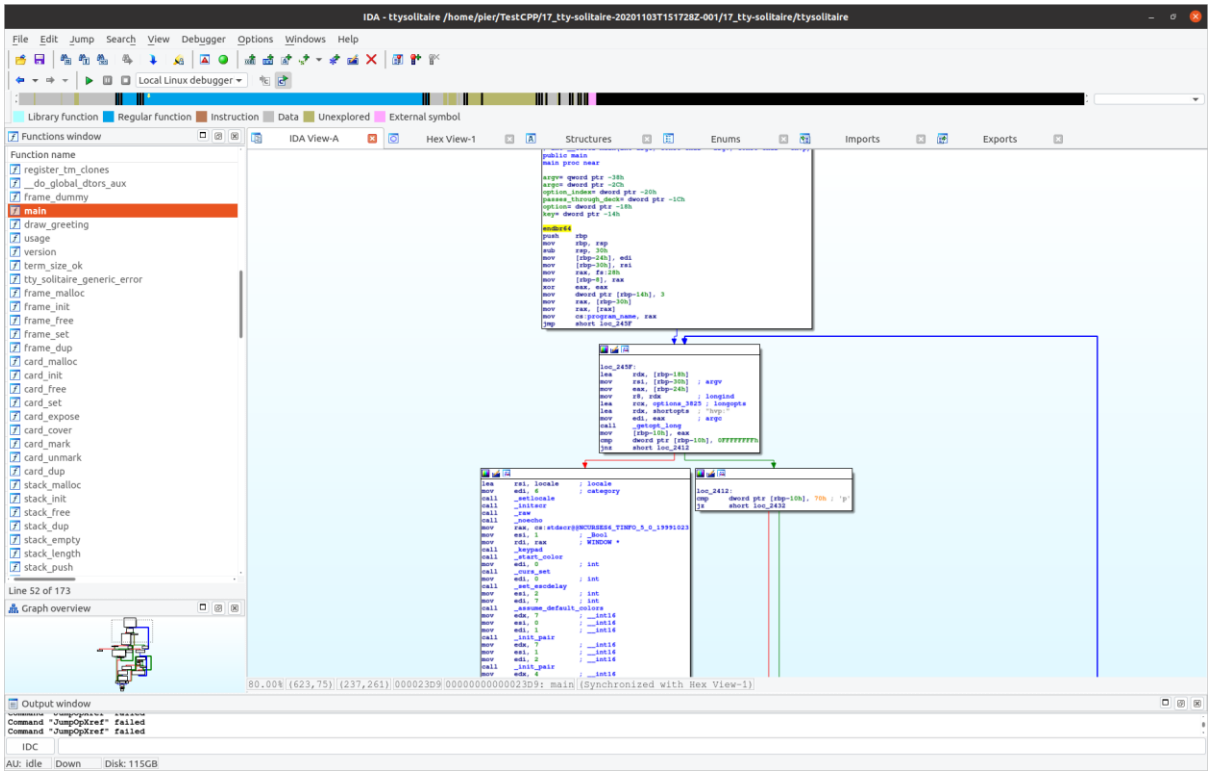
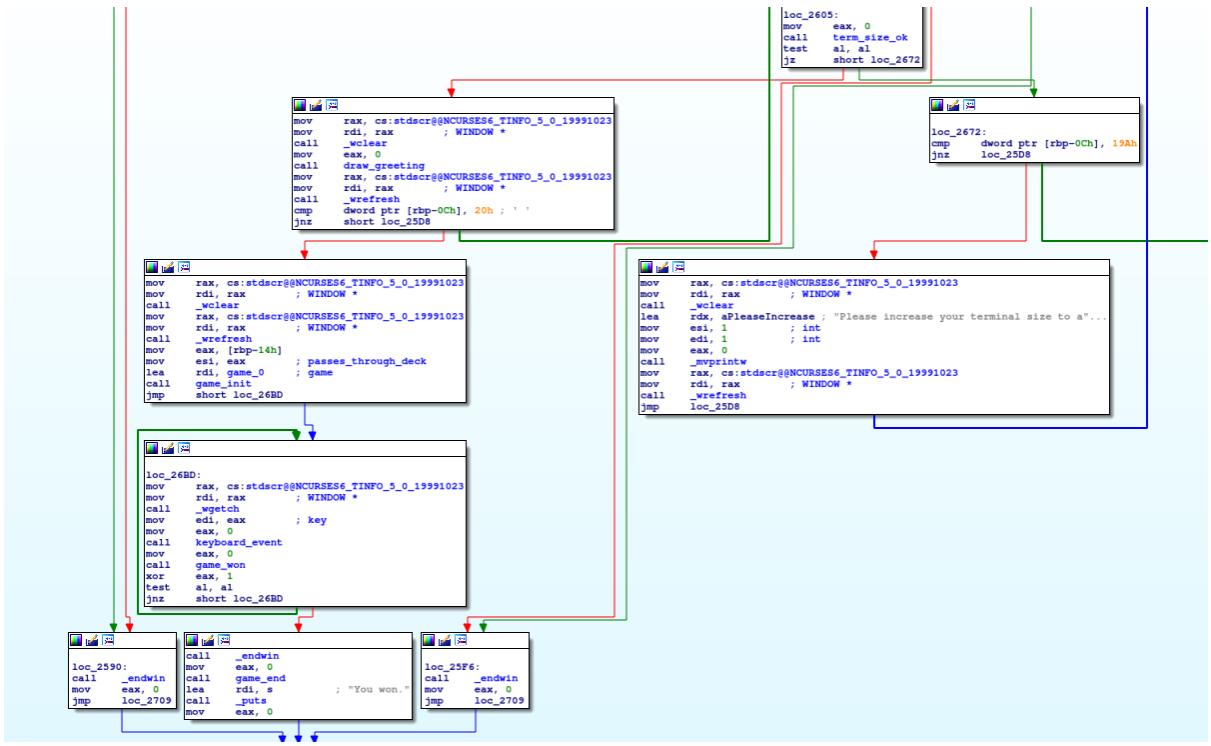
Let’s start opening the binary in IDA



