Department of Computer Science

Faculty of Engineering, Built Environment & IT

University of Pretoria

COS344 - Computer graphics

Practical 3 Specification: 3D Rendering

Release Date: 26-02-2024 at 06:00

Start By Date: 08-04-2024

Due Date: 22-04-2024 at 10:30

Total Marks: 90

Contents

1 General Instructions

- *Read the entire assignment thoroughly before you start coding.*
- This assignment should be completed individually, no group effort is allowed.
- **To prevent plagiarism, every submission will be inspected with the help of dedicated software.**
- Be ready to upload your assignment well before the deadline, as **no extension will be granted.**
- If your code does not compile, you will be awarded a mark of 0. The rendering output of your program will be primarily considered for marks, although internal structure may also be tested (eg. the presence/absence of certain functions or classes).
- Failure of your program to successfully exit will result in a mark of 0.
- Note that plagiarism is considered a very serious offence. Plagiarism will not be tolerated, and disciplinary action will be taken against offending students. Please refer to the University of Pretoria's plagiarism page at http://www.ais.up.ac.za/plagiarism/index.htm.
- You are allowed to use any standard of C++.
- The usage of ChatGPT and other AI-Related software is strictly forbidden and will be considered as plagiarism.
- No pre-build objects and textures may be used. All objects and textures that you need to use must be created by yourself.
- You must use OpenGL version 3.3 for this practical.

2 Overview

For this practical, you will need to render a low polygon 3D propeller aircraft and apply a set of transformations to the aircraft.

3 Background

Propeller aircraft have been around since the dawn of aviation, when the Wright Brothers first took flight in 1903. During the first world war, the true power of the aircraft was discovered on the battlefield where they were used for scouting, bombing and fighting off enemy aircraft. During this time period, the predominant type of aircraft was a **bi-aircraft**, which consisted of two main wings, staggered on top of each other.

Figure 1: Sopwith Camel bi-aircraft

In the closing stages of the war, the mono-aircraft was developed and experimented with. A **mono-aircraft** is an aircraft that consists of a single main wing. During the interwar period (1919 to 1939), further improvements on aircraft were made which included enclosed cockpits and cabin areas, as well as retractable landing gear. During the second world war, the propeller aircraft featured again till the closing stages of the war with only the Luftwaffe being able to form a combat squadron of jet aircraft. Unfortunately, this marked the beginning of the end of propeller driven aircraft's dominance on the aircraft market.

Figure 2: Supermarine Spitfire mono-aircraft

During the second world war two predominant, landing gear (wheels) configurations for aircraft were adopted namely, tricycle and tail-wheel landing gear configurations. **Tricycle** landing gear is the landing gear configuration used by all modern airliners and jet aircraft. This consists of a nose wheel (wheel at the front of the aircraft) and a set of main landing gear (wheels under the wings). In contrast, **tail-wheel** landing gear configurations is where the aircraft has no nose wheel but rather a wheel on the tail (back of the aircraft). The aircraft still has two main landing gear under the wings. An example of a tricycle and tail-dragger landing gear configuration can be seen in Figure 3 and Figure 4 respectfully.

Figure 3: Mooney M20 with tricycle landing gear

Figure 4: Piper J3 Cub

Figures 3 and 4 also demonstrates the different types of mono-aircraft configurations that are available. Figure 3 is an example of a **low** wing where the wing is below the cockpit of the aircraft, whereas Figure 4 is an example of a **high** wing mono-aircraft where the wing is above the cockpit of the aircraft.

In addition to the above-mentioned types, two further types of propeller driven aircraft can be found. The first is known as a **puller** propeller configuration, where the propeller effectively pulls the aircraft through the air and is mounted to point in the direction of travel. Figure 2 is an example of this type. The other type is known as a **pusher** configuration, where the propeller pushes the aircraft through the air and is mounted facing the opposite to the direction of travel. Figure 5 is an example of a puller configuration.

Figure 5: Piaggio P166s Albatross

Figure 6 is an example of an aircraft that is a hybrid between puller and pusher propeller configuration.

Figure 6: Cessna C337

Post World War 2, a complicated propeller system was developed to overcome the effects of torque produced by powerful piston engines. These types of propellers are known as **contra-rotating** propellers. With these types of propellers, each engine of the aircraft has two sets of propellers that rotate in different directions. Figure 7 is an example of an aircraft utilising contra-rotating propellers.

