

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**Computer Graphics**

**Spring 21-22**

**Section: N**

*Project Report: Sea Beach view from Hotel*

***Faculty***

**Md. Masum Billah**

|  |  |
| --- | --- |
| NAME | ID |
| MIRZA MD TAWHID | 20-42643-1 |
| MD MEHRUB KHAN LIMON | 20-42411-1 |
| GOUROV SAHA | 20-42977-1 |
|  |  |

**Date of Submission:** April 20, 2022

Table of Contents

[INTRODUCTION: 2](#_Toc90185362)

[OBJECTIVE & PROPOSAL OF THE PROJECT: 2](#_Toc90185363)

[SYSTEM IMPLEMENTATION METHOD: 2](#_Toc90185364)

[Schematic Diagram: 3](#_Toc90185365)

[**Sea Beach view: Day and Night** 4](#_Toc90185366)

[**Hotel view: Day and Night 5**](#_Toc90185367)

[Lists Of Object: 6](#_Toc90185368)

[Functions to represent the objects: 6](#_Toc90185369)

[Interactive functions: 7](#_Toc90185370)

[Task Assignment and codes of functions: 8](#_Toc90185371)

[Conclusion: 8](#_Toc90185372)

[**Beach view: Day 9**](#_Toc90185373)

[**Beach View: Night 9**](#_Toc90185374)

[**Hotel view: Day 10**](#_Toc90185375)

[**Hotel View: Night 10**](#_Toc90185376)

# INTRODUCTION:

In computer graphics, scenery or view refers to the combined application of transformation, projection, animation of elements in computer screen using OpenGL.

Here, a view or scene of a sea beach will be represented by graphical drawing algorithms. This is a mini project in computer graphics which is simple, good looking. We have mainly created some artifacts in this mini project like sky, sea, a sun, moon. There will be trees and moving ship on the sea. Some birds will be seen flying in the sky in the day mode of the view. In the night view, there will be calm nature with moon and stars. This is an overview of our project. This project is made with C++ language including glut with OpenGL graphic library.

# OBJECTIVE & PROPOSAL OF THE PROJECT:

The main objective of this project is to show the scenario of Sea beach using our knowledge on OpenGL. We have used input devices like key-board to interact with the program. In computer graphics, we learn about color, graphics characteristics like animation, transformation, projection etc. We learned to draw objects in glut display, movement of whole objects, changing color of any object, translation of an object, scaling and rotation of an object. We got all the knowledge from this course to make a real-life scenario. So, have drawn a scenery where all of our learning from the computer graphics course is implemented.

# SYSTEM IMPLEMENTATION METHOD:

This program is implemented using various openGL functions which are shown below.

Various functions used in this program:

• glutInit(): interaction between the windowing system and OPENGL is initiated.

• glutInitDisplayMode(): used when double buffering is required and depth information is required.

• glutCreateWindow(): this opens the OPENGL window and displays the title at top of the window.

• glutInitWindowSize(): specifies the size of the window.

• glutInitWindowPosition(): specifies the position of the window in screen co-ordinates.

• glutKeyboardFunc(): handles normal ascii symbols.

• glutDisplayFunc(): this handle redrawing of the window.

• glutMainLoop(): this starts the main loop, it never returns.

• glFlush(): used to flush the pipeline.

• glBegin(): delimit the vertices of a primitive or a group of like primitives.

• glPushMatrix(): push and pop the current matrix stack.

• glVertex2f (): specify a vertex.

• glColor3ub (): set the current color.

• glutPostRedisplay(): used to trigger an automatic redraw of the object.

• glMatrixMode(): used to set up the required mode of the matrix.

• glLoadIdentity(): used to load or initialize to the identity matrix.

• glTranslatef(): used to translate or move the rotation Centre from one point to another in three dimensions.

