Faculty of Science, Engineering and Technology

Introduction to Programming

Tutorial Task 4.2: Gosu Cycle

Overview

This program will allow you to use the Gosu cycle to manipulate moving widgets on the screen.

Purpose:	Modify a simple Ruby program to move a shape across the screen.
Task:	Modify the tasks in the Gosu cycle *update()* and *draw()* methods,
Submit To:	Canvas when complete
Time:	This task should be started in your fourth lab class.
Resources:	Sobkowicz, M 2015 *Learn game programming with Ruby : bring your ideas to life with Gosu*, The Pragmatic Bookshelf Gosu Ruby Documentation Gosu site Gosu game video tutorial

Submission Details

You must submit the following files to Canvas:

- Basic gosu_cycle_example.rb source code
- Screenshot of the window showing the execution of your program.

Make sure that your task has the following in your submission:

- The program must move a shape across the screen using code in the appropriate sections.
- Code must follow the Ruby coding convention used in the unit (layout, and use of case).
- The code must run and the screenshot show it working.
- This program does NOT need to have a procedure for main – the Gosu cycle is the main cycle.
- Your program must have the indicated local variables, and use them appropriately.

Instructions

You must enhance the code provided as follows:

- Add a variable in the *initialize()* method called *shape_x* with the initial value of zero.
- Add code into the *update()* method that will add 10 to *shape_x*.
- Add code into the *draw()* method that will draw a shape (square or circle of any visible colour) at the y coordinate of 30 and x coordinate of *shape_x*.
- You could use the following line to draw your shape:

```
Gosu.draw_rect(@shape_x, @shape_y, 50, 50, Gosu::Color::RED, ZOrder::TOP, mode=:default)
```

Use the code provided (from this task's resources in Canvas) to get started.

Now that the Task is complete you can submit it for assessment, which will help prepare it for your portfolio.

1. Use your preferred screenshot program to take a screenshot of the program's window, as this is one of the things you will need to submit.
2. Once you get things working you **do not** want to lose them.
   - Work on your computer's storage device most of the time... but backup your work when you finish each task.
   - Use **Dropbox** or a similar online storage provider, as well as other locations.
   - Canvas is not a Backup of your work, so make sure you keep a copy!
   - A USB key and portable hard drives are good secondary backups... but can be lost/ damaged (do not rely upon them).
3. Login to Canvas, and locate Tutorial Task 4.2
4. Change the status of the task to **Ready To Mark**
5. Upload your completed code and the screenshot.
6. If you check back later Canvas will have prepared these as PDFs for your tutor to assess.

Note: This is one of the tasks you need to **submit to Canvas**. Tutors should give guidance and perhaps feedback in the tutorial class.
Check the assessment criteria for the important aspect your tutor MAY check when assessing your finished portfolio.

End of Task