Figure 7: Avro Shackelton

The **engine cowling** is the part of the aircraft fuselage, which protects and covers the engine. Figure 8 indicates where the cowling is located by the door that is open behind the propeller. Note, this is not the standard configuration for flight, as it is an emergency if the cowling is open during flight.

Figure 8: Open Engine Cowling

Figure 9 highlights where the components are located that will become useful later in the practical.

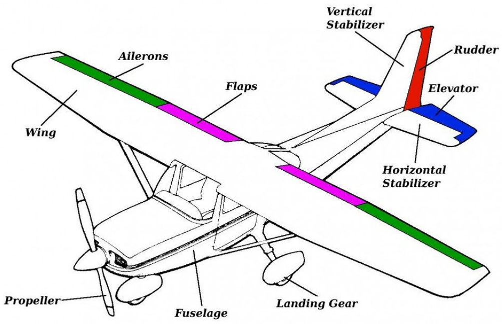


Figure 9: Components of an aircraft

4 Your Task:

For this practical, you will need to render a low polygon 3D depiction of a real-world, propeller-driven aircraft. An example of what you are expected to render is given in Figure 10. In the sections below, the different requirements are laid out. You are allowed to design any type of propeller-driven aircraft but must at least comply with the following requirements. Due to the rendering being a low polygon depiction of a real world aircraft, you will also need to submit planning photos which will also be discussed in the sections to come.

