

10.009 The Digital World

Term 3. 2018

Problem Set 11 (for Week 11)

Most recent updated: January 29, 2018

Due dates:

- **Cohort session problems** : Following week: Tuesday 11:59pm.
- **Homework problems** : Same as for the cohort session problems.
- **Exercises**: These are practice problems and will not be graded. You are encouraged to solve these to enhance your programming skills. Being able to solve these problems will likely help you prepare for the end of term examination.

Objectives

1. Develop GUI application in Kivy.
2. Create window and common widgets.
3. Invoke callback function to bind event

Note: Solve the programming problems listed below using your IDE. You may use the given template files to help you, but it is not required. Make sure you save your programs in files with suitably chosen names and in a newly created directory. For each problem, find a way to test the correctness of your program. After writing each program, test it, debug it if the program is incorrect, correct it, and repeat this process until you have a fully working program. Show your working program to one of the cohort instructors.

Problems: Cohort sessions

Complete all questions and get a checkoff from an instructors.

1. *GUI: alternate message:* Write a program that alternates between two messages displayed on a canvas, "Programming is fun" and "It is fun to program," as shown in the Figure 1. The program should alternate based on a touch down event, such as a left mouse click. Use the `bind` method with the event `on_touch_down`. Create a callback to a method called `def alternate(self, instance, touch)`.

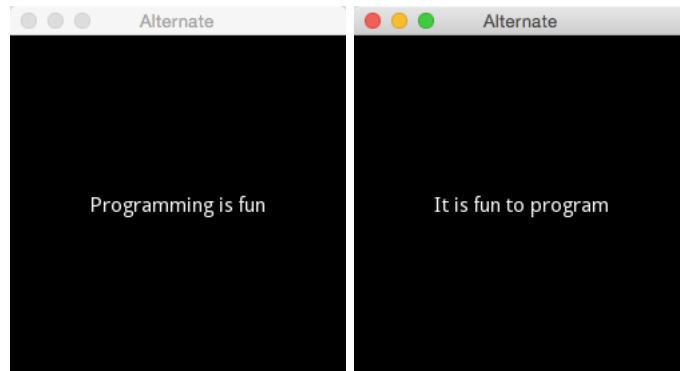


Figure 1: Alternate message on a mouse click.

2. *GUI: slide detection* Write a program that displays a label "Slide Me" at the beginning. The program should handle the `on_touch_move` event on the label by calling the method `def detect(self, instance, touch)`. The method should change the text of the label based on the direction of the slide movement. For example, when the user slides the screen to the left, it should display "Slide Left". When the user slide the screen to the right, it displays "Slide Right" (Figure 2). Do the same with up and down. You can use `touch.dx` and `touch.dy` to detect the slide movement.
3. *GUI: investment-value calculator:* Write a program that calculates the future value of an investment, at a given interest rate for a specified number of years. The formula for the calculation is as follows:

$$futureValue = investmentAmount \times (1 + monthlyInterestRate)^{years \times 12} \quad (1)$$

Use text fields for users to enter the investment amount, years, and interest rate. Display the future amount in a text field when the user clicks the *Calculate* button as shown in Figure 3.

4. *GUI: switch screen:* Write a program to switch between screens. When the program runs, it should display two buttons: one should bring the user to the Settings screen, while the

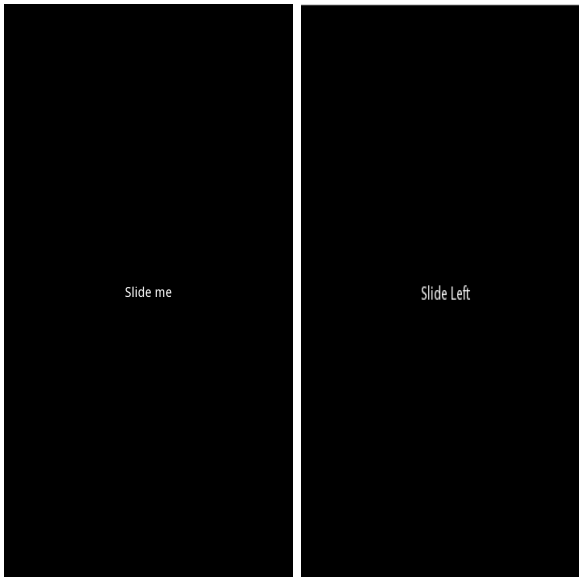


Figure 2: Slide Detection Application.

The image displays two screenshots of a software application titled "Investment". Each screenshot shows a form with four input fields and a "Calculate" button. The top screenshot has the following values: "Investment Ammount" (10000), "Years" (3), "Annual Interest Rate" (3.25), and "Future Value" (11022.66). The bottom screenshot has: "Investment Ammount" (15000), "Years" (7), "Annual Interest Rate" (5.25), and "Future Value" (21644.44). The "Calculate" button is present in both.

Figure 3: Investment calculator.

other one should allow the user to quit the application. If the user presses the Settings button, the screen will bring the user to another screen. In this second screen, it will display a label with the text "Settings", and a button "back to Menu". If the user press the button "back to Menu", it should bring the user back to the previous screen. See the snapshots to have a better idea of the question requirements.

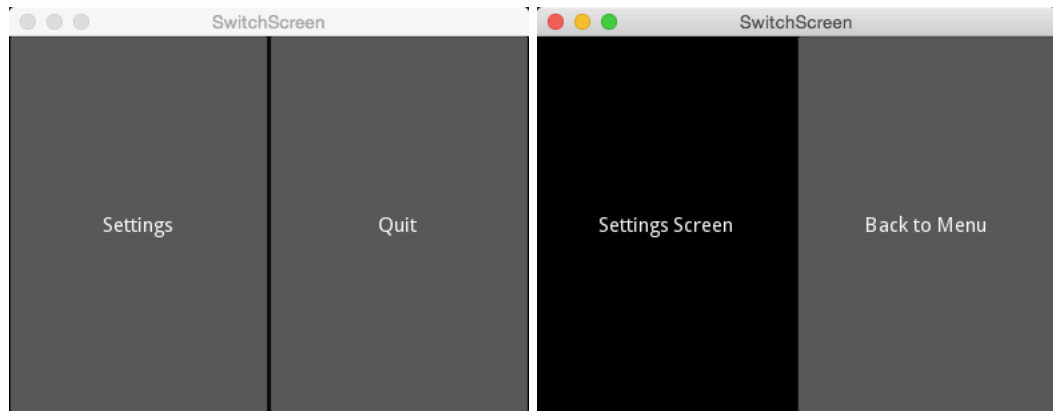


Figure 4: Switch Screen.

End of Problem Set 11.