
PROYECTO 1: BLOKUS

201731241 – Daniel Albizurez Alpirez
2991470610101

Resumen

Blokus es un juego de mesa de estrategia para dos o 4 jugadores, el cual consiste en colocar piezas de distintos colores en un tablero con la intención de cubrir la mayor parte posible del tablero con sus piezas mientras se intenta bloquear o restringir los movimientos de los otros jugadores.

La regla principal del juego es que las piezas de un mismo color no pueden tocar ninguno de sus lados, solamente sus vértices.

El proyecto trata de la simulación del juego utilizando el lenguaje de programación Python, modelando los conceptos del juego con la implementación de una matriz dispersa como representación del tablero y las piezas colocadas, junto a la representación gráfica por medio de la librería PyQt. Adicionalmente se utilizó el lenguaje de etiquetas XML para generar archivos de almacenamiento y recuperación de partidas.

El proyecto se desarrolló siguiendo los criterios establecidos por el paradigma de Programación Orientada a Objetos (POO).

Abstract

Blokus is an abstract strategy board game for two to four players, where the main goal is to occupy most of the board with their pieces. The main rule of the game is that the edges of each coloured piece played must not touch any edges of a same color piece.

The project simulates the game, using Python as a programming language, modelling the game's main concepts by using a sparse matrix alongside graphic representation thanks to Python's PyQt library. XML language was used to save and open instances of the game by saving them on a structured file.

Project was designed following Object Oriented Programming's main conventions.

Keywords

Python, Sparse Matrix, PyQt, XML

Palabras clave

Python, Matriz Dispersa, PyQt, XML

Introducción

La aplicación consiste en un juego el cual cuenta con un tablero de $m \times n$, definido por el usuario y representado a través de una matriz dispersa, el juego contará con seis diferentes piezas las cuales podrán ser colocadas a lo largo del tablero de juego, tomando en cuenta las restricciones de movimientos existentes.

Desarrollo del tema

1. El juego

El juego está diseñado para ser jugado por 2 personas por partida, al inicio del juego cada jugador podrá escoger un color (No se pueden repetir colores) y un nickname:

- a. Azul
- b. Rojo
- c. Amarillo
- d. Verde

El tablero cuenta con medidas $m \times n$ las cuales son especificadas por los jugadores al principio de una partida.

Se juega por turnos, en cada turno el jugador tendrá la oportunidad de colocar una pieza elegida de manera aleatoria, eligiendo las coordenadas donde desee agregar la pieza, si la coordenada escogida es inválida, por encontrarse ocupada, se de menor tamaño que la pieza o encontrarse junto a otra pieza del mismo color, el jugador tendrá la oportunidad de elegir una nueva coordenada para la pieza, si la segunda coordenada es inválida entonces el jugador perderá su turno.

El juego termina cuando ya no puedan realizarse más movimientos, esto puede

suceder por que el tablero ya no pueda aceptar más piezas o porque durante 3 turnos consecutivos ningún jugador ha logrado colocar una pieza.

El jugador que haya logrado colocar más piezas será el ganador de la partida.

Durante cualquier momento de la partida, esta puede ser guardada en un archivo elegido por el usuario.

2. Las piezas:

El juego provee a los jugadores con 6 piezas diferentes, las cuales le son dadas de manera aleatoria durante una partida. Las piezas son:

- a. Pieza formada por cuatro cuadros verticales, con un cuadro horizontal saliendo hacia la derecha de estos, en forma de una L
- b. Pieza formada por cuatro cuadros verticales, con un cuadro horizontal saliendo hacia la izquierda de estos, en forma de una L invertida
- c. Pieza formada por cuatro cuadros horizontales en línea recta
- d. Pieza formada por cuatro cuadros los cuales forman un cuadrado de 2×2
- e. Pieza formada por cuatro cuadros horizontales inferiores y dos cuadros horizontales sobre ellos, similar a una pirámide
- f. Pieza formada por cuatro cuadros verticales en una línea recta

3. Almacenamiento de partida:

El estado de la partida actual puede ser almacenada en cualquier momento de esta, esta es almacenada dentro de un archivo con extensión y estructura XML.

4. Matriz dispersa:

Una matriz dispersa es un tipo de estructura de datos, más específicamente una matriz de

medidas $m \times n$ en la cual la mayoría de sus elementos son 0, se encuentran vacíos o son inexistentes.

En programación una matriz es almacenada como un arreglo bidimensional, a cuyos datos se pueden acceder por medio de dos índices correspondientes a las “filas” y “columnas” de la matriz, cuando se trata de una matriz dispersa en cambio los espacios de la matriz no existen, hasta el momento en el que se desea almacenar un valor en esas coordenadas.

Existen muchas maneras de representar una matriz dispersa en este caso se optó por una representación basada en nodos los cuales cuentan con apuntadores hacia los nodos que los rodean, el valor numérico de la fila y columna a la que pertenece y la información que almacena. A la vez y para facilitar el acceso a los nodos la matriz dispersa cuenta con listas que almacenan cada columna/fila.

5. Python:

Python es un lenguaje de programación cuya filosofía se centra en la legibilidad del código. Se trata de un lenguaje de programación multiparadigma, ya que soporta la orientación a objetos, programación imperativa, modular y programación funcional. Es un lenguaje interpretado lo que le permite ser un lenguaje multiplataforma (Es posible ejecutar programas desarrollados en este lenguaje en cualquier sistema operativo).

Administrado por la Python Software Foundation. Cuenta con una licencia de código abierto, denominada Python Software Foundation License.

Python cuenta con varios principios que

definen su Filosofía, descritos por el desarrollador de Python Tim Peters:

- a. Bello es mejor que feo.
- b. Explícito es mejor que implícito.
- c. Simple es mejor que complejo.
- d. Complejo es mejor que complicado.
- e. Plano es mejor que anidado.
- f. Disperso es mejor que denso.
- g. La legibilidad cuenta.
- h. Los casos especiales no son tan especiales como para quebrantar las reglas.
- i. Lo práctico gana a lo puro.
- j. Los errores nunca deberían dejarse pasar silenciosamente.
- k. A menos que hayan sido silenciados explícitamente.
- l. Frente a la ambigüedad, rechaza la tentación de adivinar.
- m. Debería haber una —y preferiblemente solo una— manera obvia de hacerlo.
- n. Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.²³
- o. Ahora es mejor que nunca.
- p. Aunque *nunca* es a menudo mejor que *ya mismo*.
- q. Si la implementación es difícil de explicar, es una mala idea.
- r. Si la implementación es fácil de explicar, puede que sea una buena idea.
- s. Los espacios de nombres (*namespaces*) son una gran idea ¡Hagamos más de esas cosas!

6. PyQt:

Se trata de una biblioteca gráfica basada en la biblioteca gráfica Qt, originalmente desarrollada en y para el lenguaje de programación C++, la cual permite la implementación de interfaces gráficas utilizando código escrito en Python para una mejor representación de los procesos desarrollados en el lenguaje.

7. La solución:

Para la solución de este proyecto se desarrolló una matriz dispersa utilizando listas de cabecera para las filas y columnas, las cuales almacenan a su vez los nodos que contienen la información almacenada en la matriz.

En la representación gráfica se optó por una tabla donde se colocan y pintan las piezas que son a su vez almacenadas en la matriz según el jugador en turno al momento de colocarlas.

Para almacenar las partidas se utilizó el lenguaje XML y los métodos provistos por la librería Element Tree