

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

Diseño de Software

CI-0136

Segunda iteración del MARDA

Profesor:

Alan Calderón Castro

Integrantes:

Rodrigo Li Qiu B94263

Axel Matus Vargas B74597

Jorim G. Wilson Ellis B98615

Asdrúbal Villegas Molina B88583

II Ciclo 2021

Justificación:

Este trabajo sigue el patrón de arquitectura modelo-vista-controlador, el motivo es que el objetivo de este trabajo es hacer código genérico a partir del cual se pueden crear diferentes juegos de mesa. Este patrón tiene la ventaja de que si los datos cambian la vista no se ve afectada ni viceversa, por lo que es ideal para este trabajo.

Se decidieron utilizar clases abstractas (BoardMaker y Pieces) para que a partir de estas se pudiera crear un juego de mesa que necesite tablero y piezas. Para demostrar su utilidad se crearon las clases CheckersBoard y CheckersPiece para hacer un tablero compuesto por piezas de damas inglesas. El controlador utiliza estas clases para generar el tablero, el cual la clase Data (indica la información de cómo se debe jugar el juego) y la clase Board (se encarga de la representación visual). La clase MovesMade es una clase auxiliar que ayuda a comunicar la información sobre el movimiento de las piezas. Otra auxiliar es la clase player que da información acerca de los jugadores.

Patrones de diseño:

1. Patrón Constructor: El patrón de diseño constructor permite que se puedan construir clases complejas a partir de una sola. Este patrón se utilizó para este proyecto en donde se crearon clases abstractas como BoardMaker, la cual sirve para construir un tablero general (builder). Luego la clase CheckersBoard es una clase más concreta que funciona para construir específicamente un tablero de damas inglesas (concrete builder). Luego la clase controller hace uso de la clase anterior para poder generar y controlar un tablero de damas.

2. Patrón de plantilla: El patrón plantilla permite definir el esqueleto de un programa mediante clases abstractas de las cuales se pueden crear clases más concretas para realizar lo que se necesita. Un ejemplo de lo anterior es que para este trabajo se tiene las clases abstractas BoardMaker y Pieces las cuales sirven para definir un juego de mesa que necesita un tablero y piezas. Estas clases tienen métodos definidos y otros abstractos que las clases concretas deben definir, en este caso, las clases CheckersBoard y CheckersPiece que heredan las anteriores para así poder formar un juego de damas inglesas.

Patrones de Arquitectura:

Modelo-Vista-Controlador (MVC): El trabajo se puede considerar como un ejemplo de Modelo-Vista-Controlador ya que unas clases se encargan de ser el modelo (dictar las reglas del juego), la clase Board se encarga de ser la representación

visual (vista) y la clase Controlador se encarga de la comunicación de los anteriormente mencionados.