Individual Project 1

| Rui(Terry) Tang  Vancouver, Canada  Rui.tang@email.kpu.ca |
| --- |

# Overall project summary

I’ve designed a simple program, or mini game is you would like, or a submarine simulator. The main component of the program consist with uses of loop, if-else if-else conditionals, and switch functions.

# INTRODUction of fuction blocks

The very first function block I used is for the formula of constructing the submarine. I set out an individual function block for such formula is mainly for convince. Since the formula is going to be repeated number of times, it is easier and tidier if I can call out the function rather than typing it out each time. The function itself is very simple, I indicated the strings global value outside all of the blocks and simply compile them together inside the function block.

Second function block is submarine customization. To better customize the submarine, I decided to use replace only portion of the string value instead of the whole line. In that way, you don’t have to count and type out all of the space corresponding to every character. Then I assigned the customized submarine with its own function value, so I can easily alter the background based on the customized submarine that was already done by the user.

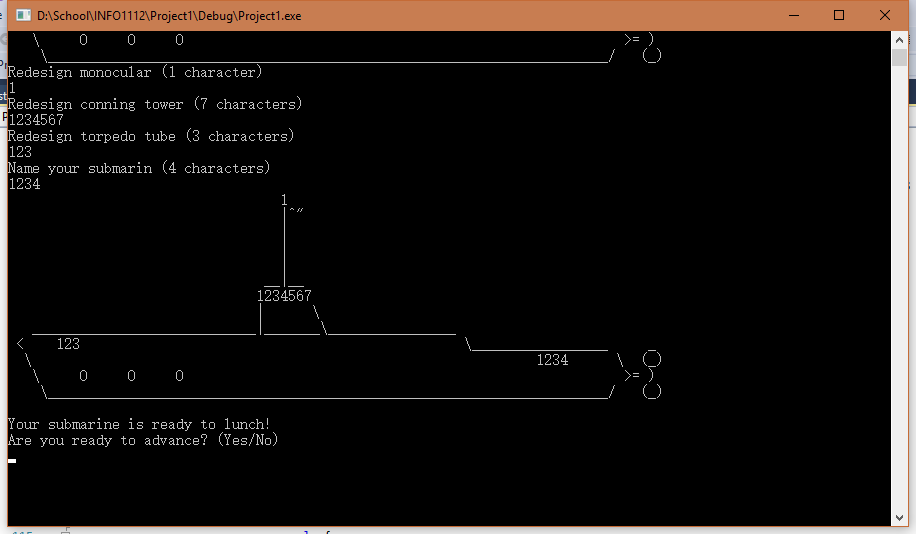
The third function block outside of the main block is my background generator. The background generator’s purpose is to construct 2 different background image that will be displaying upon the user inputs, which I’ll explain later on. The first background is when your submarine encounters an enemy destroyer; and the second background is when no enemy insight. The background plays a key part of the simulator, since the visual aids it provides help the user better understand the current scenario. First, for the first background, I defined all the string identifier that construct an enemy destroyer. Then I use the replace function to replace the blank space with fishes, which indicate the submarine is fully online underwater. After that I return the new and replaced value to the submarine function and ultimately, combine the destroyer and the submarine together. As for the second background, I just simply put some waves on top and leave some spaces, which indicate the vacancy of enemy ship.

The main block is a bit complex, it contains all the combat simulation and main user-interactions. First, I put the 2- background scenarios into two conditions; one is you encounter the enemy; and one is you encounter no enemy. To indicate whether the user encounter enemy ship, I created a number generator with the range of 0-5, and set encounter to be true if the number generated that is greater than 4, which gives you a 1/6 chance of entering a battle with the enemy. And to make the program more user-friendly, I added the option for user to either terminate the game or process on-to the next random background if there’s no encounter. When entering combat with enemy ship, I then define the hit-points and damages of everyone. The player submarine has 5 hit-points and torpedo does 10 damages; the enemy destroyer has 10 hit-point and does 2 damages with its dud-mines. To make this a combat more realistic, I use the RNG to greater simulate whether the weapon hits and cause catastrophic damages, hits and cause moderate damages, hits the armored area, and missed; therefore, the damages dealt are going to be completely random and hit-point of individual are subtracted after each turn. To give the user more freedom of action, during battle the user is given 2 choices. First choice is to attack enemy ship, which derive 2 possible outcomes. One is you sunk the enemy ship and the other one is you damage the enemy ship and now it can attack you back with dud-mines. I use switch function to indicate the 2 choices, and prompt the user input to corresponding calculation and display scenario outcomes. Second choice is to run away from battle. Initially I used the RNG that gives out a 2/3 chance of a successful run away. However, I then felt like the chances are too high I then change it to ½. In shorts there’s a loop inside a loop which are prompt by two bool variables, encounter and combat. The battle occurs only when encounter is true, that means when the no enemy insight and after enemy ship destroyed, the encounter value must set back to false if no combat is to occur immediately after. And when encounter is to be true, first it’ll pullout the background indicator, then the combat loop kicks in and will loop only when combat is set to false which is at the end when one of the side is sunk.

