Preliminary 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 at both ends. By using a linked list to represent a game’s deck, we can draw from the bottom and top quickly. |
| Player[] | The number of players per game is fixed (4), so using an array rather than a List object reduces overhead. |
| Queue<HTTPRequest> | We store all client’s requests in a queue (LinkedList) so the first requests get processed first. |

Major Classes

| **Name** | **Description** |
| --- | --- |
| Game | Handles one game, and stores current game state. Stored server side. |
| Card | Card types enum. |
| Player | Stores player ID and hand. |
| Server | Runs in a different thread and adds requests to the queue. |
| Manager | Handles creating games and managing requests. |
| HTTPConv | HTTP conversation base class. Parses headers and body. |
| HTTPRequest extends HTTPConv | HTTP request. |
| HTTPResponse extends HTTPConv | HTTP response. |
| QueueWindow extends JFrame | GUI window when waiting to join a game. |
| GameWindow extends JFrame | GUI window when playing. |