Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2

Inga. Claudia Liceth Rojas Morales
Tutor de curso: Monica Raquel Calderon Muñoz

PROYECTO 1

OBJETIVO GENERAL

Se busca que el estudiante sea capaz de desarrollar una solución integral que implemente tipos de datos abstractos (TDA) bajo el concepto de programación orientada a objetos.

OBJETIVOS ESPECÍFICOS

- Implementar POO para el desarrollo de la solución a través de lenguaje Python
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Utilizar listas ordenadas para implementar una matriz dispersa utilizando memoria dinámica.
- Visualizar TDA'S por medio de la herramienta Graphviz.
- Utilizar archivos XML como insumos para la lógica y comportamiento de la solución.
- Utilizar código HTML para mostrar los resultados de la información obtenida y trabajada.

DESCRIPCIÓN GENERAL

La aplicación consiste en un juego el cual contará con un tablero de mxn representado a través de una matriz dispersa, dicho juego contará con seis diferentes piezas las cuales pueden ser colocadas a lo largo del tablero de juego, tomando en cuenta las restricciones de movimientos que serán especificados a lo largo del enunciado.

IMPLEMENTACIÓN

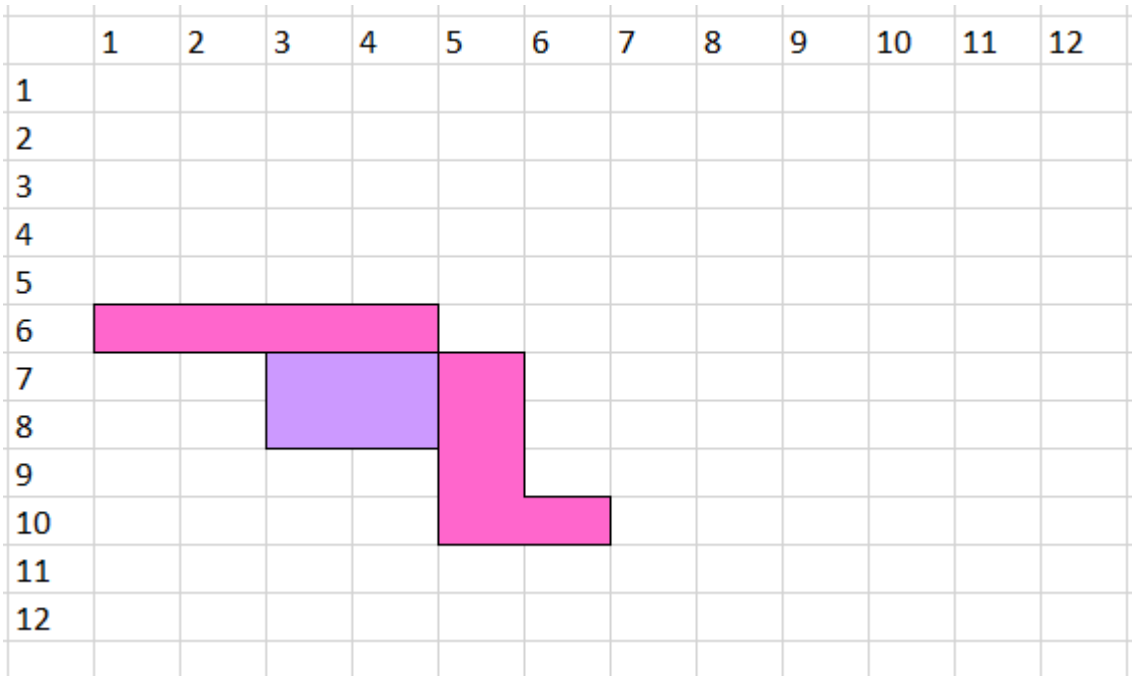
MODO DE JUEGO

Únicamente podrán jugar 2 personas por partida; Al inicio del juego, cada jugador podrá escoger entre cuatro diferentes colores:

- Azul
- Rojo
- Amarillo
- Verde

No se puede repetir color entre los jugadores de una misma partida. Así mismo, se debe de poder almacenar un nombre o alias como identificador para que, dado sea el caso no se pueda continuar con la partida, pueda almacenarse en un archivo XML y pueda ser cargada desde el punto del último movimiento.

El tablero será representado a través de una matriz dispersa, el tamaño del tablero será indicado antes de iniciar la partida. El tablero puede o no ser cuadrado.



Cada una de las posiciones será representada por las coordenadas (x,y).