# Reference of ascII arts

Different from the previous lab which I design a program to modify the ship, I picked the submarine as my main component while corporate the ship element as the enemy into the simulator.

# (n.d.). Retrieved October 25, 2017, from http://www.chris.com/ascii/index.php?art=transportation%2Fnautical



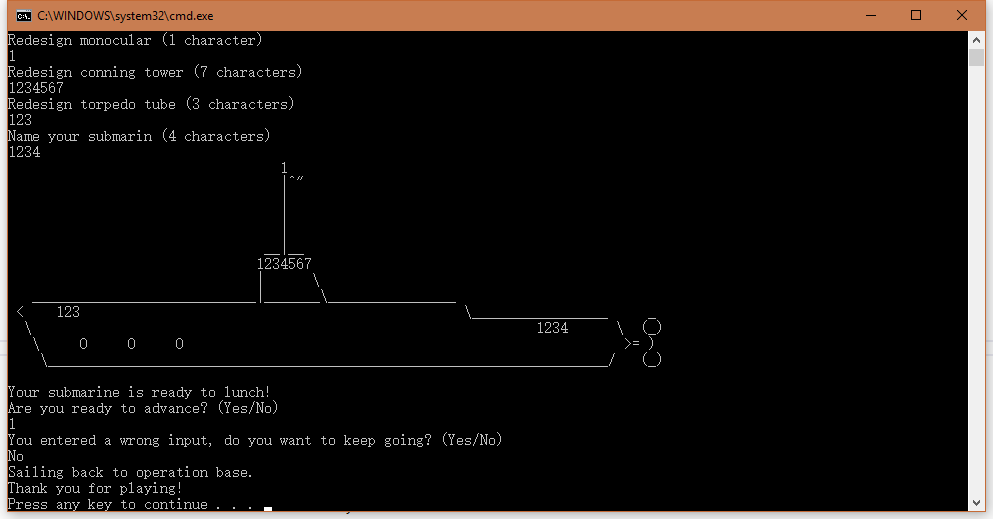
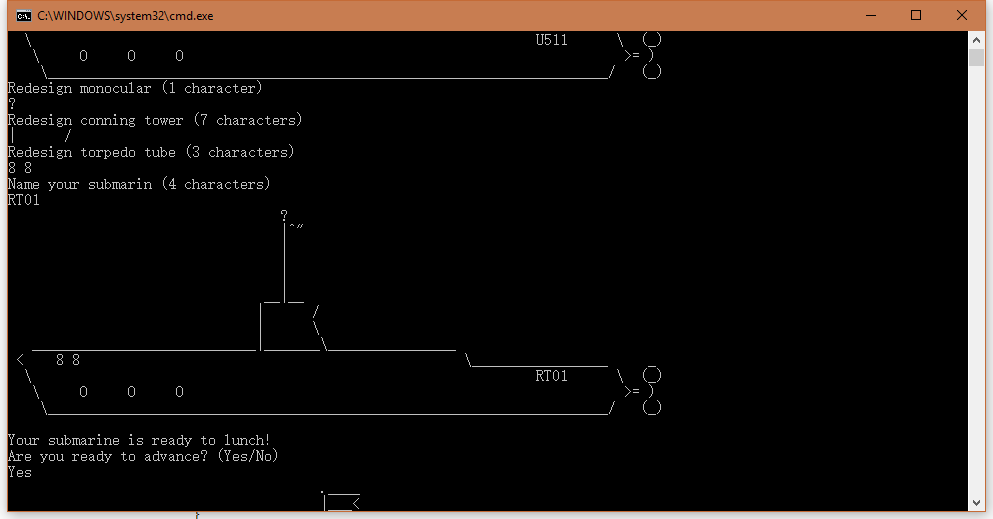
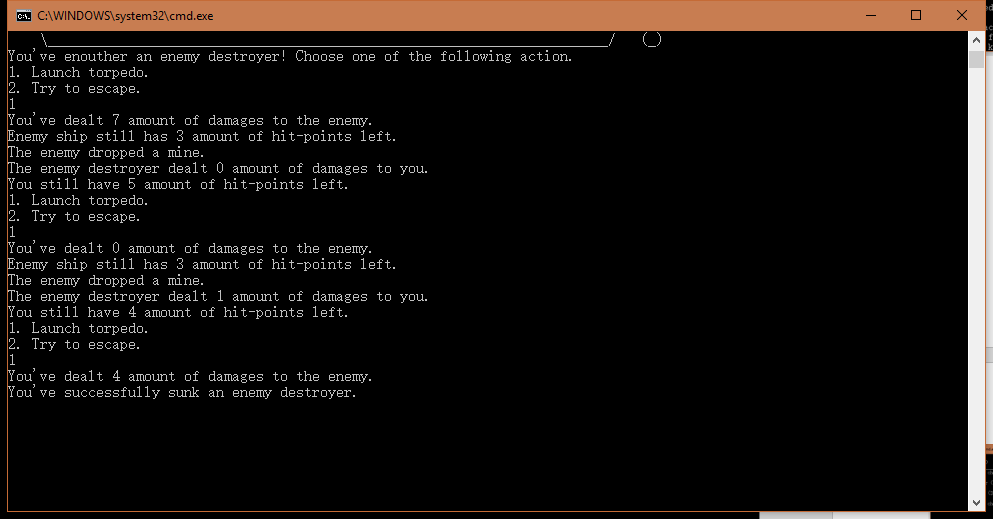


