Andres Chaves y PabloHidalgo Cliente.java Noviembre 15,2021 CronoThread CronoThreadSopa Personaies EscogerPersonaje.java Juego.java boolean inmovil.repite -CollectCoins refPantallaCC; threadCliente.java -SopaLetras refPantallaCC; - String nombre; boolean isRunning = true; boolean isRunning = true; JButton refButton; - int seconds = 0: int seconds = 0: Juegos - int numero: Cliente Juego int minutes = 0; - int minutes = 0; - int casilla.turno.cantInmovil: Cola.java Imagelcon imagen; + String: IP_SERVER + Cliente: cliente Tablero.java + run():void + run():void + Juego: ventanaCliente; - setNiceFormat(int number): String TiroDadosInicio.java - setNiceFormat(int number): String -DataInputStream: entrada; + avanzarCasillaActual(int avance):void regresarJugador.java + setIsRunning(boolean estado):void + setIsRunning(boolean estado):void -DataOutputStream: salida; + Juego():void + aetNum():int + getSeconds():String + getSeconds():String -Socket: cliente; + getName():String + getMinutes():String + getMinutes():String Mario Thread -EscogerPersonajes: vtnEscoger; ⊦ getIcon():ImageIcon + getIntSeconds():int + getIntSeconds():int -String: nomCliente; + getTurno():int + getIntMinutes():int + getIntMinutes():int -ArrayList<Integer>: disponibles; + getCasillaActual():int + setSegundos(int seg):void + setSegundos(int seg):void -int: miPersonaje; CronoThread.java + getRefButton():JButton + setMinutos(int min):void + setMinutos(int min):void -ObjectInputStream: entradaObj; CronoThreadSopa.java + getInmovil():int -ObjectOutputStream: salidaObj; + getCantInmovil():int + getRepite():boolean Games Factory + setRepite(boolean repite):void + setCantInmovil(int cantInmovil):void + conexion(): void CollectCoins.java + getNombre(): String + setInmovil(boolean inmovil):void GuessWho.java + getListaDisponibles(): + setRefButton(JButton refButton):void SopaLetras CollectCoins JuegoGato.java setTurno(int turno):void ArrayList<Integer> threadCliente JuegosFactory.java - int size[] = new int[3]; + setCasillaActual(int casillaActual):void + getMiPersonaje():int + int DIMENSIONES = 25; MarioCards.java - int tam1 = 10; + setNumero(int numero):void isDigit(String num):boolean + JButton[][]: botones +JButton[DIMENSIONES][DIMENSIONES]; - DataInputStream: entrada; Memory.java - int tam2 = 15;setImage(int num):void + int[][] valoresM: int[DIMENSIONES][DIMENSIONES]; MemoryPath.java ObjectInputStream: entradaObj; - int tam3 = 20:SopaLetras.java Servidor DataOutputStream: salida; + ImageIcon: iconoM - int tamMatriz,ajuste; TiroDadosInicio: inicio; + boolean: activo; int arrayLetras[][]; + CronoThread: cronometro; - String: name: - ArrayList<String> array4Palabras; int: num,acceso; Escoger Personaje + JButton[][] tableroLabels; FrameServidor.java - boolean: juegoActivo,resultado; + CronoThreadSopa cronometro; + setCronometro():void ServidorMarioParty.java + ArrayList<Personajes>: jugadores; + generarMonedas():void threadServidor.java ArrayList<Integer>: personajes; float letra=8: + ArrayList<String>: infoCasillas; + clickMoneda(java.awt.event.MouseEvent evt):void int contador=0,y=0,x=0; +Tablero: tablero; int : seleccionado; + finJuego():boolean - ArrayList<String> arrayList = new ArrayList<>(); Personajes String palabra = "" - initComponentes(): void + run(): void + int buenas = 1; - actualizar(int num): void + esperar():void boolean activo; + getPersonajes(): personajes.java + continuar():void - String letras[] ArrayList<Integer> + abrirHielo():void + getOut(): int + setCronometro():void - abrirCola():void + getSeleccionado():int - setTextToCrono(String newTempo):void abrirFuego():void + setOut(int out):void Imagenes - initTablero():void - actualizarme(int turno):void initMatriz():void - actualizarJuego():void ajustarBotones(int tamMatriz):int - collectCoins():void + leerTxt():void - memoryPath():void Imagenes - validarPosicion(String a, int x, int y, String m):boolean guessWho():void + palabraCorrect():void sopaLetras():void + PalabrasIguales(String a, String b):boolean - cards():void - revisarValores(int valores, int tipos, int valorJ, int tipoJ):boolean - memory():void ServidorMarioParty FrameServidor - juegoGato():void + FrameServidor ventana; JuegosFactory turnoP1Gato(JuegoGato ventanajuego,String enemigo):void + ServidorMarioParty servidor - ArrayList<Socket> cliente: ArrayList<Socket>(); turnoP2Gato(JuegoGato ventanajuego, int columna, int fila):void MemoryPath - ArrayList<threadServidor> user: ArrayList<threadServidor>(); ganadorGato():boolean + mostrar(String txt):void ArrayList<Integer> disponibles: ArrayList<Integer>(); + int fila =3: - perdida():void + static javax.swing.JFrame crearJuego(Games tipo) ArrayList<String> casillas: ArrayList<String>(); + int col =6; isDigit(String num):boolean - int cant,jugadores=6; + int size =70; Random r=new Random(); boolean activo=true; int filas=5: + runServer():void + JButton[][] botones = new