```
// Generic includes
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>
#include <stdbool.h>
#include <stdint.h>

// Socket/network includes
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <fcntl.h>

#include <pthread.h>

// Mqueue include
#include <mqueue.h>

// Error handling
#include <errno.h>

#define MAX 50
#define PORT 8000
#define SA struct sockaddr
```

```
int clientGame()
{
	int clientSocket, ret;
	struct sockaddr_in serverAddr;
	char buffer[1024];

	clientSocket = socket(AF_INET, SOCK_STREAM, 0);
	if(clientSocket < 0)
	{
		printf("ERROR: Cannot create client socket.\n");
		exit(1);
	}
	printf("CONSOLE: Created client socket.\n");

	memset(&serverAddr, '\0', sizeof(serverAddr));
	serverAddr.sin_family = AF_INET;
	serverAddr.sin_port = htons(PORT);
	serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");

	ret = connect(clientSocket, (struct sockaddr*)&serverAddr, sizeof(serverAddr));
	if(ret < 0)
	{
		printf("ERROR: Cannot connect to server.\n");
		exit(1);
	}
	printf("CONSOLE: Connected to Server.\n");

  while(1)
  {
```

```
printf("Welcome to the word game.\n");
printf("Please choose an option below (1 for Singleplayer, 2 for Multiplayer, 3 for exit)\n");
printf("1) Singleplayer\n");
printf("2) Multiplayer\n");
printf("3) Exit\n");
printf("> ");

scanf("%s", &buffer[0]);
send(clientSocket, buffer, strlen(buffer), 0);
printf("Input: %s\n", buffer);

if (strcmp(buffer, "1") == 0)
{
   printf("\nSingle Player Mode\n");
   printf("Enter your first name: ");
   bzero(buffer, sizeof(buffer));
   scanf("%s", &buffer[0]);
   send(clientSocket, buffer, strlen(buffer), 0);

   printf("\nEnter your last name: ");
   bzero(buffer, sizeof(buffer));
   scanf("%s", &buffer[0]);
   send(clientSocket, buffer, strlen(buffer), 0);

   printf("\nEnter your country: ");
   bzero(buffer, sizeof(buffer));
   scanf("%s", &buffer[0]);
   send(clientSocket, buffer, strlen(buffer), 0);
```

```
int first = 1;
int pass = 0;
while(pass < 4)
{
   bzero(buffer, sizeof(buffer));
   recv(clientSocket, buffer, 1024, 0);
   printf("Letters: %s\n", buffer);

   int resets = 0;
   while (resets < 3)
   {
      if (first == 1)
      {
         // Recieves starting character
         char starting_char = '0';
         bzero(buffer, sizeof(buffer));
         recv(clientSocket, buffer, 1024, 0);
         strcpy(&starting_char, buffer);
         printf("The starting character is: %c\n", starting_char);

         // First words submission
         printf("\nEnter your word: ");
         bzero(buffer, sizeof(buffer));
         scanf("%s", &buffer[0]);
         send(clientSocket, buffer, 1024, 0);

         if (strcmp(buffer, "pass") == 0)
         {
            pass++;
```

```
            break;
        }

        // Receives answer
        bzero(buffer, sizeof(buffer));
        recv(clientSocket, buffer, 1024, 0);

        if (strcmp(buffer, "INCORRECT") == 0)
        {
            printf("INCORRECT\n");
            resets++;
            continue;
        }
        if (strcmp(buffer, "CORRECT") == 0)
        {
            first = 0;
            pass = 0;
            printf("USER SCORED\n");
            // Check for bonus points
            break;
        }
    }
    else
    {
        // Recieves number of used words
        // Recieves used words
        char usedWords[100][100];
        uint32_t converted = 0;
        recv(clientSocket, &converted, sizeof(converted), 0);
```

```
uint32_t noUsedWords = htonl(converted);
printf("NUMBER OF WORDS: %d\n", noUsedWords);

printf("WORDS USED: ");
for (int i = 0; i <= noUsedWords; i++)
{
   bzero(buffer, sizeof(buffer));
   recv(clientSocket, buffer, sizeof(buffer), 0);
   printf("%s ", buffer);
}
printf("\n");

// First words submission
printf("\nEnter your word: ");
bzero(buffer, sizeof(buffer));
scanf("%s", &buffer[0]);
send(clientSocket, buffer, 1024, 0);

if (strcmp(buffer, "pass") == 0)
{
   pass++;
   break;
}

// Receives answer
bzero(buffer, sizeof(buffer));
recv(clientSocket, buffer, 1024, 0);

if (strcmp(buffer, "INCORRECT") == 0)
```

```
            {
                printf("INCORRECT\n");
                resets++;
                continue;
            }
            if (strcmp(buffer, "CORRECT") == 0)
            {
                pass = 0;
                printf("USER SCORED\n");
                break;
            }
        }
    }
    // Computer plays
    // Recieves if computer scored or not
    bzero(buffer, sizeof(buffer));
    recv(clientSocket, buffer, sizeof(buffer), 0);
    printf("COMPUTER BUFF: %s\n", buffer);
    if (strcmp(buffer, "COMP CORRECT") == 0)
    {
        pass = 0;
    }
    if (strcmp(buffer, "COMP PASSED") == 0)
    {
        pass++;
    }
}

// SCOREBOARD OUTPUT HERE
```

```
        }
        if (strcmp(buffer, "2") == 0)
        {


        }
    }
}

int main()
{
    clientGame();
    return 0;
}
```