

# Monopoly-morphic

## Objetivos

- Analizar el problema planteado correctamente.
- Diseñar un modelo general de solución utilizando correctamente los conceptos de programación orientada a objetos.
- Demostrar el conocimiento de patrones de diseño de y de principios de diseño utilizándolos donde sea conveniente.
- Seguir las buenas prácticas de programación

## Dominio

## Jugador

Los jugadores del Monopoly-morphic buscarán ganar a sus adversarios. Cada uno buscará o bien lograr que el resto de los jugadores quiebren (se queden sin dinero) o bien tomar suficiente poder en el juego como para considerarse ganador (ver sección [Quiebra, Cárcel y Victoria](#)). La cantidad máxima de jugadores es de 4.

## Tablero



El tablero del Monopoly-morphic es un tablero circular. Esto quiere decir que el punto de partida y de llegada es el mismo, y que al completar la vuelta, se comenzará a recorrer nuevamente desde el inicio.

El tablero consiste de muchos casilleros, de distintos tipos (ver sección Tipos de Casilleros). El tablero cuenta con un punto de partida y llegada, y con una dirección, que es hacia donde se moverán los jugadores durante cada turno.

## Tipos de Casilleros

- **De paso:** No tienen ningún efecto, son solo parte del camino del tablero.
- **De propiedad:** Representan una propiedad “comprable” por los jugadores. Las casillas de propiedad se agrupan por un determinado grupo: barrio, rubro, u otro. Cada propiedad tiene un costo de compra. Un jugador al caer en un casillero de propiedad puede hacer las siguientes acciones:
  - Si no esta comprada por nadie:
    - Puede no comprarlo
    - Puede comprarlo al valor de compra que tenga la propiedad
  - Si está comprada por otro jugador:
    - Tiene que pagarle lo que corresponda al jugador poseedor de dicha casilla el valor. El valor estará determinado por la renta que tiene dicha propiedad en ese momento (que está sujeto a las construcciones que tenga)
  - Si está comprada por el mismo jugador:
    - No ocurre nada
- **De lotería:** Cuando un jugador aterrice sobre esta casilla, cobrará del banco un determinado monto, establecido por la casilla.
- **De multa:** Cuando un jugador caiga en una de estas casillas, deberá abonar al banco el valor de multa establecido por la casilla.
- **Ir a la cárcel:** Si un jugador cae en esta casilla, se lo traslada a la Cárcel y pasa a estar Preso. La condena será de una determinada cantidad de turnos, deberá permanecer en la casilla de Cárcel y tendrá las repercusiones de estar Preso.
- **Cárcel:** Caer en esta casilla no tiene ningún efecto. Siempre debe haber una cárcel en cada tablero, que será única.
- **Llegada/Partida:** Es la casilla única e inicial donde arrancan todos los jugadores. Cada vez que los jugadores vuelvan a pasar por esta casilla, recibirán un determinado monto por parte del banco. Al caer en dicha casilla no se genera ninguna acción sobre el jugador.
- **Estaciones de transporte:** son similares a las casillas de propiedad ya que son susceptibles a ser compradas. Sin embargo, no puede construirse sobre ellas. Por el contrario, su efecto es el siguiente:
  - Luego de adquirir una estación, los demás jugadores deberán abonar al jugador un monto cada vez que caigan en dicha casilla.
  - El monto a pagar se multiplicará por cada casilla de estación de transporte que posea el jugador propietario de dicha estación.
  - Los jugadores que posean alguna estación, están exentos de abonar el monto al caer en la estación de otro jugador.

## Banco y dinero

El principal actor para el desarrollo del juego es el banco.

El banco es quien inicialmente le otorga el dinero base a los jugadores, y luego quien se encarga de repartir dinero por determinadas acciones y a quien se le deberá pagar las inversiones y multas.

