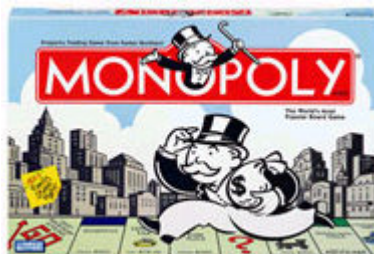


Instituto Tecnológico de Costa Rica	Tercera Tarea Programada
Departamento de Computación.	MINIPOLY
CE-1401 Taller de Programación	Prof: Ing. Jeff Schmidt Peralta
II Semestre 2014 5 de noviembre de 2014	Consultas: grupo facebook correo: jschmidtcr@gmail.com

1. Introducción.

El juego del Monopoly fue inventado y patentado por Elizabeth Phillips (Elizabeth Magie) en 1904. La patente expiró en 1922 y durante esos años se convirtió en un juego popular en muchos hogares de Atlantic City. Charles B. Darrow, a quien suele atribuírsele la invención (hacia 1933) simplemente lo redescubrió, modificó ligeramente, repató y vendió a Parker Brothers en 1934. Parker Brothers, ahora propiedad de Hasbro ha mantenido siempre la versión de que fue Darrow el inventor, tal vez para darle un toque interesante a la historia del juego.



En la presente tarea se va a realizar una herramienta MUY básica llamada MINIPOLY basada en algunos conceptos generales del Monopoly pero adaptado a un nivel muy sencillo utilizando el paradigma de Programación Orientada a Objetos.

Otro concepto de importancia para un *futuro ingeniero* es desarrollar la capacidad de **trabajar** en grupo (en nuestro caso 2 personas), en el cual tenga definidas funciones, roles y responsabilidades con el objetivo de conseguir el éxito del proyecto.

2. Descripción Funcional del Programa.

Minipoly es un juego de mesa (ahora en su versión Costa Rica para PC!!!) que utiliza un tablero con 24 casillas a sus lados que muestran sitios turísticos de Costa Rica, los cuales pueden ser adquiridos por un jugador.

Existe un banco (el programa) que maneja las transacciones de compra y administra los pagos de multas o beneficios para el jugador. (ver *ojalá jugar* juegos similares como Monopoly o Gran Banco)

La presentación de la interfaz con el usuario es **totalmente libre**, puede ser cualquier componente que se desee utilizar y será un elemento importante dentro de la calificación del proyecto.

La mecánica del juego es la siguiente: para iniciar se hace un lanzamiento de un dado para determinar el orden de lanzamiento. En caso de empate se vuelve a lanzar el dado hasta definir un orden.

Según el orden definido cada jugador lanza el dado y avanza el número de casillas indicado. En caso de llegar a una posición ocupada por una propiedad puede ser que esté a la venta (por el Banco) o pertenezca a otro jugador. Si está a la venta, el jugador puede decidir entre comprarla (deberá pagar al Banco el monto respectivo) o no comprarla. Si la propiedad ya ha sido comprada, se deberá pagar un monto al dueño por derecho de paso según se indique en la propiedad.

El juego no tiene un final como tal, el ganador está definido por la persona que posea más dinero entre efectivo y propiedades y puede ser determinado en cualquier momento. A continuación se muestra un ejemplo de cómo podría ser el “tablero” del juego.



Las casillas 6 y 18 son especiales y cuando un jugador llega a una de esas posiciones se obtiene una carta de la baraja de sorpresas, que indicará un monto a recibir, a pagar o el traslado a alguna posición dentro del tablero. La casilla 12 o cárcel hace que el jugador pierda un turno y además tenga que pagar una multa definida.

Además la casilla 0 también es especial. Además de ser el punto de inicio de los jugadores, cada vez que un jugador pasa sobre ella (completa una vuelta al tablero) recibe un monto especificado por configuración.

Modelo de objetos.

Pueden existir algunos métodos que se requieran y no estén descritos en la presente definición. Pueden realizarse modificaciones al modelo de objetos enunciado, siempre y cuando las mismas se documenten.

Se modelarán tres objetos para definir el juego: banco, jugador y propiedad. A continuación se describen los datos y métodos *básicos* de cada uno.

Banco.

Datos

Número de juego: consecutivo de cada partida.

Juego: vector de 24 posiciones conteniendo el estado actual de la partida.

Baraja: lista conteniendo cada elemento de la baraja sorpresa.

Monto: cantidad de dinero manejada.

Dado: valor actual del dado

Métodos

Iniciar juego: pide los datos para iniciar un nuevo juego.

Carga baraja: lee el archivo de baraja y carga en forma totalmente aleatoria las cartas existentes.

Carga configuración: carga del archivo de configuración los parámetros del juego.

Jugada: obtiene el resultado de “tirar” el dado

Comunicar resultado: “informa” a los jugadores el valor obtenido del lanzamiento del dado.

Realizar cobro: cobra los montos correspondientes a la venta de propiedades y multas.

Realizar pago: paga a los jugadores.

Determina ganador: revisa los acumulados de efectivo y propiedades de los jugadores y determina el ganador.

Jugador.

Datos

Nombre: nick del jugador.

Turno: posición para tirar.

Acumulado: valor que posee el jugador (se inicializa en cero).

Posición: casilla actual.

Propiedades: lista de propiedades que posee el jugador.

Métodos

Realiza compra: permite al jugador comprar una propiedad.

Pagar por paso: permite pagar a otro jugador por el derecho de pasar por una propiedad.

Pagar multas: permite pagar multas

Cobrar: permite cobrar por derechos de paso o bonificaciones de la baraja sorpresa.

Cada uno de los posibles jugadores (de 2 a 4) será una instancia del objeto jugador, que se creará dinámicamente al iniciar cada juego.

Propiedad.