En cada turno, las piezas de juego saldrán de manera aleatoria al jugador. Previo a su movimiento, el jugador conocerá la pieza por colocar e ingresará la posición (x,y) la cual será la posición de inicio de la pieza; Si dicha posición está ocupada, o bien la pieza es lo suficientemente grande como para necesitar un espacio que ya está ocupado, saltará un mensaje de error y permitirá una única vez el cambio de coordenadas; Si la segunda coordenada ingresada es errónea, el jugador pierde automáticamente su turno.

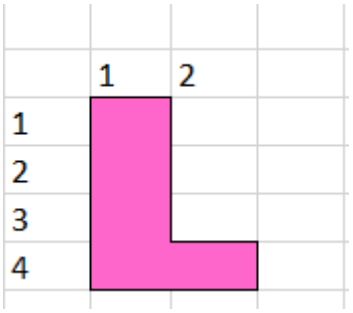
Para cada turno, el jugador contará con un tiempo estipulado para realizar su movimiento, al momento que este tiempo sea cumplido, automáticamente perderá su turno. Si el jugador realiza su movimiento antes que el tiempo sea cumplido, presionará un botón que indicará que el turno ha finalizado; Se reiniciará el contador de tiempo y pasará el mando al otro jugador.

El juego terminará cuando alguno de los dos jugadores ya no posee piezas, o bien, el tablero no permita más movimientos. Gana el jugador con la mayor cantidad de piezas colocadas.

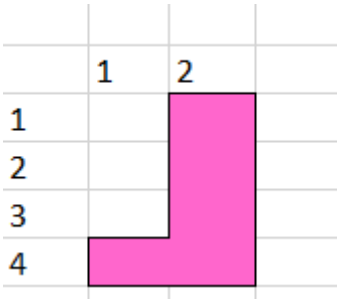
Tanto el tiempo como la cantidad de piezas serán una variable que se declarara como condición inicial del juego; Se debe de tomar en cuenta que el tiempo no debe de ser mayor a 60 segundos, para las piezas no hay restricción de cantidad.

PIEZAS DE JUEGO

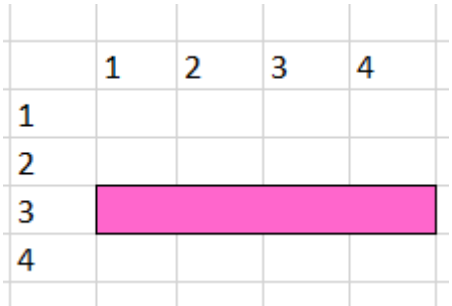
1. Pieza formada por cuatro cuadros horizontales y, saliendo del cuarto cuadro, se formará una línea vertical de dos cuadros hacia la derecha, formando una L .



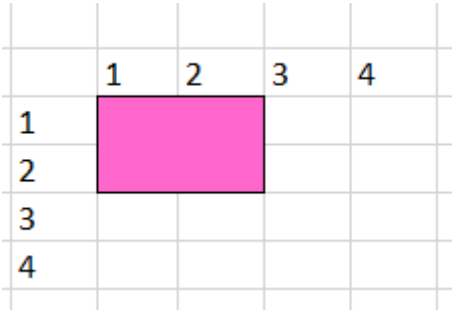
2. Pieza formada por cuatro cuadros horizontales y, saliendo del cuarto cuadro, se formará una línea vertical de dos cuadros hacia la izquierda, formando una L invertida.



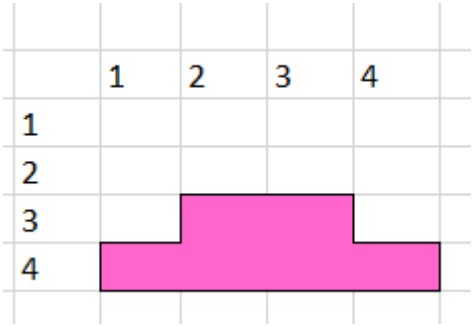
3. Pieza formada por cuatro cuadros horizontales



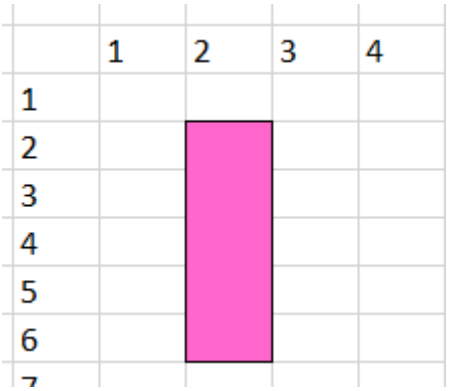
4. Pieza formada por cuatro cuadros formando un cuadrado de 2x2



5. Pieza formada por cuatro posiciones formando una línea horizontal y, sobre esas cuatro posiciones, dos cuadros formando una línea horizontal a manera de simular una pirámide.