Figure 1&2.Test run of not advancing into the water at all.

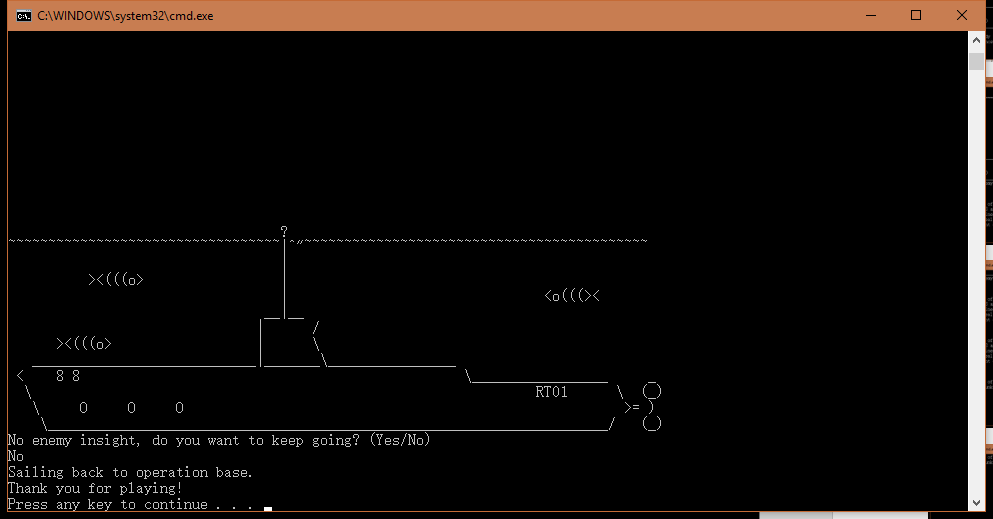
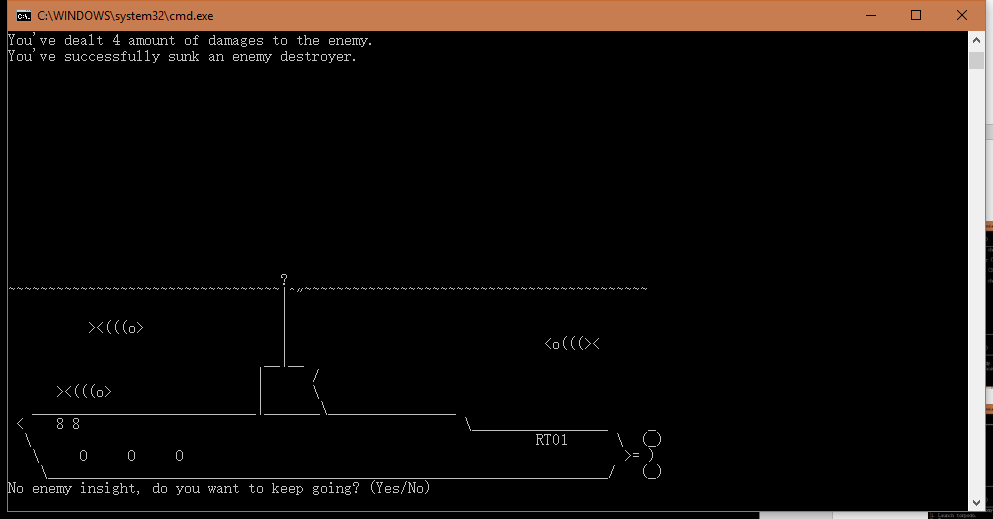
 

