|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lunes | Martes | Miercoles | Jueves | Viernes | Sabado | Domingo |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Herramientas a utilizar: IDE Momentics, Framework Cocos2d, NativeSDK

Con respecto a la parte de la comunicación:

* Investigar sobre server socket en c++
* Protocolo servidor local anidado en el cliente
* Crear la mesa y arrojar un identificador de la mesa, enviarlo por broadcast, reenviar cada cierto tiempo
* Protocolo de comunicación.
* Averiguar IP broadcast, subnet mask.

Cliente:

* Menú:
  + Crear una mesa
  + Unirse a una mesa
  + Configuraciones
* Partida:
  + Manejo de sus propias fichas
    - Cuales tienes, seleccionarlas y colocarlas
  + Tablero: las cabezas.
  + Interfaz gráfica
  + Definir si una jugada es válida: Turno del jugador válido, al escoger una ficha y desplazarla hacia algún lado de la mesa verificar si la jugada se puede realizar.

Servidor:

* Partida:
  + Manejo de turnos:
    - Anillo lógico con token
  + Se debe tener en cuenta que se tiene un jugador con quien es equipo.
  + Al realizar una jugada se calcula cuantas fichas se tienen restantes, luego se restan de haberse hecho una jugada efectiva y se verifica si gano
  + El servidor escucha por el socket correspondiente al jugador actual: recibe jugada y la reenvia.
  + Al llegar a los 100 puntos entonces debe finalizar el juego y mandarle al cliente la puntuación final.

El servidor va a mantener la cuenta de cuantas fichas quedan a cada uno y el ultimo jugador que realizo una jugada efectiva para poder reconocer si el juego esta trancado

Protocolo de comunicación

* Mensajes que envía el cliente:

- Jugada: j<numero1><numero2><ConCualJugo>

- Piezas que me quede: z<NumPiezas><Piezas(2\*NumPiezas)>

- No puede jugar: n

* Mensajes que envía el servidor:

- Jugador pasó: p<IdJugador>

- Broadcast: b<enteroIdMesa><IPPorNumeros>

- Mano inicial: e<IdJugador><Piezas(14 enteros)>

- Se sede el turno: t

- Notificación de jugada: a<idJugador><numero1><numero2><ConCualJugo>

- Fin de mano: m

- Puntuacion de la mano:

s<Score0><Score1>(<NumPiezas><Piezas (2\*NumPiezas)>){4}

**Nota**: Administrador de la mesa. En el lobby podría armar equipos y botar jugadores de una mesa.

Convenciones de C++:

* Convención de la definición de las clases,funciones y variable en inglés y Java.
* Tabulación de 3 espacios (sustituidas por espacios)
* Poner la llave abajo.
* Sobre los comentarios: descripción, input: <nombreParam> -> <Descripcion>, output: <Descripción>, sideEffects <Variables que son globales>.
* Todos los apuntadores usar malloc al iniciar y free al final, manejar todo por referencia.

Clases:

* Domino: dos enteros.