Proyecto II Programacion Facultad de Matematica y Computacion Universidad de la Habana
Curso 2022

Resumen:

Integrantes:
Alex Sánchez Saes C212
Abdel Fregel Hernández grupo C212

Sumario

En que consiste el Proyecto:

El objetivo de este proyecto es construir una librería de clases, que permita crear varias implementaciones de juegos de dominó con reglas variadas , así como distintas estrategias para jugadores inteligentes , y así propiciar un entorno donde evaluar las mismas.

Cómo Funciona

Para la creación de este proyecto separamos los elementos que forman un juego de dominó, tales como la condición de ganar una partida , la forma en que se pasan los turnos , las reglas de cuando una ficha se puede jugar , la forma en que se reparten las fichas y el criterio de cómo encajan unas fichas con otras ; para los jugadores, implementamos distintas estrategias de juego , creamos un jugador que selecciona una ficha de manera aleatoria dentro de sus fichas válidas , el jugador botagorda , el cual selecciona la ficha que más valor tenga de entre sus jugadas posibles , y el jugador heurístico,este calcula cuál es la ficha más probable que haga pasarse a los demás jugadores .En cuanto a la interacción con el usuario , implementamos una interfaz gráfica , que consta con una página web escrita en blazor , esto permite visualizar el juego en tiempo real , para esto hicimos una implementación del patrón de diseño Observer , mediante el cual se comunican la parte gráfica y el juego a través de eventos , eventos tales como ; cuándo se juega una ficha , cuándo se pasa un turno o cuándo alguien gana , se conectan los componentes del juego con la parte gráfica y esta es la que abstrae la lógica de representar el juego en pantalla . Otro patrón que utilizamos para la construcción del proyecto fue el patrón Factory mediante el cual a través de distintos parámetros abstrajo la lógica de cómo se construye el juego y se unen los componentes.

Abstracciones Hechas

Entre las abstracciones que tomamos para descomponer el juego, está el concepto de cara de una ficha,el cual se encapsula en la interfaz IFace , dicha interfaz contiene los métodos comunes que deben tener las caras de una ficha independientemente de su tipo como son;obtener su valor , obtener su representación y compararla para saber si es igual a otra, muy ligado al concepto de qué es una cara, está el concepto de una ficha , esta tiene los métodos necesarios para obtener las caras de la ficha y un método que calcula su valor ,este está representado con la interfaz IKey . En el concepto de mesa se encapsuló la idea de saber si una ficha es válida para jugar , saber que caras actualmente están en juego en la mesa , ver todas las fichas que se han jugado y por supuesto jugar una ficha, de todo esto se encargan las clases que implementen la interfaz ITable . Para definir cómo se pasan los turnos entre jugadores creamos la interfaz IPlayerSelector la cual implementaran todos los pasadores de turno , esta tiene métodos para obtener el próximo jugador a jugar . Para decidir quién gana según el criterio que se escoja , esto se define a través de la interfaz IWinCondition la cual encapsula los métodos para saber si alguien ha ganado la partida y para saber cuál es el ganador , todas las implementaciones concretas de estas interfaces se unen en la clase Manager que representa un juego funcional , es de alguna manera el armazón del juego, por último la Interfaz IScreen engloba la abstracción de un administrador de interfaz gráfica , todas las interfaces anteriormente descritas también implementan el observable que es donde se conectan los eventos y la interfaz IScreen implementa la interfaz IObserver por tanto cada componente

dispara su respectivo evento y la implementación de IScreen que se tenga los recibe y maneja en dependencia d la implementación concreta que se
haga.

Ingeniería de software:

Para mantener el código lo más limpio y moldeable posible aplicamos las técnicas estudiadas en clase para
este fin como los principios SOLID. Patrones de diseño como el ya mencionado patrón observer y factory.
El patrón observer consta de dos partes,el observable,que es el objeto el cual,al cambiar su estado notifica a un Observer
que es la otra componente de este patrón , cuando el estado interior de un observable cambia , este notifica a su respectivo observer
el cual se encargara de manejar dicho evento en correspondencia , un observable tiene los métodos attach el cual recibe un observer
y se “suscribe” a este , dettach hace todo lo contrario recibe un obsetver y lo desuscribe de este , a través del método notify es que
el observable le indica al Observer que se cambió un estado llamando a su método Update y pasándole el parámetro que requiera.
Otro patrón que utilizamos fue el patrón Factory el cual se encarga de abstraer la construcción de un objeto en dependencia de unos
parámetros específicos , en este caso utilizamos esto para la creación de los componentes según un archivo de texto , la clase Builder
se encarga de tomar las configuraciones de un juego y crear cada implementación concreta de sus componentes , inicializar un Manager
con estos y conectar el juego a la parte gráfica , por su parte la clase PlayerBuilder se encarga de dados unos parámetros de nombre y tipo
construir cada jugador por separado . Utilizando esto creamos un archivo de configuración siguiendo un formato especifico mediante el
cual se puede reconstruir el juego guardado en otro momento

