

CSC71001

Programming I

Assessment 2

Details:

Title: Practical Skills
Weighting: 15% of unit grading
Due: Monday Week 7 (16th December 2019 at 11PM)

Specifications

Overview:

Your task is to create a game in Greenfoot, with three types of elements: a **PLAYER** object, controlled by the player; **FOOD** objects that can be 'caught' by the player; and **ENEMY** objects, that can 'catch' the player. If the enemy catches the player, then the game is over.

You must **create a new scenario** and you must choose a theme for your game that is **not** crabs/worms and lobsters. All your elements should suit your theme, including the background and the actors. The movement of the actors should 'make sense' as per the theme of your game. We do not expect to see the same theme or game created by any two students – be original!

Details:

Player:

- You will create a Player class.
- At the beginning of the game, there must be one PLAYER object on the screen.
- The PLAYER must be controlled by the keyboard, and at minimum must move automatically and have left and right turn. For example, when the left arrow key on the keyboard is pressed, the PLAYER will turn to the left while moving forwards. When the right arrow key is pressed, the PLAYER will turn to the right while moving forward.

Food

- You will create one Food class.
- There must be *eight* FOOD objects on the screen at the beginning of the game.
- Each FOOD object must have random movement on the screen – that is, it must turn in random directions and move at a random speed. The food must be able to be caught by the PLAYER.
- When the FOOD is caught by the PLAYER, it should be removed from the screen.
- Later in the Portfolio, you will use the Food class to create different types of Food objects so you will need to think about the theme for your food carefully.

Enemy:

- You will create an Enemy class.
- There must be *at least one* ENEMY object on screen at the beginning of the game.
- Each ENEMY must at minimum, move at a constant speed and turn in random directions and should be different than the food (cannot use the exact same code).
- If the ENEMY catches the player, the game should end.

Sound:

- You should include sound effects that fit the theme of your game. These can be either in-built or created by you. At least one sound should be created/recorded by you in Greenfoot.
- You should include sound for when the PLAYER is caught by the ENEMY.
- You should include sound for when the PLAYER catches or eats a FOOD.
- You should include sound for when the PLAYER wins the game.

Additional Details:

You can choose to use the inbuilt media for backgrounds and actors OR you can choose to add your own (see Module 4 for how to do this), or some combination of the two. If you do add your own, make sure you use PNGs with transparency for your actors, and keep your file size small.

General criteria: playability, accuracy, careful coding, maintainability of the code, commenting, choice of names for classes, methods (and variables if necessary).

Enhancements for extra credit:

You may like to add the following features, for extra credit:

- Use alternate keys to move the player “up”, “down”, “left” and “right”.
- Add animation when the PLAYER is moving.
- Add animation when the FOOD is moving.
- Add animation when the ENEMY is moving.
- Use your own images or images sourced from the internet. These must be referenced in your documentation and commented in your code.
- Use your own sounds or sounds sourced from the internet. These must be referenced in your documentation and commented in your code.
- Add a score which displays how many FOOD pieces they have caught.

Submission:

You must export your game as both:

- a JAR file (application); and
- a Greenfoot archive (gfar) file.

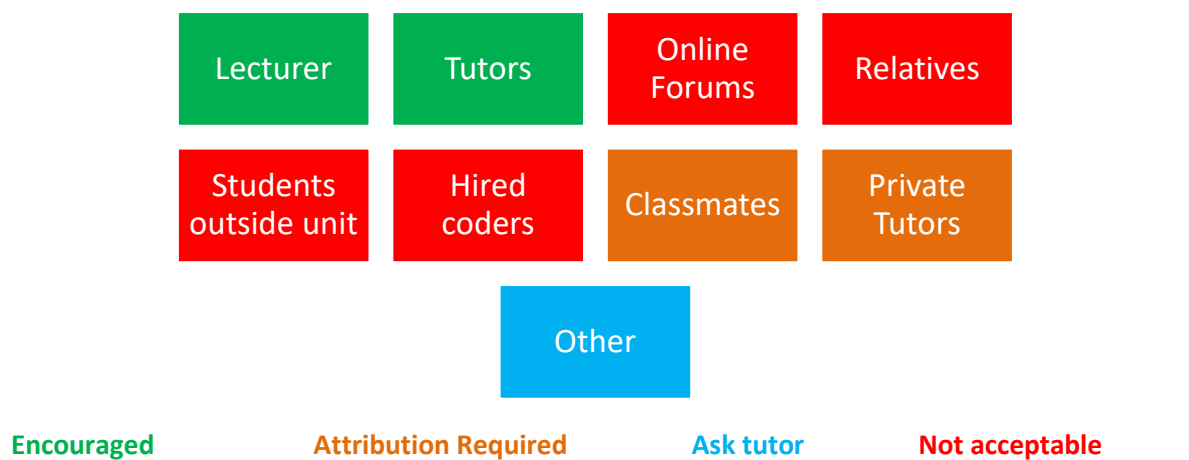
Please name your file appropriately (see below):

e.g. *yourSCUusername_Ass2.jar* and *yourSCUusername_Ass2.gfar* and
yourUsername_Enhancements.doc

If you do not submit in the above format, your assignment will not be marked. Submit both of these to MySCU site under the “Assignment 2” link. Make sure you Submit (not just Save) by the due date. Your tutor will contact you if they have any questions about your submission.

Getting Help

Who can you get help from? Use this diagram to determine from whom you may seek help with your program.



This assignment, which is to be completed individually, is your chance to gain an understanding of the fundamental concepts of object-oriented programming and coding syntax on which later learning will be based. It is important that you master these concepts yourself.

Since you are mastering fundamental skills, you are permitted to work from the examples in the study guide or textbook, but you must acknowledge assistance from other textbooks or classmates. In particular, you must not use online material or help from others, as this would prevent you from mastering these concepts.