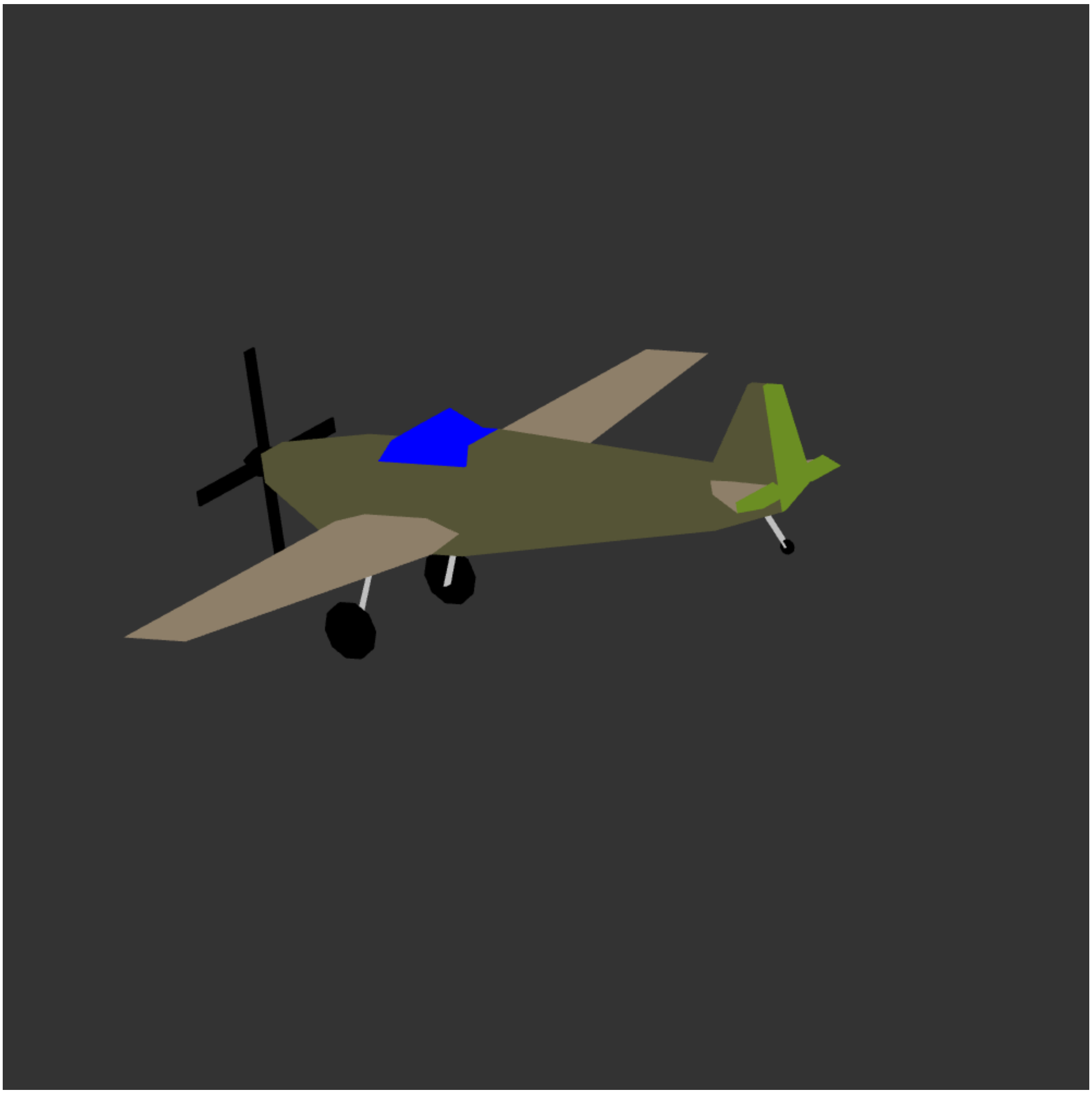


Figure 10: Example of expected render

4.1 Aircraft requirements

The aircraft should comply with the following:

- The rendering should look like the aircraft you are taking inspiration from, i.e., you cannot render a fictional aircraft.

- At least a single window should be visible from the front of the aircraft.
- At least a single window should be visible from both of the side views of the aircraft.
- Should have at least three wheels (two main landing gear and one tail/nose wheel).
- The aircraft should at least have a single engine with a single propeller. Only render the cowling, not the interior of the engine.
- If the aircraft is a high wing aircraft, the support struct for the wings also need to be rendered.

Your aircraft can contain more details, if you wish to add it, just be mindful of the delay in rendering high polygon shapes.

4.2 Shape requirements

Your aircraft is required to at least contain the following distinct[1] 3D shapes:

- Cuboid
- Triangular Prism
- Cone
- Cylinder
  - The two circles at the top and bottom of the cylinder should contain at least 8 vertices.
  - The cylinders should be used for the wheels but can be used for other elements as well.

Note, in total your 3D render should contain at least 20 3D visible shapes using the list above. You can add extra 3D shapes if needed. Also, remember some of the faces of the shapes used in the 3D aircraft will be hidden by other shapes. You are also allowed to add decorative 2D shapes (such as a star) on the faces of the 3D shapes to add extra details, but these shapes will not count towards the required 20 3D shapes.

4.3 Colour requirements

The colouring of your aircraft needs to be inspired from a real-world picture of the aircraft you have selected. Your aircraft should consist of at least 5 colours with the following requirements:

- The wheels need to be black.
- The elevator and rudder needs to be a different colour from the rest of the tail.
- The windows need to also be a different colour from the rest of the airplane.
- The background colour has to also be a distinctive colour. *Hint: there is a opengl function that can assist with this.*

[1]Distinct implies that the shape as a whole is counted and not the internal polygons used to create the shape.

- The windows need to be blue.

In conclusion. Your airplane needs to contain at least 5 different solid distinctive colours with the background also being a distinctive colour. You are allowed to add extra details to your airplane to make it more realistic.

4.4 Transformation requirements

In this section, the transformation or animation, requirements for your rendering are discussed. Note keep the animation rates small such that it allows you to press the key multiple times before going out of camera view.

4.4.1 Rotations

- When the `W key` is pressed, the aircraft needs to rotate about x-axis in an anti-clockwise direction to the positive x-axis.
- When the `S key` is pressed, the aircraft needs to rotate about x-axis in a clockwise direction to the positive x-axis.
- When the `A key` is pressed, the aircraft needs to rotate about y-axis in an anti-clockwise direction to the positive y-axis.
- When the `D key` is pressed, the aircraft needs to rotate about y-axis in a clockwise direction to the positive y-axis.
- When the `E key` is pressed, the aircraft needs to rotate about z-axis in an anti-clockwise direction to the positive z-axis.
- When the `Q key` is pressed, the aircraft needs to rotate about z-axis in a clockwise direction to the positive z-axis.