Figure3-7. Simulation run of a compete battle.

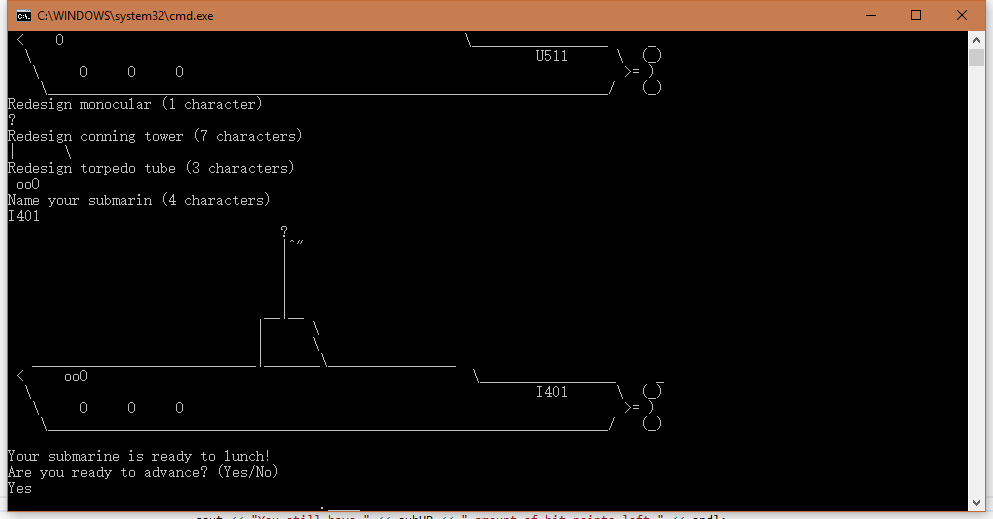
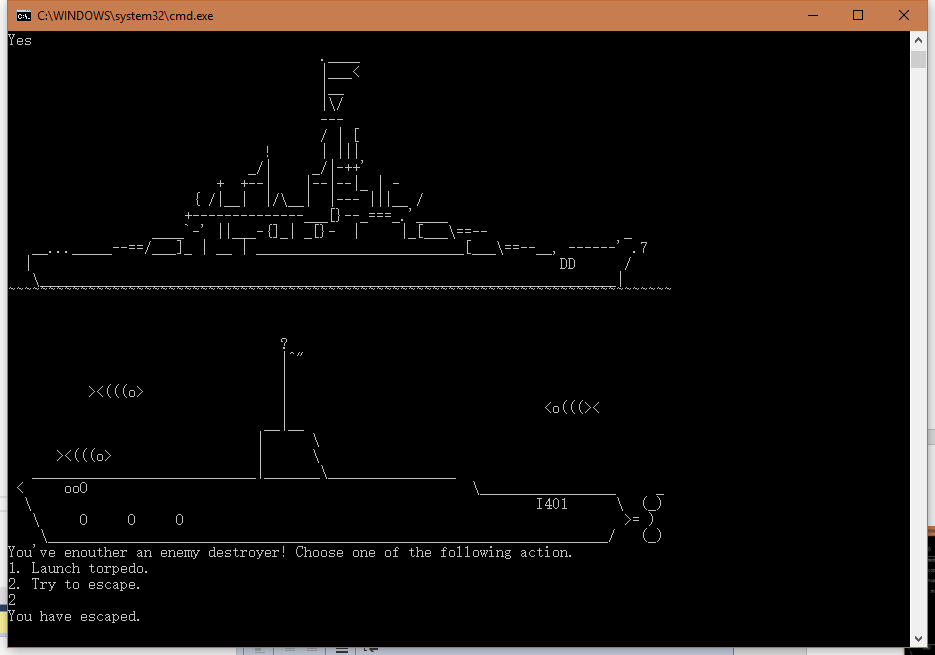
