

Proyecto: el juego final

El trono del Jokander: el juego final

Descripción

Existe un reino, *Jokander*, con un trono que los jugadores quieren obtener. Un jugador debe explorar el territorio para obtener puntos y ganar el reino. El reino, dividido en *territorios*, se puede visualizar como un tablero de dimensión $N \times N$, para acceder a un territorio se especifican las coordenadas del tablero, por ejemplo 1-1, 5-4, etc.

El juego consiste en que el jugador “visita” un territorio por sus coordenadas y obtiene o pierde puntos; también puede ser que llegue a un territorio donde está el emergente ganador o perdedor, que decreta esa condición y el juego termina. Hay un límite de intentos para lograr el objetivo. Cuando un juego ha concluido (ganado o perdido) el jugador decide si quiere jugar otra partida o si termina su participación.

El programa es para un jugador, en el futuro si su juego tiene éxito, usted puede ser contactado para que lo desarrolle para competencias de múltiples jugadores.

¿Qué hace el programa?

1. Genera un tablero de dimensiones $N \times N$ para representar el territorio del reino. N es un número aleatorio entre 8 y 10. El jugador nunca sabrá que en Python estas coordenadas empiezan en cero. Para ver el contenido del tablero, vea los detalles en [Aspectos técnicos](#).
 - 1.1. El programa debe saber cuántos puntos positivos y cuantos puntos negativos tiene el tablero, porque se usarán para determinar si se gana o se pierde.
2. Genera un número aleatorio entre 12 y 18, que son los intentos para encontrar el trono Jokander
3. Despliega un resumen de las indicaciones y reglas del juego, como se gana o como se pierde. También le dice que puede repetir el juego las veces que quiera. Le explica como seleccionar las coordenadas.
4. Le pide el nombre al jugador.
5. Despliega el “mapa del terreno” que, al inicio se ve como en la [figura 1](#).
6. Permite al jugador seleccionar las coordenadas en las que quiere explorar. Por supuesto su programa es muy robusto y valida que las coordenadas sean números y estén dentro de los límites del territorio.
7. Revisa el contenido localizado en esas coordenadas e informa lo que halló:
 - 7.1. si ganó puntos los acumula
 - 7.2. si perdió puntos los acumula,
 - 7.3. si encontró un emergente, y si ganó o perdió el juego

- 7.4. Si un jugador ya visitó esas coordenadas, le informa la situación, descuenta un intento y no hay cambio en los puntos.
8. Revisa si hay gane o pérdida del juego
 - 8.1. Se **gana** si
 - 8.1.1. un jugador obtiene 50% de los puntos positivos del tablero
 - 8.1.2. o si localizó el territorio emergente del trono Jokander
 - 8.2. Se **pierde** si
 - 8.2.1. ya acumuló 50 puntos negativos
 - 8.2.2. si localizó el emergente perdedor
 - 8.2.3. Si se acabaron los intentos
 - 8.3. Muestra el estado de visitas hechas, ver [figura 2](#)
9. Si el jugador tiene intentos repite del paso 5 al 8. Sino le pregunta si quiere seguir jugando, y repite del paso 1 en adelante.
10. Si el jugador no quiere jugar más entonces en el archivo graba un registro con el nombre del jugador, fecha y hora del juego, los puntos totales (positivos menos negativos) y si ganó por emergente. En este caso solo guarda la última partida jugada.

Note que el proyecto es muy sencillo en la interacción con el jugador, lo más fuerte es la programación del juego en sí mismo

Aspectos técnicos

1. El equipo define técnicamente en el documento de análisis del problema, si usarán listas, tuplas o diccionarios, incluso pueden utilizar una o más de esas estructuras. **Justifican** porqué.
2. Considere estos aspectos para llenar el tablero con los datos:
 - a. Recuerde que puede utilizar el operador * que repite. Para más detalles revise la presentación de 102 Secuencias Listas que está en el TEC Digital.
 - b. El tablero debe tener:
 - i. el 30% de los territorios con premios de 5 puntos, un 15% con premios de 10 puntos, un territorio con el ganador emergente que gana con solo conseguirlo.
 - ii. el 30 % de los territorios con cuevas con las cuales se pierden 5 puntos, un 15% con cuevas que hacen que pierda 10 puntos, un territorio con el perdedor emergente que pierde el juego con solo llegar ahí.
 - iii. El resto de los territorios quedarán con cero puntos: no altera el puntaje.
 - iv. Para hacer esto: piense muy bien la estructura de datos que usará, así como el algoritmo para que sea eficiente al distribuir los premios, cuevas y emergentes y que todos ellos queden distribuidos en el tablero. Debe evitar que haya grandes acumulaciones o bloques del territorio con cuevas o premios.
3. Hace la validación de datos, el manejo de errores y excepciones

4. Cuando el jugador termina una partida, en el archivo **Juegos.dat** graba: nombre del jugador, la fecha y hora del sistema, los puntos ganados, el acumulador de puntos perdidos, el estado: ganó o perdió, y si fue por puntos o por emergente. Como el archivo podría tener datos grabados, solo agrega el nuevo registro al final.