4.4.2 Translation

- When the `I key` is pressed, the aircraft's y-axis position needs to increase.
- When the `K key` is pressed, the aircraft's y-axis position needs to decrease.
- When the `L key` is pressed, the aircraft's x-axis position needs to increase.
- When the `J key` is pressed, the aircraft's x-axis position needs to decrease.
- When the `O key` is pressed, the aircraft's z-axis position needs to increase.
- When the `U key` is pressed, the aircraft's z-axis position needs to decrease.

4.4.3 Propeller

As the propeller is a vital part of the propeller aircraft, it also needs to turn to simulate the engine rotating. Meaning you will need to rotate the propeller around an arbitrary axis, which you will need to determine. Propeller rotation should appear realistic. To add extra complexity, the propeller's speed should be controlled by the user as described below:

- Initially the propeller's speed should be 0.
- When the `+` key is pressed, the propeller's speed should increase.
- When the `-` key is pressed, the propeller's speed should decrease.
- *Hint: Ensure the propeller does not rotate backwards with multiple* `-` *key presses.*

Thus, the more the `+` key is pressed, the faster the propeller should spin and visa versa for the `-`.

4.4.4 Examples

Below are examples of the result after multiple key presses, assuming the aircraft is at the original rendering position.

Note that when a key is pressed, the aircraft should **not** return the original position. Your program should be able to apply the transformation in any arbitrary order with the aircraft at any position on the screen.

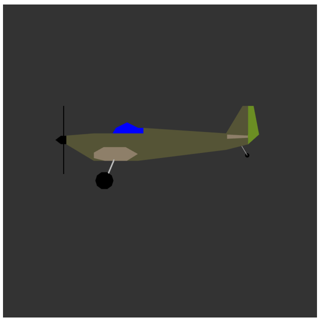
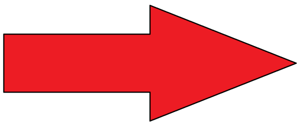


Figure 11: Original



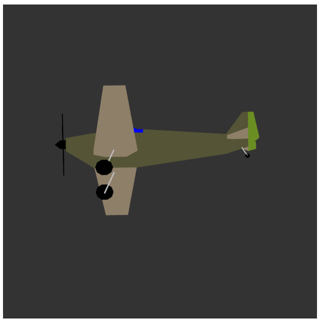


Figure 12: After a series of `W key` presses

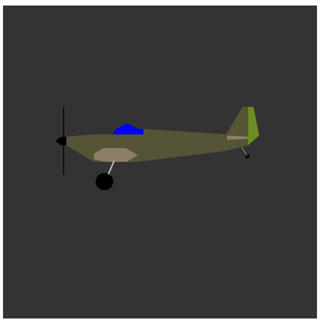
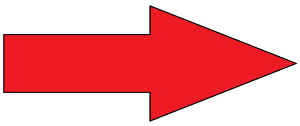


