

Proyecto Unidad 2

Carrera de sapos y tortugas

Guillermo Negrete

Gabriel Carcamo

 Programacion II | 20 de Mayo del 2017

# Introduccion

Frecuentemente en el mundo de la programación es común encontrarse con situaciones en que debe ser plasmada la realidad física en abstracciones algorítmicas que emulen dichas realidades, ejemplo de esto son los juegos de tablero. Diversos ejemplos de ellos podemos ver en el mercado, donde primero son disfrutados de forma física, conformándose por piezas, dados y un tablero generalmente que siguen un conjunto de instrucciones para dar dinámica al juego.

En la actualidad ese tipo de dinámicas son frecuentemente transportadas al mundo digital, para materializar asa, un sinfín de juegos en plataformas como consolas o computadores.

A continuación nos será develado un ejemplo de cómo se transporta a uno de estos juegos de mesa hacia la plataforma digital, para luego ser un videojuego.

# Pauta del proyecto

A continuación se exponen las especificaciones y/o requerimientos para el videojuego:

1. Desarrollar juego “Carrera de Sapos y Tortugas” bajo el paradigma de programación orientada a objetos.
2. Las Reglas del Juego “Carrera de Sapos y Tortugas” son:
   1. Existen solo 2 jugadores.
   2. Cada jugador posee 1 ficha de Sapo y 1 ficha de Tortuga.
   3. Todas las fichas deben ser ubicadas al comienzo del tablero al comenzar la partida.
   4. Se tienen 2 dados.
   5. Al principio cada jugador lanza solo 1 dado, luego parte el jugador que haya obtenido el número mayor, si hay empate ambos jugadores lanzan 1 dado nuevamente hasta que alguno saque un número mayor que el otro.
   6. En cada turno el jugador debe:
      1. Lanzar ambos dados.
      2. Escoger que pieza mover entre el Sapo o la Tortuga.
   7. La ficha Sapo se mueve el doble de la suma del valor obtenido en ambos dados.
   8. La ficha Tortuga se mueve el menor número obtenido en ambos dados.
   9. Gana el jugador que tenga ambas ficha en posición igual o mayor al final del tablero (posición 24).
3. Se deben respetar las normas de codificación.

# Restricciones del proyecto

Las restricciones para llevar a cabo el desarrollo del proyecto fueron las siguientes.

1. No Graphic model.

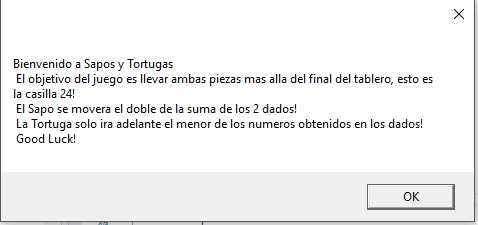
# Diagrama de clase (UML)

# 

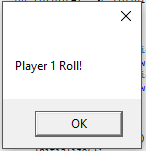
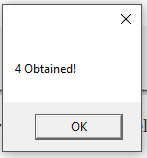
# Prueba de datos

El programa solo recibe clicks como input, asi que acontinuacion se expone el proceso de cada click.

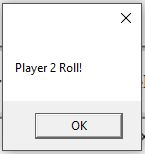
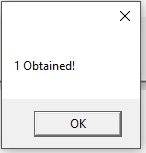
Inicio del programa:



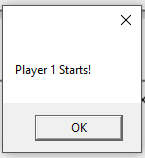
Luego para decidir quien parte, lanza un dado el jugador 1:

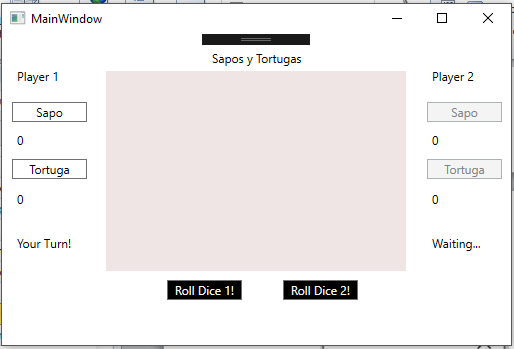
Luego lanza el jugador 2:

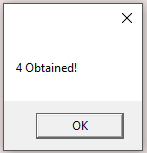
Bien, el jugador 1 obtuvo 4 y el 2 obtuvo solo 1, asique el jugador 1 debe partir:

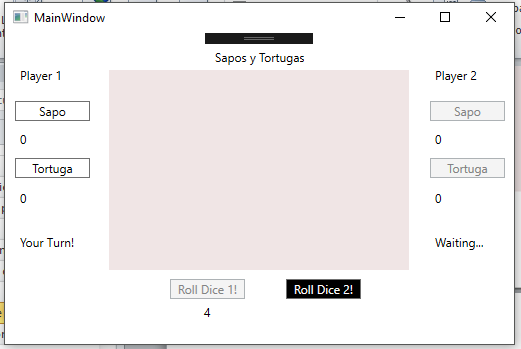


Luego se abre interfaz del juego, con todo bloqueado esperando que el jugador de turno lance los dados:

