University of Huddersfield
School of Computing and Engineering
Department of Informatics
In-course Assignment Specification

Module Code and Title: CFT2112 Studio 1	
Assignment No. and Title: 2: 2D Graphics Application	
Assessment Tutor: Dr Minsi Chen	**Weighting Towards Module Grade:** 60%
Date Set: Thursday 18/01/2018	**Submission Date:** Thursday 26/04/2018

Learning Outcome(s) covered in this assignment:
Knowledge and Understanding

1. Have a sound knowledge of basic scripting and programming concepts
2. Be aware of typical software structures and design/development procedures

Abilities

1. Design appropriate technical solutions to a problem or brief, including recognising the risks/safety issues related to their safe operation
2. Construct prototype software
3. Test and evaluate a developed software prototype

Level of Collaboration:
This is an individual assignment.
Collaboration is not allowed.

1 Introduction

This assignment consists of a set of programming exercises on building the fundamentals of a 2D graphics pipeline. You will be given a VS2015 project titled `TinyRaster` which contains a partially completed software rasteriser for 2D graphics. Your objective is to complete the exercises specified in Section 2. Each exercise has a maximum score you can achieve as specified in the brackets. This score also indicates the relative difficulty of each challenge.

In order to develop a robust and correct solution to each exercise, you must investigate the relevant topics using both lecture materials and your own reading. You will be given guidance on creating solutions to Ex1.1, Ex2.1 and Ex2.2 as part of the tutorials. Please make sure you carefully follow the tutorial materials when they are released.

2 Part I: Programming Exercises (65%)

2.1 Ex1: 2D Line Rasterisation

Ex1.1 (Max 10pts) Complete the implementation of `Rasterizer::DrawLine2D` method. This method currently consists of a partially implemented Bresenham algorithm. You must extend its implementation so that it is capable of drawing 2D lines with arbitrary gradient (slope).

Ex1.2 (Max 5pts) Extend the implementation of `Rasterizer::DrawLine2D` so that it is capable of drawing lines based on a given thickness.

2.2 Ex2: 2D Polygon Rasterisation

Ex2.1 (Max 5pts) Implement the `Rasterizer::DrawUnfilledPolygon2D` method so that it is capable of drawing an unfilled polygon, i.e. only the edges of a polygon are rasterised. Please note, in order to complete this exercise, you must first complete Ex1.1 since `DrawLine2D` method is reusable here.

Ex2.2 (Max 10pts) Implement the `Rasterizer::ScanlineFillPolygon2D` method method so that it is capable of drawing a solidly filled polygon.

2.3 Advanced Operations

Ex1 (Max 5pts) Extend the implementation of `Rasterizer::DrawLine2D` so that it is capable of interpolating colour across a line when each end-point has different colours.

Ex2 (Max 5pts) Extend `Rasterizer::ScanlineFillPolygon2D` method so that it is capable of alpha blending, i.e. draw translucent polygons.

Ex3 (Max 15pts) Implement `Rasterizer::ScanlineInterpolatedFillPolygon2D` method so that it is capable of performing interpolated filling.

Ex4 (Max 5pts) Implement `Rasterizer::DrawCircle2D` method so that it can draw a filled circle.

3 Part II: Report (35%)

In addition to completing the programming tasks, you are also to required to summarise and discuss the results of your work in a short technical report. The word count limit of your report is 500 words and the smallest font size is 11pts.

Your report should roughly consist of the following core sections:

Introduction: Inform your readers what this report is about.

Methods and Results: Present how you approach to the programming tasks along with results.

Discussion: Provide a thoughtful analysis of your work based on the presented results. What are the strengths and weaknesses?

References: List the materials including books, journal papers and website that you used in your work. Please note, this section does not count towards the word limit.

The writing style must be appropriate to university level work. You should use concise and precise language to avoid verbosity. Timely citation must be used if claims and ideas are not your own work.

APA referencing style is recommended for this assignment. Please refer to `https://library.hud.ac.uk/pages/apareferencing/` for more detail.

