DECLARATION

WE AYESHA SHEIKH (4HG20CS001) AND BHAGYA RAJU NAIK (4HG21CS403) student of 6th semester B.E, CSE, Government Engineering College, here by declare that the project entitled **"ATOM SIMULATION"** has been carried out by me, under the supervision of **MADHURI H D** faculty, Dept of CSE submitted in partial fulfilment of the requirements for the award of the degree of computer science and engineering by the Visvesvaraya technological university during the academic year 2022-23. This report has not been submitted to any other organization/university for any award of degree certificate.

BY:
AYESHA SHEIKH
BHAGYA RAJU NAIK

ABSTRACT

Everything you see around you is made up of atoms, and all atoms consist of subatomic particles. In the Atom simulation, you will learn the names of the basic subatomic particles and understand.

As a part of the project, you'll see how the electrons are revolving around the nucleus in their respective orbits. One can see and spot the nucleus, atoms and electrons and can understand how an electron revolves around the nucleus. The project has made in such a way that one can easily understand the simulation of atoms.

This project has been developed in Ubuntu OS with interfacing keyboard and mouse with menu driven interface. And plans to include lighting, shading and other features in future enhancement.

This project is written in C and used OpenGL (Open Graphics Library). Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit, to achieve hardware-accelerated rendering

ACKNOWLEDGEMENT

The completion of any project involves the efforts of many people. We have been lucky to have received a lot of help, support from all directions during this project, so with the gratitude we take this opportunity to acknowledge all those who guide and encouraged us.

We are extremely grateful to our principal **Dr.T Rangaswamy** B.E, M.Tech, Ph.D for creating an excellent and technically sound environment in our institution.

We would like to express our profound sense of gratitude to our department HOD **Dr.K C Ravishankar**, B.E, M.Tech, Ph.D and our co-ordinater **Dr.Raghu M.E**, B.E, MTech, Ph.D of CSE and our guide **Madhuri H D** for their guidance. It's a great pleasure to acknowledge for their help and encouragement rendered towards the successful completion of project.

BY:
AYESHA SHEIKH
BHAGYA RAJU NAIK

Contents

List of Figures

Chapter 1

Introduction

Our project entitled "ATOM SIMULATION" The aim of this project is to develop a 2-D atom simulator, which contains options like selecting the user desired element, simulating the selected element. And also stopping the simulation when user wants. The interface should be user friendly and should use mouse and keyboard interface for the interaction with the user. The main goal is to show the users how an element structure is and how the electrons revolves around the nucleus so that one can easily get the knowledge of atom.

This project is written in C and used OpenGL (Open Graphics Library). Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit, to achieve hardware-accelerated rendering

1.1 Objectives

1. The objective is to build an atom simulator which can convince the audience about the structure of an element.

2. The coding is implemented for the atoms from Hydrogen to Neon, that is for 10 elements.

3. In this simulation importance is given on a structure of a element and how the electrons revolve around the nucleus

1.2 Advantages

1. Cost-Effectiveness:
Atom simulation computer graphics projects offer a cost-effective way to study molecular systems. Traditional experimental methods can be time-consuming and expensive, requiring resources for sample preparation, equipment, and data collection. Computer simulations eliminate or reduce these costs by providing a virtual environment for experimentation.

2. Educational Tool:
Atom simulation computer graphics projects serve as effective educational tools. They can be used to teach students about the principles of chemistry, molecular biology, and physics in a more engaging and interactive manner. Students can interact with the simulated atoms, manipulate parameters, and observe the resulting changes, enhancing their understanding of complex concepts.

1.3 Organization of Report

This report is divided up into chapters, each dealing with different aspects of the project. Each chapter has a short introduction, explaining the subject of each chapter, and then the details each module is explained separately. The following is a short overview of each of the chapters:

• Chapter 1: Outlines some of the research made on the project in the beginning. More research was made as this project report was being developed, as new areas had to be investigated. This research is summarized in the various chapters according to the different modules.

•Chapter 2: Specifies the requirements Analysis of the Project.

•Chapter 3: This section specifies the Proposed Methodology of the project.

•Chapter 4: Specifies implementation of the project.

•Chapter 5: The conclusion of the project is discussed.

