Course	BSCH & BSCO Year 3
Stage / Year:	3
Module:	HCI & GUI Programming
Assignment No:	1 Building a working Projectile Calculation Tool
Deadline:	Sunday 23nd October at 23:55 (any changes to this date will be on Moodle)
Submission:	Upload to Moodle
Weighting:	10% of module

Introduction

In this assignment you are required to produce a working application, which will perform calculation about the firing of a projectile. The GUI should follow the basic structure indicated in Figure 1a & 1b but you are encouraged to make improvements to the interface by using some of the principles from the lectures.

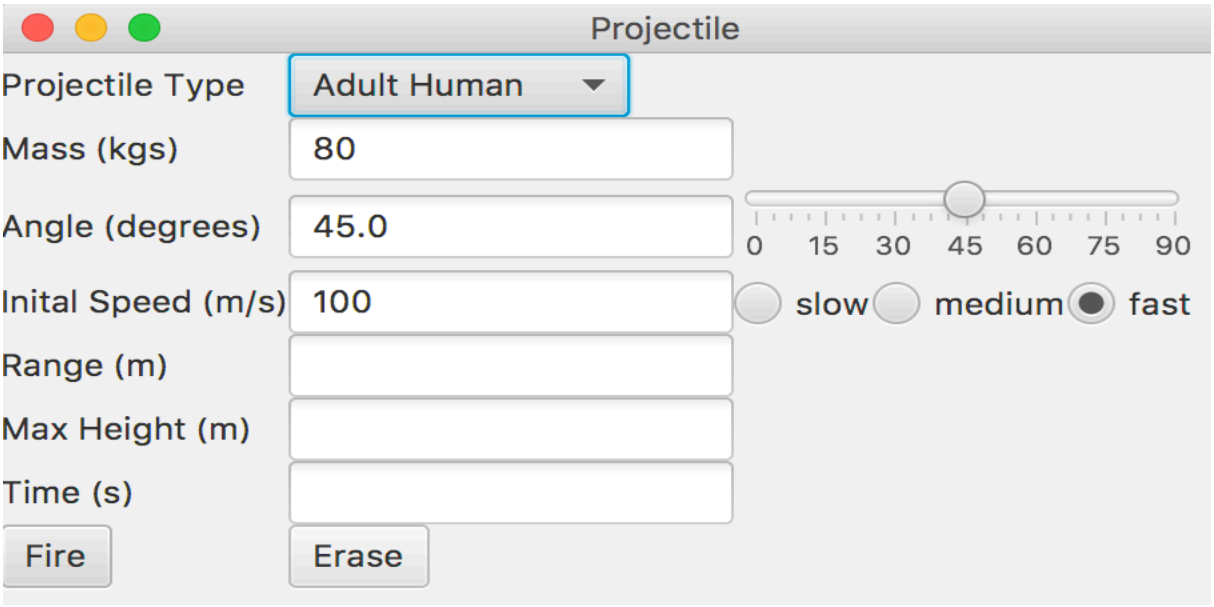

Figure 1a: Initial launch state of the application

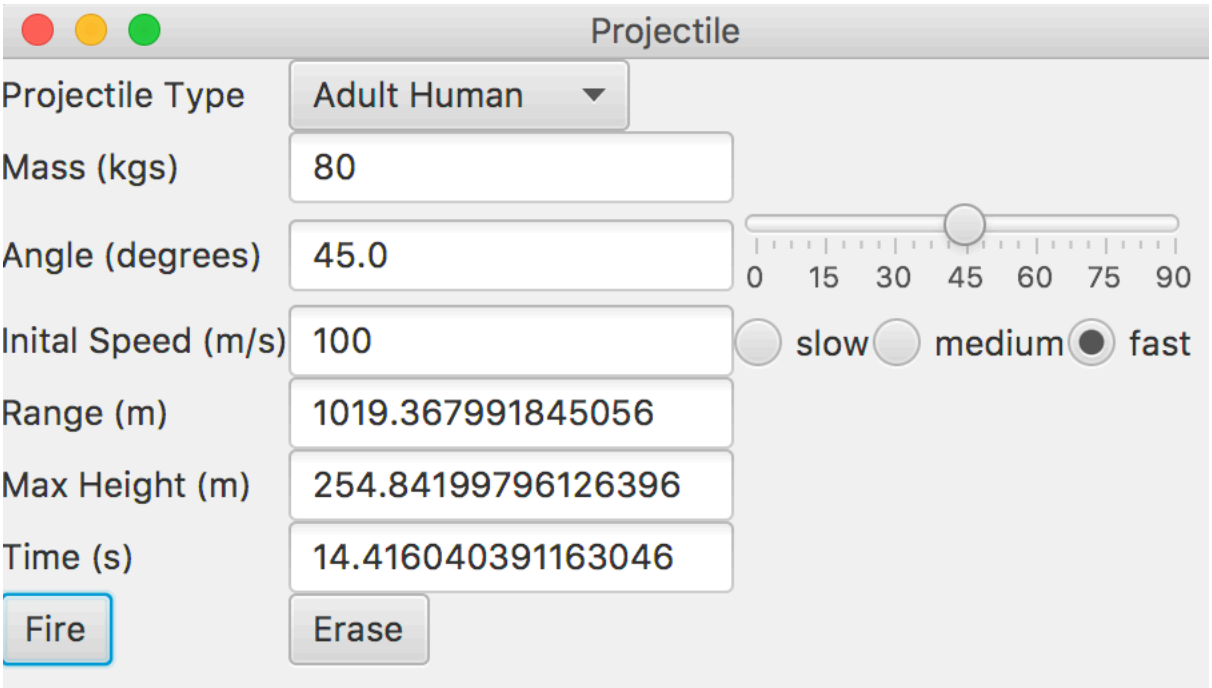

Figure 1b: Application state when the *Fire* button is pressed directly following the launch of the application.