Las acciones por las que el banco otorga dinero a algún jugador son:

- Al momento de iniciar la partida.
- Cuando un jugador da la vuelta al tablero, es decir, cruza la línea de llegada.
- Cuando un jugador cae sobre un casillero de cobro.
- Cuando un jugador hipoteca una de sus propiedades.
- Cuando un jugador vende casas/hoteles sobre una propiedad

Las acciones por las que el banco recibe dinero de un jugador son:

- Cuando un jugador construye en un casillero de propiedad, paga el monto de construcción al banco.
- Cuando un jugador paga para mejorar la propiedad en uno de sus casilleros.
- Cuando el jugador cae sobre un casillero de multa.
- Cuando el jugador decide pagar la fianza para salir de la Cárcel.

## Propiedades y Construcciones

### Hipoteca

Cada jugador puede hipotecar sus propiedades. Al hacerlo el banco le paga al jugador el precio de la hipoteca y ese casillero pasa a no tener efecto alguno, esto significa que si otro jugador cae en dicha casilla, no tiene que pagarle nada al jugador poseedor de la propiedad. El jugador en su turno puede des-hipotecar la propiedad pagando la hipoteca al banco. Cada propiedad tiene su propio valor de hipoteca.

### Casas y hoteles

Al completar todo un grupo de propiedades, esto es que un mismo jugador posee todas las propiedades de un mismo color, el jugador puede poner casas y hoteles sobre las propiedades.

El valor de cada casa/hotel depende de la propiedad. La diferencia de casas entre propiedades del mismo grupo no puede ser mayor a 1, lo que significa que una propiedad no puede tener 0 casas y otra 3. Al tener 4 casas en una propiedad, se pueden reemplazar por un hotel pagando el valor del hotel (depende de la propiedad). Cada propiedad tiene un valor que tienen que pagar los otros jugadores al caer en esa casilla, dicho valor aumenta según la cantidad de casas y hoteles, dichos valores también dependen de las propiedades. No se puede poner más de un hotel por casillero y tampoco se pueden tener hoteles y casas en una misma casilla en simultáneo. Si el jugador quiere vender las casas/hotel, lo puede hacer y el banco le pagará el precio de la casa/hotel, depende del casillero y es

necesariamente menor al dinero que invirtió el jugador para hacer la compra original. Las propiedades con casas/hoteles no se pueden hipotecar.

## Quiebra, Cárcel y Victoria

Los distintos estados por los que puede pasar un jugador son:

- **En juego:** el estado base del Jugador. Este es el estado donde va a poder avanzar en el tablero, usar su dinero para realizar acciones y cobrar renta de sus propiedades.
- **Preso:** cuando un Jugador está preso, conserva sus propiedades, pero no podrá realizar acciones en su turno (a excepción de pagar su fianza o terminar el turno) ni cobrar la renta de quienes pasen por sus propiedades. El Jugador puede salir de la Cárcel si obtiene un valor del randomizador mayor al de la cantidad de turnos que le quedan de condena, o bien si decide pagar la fianza al comienzo del turno.
- **Quebrado:** aquellos jugadores que entren en deuda (es decir, saldo negativo) se considerarán quebrados a partir de ese momento. En ese punto, los jugadores ya no participarán del juego y todas sus propiedades serán removidas del tablero.

Se considerará victorioso a aquel jugador que primero logre colocar todas sus posibles construcciones, o bien sea el último jugador en no quebrar.

## Desarrollo e Implementación

### Jugabilidad

El juego consistirá en una partida de 2 a 4 jugadores.

El orden de los turnos de los jugadores se sorteará al comenzar la partida, y se mantendrá circularmente esa secuencia a lo largo de toda la partida.

Durante su turno, el jugador “lanzará” un randomizador, que generará un número al azar. A menos que el jugador esté preso, el jugador avanzará la cantidad de casillas correspondiente a dicho número. De lo contrario, el número servirá para determinar si puede salir o no de la cárcel.

Una vez obtenido el número y/o avanzado la cantidad de casillas correspondientes, se aplicará el efecto correspondiente a la casilla donde caiga el jugador, en caso de haberlo.