1.4 OpenGL

OpenGL is an open specification for an applications program interface for defining 2D and 3D objects. The specification is cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. It renders 3D objects to the screen, providing the same set of instructions on different computers and graphics adapters. Thus it allows us to write an application that can create the same effects in any operating system using any OpenGL-adhering graphics adapter.

1.5 GLUT

GLUT, short for OpenGL Utility Toolkit, is a set of support libraries available on every major platform. OpenGL does not directly support any form of windowing, menus, or input. That's where GLUT comes in. It provides basic functionality in all of those areas, while remaining platform independent, so that you can easily move GLUT-based applications.

Chapter 2

Requirement Analysis

2.1 Hardware Requirements

Processor : i3 or i5 2Ghz

RAM : 4GB

Hard Disk : 100Gb

2.2 Software Requirements

Operating System : Windows11/Ubuntu

Coding Language : C/C++ using OpenGL

Graphics Library : GL/glut.h

2.3 Functional Requirements

A Functional is nothing but inputs the the software system.A Functional Requirement is description of the service that software must offer.It describes a Software system or its component.It can be calculation ,data manipulation ,user interaction or any other specific functionality with

defines what function a system is likely to perform.

1.User Interface - User should be able to select an element and start the simulation on their own. They can start, stop and change the elements of their choice and after this they can exit the simulation.

2.Element Selection- User can select the element from Hydrogen to Neon for the simulation, that is from atomic number 1 to atomic number 10.

3.Start/ Stop Simulation- User after selecting an element from the mentioned list he/she can start the simulation. As soon as they select start, the electrons around the nucleus starts revolving around the nucleus within their orbit. If they select the stop simulation option, the simulation will be stopped.

2.4 Non Functional Requirements

Non Functional Requirements(NFRs) define system attibutes such as security, reliability, performance, maintainability, scalability and usability.They serve as constraints or restictions on the design of the system across the different backlogs.

Performance: The system should be able to handle large-scale atom simulations with a significant number of atoms while maintaining real-time or near-real-time performance.

Scalability: The system should be able to scale efficiently to accommodate an increasing number of atoms without significant degradation in performance.

Accuracy: The simulation should accurately represent the behavior and interactions of atoms based on established scientific principles and empirical data.

Usability: The user interface should be intuitive and user-friendly, allowing researchers and scientists to easily interact with and control the simulation parameters.

Reliability: The system should be stable and robust, minimizing crashes, data corruption, and other errors that could disrupt the simulation process.

Extensibility: The software should be designed in a modular and extensible manner, allowing for easy integration of new simulation algorithms or visualization techniques.

Chapter 3

Proposed Methodology

3.1 Mouse interface

• Select element:

When the user clicks the right click button on the mouse, the screen will be prompted with the list of options. First option is to select the user desired element from the list. The list contains elements from Hydrogen to Neon for the simulation, that is from atomic number 1 to atomic number 10.

• Simulate:

After user selecting an element, when he/she clicks on the simulate option, the electrons around the nucleus starts revolving around the nucleus within their orbit.

• Stop simulation: If a user selects this option, the simulation will be paused.

• Exit: The program execution will be terminated and the window will be destroyed after selecting this option.

3.2 Keyboard Interface

Three functionalities are implemented using the keyboard function.

• After selecting an element, if a user presses spacebar the simulation will be started.

• After starting the simulation if the user clicks on 'S' key, simulation will be paused.

• If the user clicks on the 'Q' key, program execution will be terminated and the window will be destroyed.

3.3 User Interface

User should be able to select an element and start the simulation on their own. They can start, stop and change the elements of their choice and after this they can exit the simulation.

3.4 Element Selection

User can select the element from Hydrogen to Neon for the simulation, that is from atomic number 1 to atomic number 10.

3.5 Start/ Stop Simulation

User after selecting an element from the mentioned list he/she can start the simulation. As soon as they select start, the electrons around the nucleus starts revolving around the nucleus within their orbit. If they select the stop simulation option, the simulation will be stopped.

