Computer Graphics - Mini Project 21118

Shubham Chemate.

Group Members:

1> Shubham Chemate (21118) →

2> Amey Bhattad (21113) →

The entire project is collectively developed by the ideas from me & Amey. Most of the graphics part was done by me & animation of objects was done by Amey.

Problem: Design & Implement ~~Simple~~ ~~Gears~~ game/Animation clip in open source library.

Abstract: Our group has made simple animation of gears using open source graphics library open GL.

Hardware Required:

Manufaturer & Model: Acer Swift-3.
Processor: Intel core i5-8th gen (8265U @ 1.6GHz)
Installed Memory: 8GB RAM, 512GB SSD.
Architecture: 64-bit

Software Requirement:

Operating System: Ubuntu 20.04 LTS on oracle virtual machine (3 processors & 4096MB base memory is allocated)
C++ version used: C++ 14.
Compiler for C++: g++ (version: 10.1.0)
Code-editor: Sublime Text (Build: 2011)
Open GL libraries used → glut.h, glu.h, gl.h.

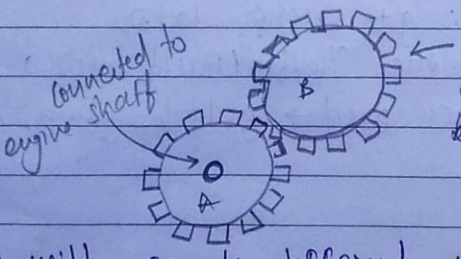
Theory.

Concepts, Idea & Topic Introduction:

C++ language: It is a general purpose programming language. Modern C++ is equipped with object oriented, generic & procedural programming features. The C++ is famous for It's close connection with low level languages, it's portability & efficiency. C++ has also been found useful in many other context such as SQL servers, game engine development etc.

OpenGL library: OpenGL is a cross language, crossplatform application programming interface for rendering 2D & 3D vector graphics. The API is typically used to interact with the graphics processing unit to achieve hardware accelerated rendering. It is written in C. OpenGL " stands for 'Open Graphics Library". provides a common set of commands that can be used to ~~manage~~ manage graphics in difference applications & a multiple platforms.

Idea & Concept: The idea is to create animation of rotating gears. This animation can be used to demonstrate how torque on one gear can accelerate rotation of other gear.



← this gear gets motion due to torque provided by gear 1.

We will provide different view options so that animation gets as real as possible.

~~Some~~ Controls of Animation:

Note: to reverse the rotation use mouse click (on gear model).

keyboard keys.	function.
'w', 's' ~~s~~	rotate ~~its~~ ~~a~~ the animation view ll'lel to ~~along~~ +x axis (antickwise direction using 'w', clkwise direction using 's')
'q', 'd'	rotate animation ll'lel to ~~along~~ +y in clkwise & anticlkwise direction.
l	~~for~~ simple light effect.
+	To increase speed of gears
−	To decrease speed of gears.

The methods used are:

Implemented By Us!

init(): set's initial environment for animation

drawScene(): Draws the gears configuration after each chage.

display(): Calls draw Scene after particular change in variables.

processSelection(): select clkwise/anticlkwise rotation of gears

reshape(): used it to change view of our animation

mouse_callback(): It calls process selection on mousepress event

keyboard(): detect keyboard keys for specific functionalities.

Provided by GL library:

glutInit(): Initialize GLUT

glutCreateWindow(): creates the window

glutInitWindowSize(): initial window height & width.

glutInitWindowPosition(): position the window.

glutDisplayFunc(): registers the callback Function.

glutMainLoop(): enters a processing loop

glutKeyboardFunc(): registers callback function for keybard event

9. Result:

The working animation clips/ photos are attached.

Conclusion:

The openGL is a multi-functional graphics API which can be used to create simple/ complex animations. We used API to create gears animation & learned different features of it we also learned how different oop concepts can be used for real world project/ animation development.

References & mentions:

The openGL API is new for us & we ~~get~~ learned lot of things throught the project. Our ~~sa~~ lab co-ordinator Prof. Amit sharma provided us with their valuable suggestion. Other than that we used following references:

i> openGL documentation present on following websites

a> www.khronos.org

b> www.opengl.org

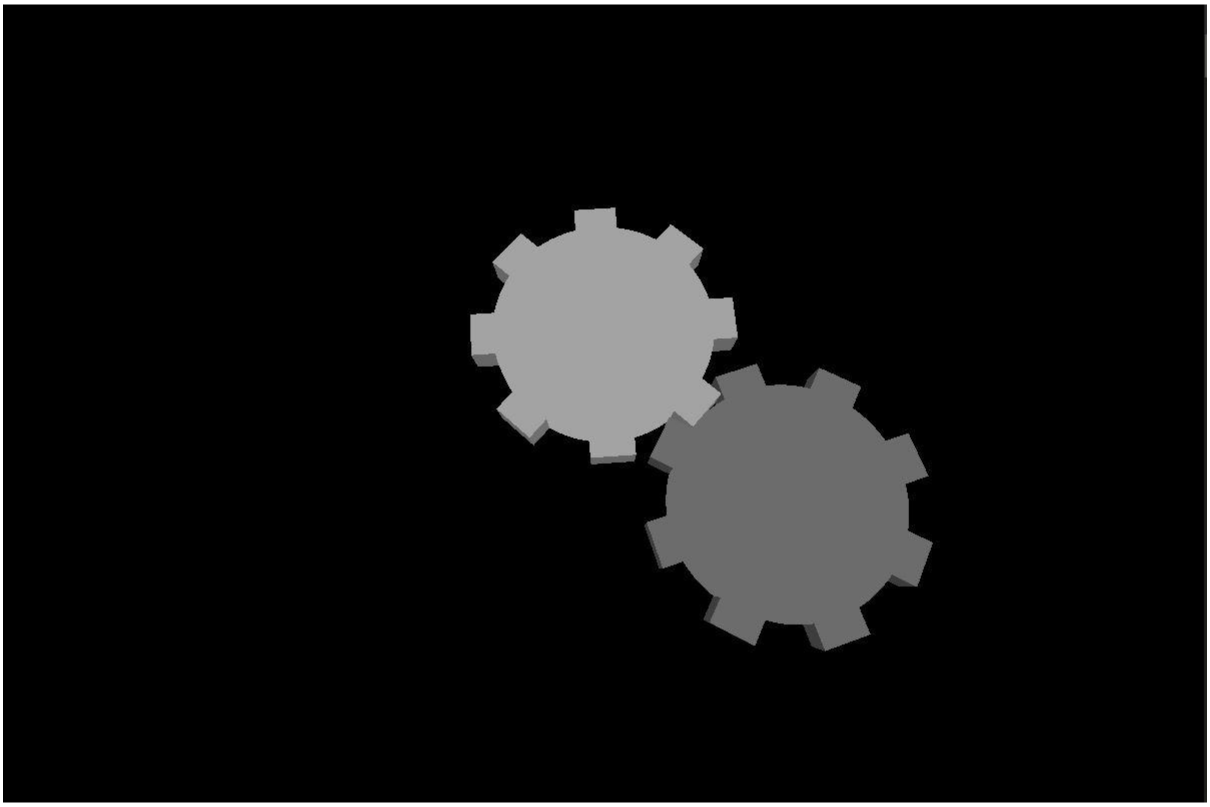
ii> wikipedia.

iii> learnopengl.com.

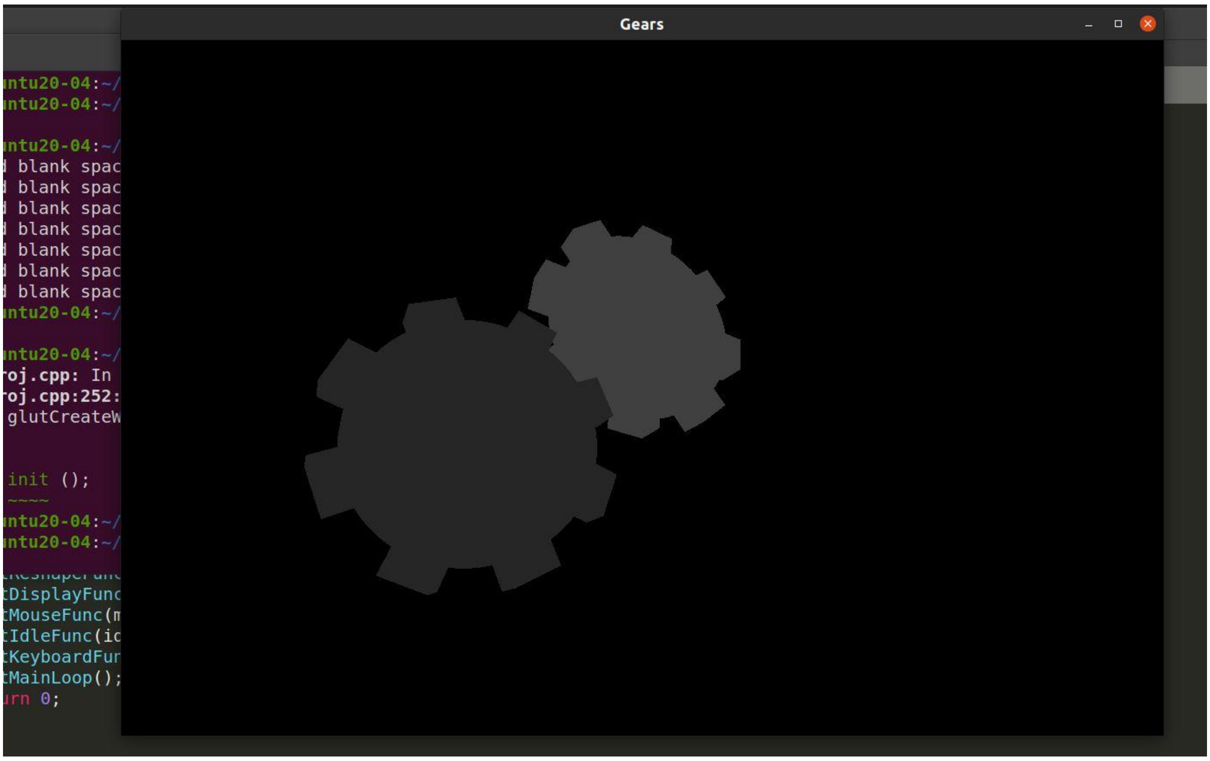
Thank You.!

