Behavioral Animation PROJECT

By. Liangyi Hu

1. Project description

This is a project made for Lab4 of Computer Animation CS6555 in George Washington University.

The deliverables contain a video file demonstrating how hierarchical movement works, a Visual Studio project containing source code and other attachments (pictures, 3D objects). However, the library and dependencies are not included in the deliverables (relatively large), they are available upon request.

2. Features.

Per the project requirements, there are required features and additional features.

Features:

- Visible boids in bird shape, with its own flying animation.
- Physics system for boids, boids' acceleration reflect the force applied.
- Group behavior, boids tend to stay with group but not too close to other individuals.
- Group behavior, group follows a controllable destination.
- Obstacle avoidance, boids will avoid crashing obstacles and ground. Code Structure

The project is using OpenGL as its environment, the API been used is GLFW.

Library Packages

The none standard libraries used are listed as below:

Package Name	Description
GLFW	OpenGL development tool, create Simple API for users to use OpenGL functions.
GLEW	The OpenGL Extension Wrangler Library is a simple tool that helps C/C++ developers initialize extensions and write portable applications.
GLM	OpenGL math library, provide some data structure and method in calculations
SOIL	Used to import image and convert to texture data
ASSIMP	Open Asset Import Library (short name: Assimp) is a portable Open Source library to import various well-known 3D model formats in a uniform manner.

Please check and install proper libraries if user want to run the program.

Source code header files

There are five header files written, the name and its description have showed in below table:

Name	Description

MatrixTransformatin.h	Containing almost all the functions and variables needed to build and use for key framing interpolation.
Background.h	Code construct background coordinate pictures and connected lines of key points for key framing.
Camera.h	Containing the method building View/Projection matrix, allows camera to respond from user actions.
Shader.h	Creating Shaders, source Shader code, compile Shaders. Also provide some methods to easily using shaders.
Model.h	Import 3D objects, extract information and provide some method to easily visualize them. Additionally, each part is a Model object that can compute its position and rotation during simulation. Collision are also included in this header file. Lots of code related to project 3 are in here.
Mesh.h	Working with Model.h, create mesh vertices based on .obj file.
Boids.h	Construct each boid and its logic, define floor, obstacles and destinations. All the boid-related logics are here.

Note: Camera.h, Shader.h and Mesh.h are customized from source code written by Joey De Vries at www.learnopengl.com.

## Source Files

The source file shows as below:

Source File Name	Description
Main.cpp	Main program entry
VertexShader.txt	Vertex Shader program for general object visualization.
FragmentShader.txt	Fragment Shader program for general object visualization.
bgVertexShader.txt	Vertex Shader for background pictures.
bgFragmentShader.txt	Fragment Shader for background pictures.

3. User Guide

The program could run under Visual Studio environment, after all the library dependencies are installed.

The steps to have it run is (in addition to Part 1):

- Define number of boids (NUM_BOIDS).
- When the program starts, user should be able to see following screen:
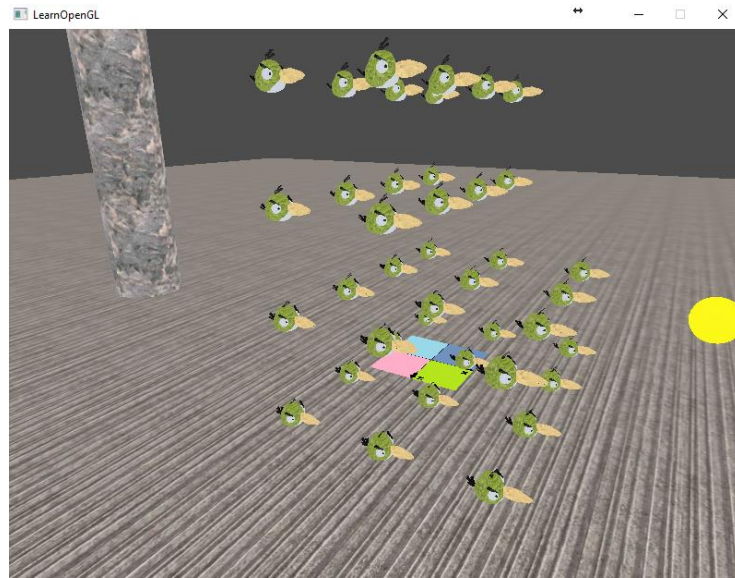
Press "B" key, all the boids will start move accordingly, staying with the group, facing the destination, avoid collision with ground/obstacle/other boids.

Left/Right/Up/Down/J/K key controls the yellow sphere destination to move in 6 directions.