Chapter 4

Implementation

glRasterPos3f(x, y, z): OpenGL maintains a 3-D position in window coordinates. This position, called the raster position, is maintained with subpixel accuracy. The current raster position consists of three window coordinates (x, y, z), a clip coordinate w value, an eye coordinate distance, a valid bit, and associated color data and texture coordinates.

glutCreateMenu(menu): glutCreateMenu creates a new pop-up menu and returns a unique small integer identifier. The range of allocated identifiers starts at one. The menu identifier range is separate from the window identifier range. Implicitly, the current menu is set to the newly created menu. This menu identifier can be used when calling glutSetMenu.

glutAddMenuEntry(args): glutAddMenuEntry adds a menu entry to the bottom of the current menu. The string name will be displayed for the newly added menu entry. If the menu entry is selected by the user, the menu's callback will be called passing value as the callback's parameter.

glutAttachMenu(button): glutAttachMenu attaches a mouse button for the current window to the identifier of the current menu; glutDetachMenu detaches an attached mouse button from the current window. By attaching a menu identifier to a button, the named menu will be popped up when the user presses the specified button.

glutMouseFunc(args): glutMouseFunc sets the mouse callback for the current window. When a user presses and releases mouse buttons in the window, each press and each release generates a mouse callback.

glutKeyboardFunc(args): glutKeyboardFunc sets the keyboard callback for the current window. When a user types into the window, each key press generating an ASCII character will generate a keyboard callback. The key callback parameter is the generated ASCII character. The state of modifier keys such as Shift cannot be determined directly; their only effect will be on the returned ASCII data. The x and y callback parameters indicate the mouse location in window relative coordinates when the key was pressed.

Chapter 5

DESIGN

Atom simulation (Main window): This window contains all the contents that is menu bar and simulation display. This is window used for all the events and functions in this project. In this window we display simulation of first 10 atoms in the periodic table. And all mouse and keyboard events triggered in this window. All the labels and Information about the model will be displayed on this window.

Menu bar: Menu bar is designed so that one can easily access the various options like selecting the elements or starting the simulation or stopping the simulation etc.

Simulation display: As soon as user selects the element and click on the simulate option, the electrons around the nucleus starts revolving around the nucleus within their orbit.

Hydrogen to Neon atom: In the menu bar the select element option consist of 10 elements that is hydrogen to neon. as the user select the desired element the orbit will calculate using elipse functions lise.void eleright(float rad),void eleleft(floatrad),void eletop(float rod),void eledown(float rod),void eletr(float rod),void eletl(float rod),void eledl(float rod),void eledr(float rad) The default nucleus will be calculated by the function void nuc(float rod) After creating nucleus and arbics the respected number of atoms start revalving arond nucleus.

Chapter 6

RESULT

6.1 HOME PAGE

The figure 5.1 shows the home page that display the details about project developers with instruction to move next step.