¿Cómo abordar el proyecto y tener éxito?

1. Empiece poco a poco y con **anticipación**. Aprenda de las lecciones que le han dejado los proyectos anteriores.
2. Inicie por **comprender el problema** y expresarlo con sus propias palabras, no aborde el proceso de programación si no ha logrado este paso, de lo contrario podría implicar que pierda tiempo y no avance en la solución.
3. **Analice** pensando en las entradas, salidas y restricciones del caso y posteriormente el proceso.
4. Analiza opciones de **estructuras de datos** para guardar la información. En este ejercicio esto es de suma importancia, ya que, si empieza a programar y luego debe cambiar la estructura de datos, esto puede generar un alto consumo de tiempo. *Es como empezar una construcción y luego deba botarla para volver a empezar.*
5. Este problema se resuelve mejor si aplica la técnica **“divide y vencerás”** y hace funciones con propósito específico que puedan ser utilizadas varias veces, posiblemente las funciones reciben parámetros y retornan un valor. **Puede programar funciones y probar su correcto funcionamiento.**
6. Desarrolle un programa principal (**main**) que articula el funcionamiento del programa y llama a las funciones según lo requiera. Revise el paso de parámetros y valores de retorno.
7. Permita el ingreso de datos por parte del usuario. El alambrear el programa, o sea que el programador define previamente los valores para que el programa funcione implica un rebajo del 80% de la nota de su tarea.
8. Sea claro a la hora de pedirle una entrada de datos al usuario. Cuide la redacción y ortografía.
9. Aplique las normas de programación estudiadas en clase, por ejemplo, con nombres de variables y de funciones, uso de comentarios, eficiencia entre otros.

Aspectos administrativos

1. Trabajo en parejas.
2. Lugar de trabajo: **Teams**, en el canal de cada grupo, en archivos. Conversaciones, archivos word y archivos .py, así como otros que requieran:
 - a. de texto con el análisis del problema, entradas, salidas, limitaciones, restricciones y explicación del proceso, indican si la solución es de autoría de los integrantes del grupo o si recibieron ayuda externa. Cómo usan de la técnica **divide y vencerás**
 - b. archivo con el programa en Python. Guarde versiones del programa Python, para que muestre el avance que van teniendo.
 - c. puede usar el chat y puede hacer por este medio las reuniones de grupo
 - d. Se evalúa el desarrollo a lo largo del tiempo asignado, hay penalización si no muestra la evolución del trabajo.
 - e. Puede agregar imágenes que representen la solución
3. Los estudiantes defenderán su proyecto en una sesión sincrónica con la profesora o con los asistentes, de aproximadamente 45 minutos. Ambos integrantes participan y encienden la cámara. Luego se dan los detalles, se publican los horarios para que cada grupo reserve su espacio. Esta actividad es un Quiz con calificación individual.
4. **¿Cómo se entrega?** Por el **TEC Digital en Evaluaciones** entrega
 - a. en **Proyectos**, Proy Final, hace dos entregas
 - i. el link al canal de Teams del equipo
 - ii. programa en Python
 - b. Si después de la fecha de entrega hay cambios en el canal de Teams se invalida el Miniproyecto con nota cero para los estudiantes.
5. Fecha de entrega: martes 15 de noviembre de 2022 a las 23:50 horas. No se aceptan entregas tardías ni se extenderá el tiempo para su realización, excepto por situaciones extremas que la profesora valorará.
6. El Proyecto debe ser inédito y de autoría de los integrantes del grupo. Si se detecta intento o confirmación de fraude se aplica lo que indica el Reglamento del Régimen de Enseñanza Aprendizaje del TEC.
7. Hay dos rúbricas para evaluar: una de la defensa y otra de la programación, uso de Teams, análisis del problema y demás aspectos.

Figura 1: Terrero sin marcas (solo muestra el recuadro y unas líneas que dan la idea de las coordenadas)

	1	2	3	4	5
1					
2					
3					
...					

Figura 2: ejemplo de marcas en el terreno cuando el jugador ha visitado territorio. Indica el valor que tiene cada territorio visitado. Si el territorio no ha sido visitado, no muestra nada. Las líneas son solo una idea para que el jugador recuerde las coordenadas visitadas.

	1	2	3	4	5
1		5			
2			-10		0
...			5		

Acumulado de puntos: xxx

Acumulado de puntos perdidos: xxx

Intentos totales: ---- intentos restantes: ____