

Assignment 2 – Phase 1

1. Overview of the Assignment

1.1 Learning objectives

By doing this assignment you will learn how to:

- Drawing using Graphics2D and geom's shape objects (Phase 1)
- Handle collisions between objects using Shape/Area objects (Phase 2)
- Search an item within a collection using linear search algorithm (Phase 2)

1.2. Programming Requirements

Phase 1 requirements

You will continue working on your simulation from Assignment 1 Phase 2

- 1) Redraw your creature, food and environment using geom's geometric primitive objects. **No Graphics' old drawing methods** should be used for drawing any of these visual objects' components.
- 2) Add food using mouse by **double-clicking** at a location. The **longer** you hold the mouse button pressed, the **larger** the food portion should be.
 - **PPP required here:** Please write a block comment above the real-code method, which outlines the execution logic for how food's method gets mouse position from its caller object, and pass it to the food's draw method. You should do it before doing your real code. NO separate PDF needed.
- 3) When you **Ctrl-Click** on a food it should be removed from the location (disappear).
- 4) You must use **MouseAdapter** rather than the interface for handling *MouseEvents*.
- 5) You should be able to create **several** food objects at the same time, and store them in an **ArrayList**. When food is created, you add it to the list, when food is eaten or Ctrl-Clicked, remove it from the list.
- 6) For food pursuing, let your creature always pursue the food that is currently the first in the ArrayList, regardless distance and/or size (we will make it smarter in Phase 2). It will turn to the next food only when the current top one in the list is gone.
 - **PPP required here:** Please write the pseudocode as a block comment above the real-code method, which outlines the steps for identifying the current top food in a list and then approach and eat it. You should do it before doing your real code. NO separate PDF needed.
- 7) When your creature eats all the foods you placed, it should continue moving along an angular direction (i.e. involve velocity along both X and Y), turning at the edges of the environment when hitting one. **Please Note:** For any collision detection, you can still use

the old way, as what we have recapped in week 4 lab, to deal with it, and we'll leave the advanced approach to Phase 2.

- 8) All newly added fields must be **private** (except for constants) and provided **minimal public interface methods** when necessary. Also the names of the class, fields and methods must follow the **naming convention** with appropriate upper- or lower case for the **initial** and **CamelCase** thereafter.

2. Submission and Grading Instructions

2.1 Phase 1

- The project must be named (**in Eclipse**) with the following format **LabNumber_FirstName_LastName_AssignmentNumber_PhaseNumer_StudentNumber**, e.g.
D104_Jim_Silvester_Assignment2_Phase1_1234567
- For the code the **entire project** MUST be submitted (not just the source files)
- **To submit, export the project** (including all the **libraries used**) **into a zip file** (Archive File) and name it exactly **the same as the project name**

(Please note failure to meet any of these submission requirements above would result in a penalty of 0.25 pt each)

- **No late submission will be accepted.** If you do not complete the assignment by the deadline, you will receive **0**. For a legitimate reason a late submission might be allowed pending **discussion with your TA before the deadline**. You may be required to provide supporting documents.
- For the coding, **Make sure your code is syntax error free so that it runs properly on the lab machine.** You would receive **0** for the coding part if your code failed to run due to syntax and/or errors.

You are graded on **completeness**, **correctness**, and **visual style**.