

PROYECTO DE ASIGNATURA FUNDAMENTOS DE PROGRAMACIÓN 2022-2

OBJETIVO

Aplicar los conocimientos adquiridos en la asignatura fundamentos de programación, en el desarrollo de un juego que permitan al(la) estudiante, avanzar en su proceso de aprendizaje, pasando de la solución de ejercicios, a la escritura de pequeños programas o aplicaciones en el lenguaje de programación Python.

ALCANCE

El proyecto que se pide desarrollar está dentro del alcance de los conceptos que se ven en la asignatura fundamentos de programación y tiene las siguientes particularidades.

- Lenguaje de programación: Python.
- Modo texto: no se requiere usar ningún aspecto gráfico avanzado en el desarrollo de la aplicación.
- Un solo archivo de código fuente es suficiente para contener a todo el proyecto.

ENUNCIADO: CUATRO SEGUIDAS

Este juego para dos participantes consiste en intentar obtener cuatro fichas seguidas ubicadas de manera vertical, horizontal o diagonal en el tablero de juego. Cada participante intentará obtener sus cuatro fichas mientras impide al oponente obtener las suyas. Gana el participante que primero logre sus cuatro fichas seguidas. Cada participante juega un turno a la vez y en caso de que el participante quiera tentar a la suerte, se le permitirá la opción para que el computador determine de manera aleatoria la ubicación de la ficha de su turno.

Al inicio de la partida se debe preguntar el nombre de cada participante. Al primer participante se le debe permitir elegir la ficha con la que jugará. Un participante juega con la ficha representada por el carácter **X** y el otro participante juega con la ficha representada por el carácter **O**. Se determina de manera aleatoria cuál de los dos participantes inician la partida.

El tablero del juego consta de 6 filas y 7 columnas en donde los participantes pueden ubicar cada una de sus fichas. Se debe tener presente que las fichas se irán apilando en orden en cada una de las columnas, comenzando desde la fila inferior, de acuerdo con la elección del jugador en su respectivo turno.

Al finalizar la partida se debe mostrar el Top-5 de lo(a)s participantes con los mejores puntajes en todo el historial del juego y preguntar si desean volver a jugar. La partida finaliza cuando alguno de lo(a)s participantes consigue sus cuatro fichas seguidas o cuando se alcanzó la capacidad de fichas en el tablero sin que nadie hubiera conseguido vencer a su oponente.

Lo(a)s estudiantes son libres de decidir el esquema de puntajes (y si lo considera, de penalizaciones). Por ejemplo, podrían obtener más puntos lo(a)s participantes que logren vencer a su oponente en

el menor número de jugadas y otorgar menos puntos a los participantes que ganan las partidas tentando a la suerte, es decir, sin esfuerzo intelectual ni estrategia alguna para vencer a su oponente.

A continuación, se ilustra la lógica de una partida de ejemplo. Las letras de color rojo se han puesto deliberadamente en ese color para identificar la entrada del (la) participante. Las indicaciones en verde se han puesto deliberadamente en ese color para indicar que en la partida suceden otros eventos que no se muestran en el ejemplo .

EJEMPLO # 1

*** CUATRO SEGUIDAS ***

- > Por favor indique nombre de participante #1: **Andrea**
- > Andrea, por favor indica con qué ficha deseas jugar [X] o [O]: **X**
- > Por favor indique nombre de participante #2: **Tatiana**
- > Tatiana, te toca jugar con la siguiente ficha: O

- > Lanzando una moneda al aire para determinar quién inicia la partida...
- > La partida la inicia Tatiana

```
1234567
+-----+
| ..... |
| ..... |
| ..... |
| ..... |
| ..... |
| ..... |
+-----+
```

- > Tatiana, indica un número de columna o pulsa [S] para tentar a la suerte: **3**

```
1234567
+-----+
| ..... |
| ..... |
| ..... |
| ..... |
| ..... |
| ..0.... |
+-----+
```

> Andrea, indica un número de columna o pulsa [S] para tentar a la suerte: 3

```
1234567
+-----+
| .....|
| .....|
| .....|
| .....|
| ..X...|
| ..0...|
+-----+
```

> Tatiana, indica un número de columna o pulsa [S] para tentar a la suerte: 7

```
1234567
+-----+
| .....|
| .....|
| .....|
| .....|
| ..X...|
| ..0...0|
+-----+
```

> Andrea, indica un número de columna o pulsa [S] para tentar a la suerte: S

> mmm...suerte con eso, se eligió aleatoriamente la columna 5 para tu ficha

```
1234567
+-----+
| .....|
| .....|
| .....|
| .....|
| ..X...|
| ..0.X.0|
+-----+
```

<Otros turnos que no se muestran en este ejemplo>

```
1234567
+-----+
| .....|
| .....|
| XXX.X0.|
| 0000X0.|
| 000X0X.|
| 0XXX0XX|
+-----+
```

> ¡Felicidades, Tatiana, has ganado la partida!

> Has sumado 40 puntos en esta partida

*** TABLA DE POSICIONES ***

1. Andrea 110 puntos acumulados. Última partida en 2022-08-16 a las 13:50
2. Camilo 80 puntos acumulados. Última partida en 2021-02-25 a las 10:15
3. Tatiana 75 puntos acumulados. Última partida en 2022-08-16 a las 13:50
4. Juana 70 puntos acumulados. Última partida en 2022-05-10 a las 14:00
5. Johana 60 puntos acumulados. Última partida en 2022-05-11 a las 14:00

> ¿Desean volver a tomar la partida Si [S] No [N]: **N**
> ¡Gracias por jugar!

REQUISITOS FUNCIONALES

- El juego debe ser desarrollado completamente en Python y en un solo archivo de código.
- Cada partida debe incluir los elementos y la lógica que se indican en el enunciado y en la partida de ejemplo.

ENTREGABLES Y SUSTENTACIÓN

- Se realizan tres sustentaciones parciales del 8% cada una en las semanas 5, 10 y 15.
- En cada una de las sustentaciones se debe tener un avance en el desarrollo del juego y se debe sustentar el avance en términos del código y planteamiento lógico del problema.
- En la semana 15 se debe presentar el juego completamente funcionando y entregar el código de este.

RECOMENDACIÓN

Su producto debe reflejar su proceso de aprendizaje. No copie código de ninguna parte ni de ningún(a) compañero(a) o grupo. Acérquese a las asesorías del docente o de los monitores para que vaya avanzando en el desarrollo y planteamiento del juego. Si se encuentra que su código y/o planteamiento lógico es igual al de otro(s) grupo(s), su trabajo y el del (los) grupo(s) copia se anula(n).