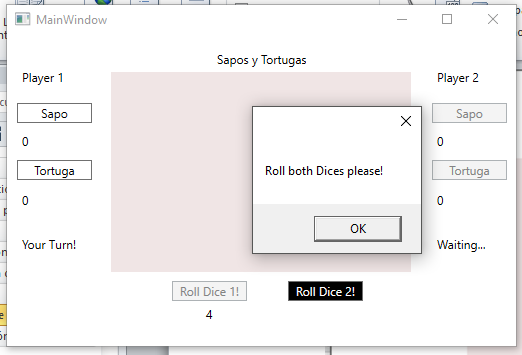


Al lanzar un dado el sistema indica el número obtenido y bloquea el dado:

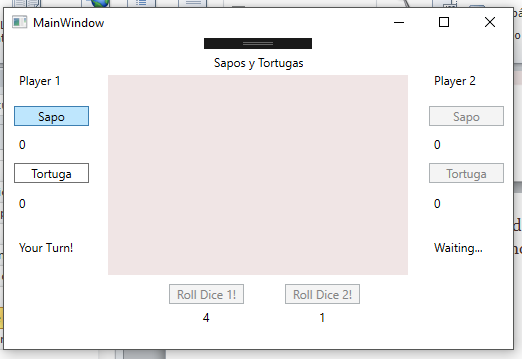




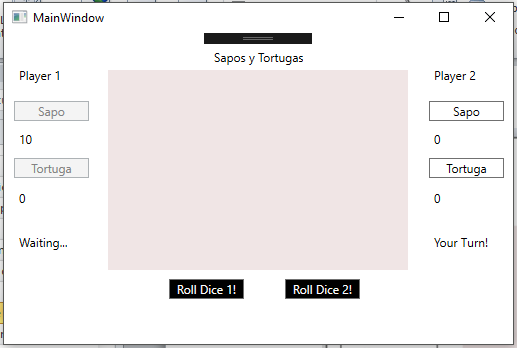
Si intentamos presionar los botones de asignación de Sapo o Tortuga, el sistema nos dirá que es necesario lanzar ambos dados antes de asignar movimiento a alguna de las piezas:



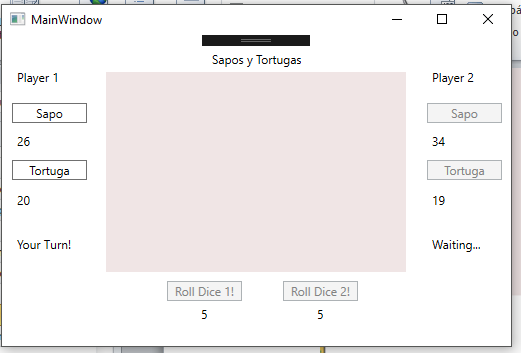
Una vez lanzados ambos dados, podemos dar movimiento a la pieza Sapo o Tortuga según le parezca al jugador de turno:



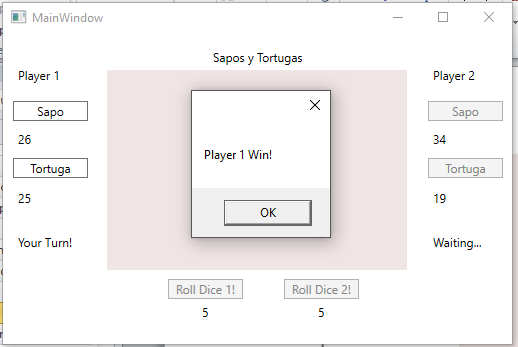
Una vez asignada la pieza el turno pasa al otro jugador:



El proceso se repite hasta que uno de los 2 jugadores deja ambas fichas en posición igual o mayor a 24.



Jugador 1 está a punto de ganar…



# Conclusión

El implementar juegos que comúnmente se disfrutan en la realidad tangible, a un entorno digital parece ser un muy buen ejercicio para desarrollar algoritmos relacionados con la interrelación de objetos y secuencias de orden.

Desde un punto de vista más simplista, este desarrollo nos ayudó a conceptualizar mejor algunas de las ventajas que ofrece el paradigma de programación orientada a objetos, aunque hasta el momento experiencias con otros lenguajes, dígase python, ruby y javascript, nos dejan con una sensación de poca escalabilidad del lenguaje C#, aunque la plataforma en que se utilice, pareciese tener un sinfín de herramientas.

Y luego como estudiantes creemos que con solo el ejercicio de desarrollar un juego con estas características a nivel back-end bastaba para absorber conocimientos de la programación orientada a objetos, luego la representación gráfica estaba algo de más en el ejercicio, y más aún la representación gráfica con curvas.