Flashcard Game ☺

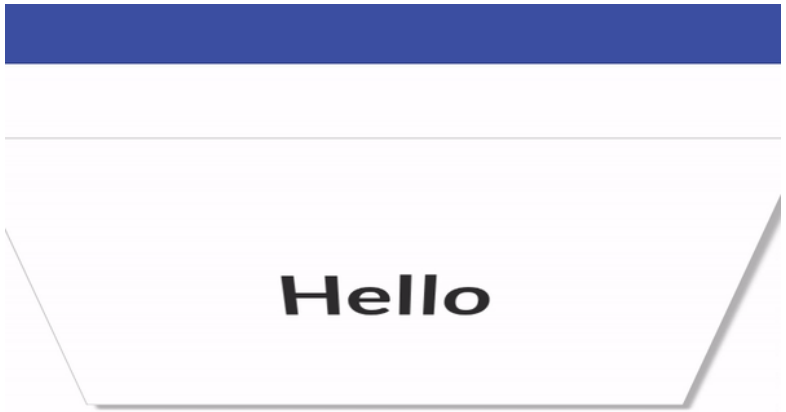
CSC 213 Final Project

Ella Kim, Caitlin Abreu

December 16, 2022

What did we build?

- A game using digital flashcards to study
- User imports file with word-definition pairings
- Practice memorizing vocabulary or concepts by playing the game
- Individually, or multiplayer!



Demo

Files and Filesystems

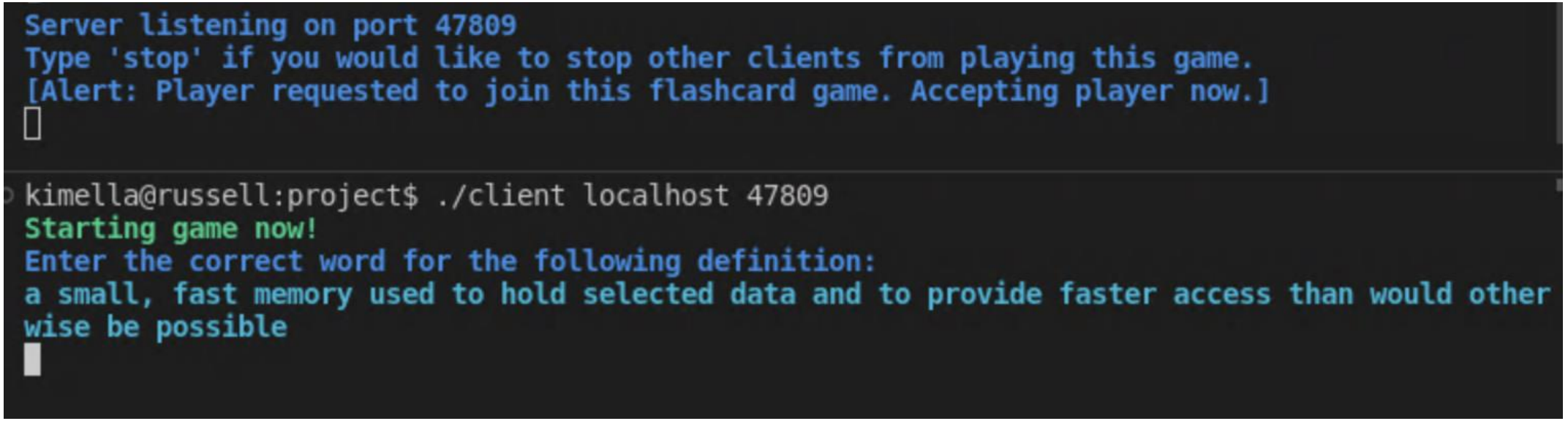
```
printf("Enter an absolute path to the text file. Type 'q' to quit:
printf("\e[0m");

// Get input path
fgets(path,200,stdin);
```

```
// Open this file to read
FILE *fp = fopen(token, "r");
if (fp == NULL){
    printf("\033[0;31m");
    printf("Error opening file. Try again. \n\n");
    printf("\033[0m");
    continue;
}
```

