Snake Game

# Design Document

### By Mal (Daniel) Lambert for CSC221 Final Project

## **Introduction**

**Project Functionality**

While my game will work according to several logical conditions under the hood that the player won't be aware of, the ideas on the surface are very simple: navigate a snake around a limited arena, eat food to grow the snake, and avoid navigating into the arena's walls or the snake's own growing tail.

**Design Process**

After outlining the logic my game needs to follow and writing the pseudocode for it, my plan is to write one function at a time before moving to the next one. I expect each piece of logic I outlined will take one function to implement. As of yet haven’t learned how the graphical elements will be implemented within each function, or if in a separate function.

## **Project Development**

**Pseudocode**

**Environment functions:**

arena\_size(height, width)

* define tile size within

food\_placement()

* generate 1 food at a time on random tile (if 0 food, place 1 food)
* if food would be placed where snake head or tail currently is, do not place food there
* remove food when snake head enters tile with food, add 1 to snake\_length()

**Snake functions:**

snake\_length()

* default: 3
* add 1 length for 1 food eaten

snake\_location()

* default: near arena’s center-left
* track which tile snake head is on within arena
* move 1 tile forward at the pace of snake\_speed(snake\_length())
* if snake head moves outside of arena\_size(), move to game\_over(‘You ran into the wall’)
* if snake head moves into snake\_tracking(), move to game\_over(‘You ran into your tail’)

snake\_speed(snake\_length())

* set speed to (snake\_length() \* 0.5) tiles per 1 second (will adjust through playtesting)

snake\_direction()

* default: right (or 2, if directions are 0,1,2,3)
* player input changes snake's direction

snake\_tracking(snake\_length())

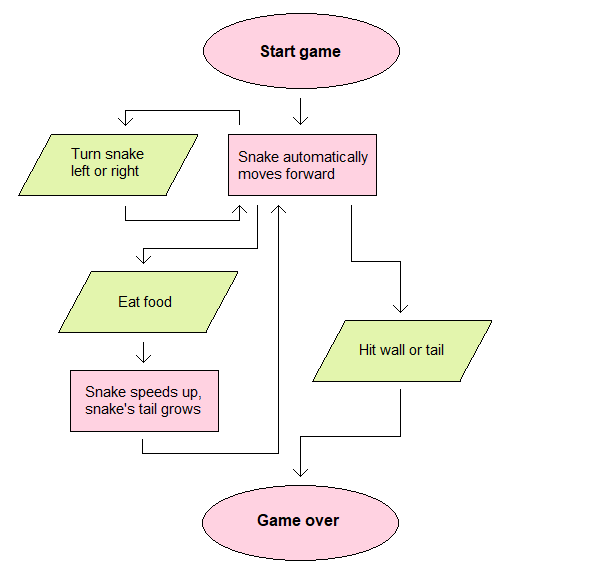
* keep track of previous turns snake has made to correctly display snake's growing tail

**Game over functions:**

game\_over(reason\_for\_loss)

* end gameplay loops
* display snake\_length() for player's score
* type y to play again, type n to exit game

**Flowchart**



**Requirements**

**• Includes user interaction (e.g., command-line inputs, GUI, keyboard movement,**

**etc.).**

Not accomplished

**• Performs some processing (e.g., conditional logic, loops, function calls, math**

**operations).**

Not accomplished

**• Generates cleanly formatted output based on input/processing.**

Not accomplished

**• Implements at least one custom function/method/module.**

Not accomplished

**• Use an imported Python library.**

Not accomplished (but will be as soon as I start work, as pygame fulfills that requirement)