

**AMERICAN INTERNATIONAL**

**UNIVERSITY-BANGLADESH**

**COMPUTER GRAPHICS – PROJECT**

|  |  |
| --- | --- |
| **Course Name** | Computer Graphics |
| **Section** | L |
| **Course Tutor** | ANEEM AL AHSAN RUPAI |

|  |  |
| --- | --- |
| **Members** | |
| Name | **ID** |
| RIFAT AHMED | 19-41107-2 |
| MEDEDI HASAN EMON | 19-41118-2 |
| MRIDUL KUMAR ROY | 20-43517-1 |
| SIMUR AHAMED SEZAN | 20-43473-1 |

**GROUP MEMBERS**

**Table of Content:**

|  |  |
| --- | --- |
| **Content List** | **Page No** |
| Introduction | 04 |
| Proposal | 05 |
| Schematic Diagram | 05-07 |
| List of Objects | 08 |
| Functions to Represent the Objects | 08 - 09 |
| Interactive Functions | 09-10 |
| Task Assignment and Codes of Functions | 10-13 |
| Output | 14-17 |
| Conclusion | 18 |

**Introduction**

**OpenGL** (Open Graphics Library) is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics. These can be used to create very simple to very complex architectural scenes. Here, **GLUT** (OpenGL Utility Toolkit) is a library of utilities for OpenGL programs, which primarily perform window definition, window control, and monitoring of keyboard and mouse input. Basic **Transformations (translation, rotation, scaling)** play a very important role in manipulating objects on screen. There is mouse interaction which will let swap between the Day and Night view.

**Proposal**

In our project we will use OPENGL to show the rainy day scenario by the city road. At first, vehicles are moving on the road daynight. For the rainy day scenario, we will try to show the rain fall. We will also show the animation of sunset during the day, vehicles are moving on the road, train movement. Windmill rotating scenario.

Press d “Day view”

Press n “Night view”

Press r “Rain start”

Press s “Rain” exist

**Schematic Diagram**

**Full Schematic Diagram**

**Timeline

Description automatically generated with low confidence**

**A picture containing diagram

Description automatically generated**

**LIST OF OBJECTS**

Sky

Star

Ground

Sun

Clouds

Cloud R

Moon

Tree

House1

House2

House3

House N

School

National Flag

Roadside

Traffic Light

Electric poll (3)

Electric wire(3)

Cellular Tower

Cellular Tower (N)

Windmill (2)

Lamppost (2)

Railing

Rail Track

Train

Boat

River

Road

Truck

Boat

**Functions to Represent The Objects**

|  |  |  |
| --- | --- | --- |
| **Object** | **Function** | **ID** |
| House | void house() | 1010 |
| House2 | void house2() | 1011 |
| Truck | void truck() | 1012 |
| Car1 | void car1() | 1013 |
| Cell phone towerN | void cellphonetowerN() | 1014 |
| Car2 | void car2() | 1015 |
| Moon | void moon() | 1016 |
| Clouds | void clouds() | 1017 |
| CloudsR | void cloudsR() | 1018 |
| Display | void display() | 1019 |
| Display1 | void display1() | 1020 |
| DisplayR | void displayR() | 1021 |
| Flag | void flag() | 1022 |
| Cell phone tower | void cellphonetower() | 1023 |
| Sky | void sky() | 1024 |
| SkyN | void skyN() | 1025 |
| School | void school() | 1026 |
| HouseN | void houseN() | 1027 |
| School | void School() | 1028 |
| Init | void init() | 1029 |
| Lamb post | void Lambpost() | 1030 |
| Electric poll | void epoll() | 1031 |
| Electric wire | void epoll() | 1032 |
| Objects | void objects() | 1033 |
| Rail | void rail() | 1034 |
| Ralling | void railing() | 1035 |
| Windmill | void Windmill () | 1036 |
| Windwheel | void windwheel() | 1037 |
| Rain | void rain() | 1038 |
| Rail Track | void rail Track() | 1039 |
| Road | void road() | 1040 |
| Trafic light | void traficlight() | 1041 |
| Sky | void sky() | 1042 |
| SkyN | void skyN() | 1043 |
| SkyR | void skyR() | 1044 |
| Sun | void sun() | 1045 |
| SunN | void sunN() | 1046 |
| Tree | void tree() | 1047 |
| TreeN | void treeN() | 1048 |
| Truck | void truck() | 1049 |
| Boat | void Boat() | 1050 |
| River | void river() | 1051 |

**Interactive Functions**

|  |  |  |
| --- | --- | --- |
| **Interactive Functions** | **Interaction** | **ID** |
| Updatecar1 | void updatecar2 (int value) | 1001 |
| Updatecar2 | void updatecar2(int value) | 1002 |
| Updatecar2 | void updatecar2(int value) | 1003 |
| Updateclouds | void updatecloud(int value) | 1004 |
| Updatecloud1 | void updatecloud1(int value) | 1005 |
| UpdateCloudsN | void updatecloudsN(int value) | 1006 |
| Updatesound | void updatesound(int value) | 1007 |
| updatetrain | void updatetrain(int value) | 1008 |
| updatesun | void updatesun(int value) | 1009 |
| updaterain | void updaterain(int value) | 1010 |
| Updateboat | void updateboat(int value) | 1011 |
| updatewinwheell | void updatewinwheell(int value) | 1012 |
| Updatewinfan2(int value) | void updatewinfan2(int value) | 1013 |

**Task Assignment and Codes of Functions**

**Contribution Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Member-1** | **Member-2** | **Member-3** | **Member-4** | **TOTAL** |
| 25% | 25% | 25% | 25% | 100% |

|  |  |
| --- | --- |
| **Name**  **ID** | **Contribution in Project** |
| **Member-1**  **Rifat Ahmed**  **19-41107-2** | clouds();  House()  House2();  car1();  ground();  epoll2();  objects(); |
| **Member-2**  **Mededi hasan Emon**  **19-41118-2** | rail();  railing();  railTrack();  river();  boat();  truck();  road();  epoll();  epoll3();  objects(); |
| **Member-3**  **Mridul kumar Roy**  **20-43517-1** | Sky();  skyR();  clouds1();  cloudsR();  House3() ;  Car2();  road();  rain()  star();  objects(); |
| **Member-4**  **Simur ahmed Sezan**  **20-43473-1** | eware1();  eware1N();  ewareN();  tree();  lambpost();  lambpost1();  sun();  school();  flag();  objects(); |

**OUTPUT**

**DAY VIEW**

Diagram

Description automatically generated

NIGHT VIEW

Diagram

Description automatically generated

**RAINY DAY VIEW**

Diagram

Description automatically generated

**RAINY NIGHT VIEW**

****

**Conclusion**

We have implemented a view of city road view, a train appear view and an complete riverside view.We've shown that using Code blocks (version 17.12) and different functionalities where day, night, rainy view appears and exists through keyboard interaction. We have used both keyboard(d, n, r, s) interactions to change the view. We implemented the color of the objects by following RGB color code. The program runs everything perfectly as we planned. In future we would like to develop this project into a 3D architecture