Functional Specification

Description

This project creates an online version of the game Exploding Kittens in Java.

The networking is done with HTTP protocol, which is a standardized protocol. This allows future applications to communicate with the server, because HTTP libraries exist for many programming languages.

The backend manages game logic and receives requests from the players. The game logic is handled here to avoid clients messing with the code.

The frontend is a GUI which will display cards, and allows the user to play cards by using the GUI.

Data Structures

| **What** | **Why** |
| --- | --- |
| Map<String, Game> | In order to access games by their ID in *O(log n)* time, we use a map of Game ID to Game. |
| LinkedList<Integer> | A linked list has fast insertion and deletion. By using a linked list to represent a game’s deck, we can easily insert cards within the deck and access cards at certain indices. |
| ArrayList<Player>(4) | The number of players in the game (4) is fixed, so we chose an ArrayList initialized to size 4 to store the player objects. The reason we chose an ArrayList rather than an Array so that we can easily change the amount of players if needed. The game’s backend reflects this and will support any number of players in the ArrayList |
| ArrayList<Integer> | A players hand is stored by an ArrayList of integers as it allows for easy resizing of the player hand and easy addition and removal of cards. Order does not matter, which is why we chose an ArrayList over a LinkedList. |
| Queue<HTTPRequest> | We store all client’s requests in a queue (LinkedList) so the first requests get processed first. |
| Map<String, String> | We use a map for HTTP headers, because it represents how the headers are (map of key to value). |

Major Classes

| **Name** | **Description** |
| --- | --- |
| Game | Handles one game, and stores current game state. Stored server side. |
| Deck | Handles and manages the deck. Stored in Game |
| Player | Stores player ID and hand. Stored within Game |
| Server | Runs in a different thread and adds requests to the queue. |
| Manager | Handles creating games and managing requests. |
| QueueWindow extends JFrame | GUI window when waiting to join a game. |
| GameWindow extends JFrame | GUI window when playing. |

Class Diagram

