A5 Project Proposal
Title: Angry Birds 3D
Name: Matthew Yang
Student ID: 20454809
User ID: m45yang

Final Project:

1 Purpose :

The proposed system will provide the user with 3D version of the popular mobile game, Angry Birds. The goal of the game is to hit all targets (pigs) with projectils given (birds). Topics included are:

1. Projectile motion
2. Collision detection
3. Particle systems
4. Keyframe animation
5. Shadow maps

2 Statement :

The system will provide functionality to define a game scene and play the game with the defined scene.

The user will start by importing the implemented meshes in the form of an .obj file and specify positions and ranges of motion of the targets in a Lua script. The user then runs the implemented executable and passes the Lua script's path as an argument to play the game with that scene.

One main problem of the project is building everything around the physics engine that defines gravity, and there are a number of ways to do so. Another interesting problem is modelling explosions, and the game will make use of particle systems to do so.

In building this game, I hope to learn more about:

1. Implementing a physics engine in computer graphics
2. Implementing collision detection systems
3. Keyframe animation with linear interpolation
4. Modelling explosions with particle systems
5. Implementing shadows using shadow maps

3 Technical Outline :

3.1 Modelling the Scene :

The system will support a hierarchical data structure similar to that of Assignment 3 to model and subsequently render a scene with pig-shapes objects as targets. The user will be able to define the position, color and 2 keyframes for animating movement of each pig. The launchpad for the projectiles will be rendered by default.

The user is also able to specify images to use for the background, as well as the sky-box.

3.2 Projectile motion :

Projectile motion is described by

$$r = v_0 cos\theta$$

$$y = -\frac{1}{2}gt^2 + v_0 sin\theta t$$

Gravity is assumed to act in the negative y direction and r is a radial coordinate along the direction in which the projectile is flying. These two equations will allow us to track the motion of the projectiles over time and render it appropriately. Additionally, eliminating t in the two equations give us

$$y = tan(\theta) \cdot r - \frac{g}{2v_0^2 cos^2\theta} \cdot r^2$$

This equation models the parabolic path of the bird, and allows us to draw a curve in the UI and represent the path that will be taken by the bird.

3.3 Collision detection :

The projectiles launched by the user will interact with other objects in the scene in a realistic way, bouncing or falling away after a collision. This can be implemented efficiently with the use of a ”sweep test” as described in the paper *Improved Collision detection and Response* by Fauerby, K.

Collision detection will also be implemented when drawing the trajectory that will be taken by the projectile. For example, the path would change direction if interrupted by a wall.

3.4 Particle systems :

The user will see a visible interaction between the projectile and an object upon collision detection, and this will be implemented using particle systems. The emitter will be positioned at the point of collision, and will only exist for a short period after the collision. The spawn rate of the emitter is determined by the ”intensity” of the collision, which can be measured in terms of change in momentum of the projectile.

3.5 Keyframe animation :

The user will be able to define 2 keyframes for each target in the scene. The game will make use of these two keyframes to animate movement of each target. Only straight line movements are supported, so the keyframes will be interpolated linearly.

3.6 Shadow maps :

To allow the user to visualize the projectiles flying through the air more easily, shadows will be implemented using shadow mapping with the depth map test.

Bibliography :

1. Department of Computer Graphics, ”CS488/688 Course Notes”, Spring 2017.
2. DeVries, J. *Shadow Mapping*, https://learnopengl.com/#!Advanced-Lighting/Shadows/Shadow-Mapping.
3. Ferguson, R. S. *Practical algorithms for 3D computer graphics.* (2014), pp. 237-245.
4. Reeves, W. T. *Particle Systems - A Technique for Modeling a Class of Fuzzy Objects.* Computer Graphics 17:3 (1983), pp. 359-376.
5. Fauerby, K. *Improved Collision detection and Response.* (2003).
6. Peachey, D. R. *Solid texturing of complex surfaces.* In SIGGRAPH 85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques (1985), pp. 279-286.

Objectives:

Full UserID: m45yang

Student ID: 20454809

___ 1: **User Interface:** User is able to control angle of trajectory, initial velocity and landing position of birds.

___ 2: **Texture mapping:** Texture mapping is implemented.

___ 3: **Collision detection:** Collision detection is supported for when a bird hits an object.

___ 4: **Particle systems:** Impact when a bird hits the ground is implemented with particle systems.

___ 5: **Sound:** Sound feedback is given when the cannonball hits an object.

___ 6: **Animation:** Keyframe animation is implemented with linear interpolation to move the pigs around the scene.

___ 7: **Physics Engine:** Gravity and friction (air resistance) is implemented with a physics engine to model the projectile motion of an object.

___ 8: **Reflection Map:** Reflection map (skybox) is implemented.

___ 9: **Shadows:** Shadows are implemented using shadow maps to allow users to better visualize a bird being launched into the air.

___ 10: **Scene Modelling:** At least one scene is modelled.