# Task 2

**Complete the Injectee\_Template Source Code**

**Team Members: Sayed Md Abu, Siyu Deng, Ige Tosin O**

We need to find out the value for following instruction

int numBytes

Number of bytes we would like to write to the memory. In other words, the bytes of inject instruction.

char cheat\_Instr[-1]

Modified instruction which implements the cheat function.

char original\_Instr[-1]

Original hex instruction

DWORD cheat\_InstrAddress

The memory address where we would like to modify.

Abstractly, the cheat function is implemented by overwriting Zelada.exe’s rupee collection counting instruction. Therefore, we can solve this problem by two steps:

1. Find the location of such instruction by using cheat engine.

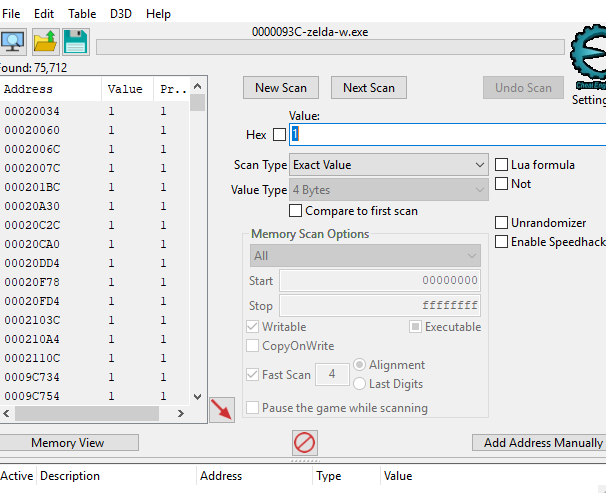
2. Turn on the cheat function and check what instruction overwrites the original one.

STEP 1:

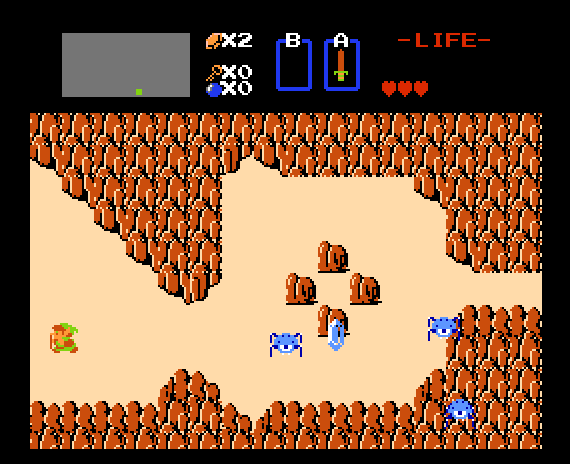
Start playing game, get score 1.



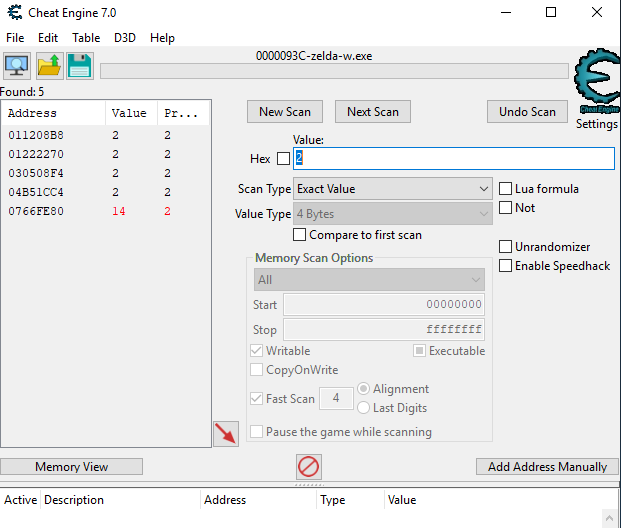
Search this exact value (1) in cheat engine.



Get another score (2)



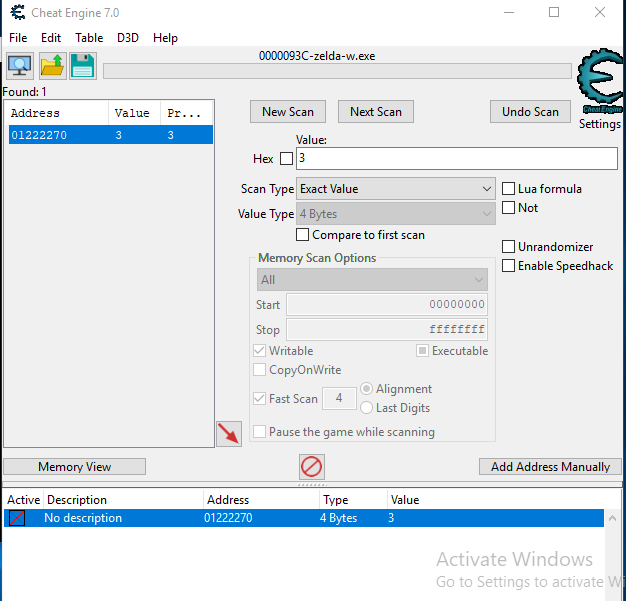
Search this value (2) in cheat engine by next value.



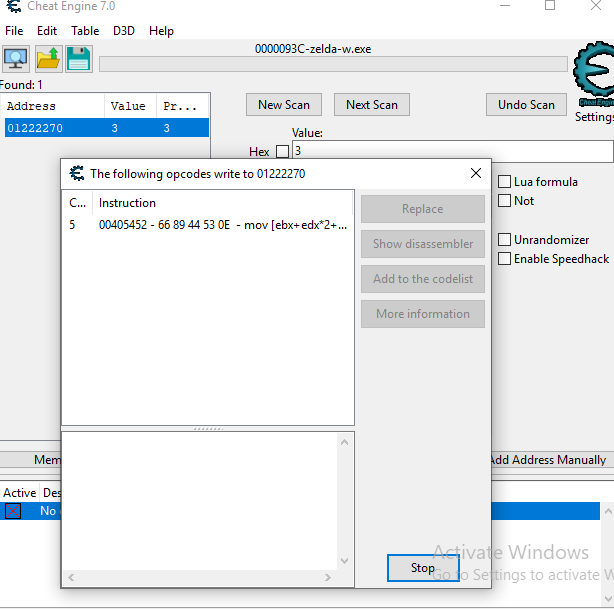
Collect another points (3)



The number of candidates is narrowed to 1. So, this is the address which holds the rupees value.



After that, we need to find out what instruction updates it and where does it locate. Let’s now turn on “Find out what writes to this address” function provided by cheat engine and collect one more rupee. Then, the cheat engine catches the instruction we want.



Now, we know:

char original\_Instr[6] = {'\x66', '\x89','\x44', '\x53', '\x0E', '\x00'};

DWORD cheat\_InstrAddress = 0x00405452

Then, let’s process to step 2 by turning on the cheat function. It shows the original function has been changed to “nop”.

Graphical user interface, text

Description automatically generated

Therefore:

char cheat\_Instr[2] = {'\x90', '\x00'};

When we try to turn on the cheat function, numBytes=2 (length of cheat instruction); when we try to turn it off, numBytes=6 (length of original instruction).

At last, the overwriteMemory function is:

memcpy(&addressToWrite, valueToWrite, numBytesToWrite);

Code of this task is stored at ./injectee/injectee.cpp.