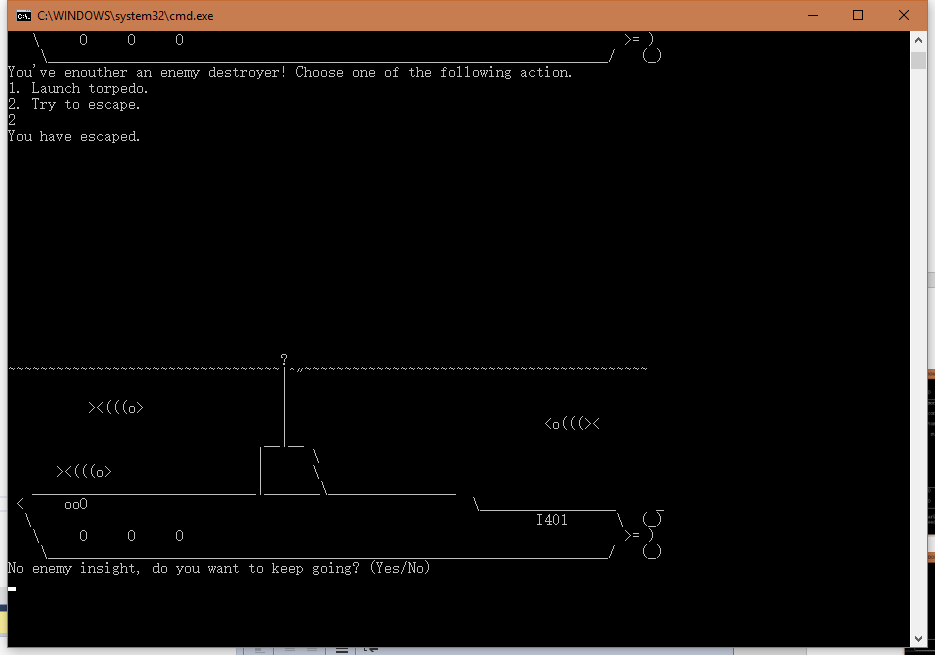
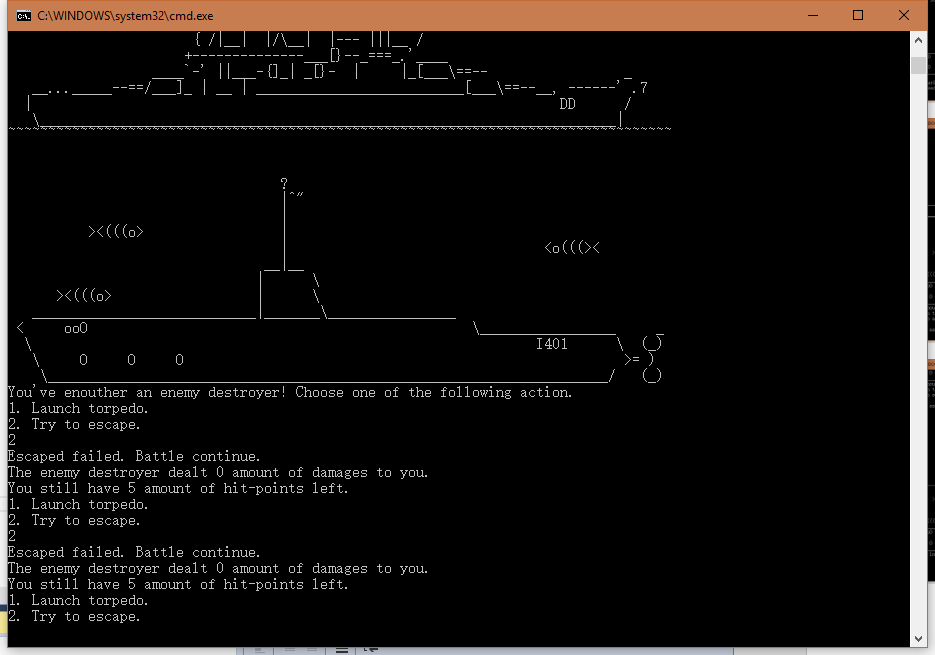
   

Figure 8-11. Complete simulation run of successed and failed escapce