Parte Grafica

Para la creación de la parte gráfica utilizamos Blazor,html,css y javascript , a través de un Objeto que implementase IScreen recibíamos
los cambios internos del estado del juego y los hacíamos reflejar en la pantalla , a través de html dimos estructura a las diferentes páginas
que componen la interfaz gráfica y marcamos sus estilos con css , javascript fue utilizado para la modificación de DOM

Implementaciones de domino hechas

Entre las implementaciones que hicimos están :
Mesas:
mesa normal: la típica mesa de dos salidas que se usa en la mayoría de juegos de domino
mesa de cartas: mesa que se utiliza para el juego del Shangai o uno
mesa de longana: una mesa que tiene la implementación de las reglas de la longana para jugar fichas
mesa múltiplo de 5: en ésta mesa el jugador gana puntos si la suma de las fichas que están en las cabezas es múltiplo de 5.

Pasadores de Turnos:
Ordenado: pasa los turnos de manera consecutiva siguiendo un orden (uno detras del otro)
Aleatorio: pasa los turnos de manera aleatoria
Turnos Juego de Cartas: implementa las reglas de pasar turnos del shangai donde si se juega una ficha con valor 2 ,3 o 5 se cambia el orden de los turnos

Condiciones de ganar:
normal: gana el primero en quedarse sin fichas o el que menos puntos tenga cuando no haya mas jugadas disponibles
multiplo de 5: gana el jugador que más puntos acumule.
Shangai: contiene las reglas de ganar del shangai donde gana el primero en quedarse sin fichas o el que menos fichas tenga cuando se acaben las fichas de la banca (las que están afuera)
Robaito: gana el primero en quedarse sin fichas o el que menos puntos tenga al quedarse sin fichas la banca

Tipos de fichas
Numéricas doble nueve y doble 6: fichas cuyas caras son números típicas fichas de domino su valor es la suma de los números en sus caras
cartas : fichas que contienen una carta se enganchas unas con otras si tiene el mismo número/letra o el mismo símbolo
fichas de colores: fichas cuyas caras son colores se enganchan si tiene alguna cara del mismo color

Tipos de jugadores:
Jugador Aleatorio: selecciona una ficha de entre las fichas jugables de su mano de forma aleatoria y la juega
Jugador Botagorda : selecciona la ficha con más valor de entre sus fichas jugables
Jugador Euristico : Calcula las caras de fichas que haces al resto de jugadores pasarse de turno y juegas la que mas probabilidad tenga de hacer a otro pasarse

Decoradores para jugadores:
Jugador Robaito: agrega a un jugador de cualquier tipo la posibilidad de robar una carta hasta que pueda jugar o hasta que el banco se quede sin cartas en cada turno

Jugador Shangai : agrega al jugador las propiedades de un jugador de shangai como robar una ficha en cada turno si no puede jugar robar cartas del banco si se juega alguna carta como el 2 o el 3 y jugar continuamente mientras tenga cartas jugables si juega algún 7

Juegos pre-creados:
Shangai: uniendo la mesa de cartas y los pasadores de turno y condición de ganar del shangai se crea este juego utilizando las caras como tipo de fichas

Robaito utilizando las fichas numéricas el pasador de turno organizado y la condición de ganar del Robaito se crea este juego

Juego aleatorio: utilizando las fichas aleatorias la mesa normal , el pasador de turnos aleatorio y cualquier condición de ganar se crea un juego completamente loco

Conclusiones

Trabajando en este proyecto aprendimos de los patrones de la programación orientada a objetos , cómo hacer código fácil de mantener y robusto a cambios , la forma correcta de abstraer y separar las distintas implementaciones de los componentes de un problema que queramos modelar a través de datos y como unir distintas tecnologías como fueron las aplicaciones de consola y frameworks de programación web como blazor , html css javascript que fueron muy útiles en la creación de la parte grafica