WASHINGTON STATE UNIVERSITY
VANCOUVER

Software Design Document

for

<MOS 6502 Virtual Machine>

Version <Beta 1.01>

Prepared by

Group Name: ***<GIT her done>***

Shanpreet Singh	**011586218**	**Shanpreet.singh@wsu.edu**
Aleksandr Khavarovskiy	**011600326**	**a.khavarovskiy@wsu.edu**
Daniil Kernazhytski	**011144879**	**daniil.kernazhytski@wsu.ed**
Saeed Al-Sarhi	**011591809**	**saeed.alsarhi@wsu.edu**
Josh Feener	**011579239**	**joshua.feener@wsu.edu**

Date: **November 5th, 2017**

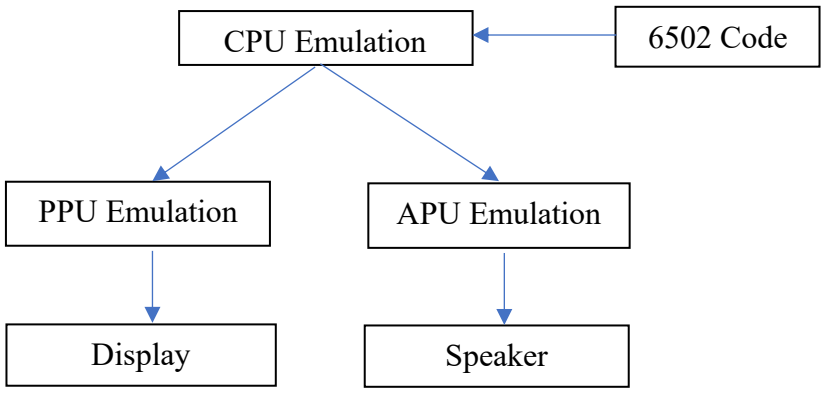
1 Preliminary Design

1.1 Document Purpose

The following document is a Software Design Document (SDD). In this document, you will find the software requirements translated into a representation of software components, interfaces, and data necessary for implementation phase. The software system will be structured to satisfy the requirements. The document is broken up into two stages of design. The stages are the Preliminary Design and System Modeling. The purpose of this document given those two stages is to provide a general understanding of how and why the system was decomposed, and how the individual parts work together.

1.1 Architectural Design

Our system relies on 3 different subsystems to achieve full functionality. These systems mimic the original functionality of the Nintendo Entertainment System. Once this is complete, our goal is to execute 6052 8-bit microprocessor code on a processor that does not natively support it.



6502 Code

- Machine code of a complied NES application.

CPU Emulation:

- The CPU Emulation is an abstraction layer between the native hardware and 6502 instruction code. Modern processors do not support the 8-bit instruction set and our emulator uses native hardware to simulate these commands and their desired output.

PPU Emulation:

- The PPU is a Picture Processing Unit which runs in parallel with the 6502 to generate images at 30 frames per second. Normally encoded with a NTFC or a PAL video output.

APU Emulation:

- The APU is an audio processing unit generating 8-Bit audio from the game programming.

Display:

- To generate our image onto the display we use the OpenGL rendering library to render images produced by the PPU.

Speaker:

- Once the audio signal is generated at the APU we send the signal into an external speaker.

1.2 Data Design

The Nintendo Entertainment System was released in the US on October 18, 1985 and was discontinued on August 14th, 1995. For over 10 years the console has had the same hardware specifications as technology continued to progress. The hardware limitations lead to game developers implementing secondary hardware onto the game ROM cartridges to further expand their game functionality. Secondary hardware implementations allowed for persistent memory, users were able to save their games directly onto their cartridges. Some cartridges added more RAM, ROM, CROM or even secondary APU which allowed for much richer audio quality and depth. The 6502 microprocessor is limited to 32KB of internal program memory lead to developers implementing a feature known as bank switching. Memory mapping chips were implemented to allow an executing program to switch memory banks and effectively run applications much bigger than the limiting factor of the hardware without actually exceeding the 32KB limit of the 6502. This makes every game cartridge a Unique piece of hardware and must be accurately accounted for accurate execution. However, our initial release will be limited by this factor and will stand as a proof of concept.