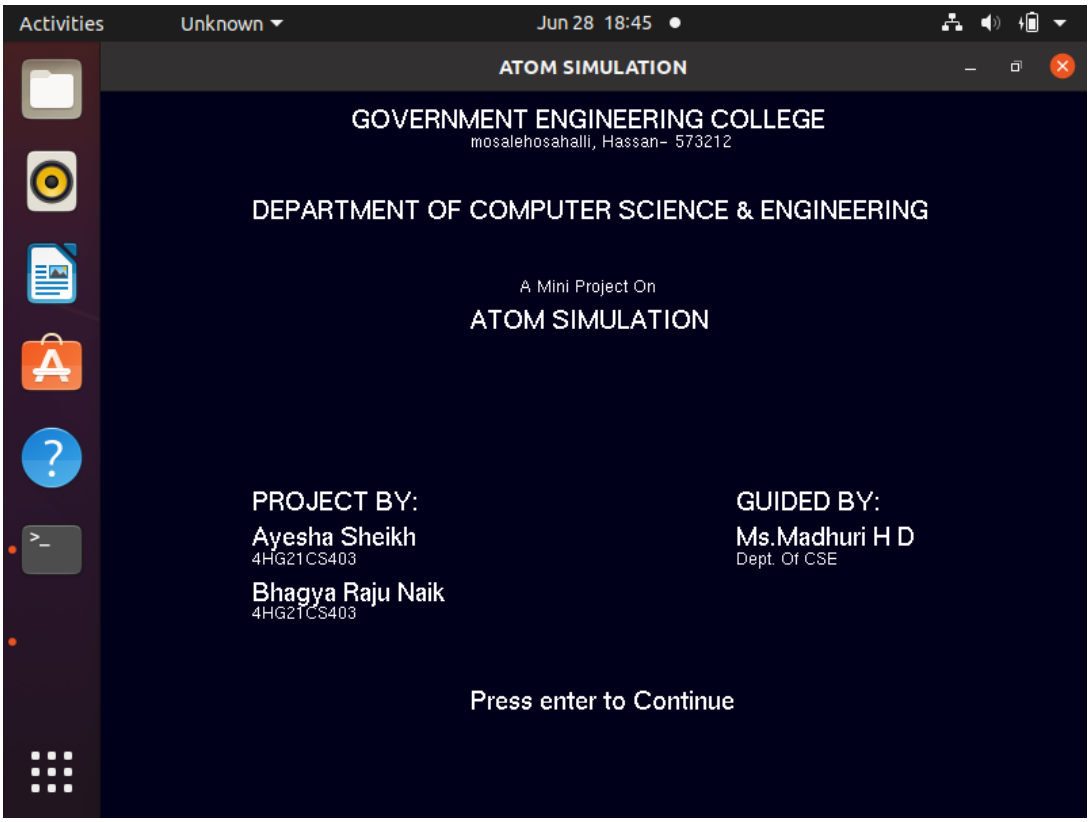

Figure 6.1: First page

6.2 MENU BAR

Menu bar is designed so that one can easily access the various options like selecting the elements or starting or stopping the simulation etc.

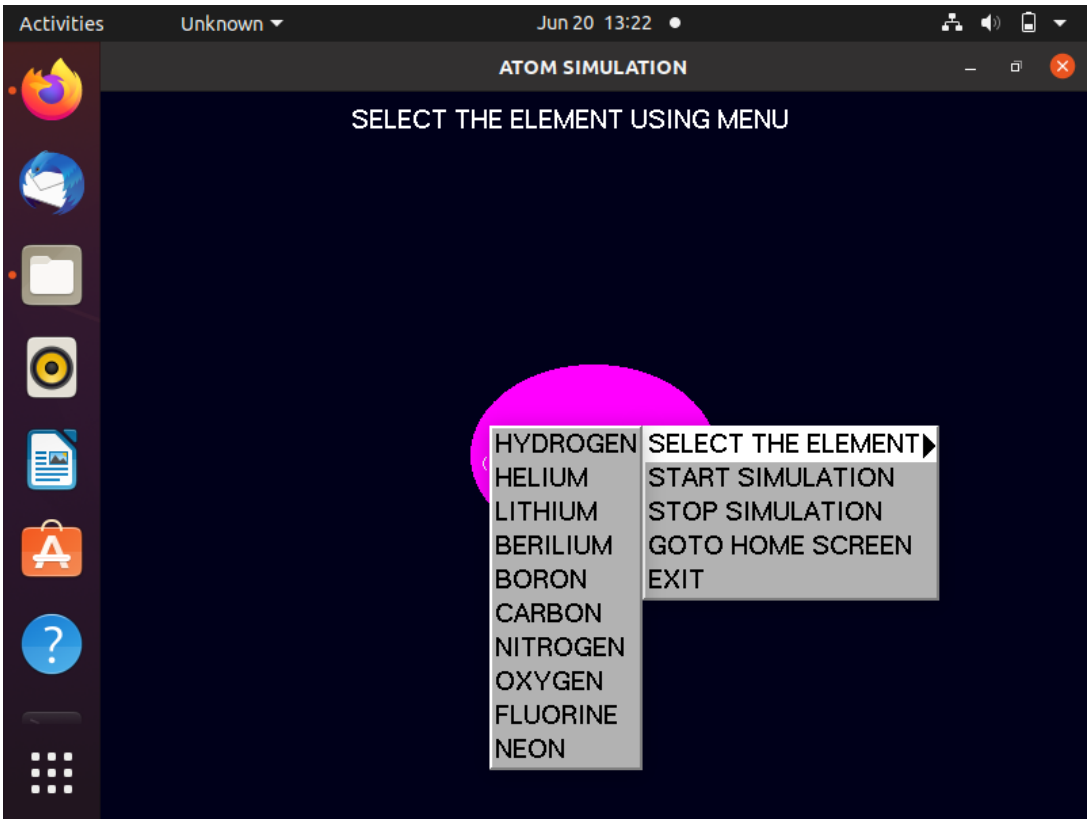

Figure 6.2: MENU BAR

6.3 Nucleus

Nucleus is default which is combination of neutron and protron. Nucleus is default for all elements.

