COMPUTER GRAPHICS

T1 – EVANDE TEAM PROJECT

소프트웨어학과

2017310457 김현주

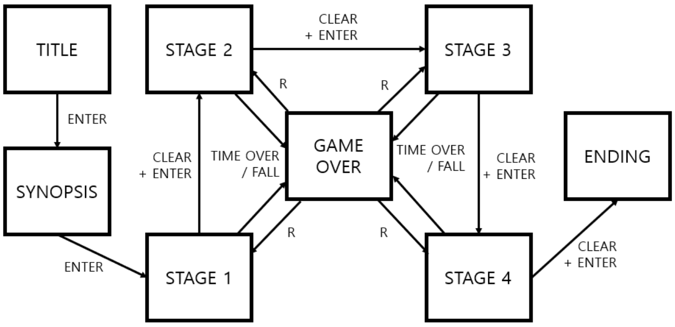
2017313099 서유영

[Objective]

The main content of this project is making a racing game with a theme of university life using OpenGL. There are 4 stages that each represents a phase of life in campus, from admission to graduation. The object of this game is to clear stages in time by avoiding obstacles and getting items.

[Algorithms and Data structure]

Game Logic and Game Control



The image shows full logic of game. To convert a scene to other scene, it needs to satisfy the conditions using flags. Player can control the sphere pressing UP, DOWN, LEFT, RIGHT buttons on keyboards. To move the sphere, press UP or DOWN, and to rotate it, press LEFT or RIGHT. There are some items which is useful to clear the stage and some obstacles which debuff the player to make it difficult to clear the stage.

Loading external 3D models (Assimp)

All maps, background, items, and obstacles are 3d meshes created from obj files except the sphere. Most of items and obstacles are downloaded from Unity Asset Stores, but other things are created by the team. We modeled objects ourselves using Unity and extracted them into obj files. After loading 3d models by Assimp library, the mesh information of models stored into game object list with body information which will be described in Physics Engine.

Physics Engine (Bullet Pysics)

By applying gravity on the world, setting internal inertia on sphere, and setting the rotation friction of sphere, the sphere falls with the gravity force and rolls on the track. When the keyboard inputs come, the sphere moves towards the according direction by applying the impulse force on its rigidbody. When it moves, due to the gravity and the force makes the sphere to acceleration movement.

Items, obstacles, and the track itself are all statics that are not affected by forces, so we set their mass as 0. On the other hand, the mass of the sphere is set by 1 so that the sphere can be in dynamic state to move around. For each model, we create the rigidbody with their mesh and classify the model into two types - one that should be collide with each other by physics and others that should not be collide. Normally, the triggers, items and obstacles (except for bus) should not be collide with sphere, but we should be able to detect whether the sphere pass those things. So, we detect the collision and set each effect for each collision.

Text rendering (Stb_truetype)

By using stb_truetype library, the texts are rendered. The inputted texts are parsed into one character so that it can be applied with wanted fonts, position, and size. The texts are used to indicate what the players can do, the stories, and some extra information. The timer of each game stage is also printed with text by rendering with changing string.

Sound rendering (Irrklang)

With the Irrklang library, the sound engine is created to output the sound. For each stage, the background music is set that fits to the stage's theme, and when game is over or cleared, the background music changes with corresponding music. Also, when sphere goes through the items or obstacles, and when keyboard input down is pressed, the sound effects are added to be more realistic.

Effect (Particle)

When each items or obstacles are triggered, the particles come out from the sphere in radial shape with different colors. By looking at each particles' color, it is possible to know what kind of item effect is currently working. Also, because sphere changes its position every time it moves, the particles are set to be follow the sphere so that it can be look like it comes out from the sphere.

Camera

The camera is set to follow the sphere when sphere moves. When keyboard input left or right is pressed, the camera turns around the sphere so that player can changes the direction with its angle. Also, when the A+ item (which has boost effect) is triggered, the camera's view changes to be wider so that the player can feel more that the sphere moves faster than before.

[Discussions]

In this project, we tried to use every factor that we learnt from the lecture and what the professor suggests as an extra. The whole game is based on the university life in SKKU, with a little fun. Creating our own 3d models and applying game logics on it was hard, but worthy.