JButton[fila][col]; cargarPersonajes():void + ImageIcon inicio; cargarCasillas():void + Imagelcon bueno; getJuego(int opcion):String setDisponibles(ArrayList<Integer> disponibles):void generarTablero():void habilitar(JButton [][] botones,int columna):void + setJugadores(int cantJuagdores):void getDisponibles():ArrayList<Integer> asignarBuenos(JButton[][] botones):void getUser():ArrayList<threadServidor> + actionPerformed(ActionEvent e):void addContrincantes():void ordenDados(int ordenT):void ordenNum(int ordenT,int corte):void Tablero MarioCards + int DIFCARTAS = 13; + int BUTTON_SIZE =70; + int TIPOSCARTAS = 4; + int BOARD_SIZE = 28; + JButton[][] cartas = new JButton[DIFCARTAS][TIPOSCARTAS]; - JButton buttonArray[]: JButton[BOARD_SIZE]; + Imagelcon iconoM; - int turno; + int tipo = -1; - int max,resultadoDados,id,tubo; + int valor = -1; threadServidor Random r: new Random(); ArrayList <Personajes> jugadores; + generarCartas():void Socket cliente ArrayList <String> infoCasillas; + darInfoCarta(java.awt.event.MouseEvent evt):void JuegoGato DataInputStream entrada - Personajes enemigo ; + getTipoCarta(int num):String DataOutputStream salida - boolean finish, repite, fuego, hielo, cola; + String amigo; + getValorCarta(int num):String ObjectOutputStream salidaObj + String enemigo; - Personajes jugadorEnTurno; + getTipo():int ObiectInputStream entradaObi + int DIMENSIONES = 3: boolean juego=false; + setTipo(int tipo):void Memory Personajes personaje + JButton[][] tableroLabels :JButton[DIMENSIONES][DIMENSIONES]; - String nombrejuego; + getValor():int + String nameUser; + int[][] tableroLogico = int[DIMENSIONES][DIMENSIONES]; - int col=3; + setValor(int valor):void ServidorMarioParty servidor + Imagelcon iconoVacio; - initUsers():void + ArrayList<threadServidor> enemigos int contador=0,k=0,l=0,parejas=0,intentos=9; + Imagelcon iconoEquiz; + ArrayList<Personajes> jugadores - initBoard():void int cartasIguales,turnoJuego=1; + ImageIcon iconoCirculo - jButtonLanzarActionPerformed(java.awt.event.ActionEvent evt) + int numeroDeJugador,opcion; int miTurno=1; + int turnoJugador=1; - int puesto=-1; + actualizarTurno():void - boolean listo; + int numeroJugador = 0; + actualizarEnemigos(int movimiento, int id,int tubo):void - JButton[][] tableroLabels= new JButton[fila][col]; + int filaA ; + run():void - verificarDados(Personajes jugador,int dado1,int dado2):boolean int math [][]= new int[fila][col]; + int columnaA; + lanzarOrden(int ordenT):void + actualizarJugadorInmovil(Personajes jugador):void + Random r= new Random(); + boolean resultado=false,full=false; + tablero(ArrayList<String>infoCasillas):void + moverFicha(Personajes jugador, int resultadoDados,int boolean activo; + getNameUser():String tipo):void - initMatrizLogica():void + getPersonaje():Personajes - verificarCasilla(JButton casilla,Personajes jugador): Personajes + getAmigo():String - initMatrizImagenes():void + getResultadoOrden():int - cola(Personajes jugador):int + setAmigo(String amigo):void eventoCartas():void + setNameUser(String name):void GuessWho encontrarTubo(String tubo):int + getEnemigo():String - voltearCartas():void + setOpcion(int num):void - atacarEnemigo(Personajes jugador,int tipo):Personajes + setEnemigo(String enemigo):void + int DIMENSIONES = 10: - mismo(JButton jButton, JButton jButton0):boolean + setResultadoOrden(int num):void + getTurnoJugador():int + int CANTBOTONES = 15; + getContador():int + compareTo(threadServidor compareUser):int + setTurnoJugador(int turnoJugador):void + int intentos; setK(int K):void + getNumeroJugador():int + JButton[[]: botones: JButton[DIMENSIONES][DIMENSIONES]; setL(int L):void + setNumeroJugador(int numeroJugador):void + JButton[] opciones :JButton[CANTBOTONES]; - getK():int + getResultado():boolean + int[][] valoresM: int[DIMENSIONES][DIMENSIONES]; getL():int + getFull():boolean +ImageIcon iconoM + setTurnoJuego(int turnoJuego):void + setResultado(boolean resultado):void +int: indicef; + actualizarIntentos():void + setFull(boolean full):void - boolean: activo; + getIntentos():int + generarTablero():void + setMiTurno(int miTurno):void + generartablero():void + marcar(int columna, int fila):void + getParejas():int + generarbotones():void + clickSobreTablero(java.awt.event.MouseEvent evt):void + aumentarParejas():void + setTransparente(java.awt.event.MouseEvent evt):void + getFilaA():int + setContador(int contador):void + generarIntentos():void + setFilaA(int filaA): void + sumarContador():void + acertar(java.awt.event.MouseEvent evt):void + getColumnaA(): int + getCartasIguales():int + setColumnaA(int columnaA):void + cartaGuardada(int cartasIguales):void + fullTablero(): boolean + getCartaGuardada():int + haGanado():booelan + conocerValor(int i,int j):int + setEnemigoL(String enem):void + getListo():boolean + getActivo():boolean

+ setListo(boolean listo):void

+ setActivo(boolean activo):void

Diagrama de clases Proyecto 2