Luego de esto, el jugador podrá realizar cualquier de las siguientes acciones ilimitadamente, hasta terminar su turno:

- Terminar su turno
- Construir o reformar en una de sus propiedades
- Vender alguna de sus construcciones
- Hipotecar una propiedad al banco
- Des-hipotecar una de sus propiedades
- En caso de estar en preso:
  - Pagar la fianza

- En caso de estar sobre una casilla de propiedad vacía:
  - Comprar la propiedad

Al terminar el turno, ya no se podrá realizar ninguna otra acción y pasará a ser el turno del siguiente jugador.

## Configuración de partida

Se debe disponer de una clase *Configuración* donde se va a poder configurar todos los parametros del juego, como por ejemplo el monto a pagar al pasar por la casila de salida, la definición del tablero y todas sus casillas, etc.

## Interfaz de usuario

Los mensajes de informacion y de error quedan para ser definidas por los alumnos. Se recomienda utilizar una libreria como [JLine](#) para manejar el input del usuario y mostrar listas de opciones para limitar la cantidad de errores posibles. Como por ejemplo:

1 . A

2 . B

3 . C

*Ingrese numero de la opcion deseada : 1*

*Seleccionado : A*

**Recomendacion:** Tener un modulo que ayude con este tipo de opciones que reciba una lista de opciones y retorna la seleccionada para tener la impresi'ón por consola en un solo lugar.

Cada jugador contará con un nombre y un color. El color se seleccionará de una determinada lista y no podrán repetirse entre los jugadores. Debe haber una pantalla inicial donde se ingresen los datos de cada jugador indicando la cantidad de jugadores.

Todos los casilleros deberán contar con un color distinto dependiendo de su tipo. Debe de quedar claro cuáles propiedades no fueron compradas y cuales sí, indicando el jugador que dispone de dicha propiedad. Además, deben ubicarse las construcciones que haya sobre cada propiedad, del color correspondiente al jugador propietario.

Un jugador debe poder ver o consultar en todo momento el estado del juego y de los otros jugadores.

Cada grupo de trabajo practico debe definir como mostrar el tablero y toda la información de cada casillero (nombre, categoria, precio, dueño, cantidad de hoteles y/o casas, etc). Debe ser de la forma mas clara y consisa posible. No olvidar mostrar en el tablero la posición de cada jugador.

**IMPORTANTE:** La *jubailidad* es un factor clave. El juego de mostrar claramente el panorama actual de cada turno y debe mostrar a los jugadores el resultado de cada acción. Cada acción debe tener una respuesta.

## Informe

Se debe entregar un informe mostrando un diagrama UML de objetos **completo** explicando las decisiones tomadas. El informe debe contener las hipótesis tomadas durante la realización de este trabajo práctico. No olvidar agregar una sección para las conclusiones.

## Formato de la entrega

Para realizar la entrega se debe mandar un mail a [algoritmos3.fiuba@gmail.com](mailto:algoritmos3.fiuba@gmail.com) indicando los integrantes del grupo adjuntando el informe e indicando cual es el branch y el repositorio de la entrega. El repositorio tiene que ser privado. El trabajo práctico debe realizarse utilizando Git y Github. Para realizar la entrega se debe crear un branch llamado "tp-x" (en este caso tp-1) y no se debe modificar mas luego de enviado el mail. Se recomienda utilizar un branch común de trabajo, por ejemplo "main", y crear el branch una vez que se decida hacer la entrega. Se debe dar permisos en el repositorio a los 3 docentes del curso. Se debe contar con un README que explique como correr el programa "desde cero", como instalar las dependencias y como correr el juego de forma clara y unica. El código debe poder correrse con un comando que compile el código y lo ejecute, **no es valido subir un ejecutable y que el comando simplemente lo corra**.

## Fecha de entrega

26/04/2024

## Criterios de aprobación

Para aprobar el trabajo práctico se deben cumplir con los objetivos mencionados previamente y además cumplir con todos los puntos mencionados en el enunciado.