Lab Assignment #3

Due Date: Week 6

Purpose: The purpose of this lab assignment is to:

- Develop Android Apps that contain Graphics and Animations
- Practice the use of frame-by-frame animations
- Practice the use of tweened animations

References: Textbook, ppt slides, class examples, and Android documentation (https://developer.android.com/guide/topics/graphics/overview.html , http://developer.android.com/reference/android/view/KeyEvent.html, https://material.io/icons/#ic_keyboard_arrow_down). This material provides the necessary information that you need to complete the exercises.

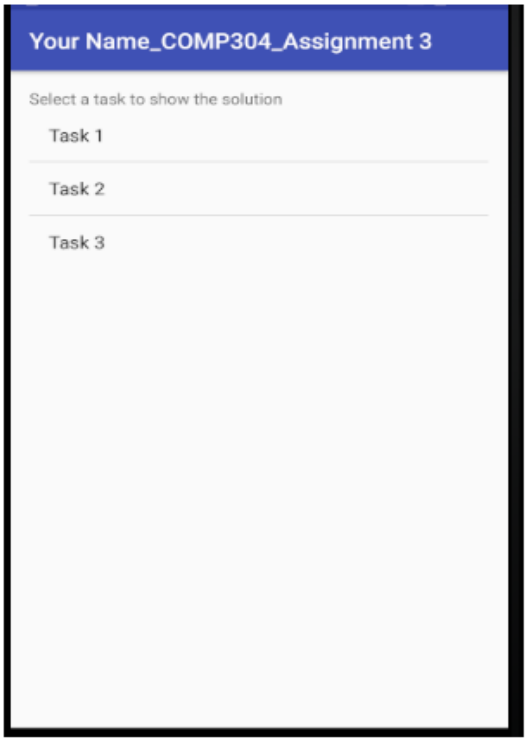
Be sure to read the following general instructions carefully:

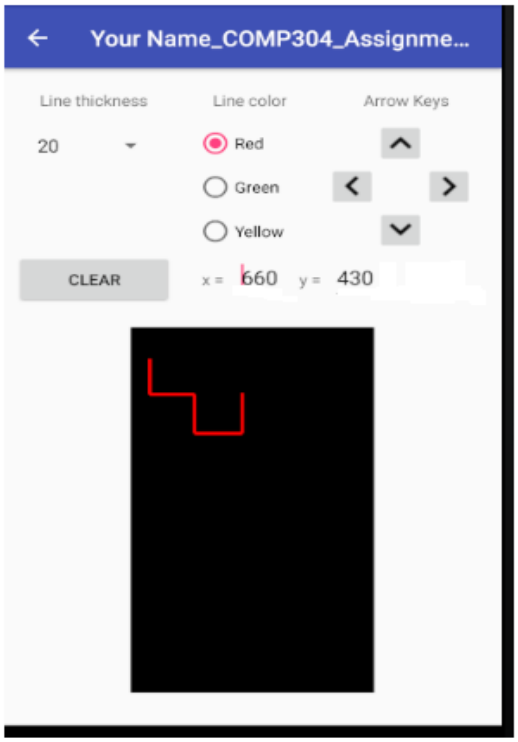
-This assignment must be completed **individually** by all the students.

- You will have to **demonstrate your solution in a scheduled lab session** and upload the solution on e-Centennial through the assignment link.

Start a new Android Application and name it according to rules given in lab assignment 1.

Task 1





The main activity of this application uses a ListView control to allow users to select a task to show the solution as shown in the picture above.

Create an activity that allows the user to draw continuous horizontal or vertical lines, starting from a given position, using up, down, left, and right **keys of Android keypad** as well as **image buttons in UI**, as shown in the picture above. Use a Canvas object to implement the drawings.

Allow the user to choose *colors*, the *thickness* of lines (At least 3 colors and 5 line thickness) and a clear button to *clear* the drawings and *restart*. The image buttons for arrow keys are provided on e-Centennial. Use code from **CanvasPaint** example from week 5 examples.

Task 2

In this task, you will develop a frame-by-frame animation. Create a series of pictures that differ slightly from each other. Your animation should be original, relate to things that you like or express an idea of yours. Allow the user to start and stop the animation. You may use code from **FramedAnimation** example of week 5.

Task 3

In this task, you will create a tweened animation similar to **TweenActivity** from Lecture 5 examples (**ShapeShifter**). Create an application to simulate a sun view. Display a png image of the sun and the earth image slightly above the sun image. Apply the necessary transformations (*rotate*, *scale*, *translate*) to make the earth revolve nicely around the sun.

Define tweening transformations as XML resource files. Allow the user to start and stop the animation.

Assignment and project naming rules are according to rules given in lab assignment 1.

Academic honesty (Plagiarism and cheating)

All students must follow the academic honesty policies regarding Plagiarism and cheating on assignments, Quizzes or Tests. Centennial college's Academic Policy will be strictly enforced. To support academic honesty at Centennial College, all academic work submitted by students may be reviewed for authenticity and originality, with utilizing software tools.

For more details, please visit the Academic Honesty site on https://www.centennialcollege.ca/mycentennial/your-support/academic-support/student-academic-advising/academic-honesty/

Evaluation Rubrics:

Criteria	Not Satisfactory 0-1	Satisfactory 2	Very Good 3	Excellent 4
Task 1: Specification and coding	The code produces incorrect results due to poor understanding.	The code produces correct results but does not display them correctly due insufficient analysis.	The code works and meets most of the other specifications. Very good code analysis.	The code works and meets all the specifications and well thought out code analysis.
Layout Design and Resources file	Poor design and no resources files used.	Good design with fair use of resources file and good user-friendly design.	Very good design with appropriate use of resources file with very good user-friendly design.	Excellent design with excellent use of resources files and excellent user-friendly design.
Task 2: Specification and coding	The code produces incorrect results due to poor understanding.	The code produces correct results but does not fully implemented.	The code works and meets most of the other specifications and good implementation.	The code works and meets all the specifications and well implemented.
Task 3: Specification and coding	The code produces incorrect results due to poor understanding.	The code produces correct results but does not fully implemented with XML animations.	The code works and meets most of the other specifications and good implementation with XML Animations.	The code works and meets all the specifications and well implemented with XML Animations.
Program Delivery	The assignment is completed (50-70%) with poor code standard and submitted within three days late with in correct format (.zip file format)	The assignment is completed (70-80%) with good code standard and submitted within two days late with correct format (.zip file format)	The assignment is completed (80-90%) with very good code standard and submitted within a 24 hour late with correct format (.zip file format)	The assignment is fully completed (100%) with excellent code standard and submitted on time with correct format (.zip file format)
Grade	0-5	10	15	20