Created by Alex Cronin edited on 2016.11.10 13:00

Template File

- You will be given a template java file to add to.
- The lecturer solution is approximately 280 lines of code including comments. The template file is approximately 160 lines.
- There are many lines of code, imports etc. missing.
- The amount of space in the template files does not indicate the number of lines of code required.
- You should not change the method or variable names as this will make your work very difficult to support and correct.
- You will learn more if you comment your code in a detailed fashion.

Applications to give you ideas

- A very fancy version of an application written using different technology is available here http://www.convertalot.com/ballistic_trajectory_calculator.html The calculations are performed in this version are a little different to they way we want to do them so do not compare their results to yours e.g. the results boxes track the height and time in real-time as opposed to displaying the final result.
- A very simple version of an application using a different technology is available here http://www.convertalot.com/ballistic_trajectory_calculator.html . The calculations results presented here can be used to double-check your answers.

You are not required to fully understand the physics behind this application. It is only necessary that you harvest data from the various input controls and use the equations below to compute and display the results.

Equations

Glossary of notation

- **d** is the distance or range of the projectile
- **V_0** is the initial velocity or initial speed of the projectile
- **g** is the acceleration due to gravity and is a constant on earth of 9.81m/s/s
- **θ** is the the launch angle (be sure to convert your input from degrees to radians before using the Math.sin() method. You will need to search to find an appropriate method.
- **t** is the flight time of the projectile
- **h** is the max height reached by the projectile

The Equations

Range:

$$d = \frac{v_0^2}{g} \sin(2\theta).$$

Max Height:

$$h = \frac{v_0^2 \sin^2(\theta)}{2g}$$

Time:

$$t = \frac{2v_0 \sin(\theta)}{g}$$

A full explanation of the equations is available here https://en.wikipedia.org/wiki/Projectile_motion

Note: Mass is not involved in these calculations. If you wish to extend your application to include a result

Created by Alex Cronin edited on 2016.11.10 13:00

involving Mass this would earn your additional marks.

User Tasks and Marking Scheme

Task No.	Detail	Marks (100)
1	The application should load and display all the relevant controls in a similar layout.	20
2	The controls should contain the default values including the *Angle* Slider and the *Initial Speed* ToggleGroup.	5
3	On pressing the *fire* Button the values for *Range, Max Height* and *Time* should be calculated and displayed.	20
4	On changing the *Projectile Type* from *Adult Human* to *Piano* the following changes should take place a. *Mass* should change from 80 to 400 b. *Angle* should change i. from 45 to 20 (in both the TextBox and the Slider) ii. from 90 degrees max to 40 degrees max (Slider) iii. from ticks every 15 degrees to ticks every 10 degrees (Slider) c. *Initial Speed* should change i. from 100 to 10 (in the TextBox) ii. from the *fast* RadioButton to the slow RadioButton d. *Range, Max Height* and *Time* should be blank. e. Changes to a, b and c should happen in reverse when the *Projectile Type* is changed back from *Piano* to *Adult Human*.	20 (5 x 4 marks)
5	On moving the *Angle* Slider the *Range, Max Height* and *Time* should be updated.	5
6	On changing the *Initial Speed* Radio Button the *Range, Max Height* and *Time* should be updated.	5
7	The only TextBox which accepts input directly is the *Mass* TextBox, all other TextBoxes should not be editable.	5
8	When the *Fire* Button is pressed and the *Mass* TextField does not contain a numerical value an appropriate error message should be displayed in the *Mass* TextField.	5
9	When the *Erase* button is pressed the application should initialize the controls based on the current state of the *Projectile Type* ComboBox.	5
10	Additional layout improvements using Gestalt Principles, the functionality of the application must not be reduced.	5
11	Additional functionality improvements make sure to clearly highlight it in the UI and comment it in the code so I can see it!	5

Submission Notes

- You will be given approximately 2 weeks to do this assignment.
- The deadline for this assignment is displayed on Moodle.
- You are required to submit a single archive file to the Moodle using the file name conventions detailed in the *Assessment, Submissions and Requirements* Lecture.
- Penalties that will be applied should your submission fail to meet certain standards. These are detailed in the above set of lecture notes.

Additional Notes

The JavaFX lab book will only give you the pieces to construct widgets and perform event handling on them. You are required to generate the rest of the application yourself. It will be a long time before you will see if your application is working fully, but be sure to test each additional code block before proceeding. As a suggestion whenever you add event handlers or methods you should test them will simple print statements to the console to ensure that you are receiving events correctly.