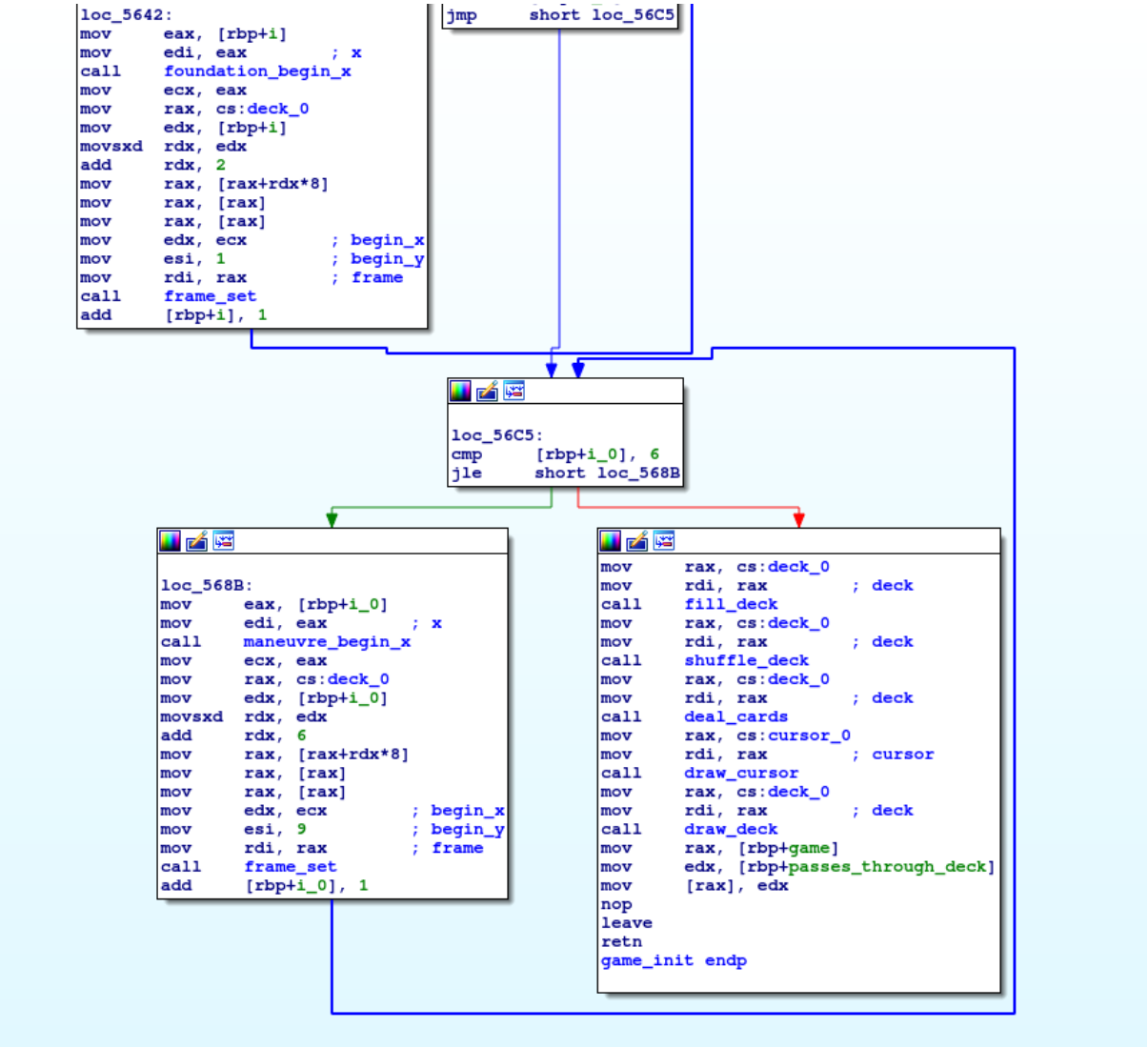
We can immediately see that we have a huge graph with a lot of functions on the left side.