Figure 13: Original



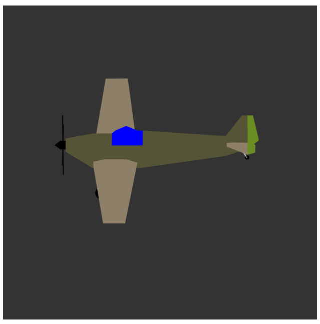


Figure 14: After a series of `S key` presses

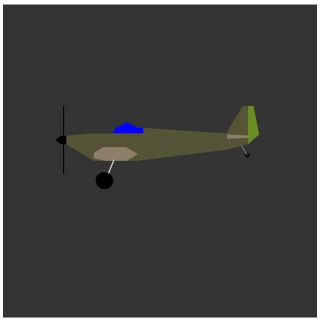


Figure 15: Original

Figure 16: After a series of `A key` presses

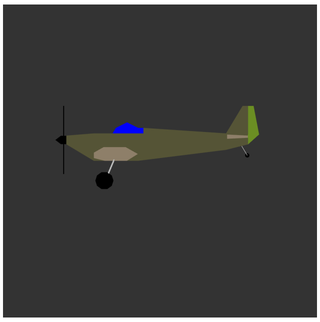


Figure 17: Original

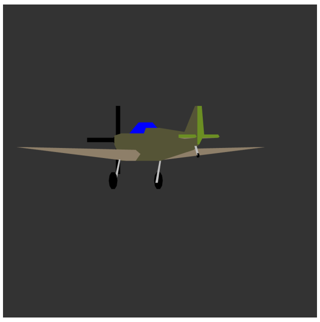


Figure 18: After a series of `D key` presses

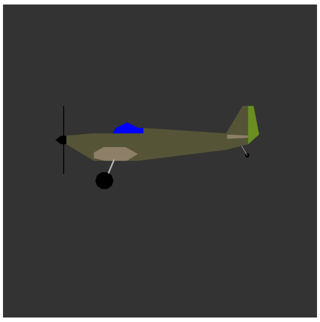


Figure 19: Original

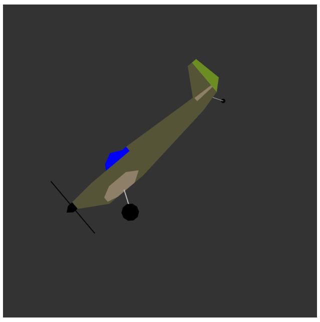
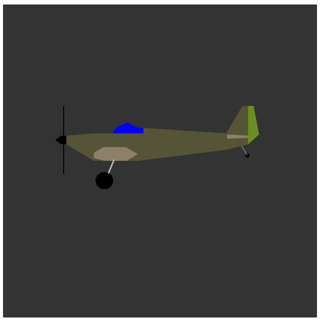
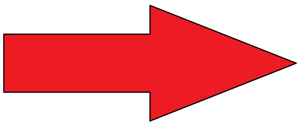


Figure 20: After a series of `E key` presses





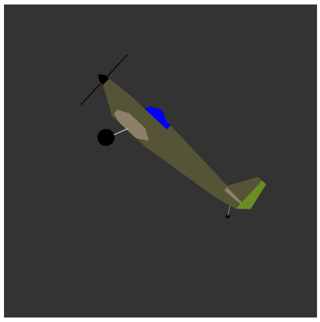


Figure 21: Original

Figure 22: After a series of `Q key` presses

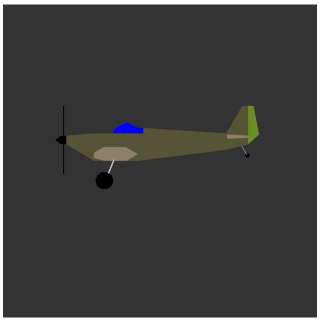
4.5 Wireframe

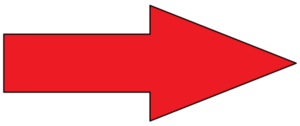
You are also required to implement a wireframe for your airplane. The wireframe should maintain the colour scheme of your airplane, such that the colours of the different shapes can be identified. The wireframe should also conform to all the transformations described earlier.

Your program should toggle between the wireframe and normal airplane when the `Enter key` is pressed. *Hint: you may need to implement a time delay between key presses for the wireframe toggling, so that the expected behaviour is achieved.*

Please note that you **must** use the `GL_LINES` to implement your wireframe. **Using the `glPolygonMode` function will result in the forfeiting of your wireframe marks.**

Please see the figure below for an example of a wireframe rendering of the airplane.





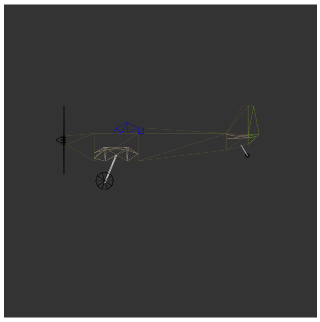


Figure 23: Original

Figure 24: Wireframe

4.6 Reference and plannning

Due to the aircraft needing to be based off of a real-world aircraft, you will need to provide some planning and reference photos that you will show during marking of the practical. You need to include the following photos with your submission:

- An augmented side view where you indicate the shapes that will be used to create the aircraft.
- An augmented front view where you indicate the shapes that will be used to create the aircraft.
- An augmented top view where you indicate the shapes that will be used to create the aircraft.

Below you will find the reference photo that was used to create the example aircraft.