**Interaction with program**

**This program includes interaction through keyboard:**

• q-> This button is for stopping the ship

• t-> It will increase the speed (ship)

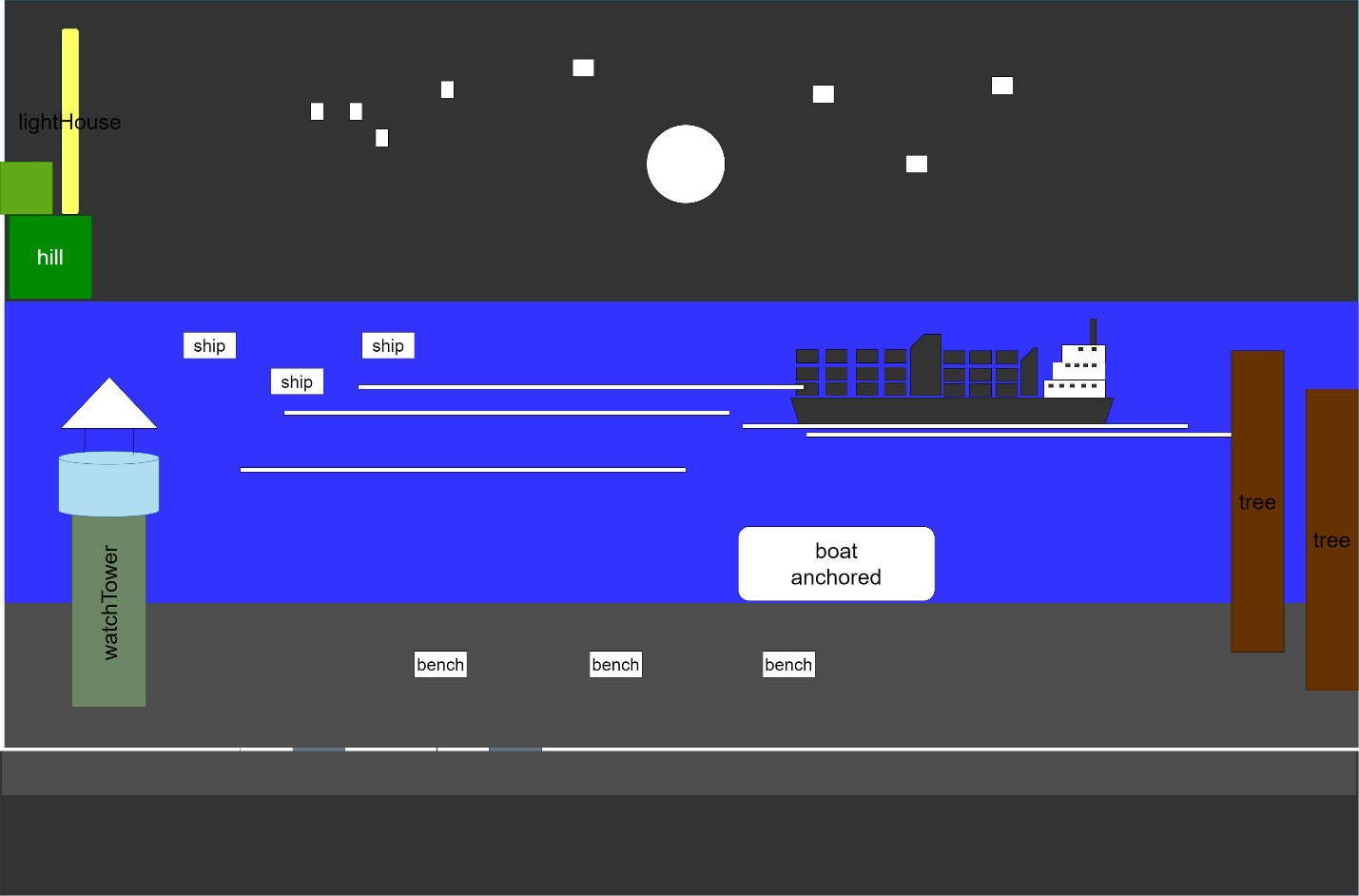
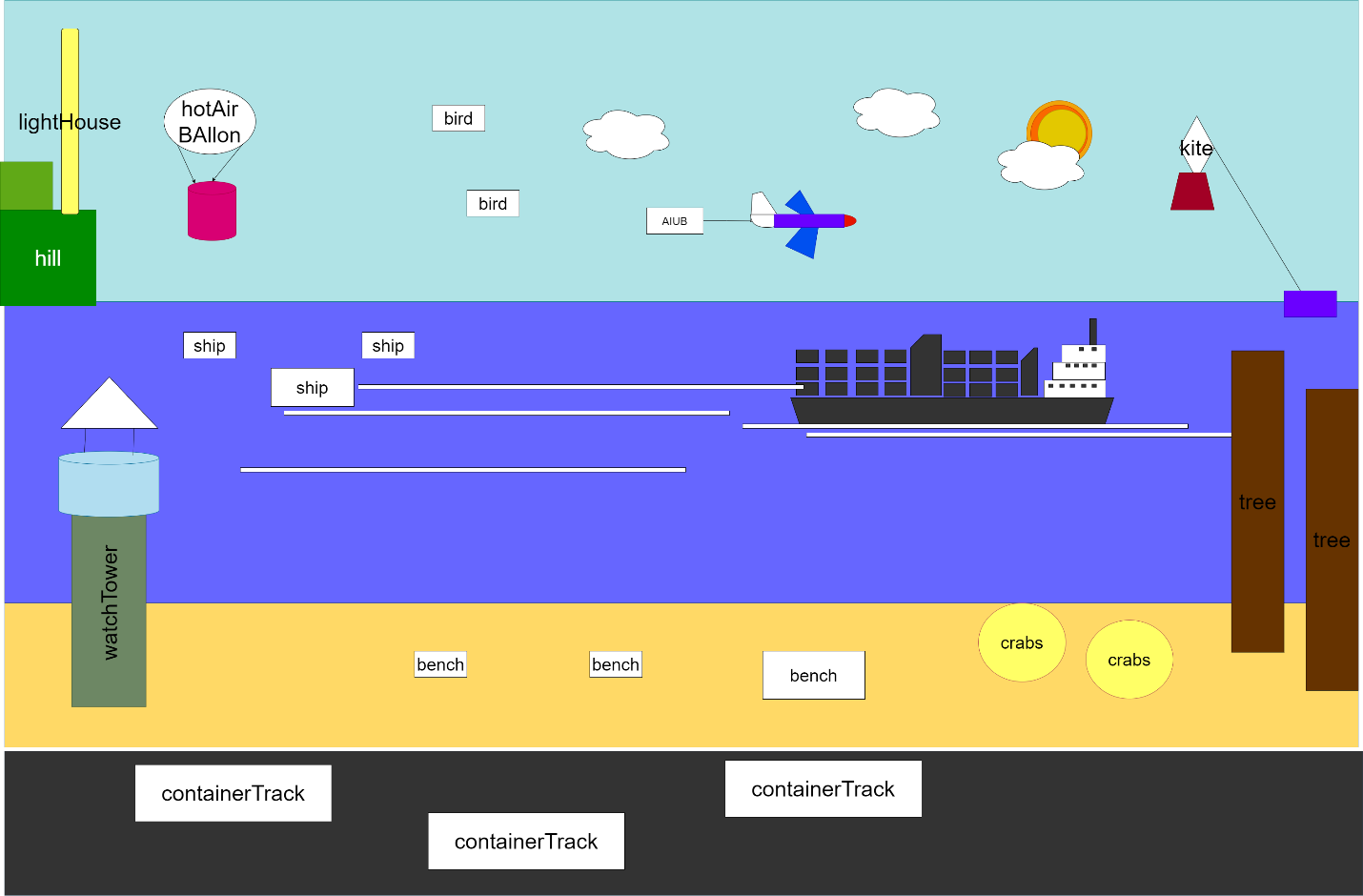
• u-> It will decrease the speed (ship)