Since we want to make the initial hand to be always the same, we might look for a call to a random function that initializes the deck.

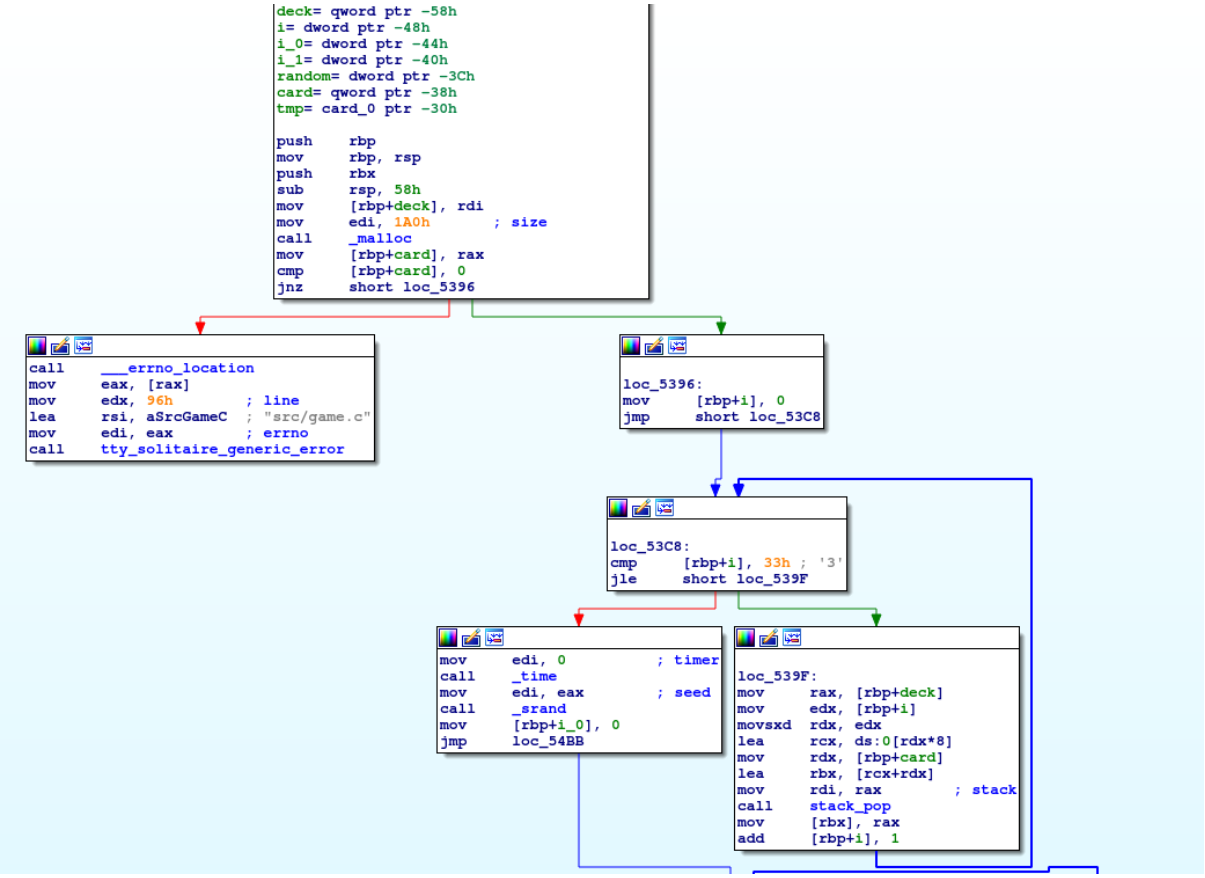
Inspecting the binary we can see a bunch of branches that check the terminal size, and an interesting loop that comprehends the string “you won”.



Before the block containing “you won” we have a loop with keyboard_event and getch, which is probably the loop that takes the user input and does stuff. The previous block contains a call to a function called “game_init”. The name suggests that it could initialize the whole game setting, maybe the deck too. Let’s inspect it!



Looking at the different calls, we have an interesting one: “shuffle_deck”. Let’s inspect it further.



Finally we found a _srand call, which means that it calls the random function to generate the cards order in the deck. Before it, we see a _time call, and the result is put into edi before the _srand call. This basically means that it uses the _time function to generate the seed to feed into the random. So, if we remove the call to _time and the mov edi,eax, in edi we will always have 0 (note the instruction right before call _time). It is enough to NOPs these two instructions (call and mov), and we are done!