Design Document

Overall Approach:

I divided my program into these parts:

* Game part: controls the game logic
* It contains global turtle screen.
* Provide a function call start\_screen to initiate the game.
* Other function like pause for other internal use.
* Prompter part: Controls turtle to print in game messages
* Provide global turtle intro.
* Add and event: add event to the mainloop to return whether it result win or lose.
* Snake part: Controls global turtle to print snake
* Provide global turtle head , new\_segments
* Provide function snake\_move: it includes condition for the new snake body to follow with the head. Condition for the snake to move accordingly with the user’s input. It also includes timepause condition to pause the timer when the game stops
* Monster part: Controls turtle to print monster
* Provide global turtle monster
* Provide function monster\_move: it includes the condition for the monster to follow snake by the difference in coordinate. If also include monster and snake contact count
* Food part: Controls turtle to print food
* Provide function consume: it includes condition to detect which food has been eaten by the snake by using indexing methode. It also includes turtle new\_segments to add the body accordingly with the food eaten

Data Type:

* Game:
* screen = turtle.Screen()
* contact: int
* second: int
* Prompter:
* Intro = turtle.Turtle()
* Context = str
* Snake:
* Head =turtle.Turtle()
* Headdirection = str
* Predirection = str
* Body = list
* New\_segments = turtle.Turtle()
* Monster:
* Monster = turtle.Turtle()
* Xcoor = int
* Ycoor = int
* Food:
* Food\_list = list
* Index = int
* Total = int
* Others:
* Temp = int
* E\_delay = int
* Counter = int
* Check = bool
* Timepause = int
* Stopgame = bool
* Nopause = int

Motion Logic

Snake:

* Set snake to position (0,0)
* Calculate snake direction accordingly with user’s prompt
* Update the displacement
* Check snake food by indexing
* Update new body segment (tail)
* Place tail coordination behind snake’s head coordination
* Check contact with monster
* Check condition if snake tail length equals to 50
* Show game win prompt

Monster:

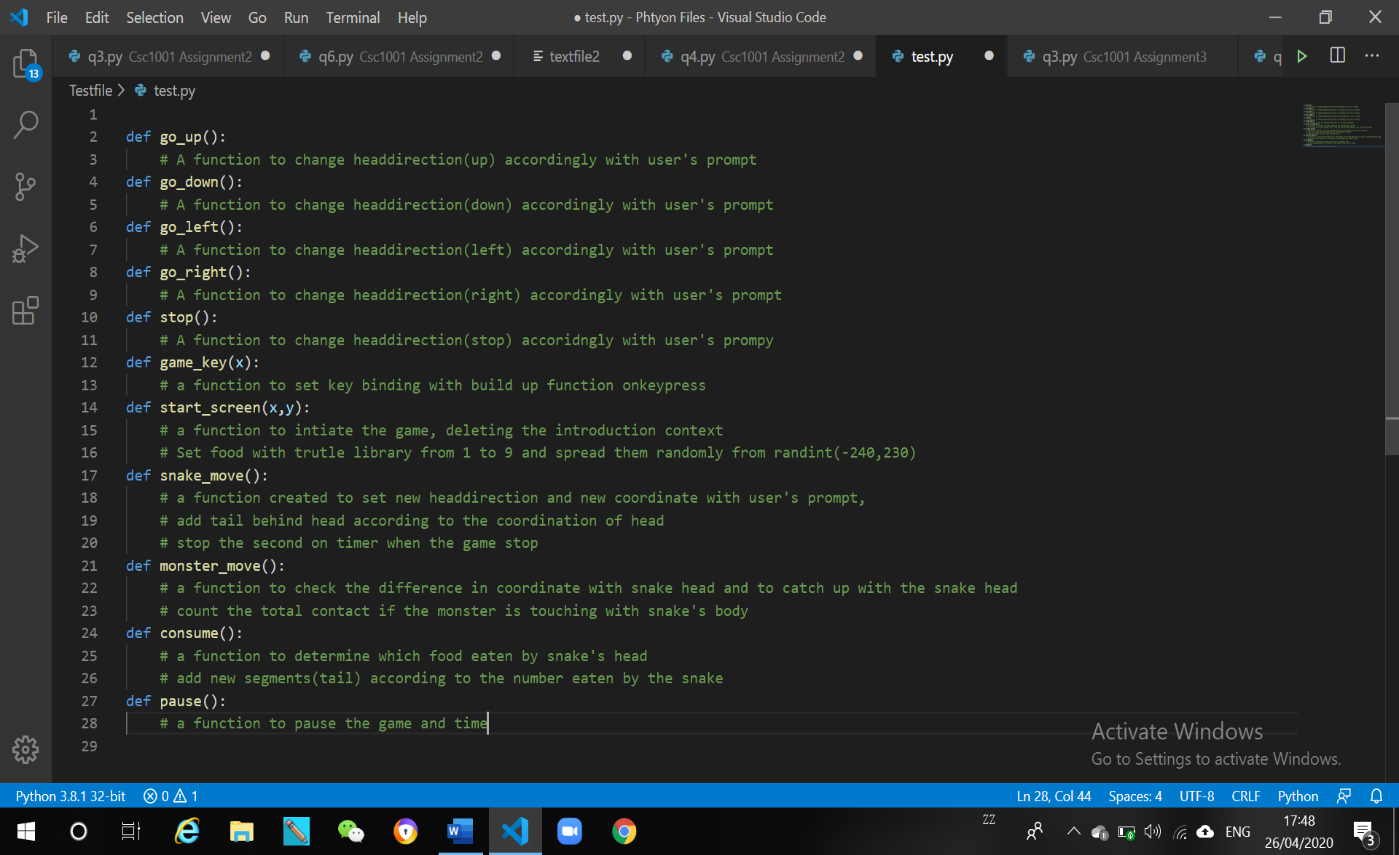
* Set position of monster with the differences with snake head coordinate
* Calculate monster displacement with user’s prompt accordingly with the snake head
* Check contact with snake head
* Show game lose prompt

