THE WII MAZE

SHAUN SALZBERG
MACK LU
BRIAN RAMOS
THE GAME
HARDWARE OVERVIEW

- Actual maze, object, goal, and character tiles
- Virtual maze and object tiles
- Maze and object descriptors
- Other signals
- Virtual screen space
- Pixel drawing scheme
Actual Tiles

Maze Tiles

Object Tiles

Character Tiles

Goal Tile
Virtual Tiles

Maze Virtual Tiles
Bits 7 - 6: Index into Maze_Tiles
Bits 5 - 4: Rotation
Bits 3 - 0: No longer used

Object Virtual Tiles
Bits 8: Index into Mto Obj_Tiles
Bits 7 - 6: Rotation
Bits 5 - 4: Color
Bits 3 - 0: No longer used

type virtual_maze_tile_type is array(0 to 14) of unsigned(7 downto 0);

constant LEFT_WALL : unsigned(3 downto 0) := "0000";
constant TOP_WALL : unsigned(3 downto 0) := "0001";
constant RIGHT_WALL : unsigned(3 downto 0) := "0010";
constant BOTTOM_WALL : unsigned(3 downto 0) := "0011";
constant TOP_LEFT_CORNER : unsigned(3 downto 0) := "0100";
constant TOP_RIGHT_CORNER : unsigned(3 downto 0) := "0101";
constant BOTTOM_LEFT_CORNER : unsigned(3 downto 0) := "0110";
constant BOTTOM_RIGHT_CORNER : unsigned(3 downto 0) := "0111";
constant HORIZONTAL_WALL : unsigned(3 downto 0) := "1000";
constant DIAGONAL_WALL : unsigned(3 downto 0) := "1001";
constant TOP_DEADEND : unsigned(3 downto 0) := "1010";
constant BOTTOM_DEADEND : unsigned(3 downto 0) := "1011";
constant LEFT_DEADEND : unsigned(3 downto 0) := "1100";
constant RIGHT_DEADEND : unsigned(3 downto 0) := "1101";
constant NO_MAZE_TILE : unsigned(3 downto 0) := "1110";

constant LEFT_WALL_RED : unsigned(3 downto 0) := "0000";
constant TOP_WALL_RED : unsigned(3 downto 0) := "0001";
constant RIGHT_WALL_RED : unsigned(3 downto 0) := "0010";
constant BOTTOM_WALL_RED : unsigned(3 downto 0) := "0011";
constant LEFT_WALL_GREEN : unsigned(3 downto 0) := "0100";
constant TOP_WALL_GREEN : unsigned(3 downto 0) := "0101";
constant RIGHT_WALL_GREEN : unsigned(3 downto 0) := "0110";
constant BOTTOM_WALL_GREEN : unsigned(3 downto 0) := "0111";
constant LEFT_WALL_BLUE : unsigned(3 downto 0) := "1000";
constant TOP_WALL_BLUE : unsigned(3 downto 0) := "1001";
constant RIGHT_WALL_BLUE : unsigned(3 downto 0) := "1010";
constant BOTTOM_WALL_BLUE : unsigned(3 downto 0) := "1011";
constant KEY_RED : unsigned(3 downto 0) := "1100";
constant KEY_GREEN : unsigned(3 downto 0) := "1101";
constant KEY_BLUE : unsigned(3 downto 0) := "1110";
constant NO_OBJ_TILE : unsigned(3 downto 0) := "1111";
Descriptors

- 2-D arrays of virtual tile numbers
- Represent relative positions of those tiles
- Max maze size is 16 by 16 tiles
OTHER SIGNALS

- `MAZE_X_OFFSET, MAZE_Y_OFFSET`
- `CUR_XPOS_PIX, CUR_YPOS_PIX`
- `END_XPOS_PIX, END_YPOS_PIX`
- `HALF_WINDOW_SIZE_X, HALF_WINDOW_SIZE_Y`
- `WHICH_KEYS, WHICH_CHAR_TILE`
VIRTUAL SCREEN SPACE

(0,0)

QXTRA_TOP

QXTRA_LEFT

ACTUAL SCREEN

SYNC, BACKPORCH, FRONTPORCH, ETC...
PIXEL DRAWING SCHEME

- If pixel is outside window, it is black.
- If pixel is on window border, it is white.
- If pixel is within character tile and entry in that tile is 1, it is yellow.
- If pixel is within object tile, and entry in that tile is 1, it is the object color (red, blue, or green).
- If pixel is within goal tile and entry in that tile is 1, it is light blue.
- If pixel is within maze tile and entry in that tile is 1, it is white.
- Else it is black.
INTERACTING HARDWARE WITH SOFTWARE

// maze tile info
// first row
the_level[DESC_MAZE_OFFSET+0][0] = TOP_LEFT_CORNER;
the_level[DESC_MAZE_OFFSET+0][1] = TOP_RIGHT_CORNER;
for(i = 2; i < MAX_MAZE_SIZE_X; i++ ) the_level[DESC_MAZE_OFFSET+0][i] = NO_MAZE_TILE;

// second row
the_level[DESC_MAZE_OFFSET+1][0] = BOTTOM_LEFT_CORNER;
the_level[DESC_MAZE_OFFSET+1][1] = BOTTOM_RIGHT_CORNER;
for(i = 2; i < MAX_MAZE_SIZE_X; i++ ) the_level[DESC_MAZE_OFFSET+1][i] = NO_MAZE_TILE;

// rest of the rows
for(i = DESC_MAZE_OFFSET+2; i <= MAX_MAZE_SIZE_Y; i++ ) {
    for( j = 0; j < MAX_MAZE_SIZE_X; j++ )
        the_level[i][j] = NO_MAZE_TILE;
}
SOFTWARE INFRASTRUCTURE

- uCLinux
- Bluez Library
- LibWiimote Library
- Compiling Apps
- Running
THE PROGRAM

- ACCELEROMETER READINGS
- WIIMOTE_UPDATE