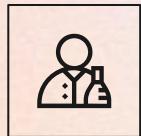


01

Live Demo!



Summary



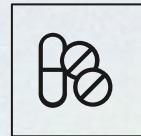
Best Part

Challenging ourselves



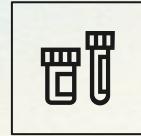
Hardest Part

Connecting
GUI/Backend



Next Time

Event cards!



BRAG

ZERO Merge Conflicts

