SimpleVania Description:

SimpleVania is a video game implemented on the ATMega1284P microcontroller, which resembles the gameplay style of the Nintendo classic, CastleVania. The player controls the character with a Super Nintendo Controller connected to the microcontroller. After receiving input from the controller the player will move across each level which appears on the 8x8 LED matrix. Additional information is displayed on the LCD screen located above the LED matrix. Game logic includes character movement, synchronized updating of the LCD screen with the game as well as a delay on the players attack, just like the original Castlevania game. The map for the game itself is also a 1-to-1 recreation of the first stage.
http://faqs.neoseeker.com/Games/NES/castlevania_level_1.png

User Guide:

Basic Controls:

- DPAD - MOVEMENT
- B - JUMP
- Y - ATTACK
- SELECT - RESET POSITION

Advanced Controls:

- B+Y - JUMP ATTACK
- B+LEFT/RIGHT - FORWARD/BACKWARD JUMP
- Y+DOWN - CROUCH ATTACK

Game Rules:

- Player loses when HP(displayed on LCD) reaches 0
- Player loses 1 HP when touching enemies(Green) or water(blue)
- Candles(white/blue) are destructible and do not harm the player
- Player whens when reaching/destroying the boss(purple)

Special Considerations:

EPILEPSY WARNING

READ THIS NOTICE BEFORE YOU OR YOUR CHILD USE ANY VIDEO GAME

A very small portion of the population have a condition which may cause them to experience epileptic seizures or have momentary loss of consciousness when viewing certain kinds of flashing lights or patterns that are commonly present in our daily environment. These persons may experience seizures while watching some kinds of television pictures or playing certain video games. Players who have not had any previous seizures may nonetheless have an undetected epileptic condition.

If you or anyone in your family has experienced symptoms linked to an epileptic condition (e.g. a seizure or loss of awareness), immediately consult your physician before using any video games. We recommend that parents observe their children while they play video games. If you or your child experience any of the following symptoms: dizziness, altered vision, eye or muscle twitching, involuntary movements, loss of awareness, disorientation, or convulsions, DISCONTINUE USE IMMEDIATELY and consult your physician.

Source: (http://www.nintendo.com/consumer/manuals/precautions_console_pak_english.jsp)

Technologies and Components

- AVR Studio 6
- ATMega1284p
- GTM2088ARGB-25 8x8 LED Matrix

- 74HC595 Shift Register (x4)
- LCM-S01602DTR/M 16x2 LCD Screen
- 330 Resistor (x8)
- SNES Controller (x1)
- Breadboard

DEMO VIDEO: (https://www.youtube.com/watch?v=MUMmozsTyTw)

SOURCE FILES:

game_objects.h
Contains character objects as well as the flag in_air which limits player actions when set to 1
LINK: (https://drive.google.com/file/d/0BxwLt8W7ewG2NzUySTQ1ck9wb2M/edit?usp=sharing)

level.c (CRITICAL)
Contains most functionality of the game, including movement, attacking, and even rendering the game.
LINK:(https://drive.google.com/file/d/0BxwLt8W7ewG2cVNVb2NfMUJhSFk/edit?usp=sharing)

SimpleVania.c
Contains the task scheduler which allows the game to function.
LINK: (https://drive.google.com/file/d/0BxwLt8W7ewG2dFdLMzliVkRhOEE/edit?usp=sharing)

bit.h source: University of California Riverside (Modifications by Jonathan Padilla)
Added functions such as setLongbitRED(), which are used by level.c in order to determine what colors to generate on the LED matrix.
LINK:(https://drive.google.com/a/ucr.edu/?usp=docs_web#folders/0BxwLt8W7ewG2cmRrWU4tXzdYTUk)

Outside Sources:
controller.c source: Johnny Do
This include contains code which is used to obtain controller input.
The function however was modified slightly and placed in SimpleVania.C
Buttons presses are placed into a vector. The position in the vector determines what buttons are pressed.
LINK: (https://drive.google.com/file/d/0BxwLt8W7ewG2ZHphaFR5N0xWMUU/edit?usp=sharing)

bit.h (UCR_includes) source: University of California Riverside
The include contains functions such as SetBit(); and GetBit();