512 The Game

Andy Cox V - 12/22/2016

Preface

512 The Game was written as a way to see how much of a complex and enjoyable of a game I could fit within the first sector of a boot-able medium. The significance of the first sector is that it is loaded by the UEFI or BIOS as a boot loader to load either a secondary boot loader or the main kernel of the operating system. The first sector is always 512 bytes long with the last two bytes set as 0x55aa (in NASM displayed as 0xaa55 since the x86 architecture is a little-endian processor), so I had a very limited 510 bytes of work space. In order to make this game playable on an IBM 5150 and compatibles I had to write in Intel 8086 assembly. The source code ended up being ten pages long and the whole game is 512 bytes in size, hence the name 512 The Game.

Game Overview

The goal of the game is to collect as many tokens (flashing ♥ characters) as possible without the player moving over his/her path. If the player moves over his/her path the game terminates. The player can move across the screen both vertically and horizontally. The player will wrap around to the other side if they reach one of the screens boundaries. When the game is over a new screen will appear with the amount of tokens the player has collected and moves the player has made. A ♥ is displayed for every token collected. A ☺ will be displayed for every move the player made. If the player had collected no tokens the amount of moves will be displayed followed by a screen filled with the token character.

Movement (Press the WASD keys to move accordingly)

W – Move Player Up One Space
A – Move Player Left One Space
S – Move Player Down One Space
D – Move Player Right One Space

Characters

☺ - Player
♥ - Token
↑ - Player Path Up (Will kill player if crossed)
↓ - Player Path Down (Will kill player if crossed)
→ - Player Path Right (Will kill player if crossed)
← - Player Path Left (Will kill player if crossed)

Game Installation on Physical Machine

In order to play the game, load the 512.bin file into the first sector of a boot-able medium. A boot-able medium may be a flash drive, floppy disk, compact disc, ect. Use a third party software in order to install 512.bin file onto the first sector. A hex editor with the ability to read and write to physical mediums would work. After 512.bin has been installed on the first sector of the boot-able medium make sure to with a marker or pen label the medium 512 The Game. Place the medium into the machine (with the machine off) then turn the machine on with the medium inside. Make sure to change the boot order of your machine to load the boot-able medium with 512.bin installed on first. If the machine uses a UEFI system make sure to set the UEFI to read legacy MBR boot-able mediums. If the instructions to the installations are followed correctly the upper right hand corner should display “512♥GAME☺APCV♥2016” excluding quotations. When this string of text appears press any key to play 512 The Game and enjoy.

Game Installation on a Virtual Machine

Set 512.bin as a boot-able medium. Make sure the virtual machines settings are IBM 5150 compatible, if all the necessary procedures are done correctly the upper right hand corner should display “512♥GAME☺APCV♥2016” excluding quotations. When this string of text appears press any key to play 512 The Game and enjoy.

Sounds may be too fast or too slow. In 512 The Game the timing made for the sound is calculated by a delay using the machines system clock. On a physical machine the system clock should iterate at a rate of 18.2Hz or around 55 milliseconds. This setting may not be changeable in the virtual machine. In the source code line 443 acts as a delay time multiplier. Change the value being moved into the cx register to a larger value if the sound is too fast or a smaller value if the sound is too slow. Compile source code and create a new 512.bin file. Note: this program was written using the NASM Assembler. Line 443 in the source code is as follows:

```
mov cx, 2 ; Change this value to 18 to make it sound normal on Bochs virtual machines.
```