• up-> For night view

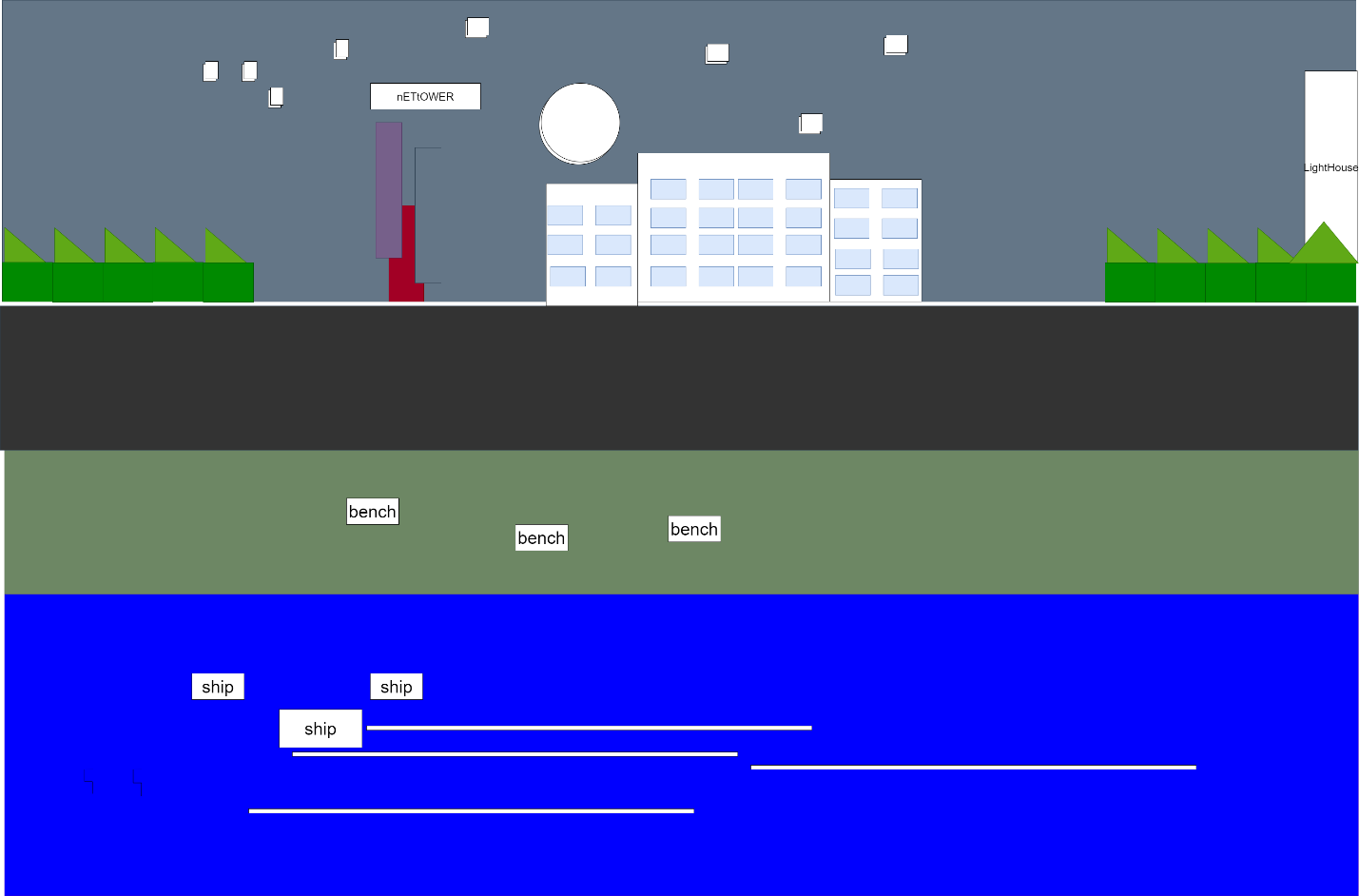
