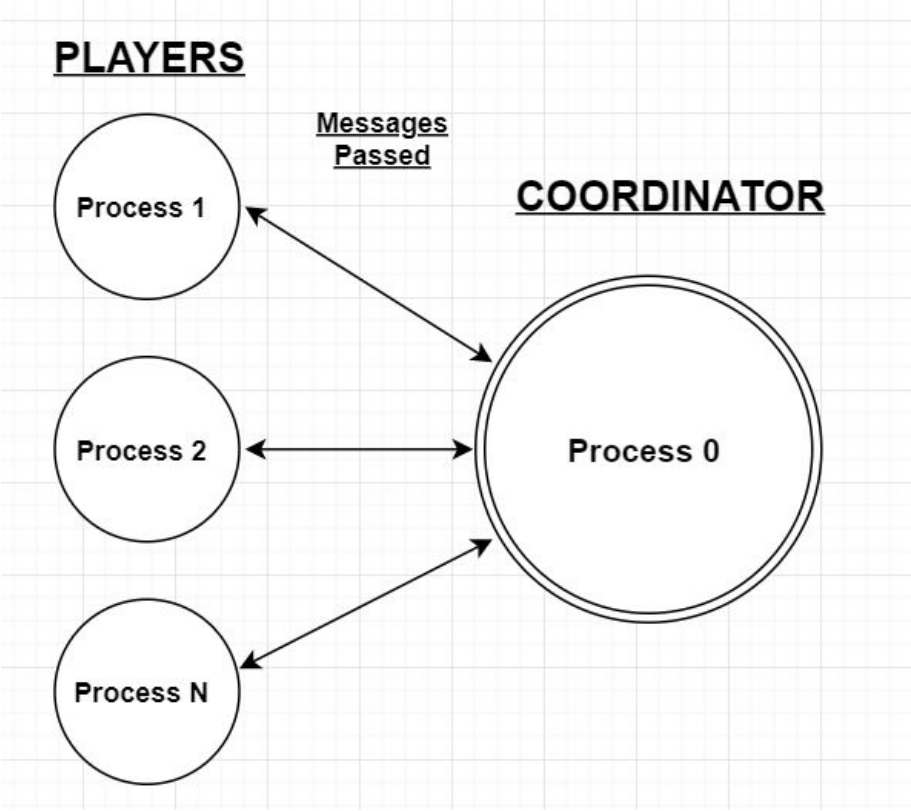
Logan D'Auria 10/18/2019

Game of Morra Project Description

Approach

The approach to create the Game of Morra using a Message Passing Interface involved a two sided unicast messaging system between a coordinating process thread and different player threads. The coordinating process handled the game logic, while the player processes passed their moves and guesses in messages sent to the coordinating process.



Assigning Processes

Each process is assigned a rank from 0 to n. The Coordinating process will always be process 0 and players will be from 1 to n. Their work is separated by an if statement.

```
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
```

```
if (rank == 0) {
    // Coordinating process work
}
else {
    // Player process work
}
```

Analysis

Using MPI to control player processes was very efficient and has high scalability. The synchronization of message sending was easy to organize due to the blocking nature of MPI_Send and MPI_Recv

The process for 500 players was timing out. This is likely due to the way the second part of the algorithm was implemented for deciding how many players each player outperformed. Since they all need to send other players messages and receive a message from every player, with 500 players that totals to ~250,000 message sends and receive requests. This could have been improved using a broadcast message instead of individually sending messages to each process.