Datos

Descripción: nombre de la propiedad.

Posición: posición en el tablero.

Valor: monto de compra.

Derecho de paso: valor a pagar cuando la propiedad ya ha sido adquirida.

Estado: indicador si la propiedad está libre (inicial) o comprada. Debe actualizarse cada vez que se inicia un juego.

Métodos

Cambia estado: cambia de libre a comprada el estado de la propiedad.

Obtener datos: obtiene los datos de la propiedad.

Cuando el programa se carga las instancias del objeto propiedad se deben crear en forma dinámica tomando los datos requeridos del archivo prop.txt que tiene la siguiente estructura:

Tabla: Propiedades

Nombre archivo: prop.txt

Campos: posición, descripción, valor, derecho de paso.

Posición	Descripción	Valor	Derecho paso
1	Hotel Marriot Los Sueños	2,500,000	50,000
2	Hotel Barceló Tambor	1,800,000	45,000

Además van a existir 2 archivos de datos: configuración que va a contener datos de importancia para poder jugar y baraja, que va a contener las opciones de la baraja para los usuarios que caen en las posiciones respectivas.

Tabla: Configuración

Nombre archivo: setup.txt

Campos: monto inicial banco, monto inicial a repartir por jugador, monto x pasar inicio, multa por cárcel.

Monto banco	Inicial x jugador	Monto inicio	Multa cárcel
1,000,000,000	6,000,000	2,500,000	250,000

Tabla: Baraja de sorpresas

Nombre archivo: baraja.txt

Campos: mensaje, acción, valor

Mensaje	Acción	Valor
"Debe pagar una multa de 10000"	0	-10000
"Reciba 20000 como bonificación"	0	20000
"A la cárcel..."	1	12

Si la acción es 0, indica un cobro o pago por parte del banco al jugador, si el valor es negativo, se cobra al jugador. Si es positivo, se paga al jugador. En caso que la acción sea 1, se traslada al jugador a la casilla indicada en la posición definida en el campo valor.

Ayuda interactiva.

Permite obtener ayuda sobre la utilización del programa y sobre las funciones de MINIPOLY.

3. Funciones a investigar.

Para realizar la tarea, debe **investigarse** el uso de algunos algoritmos y funciones. La investigación a realizar debe abarcar:

Utilización de multimedia: integración de animaciones, sonidos y otros.

Generación de números aleatorios (dato)

4. Documentación.

La documentación es un aspecto de gran importancia en la elaboración de programas. Existen 2 tipos de documentación a realizar para esta tarea: la interna y la externa.

La documentación interna se refiere a la inclusión de comentarios en el programa fuente, al menos antes de definir cada función, y en los cuales se explique lo que realiza la función, las entradas, salidas y restricciones consideradas. En las funciones que definen la interfaz gráfica únicamente se debe identificar el componente utilizado.

La documentación externa debe incluir:

- Tabla de contenidos
- Introducción
- Descripción del problema.
- Diagrama de clases: **INVESTIGAR**.
- Análisis de resultados. (incluyendo corridas de ejemplo)
- Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos
- Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción y duración.
- Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas **REALES** invertidas. Ejemplo:

FUNCION	Integrante 1	Integrante 2	TOTAL
Análisis de requerimientos	xx horas	xx horas	xx horas
Diseño de la aplicación	xx horas	xx horas	xx horas
Investigación de funciones	xx horas	xx horas	xx horas
Programación	xx horas	xx horas	xx horas
Documentación interna	xx horas	xx horas	xx horas
Pruebas	xx horas	xx horas	xx horas
Elaboración documento	xx horas	xx horas	xx horas
TOTAL	xx horas	xx horas	xx horas

- Conclusión personal (por cada integrante del grupo).

5. Opcional.

Esta tarea cuenta con un aspecto opcional al cual se podría optar para mejorar la calificación del proyecto programado. Deben documentarse **TODAS** las funciones investigadas y el método utilizado para realizar el punto pedido.

El punto opcional consiste en poder jugar el MINIPOLY en red, es decir, que cada jugador esté en una estación de trabajo. Esto tiene un valor de 15 puntos sobre la nota.

6. Evaluación.

Documentación	15%
Interna	5%
Externa	10%

Resultados (ejecución, eficiencia, manejo correcto de objetos y estructuras planteadas, presentación)

Funciones:

Manejo de la interfaz	20%
Modo de juego	60%
Ayuda	5%

7. Aspectos Administrativos.

- El proyecto está estructurado para ser realizado por **2** personas.
- La tarea se debe entregar hasta el día 20/10/2014 hasta las 11:59 pm, en forma electrónica, en un archivo comprimido con los nombres de los estudiantes, que contenga TODO lo necesario para poder ejecutarla. Al día siguiente se asignará una cita de revisión del programa. **No se aceptarán tareas después de la fecha y hora indicadas.** Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe enviar al correo danielcanessa09@gmail.com
- La defensa o revisión del proyecto es indispensable y deben asistir los 2 estudiantes que componen el grupo. En esta revisión se preguntará sobre aspectos relacionados con funcionalidad, así como sobre el código. Los estudiantes deben mostrar TOTAL dominio de estos dos temas, de lo contrario, el proyecto puede ser considerado como una copia.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- Se debe incluir en el archivo comprimido la documentación solicitada. Debe entregarse en formato electrónico (archivo .doc .odt o .pdf).
- No se aceptarán tareas cuyo archivo sobrepase 2 mb de espacio en disco.

- Se debe adjuntar la documentación solicitada. Debe entregarse en formato electrónico (archivo .doc o .pdf).
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.
- **El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada**

8. Bibliografía.

Documentación técnica de Python

9. Consultas.

Puede dirigir cualquier consulta a jschmidtcr@gmail.com , o al grupo del curso en facebook.