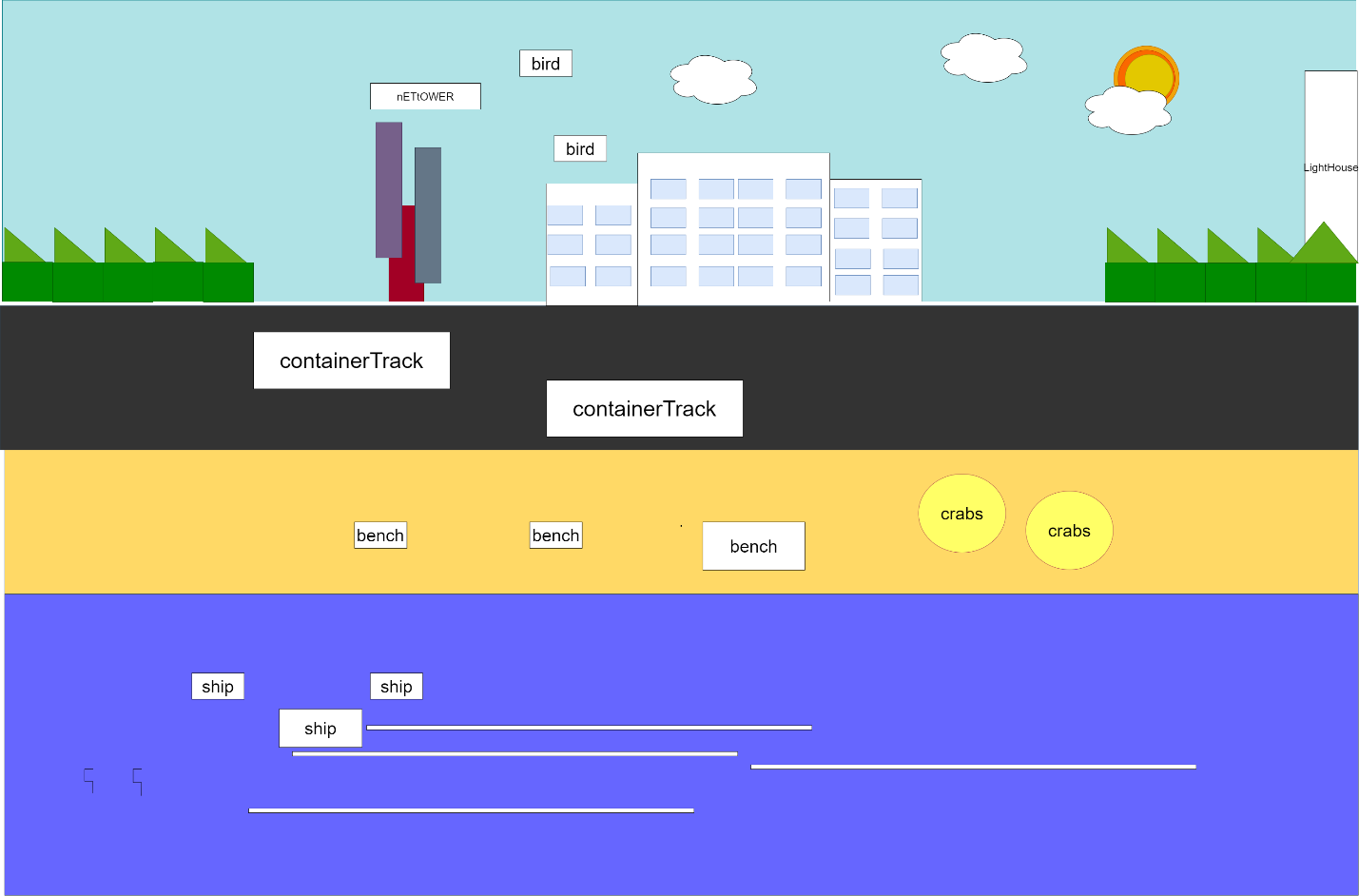
• down-> for day view