4 Submission

The solution to the exercises must be uploaded to CourseResource by **23:59 Thursday 26/04/2018**. A submission point will be made available two weeks before the deadline.

In addition, you must package your solution including source and executable according the following requirements:

- Your package must be compressed into a ZIP file using the naming convention `STUDENTID_TinyRaster.zip`.
- Your package must contain the following directories and contents
  - Source - for source and Visual Studio project files
  - Executable - for standalone executable(s)
  - Your report in PDF format (please do not submit MS Word file)
- All standalone executables must be built for RELEASE; test these thoroughly before making your submission
- The Source directory must contain the entire source code and the corresponding solution/project files
- To reduce the file size, intermediate files generated by Visual Studio should be removed from your submission

5 Assessment and Grading Criteria

5.1 Assessment Criteria

Each exercise listed in section 2 will be assessed according to criteria listed in the table below:

Criterion	**Weight**
Correctness and robustness	60%
Techniques and methods	30%
Code and Structure	10%

Table 1: The assessment critera for each exercise.

The correctness and robustness of your submitted solutions will be evaluated using the test cases defined in the `TinyRaster` framework. In turn, you can use these test cases as a means of testing your implementation.

For your code to be effectively assessed, you `must` provide clear and concise comments in your code to explain the techniques and methods adopted for solving the exercises.

5.2 Grading Criteria

Grades	Criteria
80-100	The submitted work is exceptional and shows a comprehensive awareness of the dimensions of the topic. The solution demonstrates a level of novelty and offers an extension to existing techniques. Discussion and analysis are presented in a scholarly and scientific manner. Accurate data and strong evidences are used to support the discussion of any novel approach.
70-79	The submitted work is excellent and shows a full awareness of the dimensions of the topic. The solution is highly optimised and demonstrates individuality and creativity. Discussion offers insightful information and are well presented. Analysis is supported by accurate data and resources. **Correctness:** Your provided solutions are not only correct for all test cases, but also demonstrate additional features with high complexity and a certain degree of originality. **Techniques and Methods:** Your implementation is functional and fully optimised; you are fully aware of the runtime resource management and computational efficiency; the code base demonstrates some more advanced usage of OO design paradigm and is extensible. **Code:** Excellent use of object-oriented programming paradigm; new functionalities and their implementation fully adhere to code reusability and extensibility.
60-69	The submitted work demonstrates some advanced techniques and includes other relevant aspects that shows a clear awareness of the dimensions of the topic. The implementation is efficient and scalable. Discussion and analysis are critical and well structured; data and resources are accurate. **Correctness:** Solutions and results are completely correct for all cases. **Techniques and Methods:** Your implementation is functional; you are fully aware of the runtime resource management and computational efficiency; the code base demonstrates some advanced usage of OO design paradigm and is extensible. **Code:** Very good use of object-oriented style coding; the code is highly reusable and extensible.

50-59	The submitted work demonstrates advanced features and the implementation. The solution evidences a clear understanding of the problems and solutions, but it lacks in depth analysis and discussion. **Correctness:** Solutions and results are correct for a majority of cases, and only exhibit minor flaws and limitations in rare cases. **Techniques and Methods:** Your implementation is functional; some consideration was made to runtime and resources efficiency by using adequate data structure and algorithms; the code base demonstrates a good level of object oriented concept. **Code:** Good use of object-oriented style coding; implementation is clear to the reader; little redundancy in code and good code reusability.
40-49	The submitted work is functional and exhibiting very limited results required for the pass grade. Although sufficient understanding in the topic is shown, it lacks analysis. **Correctness:** Solutions and results exhibit major flaws and limitations. **Techniques and Methods:** Your implementation is functional; little consideration was made to runtime and resource; the code base is not extensible. **Code and Structure:** Poor use of object-oriented style coding; implementation is difficult to follow for the reader; noticeable redundancy in code.
30-39	The submitted work is clearly deficient. It does not meet all the requirements for the pass grade set in the assignment specification. The work does not evidence a sufficient understanding in the subject.
0-29	The submitted work is not relevant to the topic. There is little evidence that attempts were made to complete the work to the minimum standard.
NS	Non-submission