The outputs:

In normal mode -> just after starting the animation



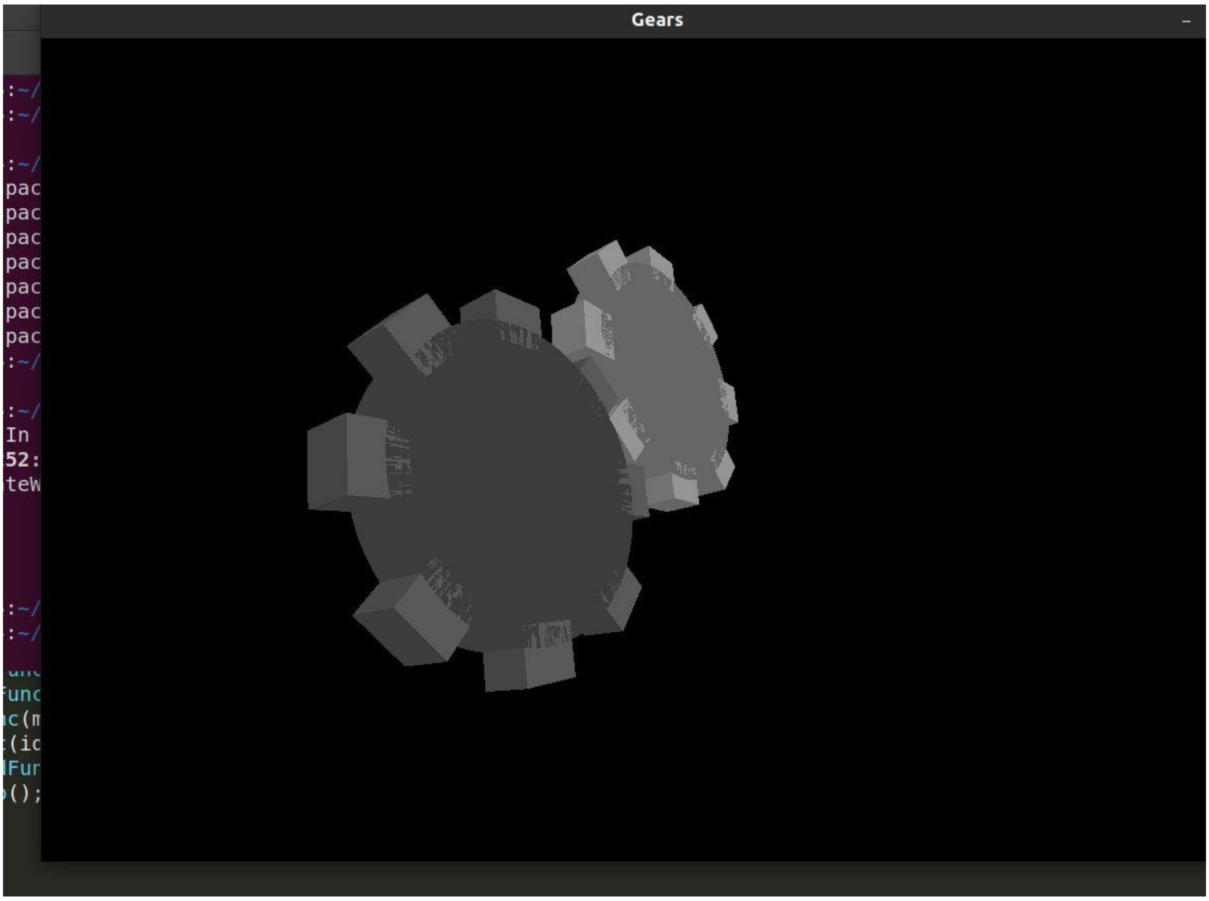
Light conditions are changed –> by pressing lowercase ‘l’



```
ıntu20-04:~/
ıntu20-04:~/

ıntu20-04:~/
l blank spac
l blank spac
l blank spac
l blank spac
l blank spac
l blank spac
l blank spac
ıntu20-04:~/

ıntu20-04:~/
oj.cpp: In
oj.cpp:252:
 glutCreateW


 init ();
 ~~~~
ıntu20-04:~/
ıntu20-04:~/
```

```
tDisplayFunc
tMouseFunc(m
tIdleFunc(ic
tKeyboardFur
tMainLoop();
urn 0;
```

View is change -> by using keys ‘a’, ‘s’, ‘w’, ‘d’ : all lowercase



```
:~/
:~/

:~/
pac
pac
pac
pac
pac
pac
pac
:~/

:~/
In
52:
teW



:~/
:~/
unc
Func
ac(m
:(ic
dFun
();
```