NES ROM:
We load our file into program memory and we its header to ascertain any pertinent information.

```
NES Header information
--------------------------------------------------------------------------
Byte     Contents
--------------------------------------------------------------------------
0-3      String "NES^Z" used to rec
4        Number of 16kB ROM banks
5        Number of 8kB VROM bank
6        bit 0     1 for vertical mirrorin
         bit 1     1 for battery-backed
         bit 2     1 for a 512-byte traine
         bit 3     1 for a four-screen VR
         bit 4-7   Four lower bits of RO
7        bit 0     1 for VS-System cart
         bit 1-3   Reserved, must be z
         bit 4-7   Four higher bits of R
8        Number of 8kB RAM banks.
         versions of the .NES format,
         byte is zero.
9        bit 0     1 for PAL cartridges,
         bit 1-7   Reserved, must be z
10-15    Reserved, must be zeroes
16-...   ROM banks, in ascending o
         512 bytes precede the ROM
...-EOF  VROM banks, in ascendin
--------------------------------------------------------------------------
```

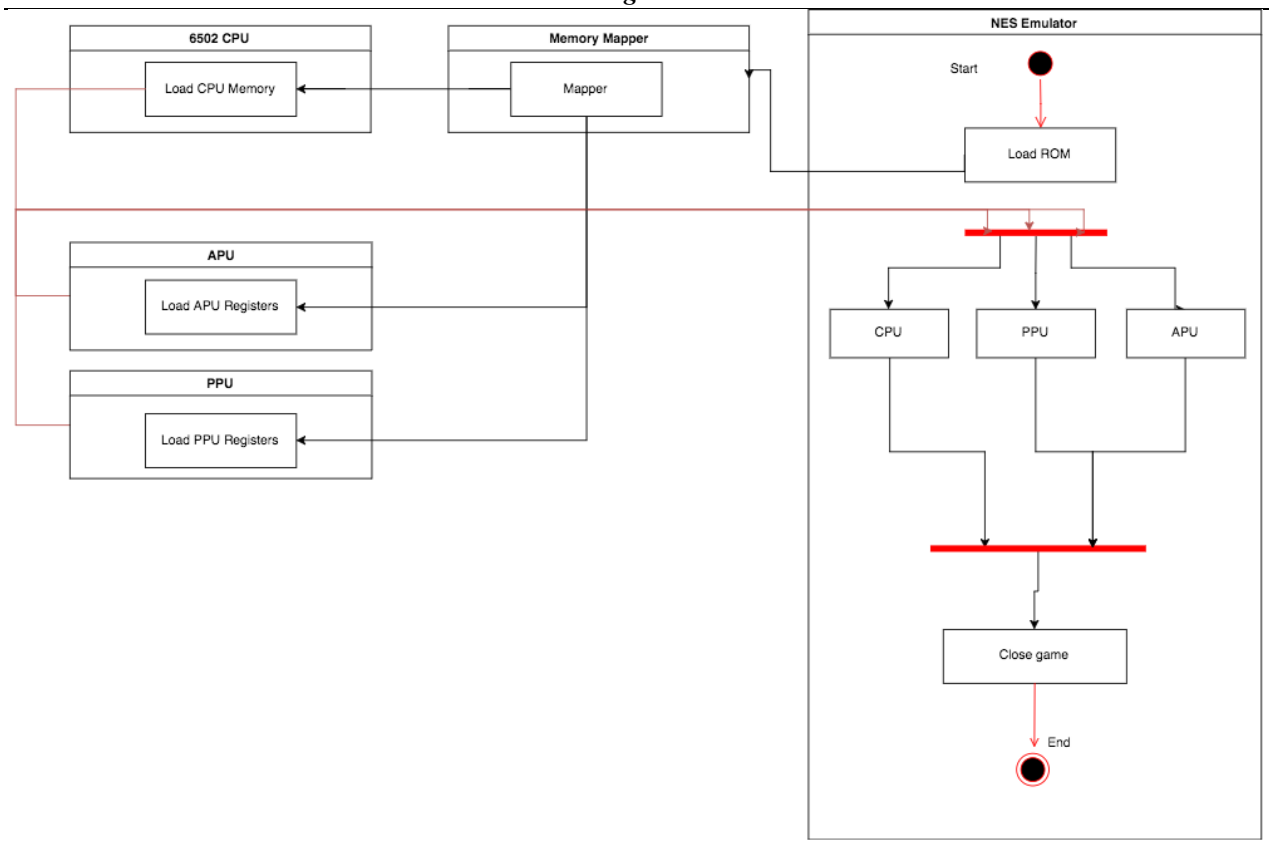
C Equivalent of the header

```
typedef struct __NES_HEADER
{
    u8 m_nes[4];
        u8 m_programRomSize;   // 16,384 Bytes * Program   Rom 16k = 0x4000
    u8 m_characterRomSize; //  8,192 Bytes * Character Rom     = 0x2000
    u8 m_mirroring        : 1;
    u8 m_batteryBacked    : 1;
    u8 m_traniner         : 1;
    u8 m_fourScreenVRAM   : 1;
    u8 m_lowerMapperNybble : 4;
        u8 m_VSUnisystem      : 1; // VS Unisystem
    u8 m_reserved0        : 3; // Reserved, must be zeroes!
    u8 m_upperMapperNybble : 4;
    u8 m_programRamSize;//8,192 Bytes * Program Ram Size - Assume 1 8KB bank on 0
        u8 m_tvSystem  : 1;
    u8 m_reserved1 : 7
    u8 m_flag10;
    u8 m_zero[5];
}
NES_HEADER
__attribute__ ((aligned(1)));
```

information, which is mapped onto the loaded memory.