6. Pieza formada por cuatro cuadros verticales



RESTRICCIONES DE MOVIMIENTO

1. Las piezas de un mismo jugador únicamente pueden tocar sus vértices, es decir, no pueden apoyar ninguno de sus lados, tan sólo deben de estar en contacto con alguna (o varias) de sus esquinas.
2. Las piezas del jugador contrario si pueden tocar los vértices como los lados de las piezas del jugador rival.
3. Si un jugador no puede colocar una pieza, pierde su turno.

GUARDANDO PARTIDA

El estado de la partida podrá ser guardado en archivos para poder retomar la partida a partir de él. Los archivos consistirán en archivos con extensión y estructura xml en el cual se limitará a utilizar únicamente las etiquetas:

- **matrices:** este será necesario para la lectura inicial del archivo, ya que será la etiqueta padre de todo.
- **matriz:** esta etiqueta será la que indica que una nueva matriz será creada para su respectivo análisis y únicamente puede estar dentro de la etiqueta matrices y puede tener los siguientes componentes hijos:
    - **nombre**: este contendrá el identificador de la partida (se deberá validar la existencia de matrices con el mismo nombre, para mantener la consistencia de los datos).
    - **fila**: será el número de filas que tendrá la matriz.
    - **columna:** será el número de columnas que tendrá la matriz.
    - **imagen**: esta etiqueta únicamente podrá estar dentro de la etiqueta matriz y contendrá los valores respectivos a cada celda de la matriz, consiste en usar cadenas de caracteres; donde el carácter “1” representa una celda con datos del jugador 1, el carácter “2” representa una celda con datos del jugador 2 y el cambio de fila en la matriz se identificará mediante el carácter salto de línea (\n). Además, se utilizará el carácter “-” para representar los espacios en blanco.

REPORTE EN HTML

Se debe poder visualizar en un reporte HTML la cantidad de partidas ganadas con los respectivos datos de cada jugador, movimientos hechos incluyendo los movimientos restringidos con sus respectivos mensajes de error.

También se debe poder visualizar a través de graphviz el estado final del tablero

VENTANA PRINCIPAL

La ventana principal será la forma visual a través de la cual se llevará a cabo el juego; queda a discreción del estudiante la disposición de cada uno de los componentes, sin embargo se calificará tanto el diseño como la facilidad de manejo.
Entre los componentes obligatorios los cuales la aplicación debe contar son:

1. **Barra de menú**
   Lo cual contará con las opciones básicas de un menú tales sean:
    - Abrir partida
    - Guardar partida
    - Ayuda
2. **Inicio de juego**
3. **Tablero**
4. **Reportes**
5. **Panel de juego**
   El cual contará con las opciones tales como:
    - Ingresar coordenadas para agregar la pieza al tablero
    - Cantidad de puntos acumulados
    - Pieza por colocar

- Contador de tiempo
- Fin de turno

CONSIDERACIONES

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma Github en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar mínimo 4 releases o versiones del proyecto. Se deberá agregar a su respectivo auxiliar como colaborador del repositorio. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.
Para la realización de la interfaz gráfica queda a discreción del estudiante que librería utilizar.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Este informe debe expresar con claridad el diseño de objetos ideado para resolver este proyecto por lo que debe expresar el diagrama de clases y los diagramas de actividades de los algoritmos más importantes.

Debe de tomar en cuenta que es un ensayo formal, por lo que se calificará tanto la redacción, como ortografía y presentación.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Se deberá utilizar una lista para el manejo de las matrices. Está lista debe ser creada completamente por el estudiante mediante clases. Caso contrario, se penalizará con el 100% de la nota.
- Uso obligatorio de programación orientada a objetos (POO).
- El nombre del repositorio debe de ser IPC2_Proyecto1_#Carnet.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Se calificará la versión del cuarto release o del último release realizado previo a la fecha de entrega. No se calificará dado se dé el caso que existan modificaciones de código en fechas posteriores a la entrega.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- De no existir una forma gráfica de poder validar el funcionamiento de la aplicación se penalizará con el 100% de la nota.
- **COPIAS TOTALES O PARCIALES SERÁN REPORTADOS A LA ESCUELA Y OBTENDRÁN NOTA DE 0 PUNTOS.**
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el día sábado 19 de junio antes de las 23:59.
- La entrega será por medio de la UEDI.
- Se deberá crear un repositorio con nombre IPC2_Proyecto1_#carnet.