# Schematic Diagram:

## **Sea Beach view: Day and Night**



## **Hotel view: Day and Night**



# Lists Of Object:

1.cargo ship

2.flagship

3.tree

4.containerTack

5. Lighthouse

6.mountain

7.watchTower

8.stars

9.moon

10.sun

11.kite

12.bannerPLane

13.road

14.beach

15.crabs

16.bench

17.cloud

18.birds

19. Umbrella

# Functions to represent the objects:

|  |  |  |
| --- | --- | --- |
| **Obj name** | **Function** | **Id** |
| **Cargo Ship** | **CargoShip**  **CargoShip01**  **cargoShip02** | **akm\_ship**  **akm\_ship01**  **akm\_ship02** |
| **Banner PLane** | **banner()**  **airPlane()**  **bannerPlane()** | **akm\_banner**  **akm\_airPane**  **akm\_BP** |
| **Watch Tower** | **watchTower\_day()**  **watchTower\_night()** | **akm\_WT\_D**  **akm\_WT\_N** |
| **Road** | **road()** | **akm\_road** |
| **Container Truck** | **truck00()**  **truck01()** | **akm\_truck00**  **akm\_truck01** |
| **Network Tower** | **netTower()** | **Akm\_netTower()** |
| **Batman Signal Light** | **batmanLogo()** | **akm\_bat** |
| **Hotel** | **buiding()/build\_windows()** | **Akm\_hotel** |
| **Light House** | **lightHouse()** | **akm\_lightHouse** |
| **flagship** | **flagship()**  **nightflagship()** | **raj\_flagship**  **raj\_nightflagship** |
| **mountain** | **mountain()** | **raj\_mountain** |
| **birds** | **bird()** | **raj\_bird** |
| **kite** | **Kite()** | **raj\_kite** |
| **daynight** | **day()**  **night()** | **raj\_day**  **raj\_night** |
| **Sky** | **sky()** | **ht\_sky** |
| **moon** | **Moon()** | **Ht\_moon** |
| **sun** | **sun()**  **movesun()** | **ht\_sun** |
| **cloud** | **cloud1()**  **cloud2()**  **cloud3()** | **ht\_cloud** |
| **stars** | **stars()** | **ht\_stars** |
| **ocean** | **ocean()** | **ht\_ocean** |
| **umbrella** | **Umbrella()** | **sia\_umbrella** |
| **seat** | **Seat()** | **sia\_seat** |
| **crab** | **crab()** | **sia\_crab** |
| **tree** | **tree()** | **sia\_tree** |
| **land** | **land()** | **sia\_land** |

# Interactive functions:

|  |  |  |
| --- | --- | --- |
| **name of interactive functions** | **ID** | **mapped with obj** |
| **move\_cargoShip\_01**  **move\_cargoShip\_02** | **akm\_ship01\_anm**  **akm\_ship02\_anm** | **Cargo Ship** |
| **move\_BP** | **akm\_BP\_anm** | **Banner Plane** |
| **rotateWT\_light()** | **akm\_WT\_anm** | **Watch Tower** |
| **update\_truck00()**  **update\_truck01()** | **akm\_truck00\_anm**  **akm\_truck01\_anm** | **Contaoner Truck** |
| **RotateBatSignal()** | **akm\_bat\_anm** | **Batman Signal Light** |
| **rotateLH\_light()** | **akm\_lightHouse\_anm** | **Light House** |
| **updateBat()** | **akm\_batmanlogo\_anm** | **Batman Logo** |
| **move\_flagship()** | **raj\_flagship\_anm** | **flagship** |
| **move\_bird()** | **raj\_bird\_anm** | **birds** |
| **move\_nightflagship()** | **raj\_nightflagship\_anm** | **nightflagship** |

