1. **IDENTIFICACIÓN DEL ESPACIO ACADÉMICO.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Programa** | Ingeniería de Sistemas | | | |
| **Espacio Académico/curso** | Programación de video juegos | | **Semestre** | 6 |
| **Área** | N/A | **Grupo** | E | |
| **Tipo de entrega** | Informe de entrega | | | |
| **Nombres completos** |  | | | |

1. **GUÍA PRÁCTICA**

A continuación, se presenta una tabla cronológica con los videojuegos más destacados hasta el momento, incluyendo detalles como su creador, lenguaje de programación, plataforma de desarrollo, tipo de juego, fecha de lanzamiento, y tipo de dispositivo. Completa las columnas solicitadas:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabla de video juegos** | | | | | | |
| **Nombre del juego** | **Creador / empresa** | **Lenguaje de programación** | **Plataforma de Desarrollo** | **Tipo de Juego/Clasificación** | **Fecha de Lanzamiento** | **Tipo (Web, Móvil, Consola)** |
| Pong |  |  |  |  |  |  |
| Space Invaders |  |  |  |  |  |  |
| Pac-Man |  |  |  |  |  |  |
| Donkey Kong |  |  |  |  |  |  |
| Tetris |  |  |  |  |  |  |
| Super Mario Bros. |  |  |  |  |  |  |
| The Legend of Zelda |  |  |  |  |  |  |
| Street Fighter II |  |  |  |  |  |  |
| Doom |  |  |  |  |  |  |
| Final Fantasy VII |  |  |  |  |  |  |
| Half-Life |  |  |  |  |  |  |
| The Sims |  |  |  |  |  |  |
| Grand Theft Auto III |  |  |  |  |  |  |
| World of Warcraft |  |  |  |  |  |  |
| Minecraft |  |  |  |  |  |  |
| Candy Crush Saga |  |  |  |  |  |  |
| The Last of Us |  |  |  |  |  |  |
| Fortnite |  |  |  |  |  |  |
| Among Us |  |  |  |  |  |  |
| Genshin Impact |  |  |  |  |  |  |

A continuación, se presenta una tabla con los motores para desarrollo de videojuegos más destacados hasta el momento. Completa las columnas solicitadas:

|  |  |  |
| --- | --- | --- |
| **Motores de videojuegos** | | |
| **Motor** | **empresa** | **Lenguajes que usa para el desarrollo del videojuego** |
| Unity |  |  |
| Unreal Engine |  |  |
| Godot |  |  |
| CryEngine |  |  |
| GameMaker Studio |  |  |
| RPG Maker |  |  |
| Defold |  |  |
| Amazon Lumberyard |  |  |
| Cocos2d-x |  |  |
| Construct 3 |  |  |
| Phaser |  |  |
| Corona SDK |  |  |
| Panda3D |  |  |
| Torque 3D |  |  |
| Ren'Py |  |  |
| Scratch |  |  |
| Monogame |  |  |
| AppGameKit |  |  |
| libGDX |  |  |
| O3DE (Open 3D Engine) |  |  |
| Pygame |  |  |

**Electiva de programación** | Programación de video juegos

1. Considera el siguiente juego de “Carrera numérica”: (5.0 pts)

**Fuente:** <https://masterwise.cl/products/4333>

Crea un *Script* en **Python** que permita simular el comportamiento del juego de “Carrera numérica” bajo las siguientes condiciones y reglas de juego:

# **Entradas:**

* 1. El sistema debe solicitar por pantalla la cantidad de jugadores (Mínimo 2, máximo 4).

**Remark**: Valida que no acepte menos de dos, ni más de 4 jugadores.

* 1. El sistema debe solicitar por pantalla el Nivel de tablero a jugar:

# **Considera el siguiente menú:**

1. Nivel básico (Tablero de 20 posiciones)
2. Nivel intermedio (Tablero de 30 posiciones)
3. Nivel avanzado (Tablero de 50 posiciones)
4. Nivel experto (Tablero de 100 posiciones)

# **Proceso:**

* 1. Una vez inicie el juego, en el turno de cada jugador, el sistema debe “lanzar los dos dados” y generar aleatoriamente las posiciones a mover. Este proceso se realizará cíclicamente por cada turno. El juego finalizará ÚNICAMENTE cuando uno de los jugadores llegue a la meta.
  2. **Remark**: Si el jugador obtiene 3 pares CONSECUTIVOS (Se refiere a dados iguales), es el ganador directamente.

Tenga en cuenta que la meta será la última posición del tablero, de acuerdo al nivel escogido en el ítem **B**. Gana quién llegue con el valor igual o mayor al de la meta o según lo indicado en el ítem d.

**El envío del Script debe realizarlo a través del STAGING con GIT/GITHUB al repositorio video-games del espacio académico.**