

Documentación Actividad n°15



Año: 2024

Profesores: York Elias MANSILLA MUÑOZ y Jorge Fabián SILES GUZMÁN

Materia: Proyecto de Implementación de Sitios Web Dinámicos

Grupo: Carlos Soliz

Documentación del Código 1: Juego de Tic-Tac-Toe

Este código implementa un sencillo juego de Tic-Tac-Toe (Tres en línea) utilizando la biblioteca gráfica de Processing.

Variables Globales:

- `int[][] board`: Matriz de 3x3 que representa el tablero del juego. Cada celda puede tener los valores 0 (vacío), 1 (X) o 2 (O).
- `boolean xTurn`: Indica si es el turno del jugador X. Si es `true`, es el turno de X; si es `false`, es el turno de O.
- `boolean gameEnded`: Indica si el juego ha terminado.

Funciones:

- `void setup()`: Configura el entorno gráfico y llama a la función `resetBoard()` para inicializar el tablero.
- `void draw()`: Se ejecuta continuamente y actualiza la visualización del juego. Dibuja el tablero y verifica si el juego ha terminado.
- `void drawBoard()`: Dibuja las líneas que separan las casillas del tablero y las marcas (X u O) en las celdas.
- `void drawX(int i, int j)`: Dibuja una X en la celda especificada por las coordenadas `(i, j)` del tablero.
- `void drawO(int i, int j)`: Dibuja una O en la celda especificada por las coordenadas `(i, j)` del tablero.
- `void mousePressed()`: Se activa cuando el jugador hace clic en el tablero. Coloca una X u O en la casilla correspondiente si el juego no ha terminado y la casilla está vacía.
- `void checkWin()`: Verifica si hay un ganador revisando las filas, columnas y diagonales. También comprueba si hay un empate (cuando todas las casillas están llenas y no hay ganador).
- `void resetBoard()`: Inicializa o reinicia el tablero, poniendo todas las casillas en 0 y restableciendo el turno del jugador X.

Flujo del Juego:

1. El juego comienza con un tablero vacío.
2. Los jugadores hacen clic en las casillas para colocar una X u O en turnos alternos.
3. El juego verifica si alguien ha ganado o si se ha producido un empate.
4. Si el juego termina, se puede reiniciar haciendo clic de nuevo en el tablero.

Documentación del Código 2: Juego de Pong en 3D

Este código implementa una versión en 3D del clásico juego de Pong utilizando Processing.

Variables Globales:

- `PVector ballPos, ballSpeed`: Representan la posición y velocidad de la pelota en 3D.
- `float ballRadius`: Radio de la pelota.
- `PVector paddle1Pos, paddle2Pos`: Posiciones de las dos paletas (izquierda y derecha).
- `float paddleWidth, paddleHeight, paddleSpeed`: Dimensiones de las paletas y velocidad de movimiento.

Funciones:

- `void setup()`: Configura el entorno gráfico en 3D, inicializa las posiciones de la pelota y las paletas.
- `void draw()`: Se ejecuta continuamente y actualiza el juego. Dibuja la pelota, las paletas, y maneja las colisiones y el movimiento.
- `void moveBall()`: Actualiza la posición de la pelota en base a su velocidad y verifica colisiones con las paredes del campo.
- `void checkCollisions()`: Verifica si la pelota ha colisionado con alguna de las dos paletas. Si es así, invierte la dirección de la pelota.
- `void drawBall()`: Dibuja la pelota en su posición actual en el espacio 3D.
- `void drawPaddles()`: Dibuja las paletas en sus posiciones actuales en el espacio 3D.
- `void movePaddles()`: Controla el movimiento de las paletas según la entrada del teclado. La paleta izquierda se mueve con las teclas 'W' y 'S', mientras que la derecha se mueve con las flechas arriba y abajo.

Flujo del Juego:

1. La pelota comienza en el centro de la pantalla y se mueve en una dirección diagonal.
2. Las paletas pueden moverse verticalmente para interceptar la pelota.
3. Cuando la pelota choca con una paleta, su dirección horizontal cambia. También rebota en las paredes superior e inferior.
4. El objetivo del juego es evitar que la pelota salga del campo por el lado contrario al de la paleta controlada por el jugador.