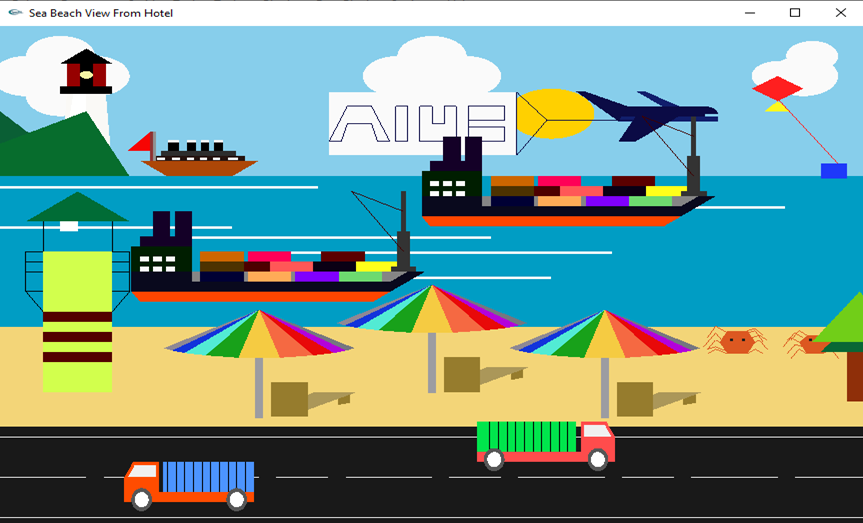
# Task Assignment and codes of functions:

|  |  |  |
| --- | --- | --- |
| Tawhid | Limon | Gourov |
| sky | boat | Lighthouse |
| moon | hill | Watchtower |
| sun | kite | bannerplane |
| cloud | birds | road |
| stars | bench | containertruck |
| ocean | crabs | hotel |
| sand | Cargo ship | Batman signal light |
| Net Tower | umbrella |  |

# Conclusion:

Here this is a mini-project with a sea beach scenario. In the future we will be able to add more features, objects and movements to this scene. For the time being we tried to implement all the basics and agendas that we learned in the computer graphics course. We tried to implement some real-time effect in this scenery. We learned a lot from the course which we tried to show in this project and this small project is our great motivation to get into the deep of graphics contents.

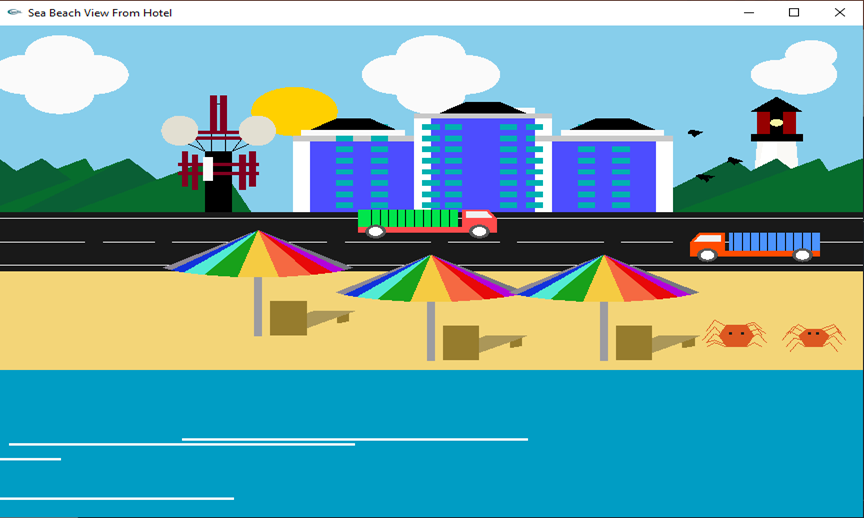
## Beach view: Day



## Beach View: Night



## Hotel view: Day



## Hotel View: Night

