CS307 Programing Assignment 2 Report

Kaan Atmaca

Pseudocode of locking algo:

```
While game not finished:
        Create threadx
        Create threado
        Lock mutex with threadx
                Play x  on table
        Unlock mutex with threadx
        Join threadx
        Check if game finished
                break if finished
        Lock mutex with thready
                Play y  on table
        Unlock mutex with thready
        Check if game finished
                break if finished
        Join thready
After game finished if breaked with threadx
        Join thready
```

Lock:

I used only one lock which I created by:

```
pthread_mutex_t mut = PTHREAD_MUTEX_INITIALIZER;
```

The purpose for this mutex is adding the variables x and o to the same table with no confusion. So that each variable will be added one by one. Since if they didn't use the same mutex they may add to the same arr[row][column] and with this there won't be synchronization. So I used the mechanism of one mutual mutex in adding variable to table function (game function). I adapted it to be locked until one thread (player x or o) finished entering their variable on to the table and after that it would be unlocked. Next the thread which was waiting (the other player) will lock, so, that they will be the only one to implement a variable on to the table. With all this I was able to add all variables to the same table one by one in order so that only one player can put its variable on to the table and only one player can win.