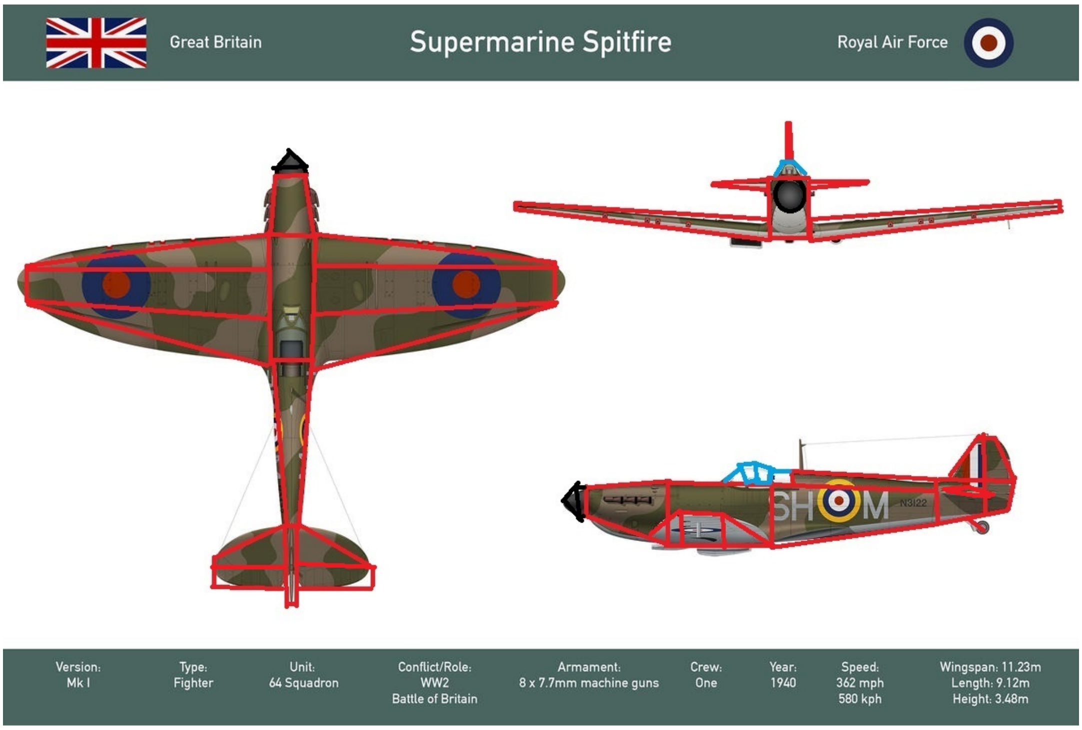


Figure 25: Augmented photo showing the planning of the 3D low polygon rendering

The planning photos will contribute **10 marks** to your final mark for this practical. For each missing photo, **3 marks** will be **subtracted** from your final mark. If no photo is included, you will receive 0 for this section.

5 Marking rubric

The following rubric will be used to mark your submitted assignment. Note you will be demoing the practical during the practical sessions on your own computer or a lab computer. Please see Table 1 for the rubric. Note: 1 mark will be subtracted for each transformation if the render is moved back to the center before a new transformation is applied. Although not included in the marking rubric, as stated in Section 4.6, the inclusion of the planning photos will contribute 10 marks to the final mark of the practical.

6 Bonus marks

There are 10 bonus marks available. Bonus marks are worth 2 for each extra thing that you render or implement. Some examples of things that count as extra are:

- Adding details on the aircraft such as roundels, squadron markings or text.
- Creating a multi-engined aircraft. A pusher-pulled combination aircraft will also count.
- Render a contra-rotating propeller aircraft. Both propellers should turn in opposite directions.
- Adding additional animations to your airplane such as the landing gear extending or retracting.

- Allowing the wheels of aircraft to turn.
- When rotating the aircraft, animate the ailerons, elevator and rudder to coincide with the real world movements to perform the rotation.

7 Implementation Details

- You need to use OpenGL version 3.3 for this practical.
- You may **not** use any of the build-in mathematical libraries within the `glm` package. This included matrix arithmetic. *Hint: You may use your practical 1 in this practical.*
- You may **not** use any of the built-in OpenGL functions to generate the shapes for you. You need to create each shape from first principles.
- You may **not** use any of the built-in OpenGL functions to perform the transformations of the shapes. You need to transform each shape from first principles either explicitly or by using the matrix arithmetic techniques discussed in class.
- You may only use the following C++ and OpenGL libraries:
  - stdio.h
  - stdlib.h
  - iostream
  - iomanip
  - cmath
  - sstream
  - GL/glew.h
  - GLFW/glfw3.h
  - glm/glm.hpp

  You may also use the `shader.hpp` and `glad.c` files that assist with compiling and linking of shaders. Your code should be able to be compiled without the assistance of an IDE, i.e. the project needs to be able to be compiled from terminal.
- All your helper classes and files needs to be in the same directory of the main.cpp.
- Ensure that the title of the window of your program is your correct student number.