Tail Expansion logic

* Detect snake’s food number by indexing on the empty list.
* Create new segment with turtle method with different color with head
* Move new segment to head’s coordinate with build up function xcor() and ycor()

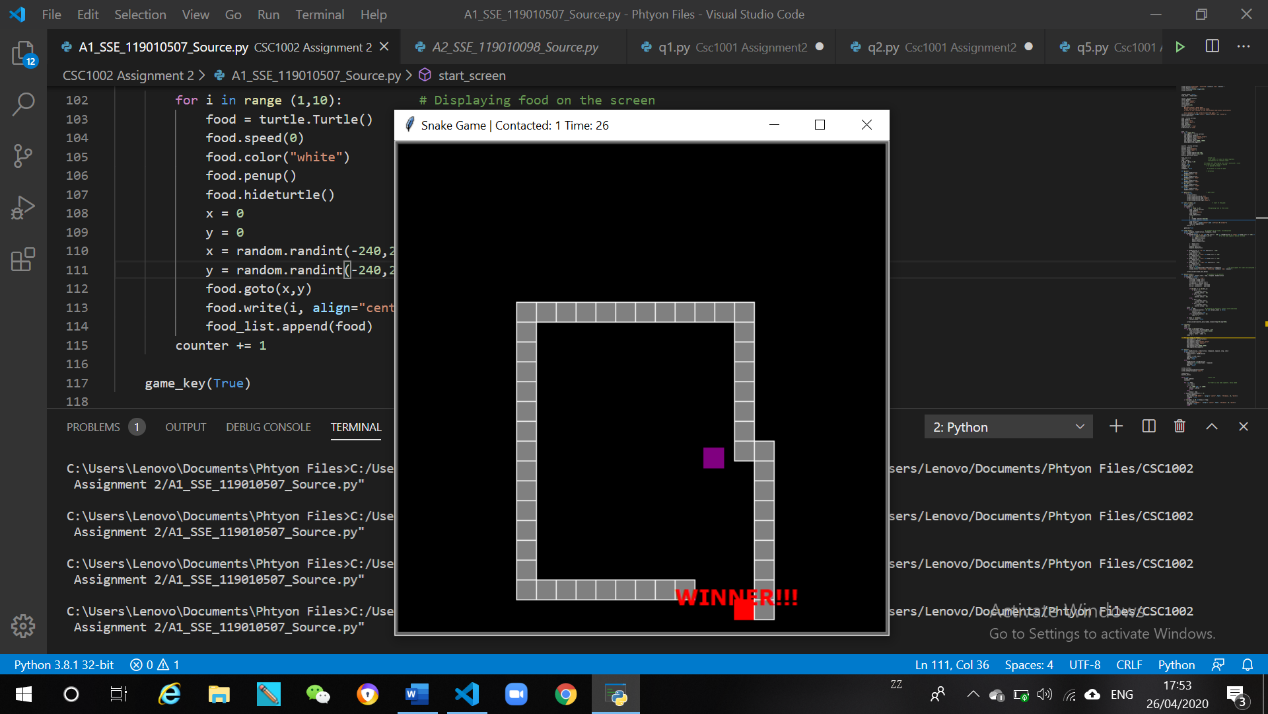
Body Contact logic

* Both monster and snake are moving in the same time with different direction
* Create condition if snake(tail) and monster coordinate are less than 20 pixels by using build up function distance() than contact number will added by 1

Function

Output

Game win



Game Lose

