**Grupo 0:** Alexis Rodriguez, Marcel Moran

**Titulo: Tetris**

1. **Descripción del juego**

El juego se juega en en un canvas en el cual debemos encajar figuras geometricas compuestas de cuadrados unidos de forma ortogonal para formar una fila o filas, estas figuras caen desde el tope del canvas y se detienen cuando chocan con la parte baja del canvas o chocan con otras piezas, el jugador puede rotar la pieza en 90 o -90 grados, las piezas que caen del canvas son aleatorias y por tanto el jugador no puede saber exactamente qué pieza le tocará jugar, normalmente el jugador toca la parte baja del canvas.

1. **Propósito del juego:**

El usuario debe tratar de conseguir la mayor cantidad de puntos. Cada vez que el usuario consiga completar una fila al pie del canvas del juego toda la fila desaparecerá y eso le otorgará al jugador 10 puntos adicionales. Si varias filas se destruyen al mismo tiempo el usuario recibe 10 puntos por cada fila que destruyó.

1. **¿Cómo se juega?**

El usuario puede mover las piezas con las flechas izquierda y derecha, con la flecha hacia abajo la pieza caerá inmediatamente a la base. Las piezas pueden rotar presionando el botón numérico 0, el juego puede ser pausado con la tecla X.

1. **Diseño del código:**

Este programa requiere las siguientes clases:

| Clase celda subclase de **pygame.Rect** | |
| --- | --- |
| Atributos | Métodos |
| \_\_libre  \_\_izquierda  \_\_derecha  \_\_arriba  \_\_abajo | limpiar  marcar |

| Clase Pieza | |
| --- | --- |
| Atributos | Métodos |
| \_\_color  \_\_forma  \_\_celdas  \_\_celdas\_requeridas | actualizar  celdas  mover |

| Clase Contenedor | |
| --- | --- |
| Atributos | Métodos |
| ancho  alto  celdas | render  aparecer\_pieza |

| Clase Tetris | |
| --- | --- |
| Atributos | Métodos |
| \_\_pantalla  \_\_contenedor | \_\_configurar  empezar  \_\_bucle  \_\_render  \_\_manejo\_de\_eventos |

Para empezar el juego se deberá crear un objeto de clase Tetris y llamar al método empezar. Este método llamará al método privado \_\_configurar, que inicializa todo lo necesario para luego llamar el metodo \_\_bucle que como su nombre lo indica corresponde a un bucle en el que el juego permanece hasta que este termine.

1. **Librerias :**

pygame

numpy

1. **Aspectos adicionales :**

* Los colores de las piezas son aleatorios

**División y planificación de tiempo**

Número de horas esperadas: **16 horas**

| **Alexis** | **Marcel** | **Dia** |
| --- | --- | --- |
| Planificacion | Planificacion | **Miércoles** |
| Aprender acerca de pygame | Aprender acerca de pygame | **Jueves** |
| Desarrollo de Celda | Desarrollo de Juego | **Viernes** |
| Desarrollo de Contenedor | Desarrollo de Player | **Lunes** |
| Integracion | Integracion | **Martes** |

| **Alexis** | **Marcel** | **Dia** |
| --- | --- | --- |
| Integracion | Integracion | **Miercoles** |
| Depuracion | Depuracion | **Jueves** |
| Preparar presentación | Preparar presentación | **Viernes** |
| Presentacion | Presentacion | **Lunes** |
|  |  | **Martes** |