8 Submission

You are required to submit on ClickUp under the appropriate submission link. In the archive that you submit, include a makefile and compiling instructions such that the program can be compiled and executed by the markers if needed. *Failure to upload to ClickUP will result in you forfeiting all marks for this practical.* No exceptions will be made on this matter.

9 Demo Instructions

1. You will first be required to download your submission from ClickUP.

2. You will then demo your practical to the tutor.

3. In the presence of the tutor, you will be required to upload the archive you downloaded from ClickUP to FitchFork. *Failure to upload to FitchFork will result in you forfeiting all marks for this practical.* No exceptions will be made on this matter.

Assessment Criteria	0	1	2	3	4
Airplane requirements [12 marks]					
Realistic aircraft	No resemblance of actual aircraft		Slight resemblance to aircraft		Realistic resemblance to aircraft
Front window	There are no windows		There is a front window		
Side windows	There are no side windows		There is a only a window on a single side	There is at least a single window on both sides of the aircraft	
Main landing gear	There are no main landing gear	There is only a single landing gear	There is at least two main landing gears		
Tail/Nose wheel	There is no tail/nose wheel	There is a tail/nose wheel			
Shape requirements [12 marks]					
At least one cubiod	There are no cuboids	There is at least one cuboid			
At least one triangular prism	There are no triangular prisms	There is at least one triangular prism			
At least three cylinders	There are no cylinders	There are cylinders but of the wrong vertex count	At most one cylinder that has the correct number of vertices	At most two cylinders have the correct number of vertices	There are at least three cylinders that has the correct number of vertices
At least one cone	There are no cones	There are cones but they do not have the correct vertex count	There is at least one cone that has the correct vertex number		
At least 20 distinct 3D shapes	There are no 3D shapes	There are at most 5 3D shapes	There are at most 10 3D shapes	There are at most 15 3D shapes	There are at least 20 3D shapes
Colour requirements [12 marks]					
Black wheels	Not correct colour			Correct colour	
Elevator and rudder of the airplane is correctly coloured	Same colour as the rest of tail			Correct colour	
Windows of the airplane is the correctly coloured	Not the correct colour or same as body of airplane			Correct colour	
Background colour is distinctive	Not distinctive colour or black			Correct colour	
At least 5 colours used	Only a single colour used			Correct colour	
Rotation requirements[12 marks]					
X-axis rotation	The airplane is not able to rotate around the x-axis		The airplane is able to only rotate in one direction around the x-axis		The airplane is able to rotate in both directions around the x-axis
Y-axis rotation	The airplane is not able to rotate around the y-axis		The airplane is able to only rotate in one direction around the y-axis		The airplane is able to rotate in both directions around the y-axis
Z-axis rotation	The airplane is not able to rotate around the z-axis		The airplane is able to only rotate in one direction around the z-axis		The airplane is able to rotate in both directions around the z-axis
Transformation requirements[12 marks]					
Horizontal translations	The airplane is not able to move along the x-axis		The airplane is able to only to move along the x-axis in one direction		The airplane is able to move along the x-axis in both directions
Depth translations	The airplane is not able to move along the z-axis		The airplane is able to only to move along the z-axis in one direction		The airplane is able to move along the z-axis in both directions
Vertical translations	The airplane is not able to move along the y-axis		The airplane is able to only to move along the y-axis in one direction		The airplane is able to move along the y-axis in both directions
Propeller rotation[8 marks]					
Propeller deacceleration	No propeller is turning		The propeller rotates at a constant speed		The propeller rotates at a increasable speed
Propeller acceleration	No propeller is turning		The propeller rotates at a constant speed		The propeller rotates at a decreasable speed
Wireframe[12 marks]					
Render	No wireframe		Partial wireframe		Correct wireframe
Colour	No wireframe		Partially coloured		Correctly coloured
Transformations	No wireframe	Partial transformations			Correct transformations

Table 1: Marking rubric