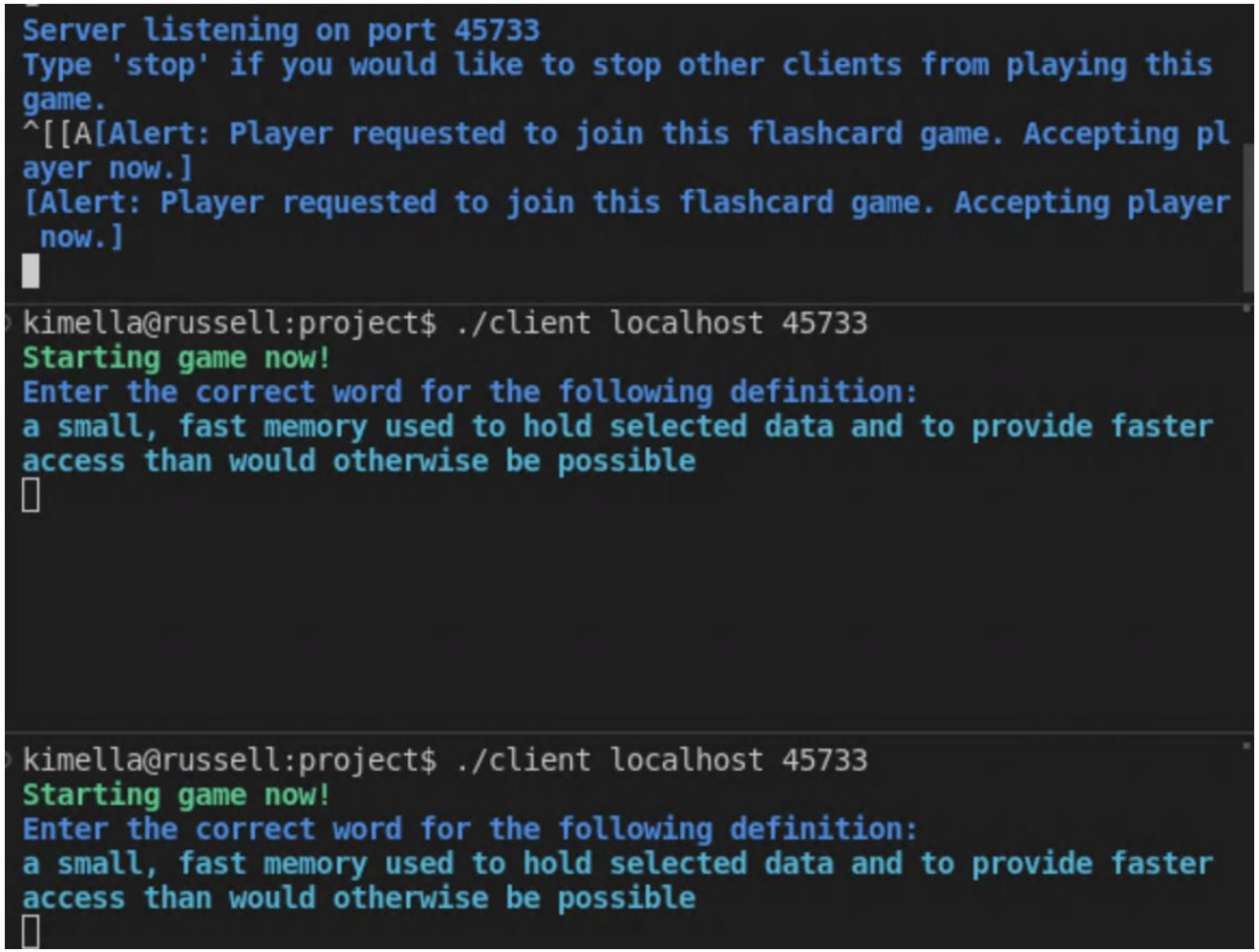
Files and Filesystems

```
Enter an absolute path to the text file. Type 'q' to quit:
test1.txt
Adding word: cache memory
Adding definition: a small, fast memory used to hold selected data and to pro
vide faster access than would otherwise be possible
Adding word: LRU policy
Adding definition: Least-recently-used; a page-replacement policy that remove
s from main memory the pages that show the least amount of recent activity
Adding word: thread
Adding definition: portion of program that can run independently of other por
tions
Finished adding flashcard set!
```

Networking


Server listening on port 47809
Type 'stop' if you would like to stop other clients from playing this game.
[Alert: Player requested to join this flashcard game. Accepting player now.]
kimella@russell:project$ ./client localhost 47809
Starting game now!
Enter the correct word for the following definition:
a small, fast memory used to hold selected data and to provide faster access than would other
wise be possible

Networking


Server listening on port 45733
Type 'stop' if you would like to stop other clients from playing this game.
^[[A[Alert: Player requested to join this flashcard game. Accepting player now.]
[Alert: Player requested to join this flashcard game. Accepting player now.]
kimella@russell:project$ ./client localhost 45733
Starting game now!
Enter the correct word for the following definition:
a small, fast memory used to hold selected data and to provide faster access than would otherwise be possible
kimella@russell:project$ ./client localhost 45733
Starting game now!
Enter the correct word for the following definition:
a small, fast memory used to hold selected data and to provide faster access than would otherwise be possible

Threads

```
printf("Type 'stop' if you would like to stop other clients from playing this game.\n");
printf("\e[0m");
// Create a thread running in the background to read the "stop" input from user
pthread_create(&server_thread, NULL, readStop,(void *)&info);
```

```
while (1) {
    // Accept new client connecting to the server
    int client_socket_fd = server_socket_accept(server_socket_fd);
    if (client_socket_fd == -1) // if server closed or is not valid
    {
        break;
    }
    // Store the new client into the array
    client_array[idx] = client_socket_fd;
    info.client_fd = client_array[idx];
    // Create a thread running in the background to send to the client the flashcard set
    pthread_create(&client_threads[idx], NULL, sendFlashcards, (void *)&info);
    idx++;
}
```

Implementation Details

Most Challenging Part of Implementation

```
static int server_socket_accept(int server_socket_fd) {
  // Create a struct to record the connected client's address
  struct sockaddr_in client_addr;
  socklen_t client_addr_len = sizeof(struct sockaddr_in);
  struct timeval timeout;
    timeout.tv_sec = 100;
    timeout.tv_usec = 0;

    if (setsockopt (server_socket_fd, SOL_SOCKET, SO_RCVTIMEO, &timeout,
                    sizeof timeout) < 0) {
        // perror("setsockopt failed\n");
        return -1;
        }

    if (setsockopt (server_socket_fd, SOL_SOCKET, SO_SNDTIMEO, &timeout,
                    sizeof timeout) < 0) {
        return -1;
        }

  // Block until we receive a connection or failure
  int client_socket_fd = accept(server_socket_fd, (struct sockaddr*)&client_addr, &client_addr_len);

  // Did something go wrong?
  if (client_socket_fd == -1) {
    return -1;
  }

  return client_socket_fd;
}
```

Thank you.
Questions?