Figure 6.3: Nucleus

6.4 BORON ELEMENT

Boron is a multipurpose element. It's a crucial nutrient for plants, an important component in the nuclear industry.

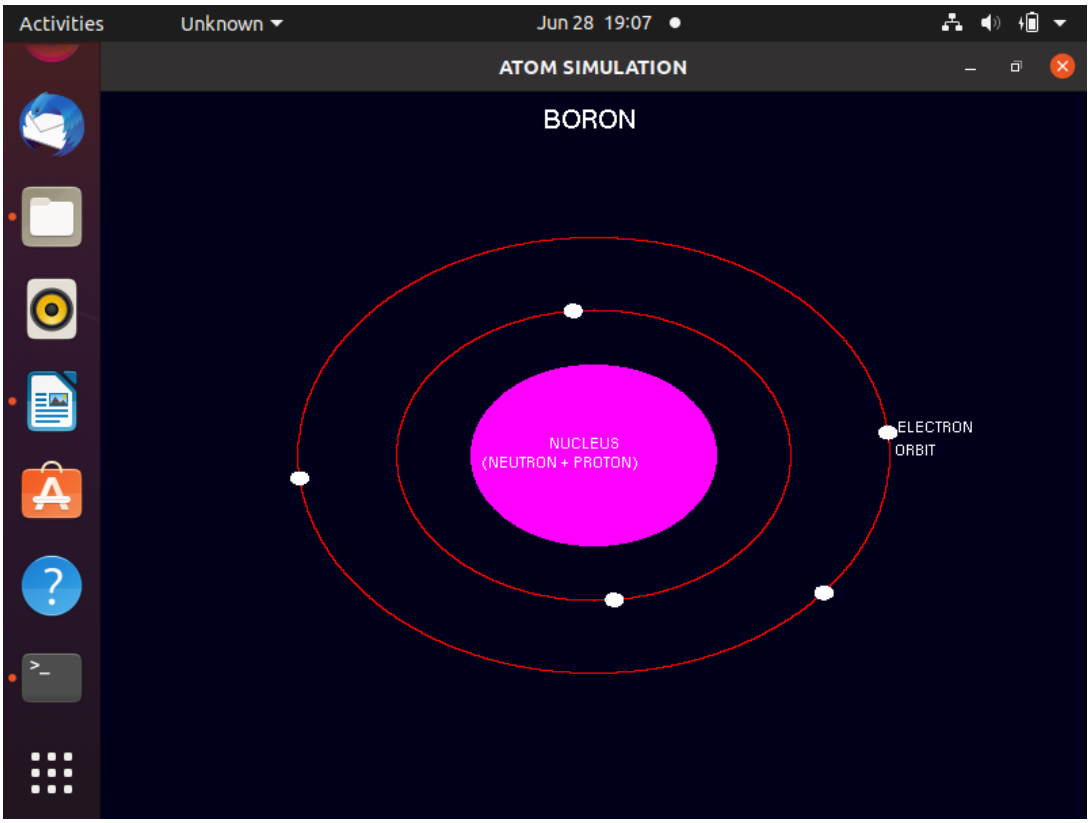

Figure 6.4: BORON ELEMENT

Chapter 7

Conclusion

This atom simulation is very good project. Users can very easily understand the structure of an element. The interface is mouse driven and the user can select a function by clicking. And also, the interface supports keyboard interface. We have tried our best to make this simulator very realistic, so that user can easily understand the concepts of electrons, orbits, atoms and nucleus etc.

7.1 Future enahancement

The following are some of the features that are planned to be supported in the future versions of the atom simulator.

1. Adding all the elements from the periodic table.

2. Features like showing the simulation with all the important details of an element.

3. Adding a search bar for selecting an element from the list of all the elements.

4. Making the simulation in 3-D.

Bibliography

[1] https://github.com

[2] https:/ /www.w3schools.com/php/default.asp

[3] https:/ /www.tutorialspoint.com

[4] https://www.researchgate.net

[5] https.openglforum.org