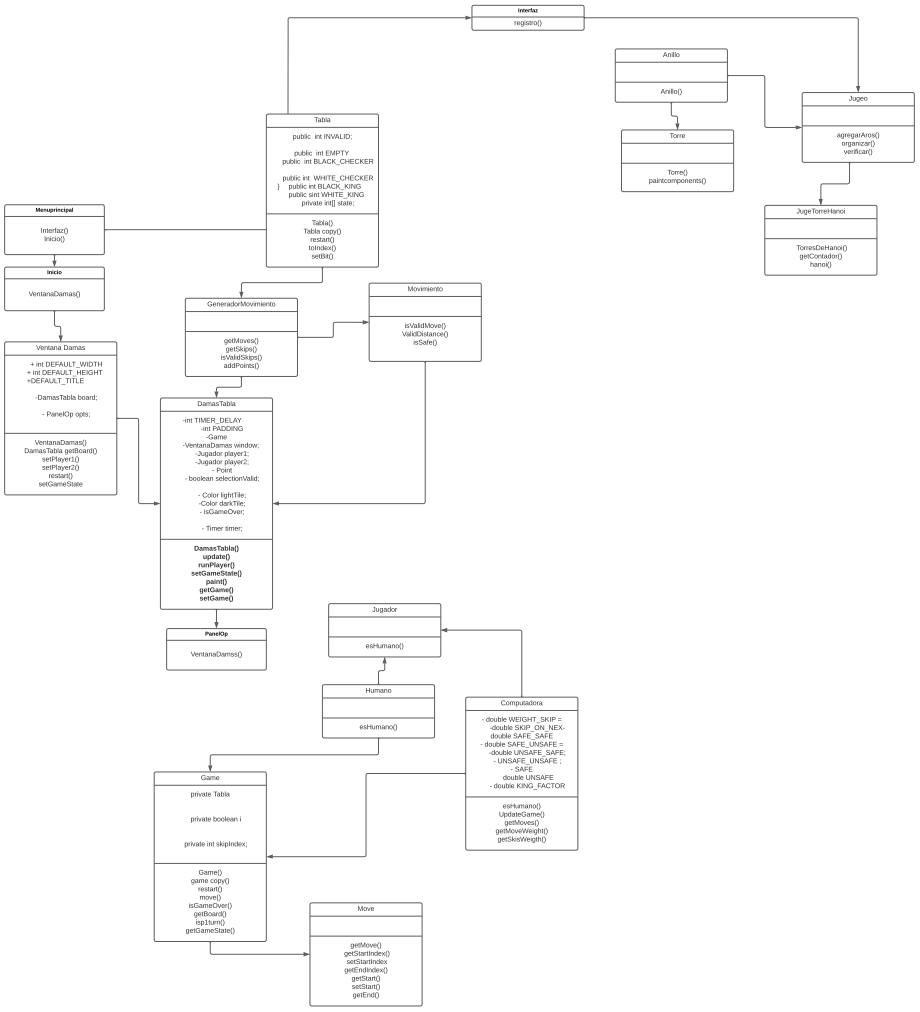
Interfaz
registro()
Anillo
Anillo()
Jugeo
agregarAros()
organizar()
verificar()
Tabla
public int INVALID;
public int EMPTY
public int BLACK_CHECKER
public int WHITE_CHECKER
} public int BLACK_KING
public sim WHITE_KING
private int[] state;
Tabla()
Tabla copy()
restart()
toIndex()
setBit()
Torre
Torre()
paintcomponents()
JugeTorreHanoi
TorresDeHanoi()
getContador()
hanoi()
Mensprincipal
Interfaz()
Inicio()
Inicio
VentanaDamas()
GeneradorMovimiento
getMoves()
getSkips()
isValidSkips()
addPoints()
Movimiento
isValidMove()
ValidDistance()
isSafe()
Ventana Damas
+ int DEFAULT_WIDTH
+ int DEFAULT_HEIGHT
+DEFAULT_TITLE
-DamasTabla board;
- PanelOp opts;
VentanaDamas()
DamasTabla getBoard()
setPlayer1()
setPlayer2()
restart()
setGameState
DamasTabla
-int TIMER_DELAY
-int PADDING
-Game
-VentanaDamas window;
-Jugador player1;
-Jugador player2;
- Point
- boolean selectionValid;
- Color lightTile;
-Color darkTile;
- isGameOver;
- Timer timer;
DamasTabla()
update()
runPlayer()
setGameState()
paint()
getGame()
setGame()
PanelOp
VentanaDamss()
Jugador
esHumano()
Humano
esHumano()
Computadora
- double WEIGHT_SKIP =
-double SKIP_ON_NEX-
double SAFE_SAFE
- double SAFE_UNSAFE =
-double UNSAFE_SAFE;
- UNSAFE_UNSAFE ;
- SAFE
double UNSAFE
- double KING_FACTOR
esHumano()
UpdateGame()
getMoves()
getMoveWeight()
getSkisWeight()
Game
private Tabla
private boolean i
private int skipIndex;
Game()
game copy()
restart()
move()
isGameOver()
getBoard()
isp1turn()
getGameState()
Move
getMove()
getStartIndex()
setStartIndex
getEndIndex()
getStart()
setStart()
getEnd()