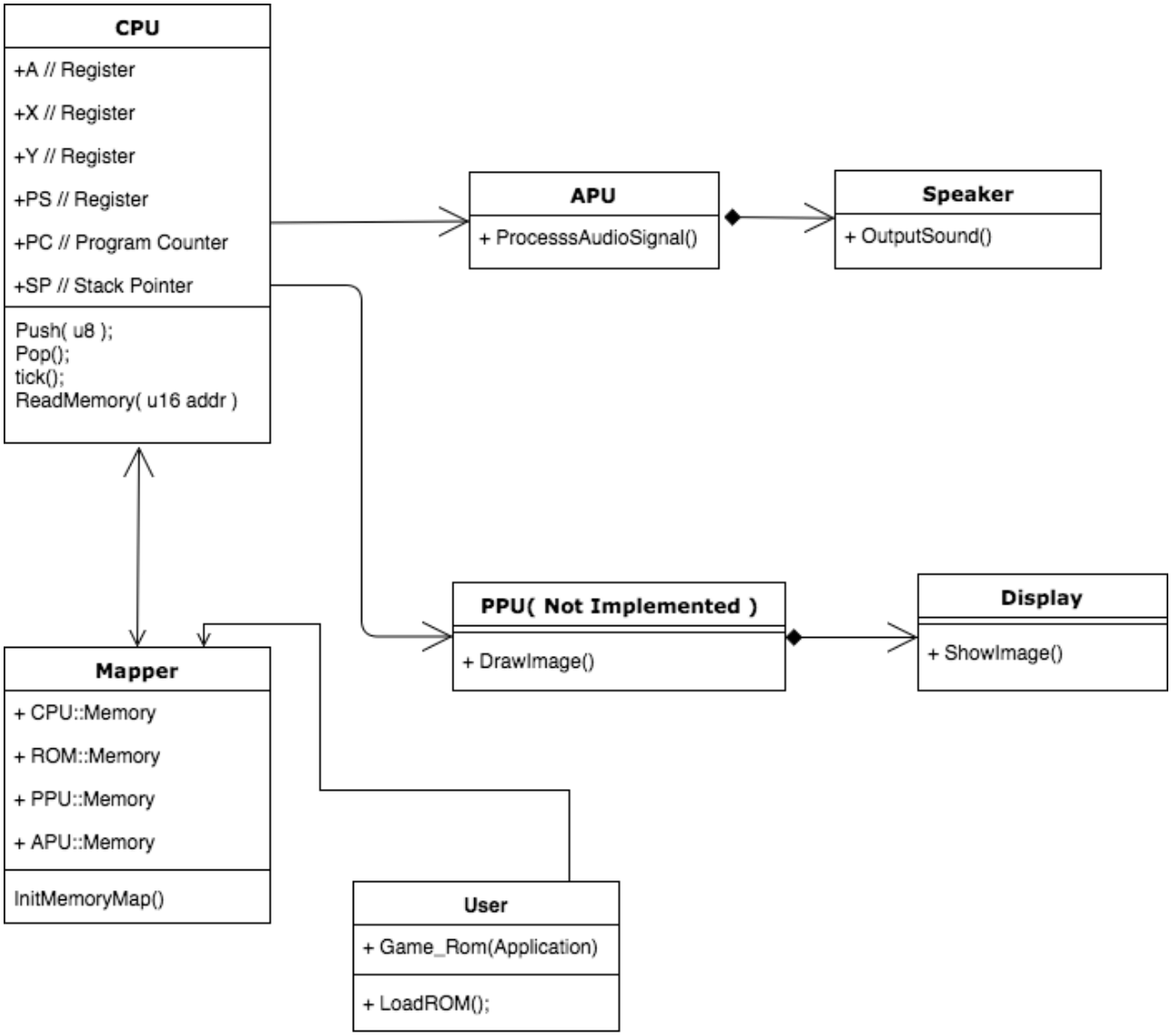
2 System Modeling

2.1 Activity Diagrams



The Previous activity describes the general use case.

2.2 Structural Modeling



2.3 Behavior Modeling