CSC4120 Principle of Computer Game Software

Program Assignment

UFO Catcher(夾公仔機)

Due: May 12, 2020(Friday)

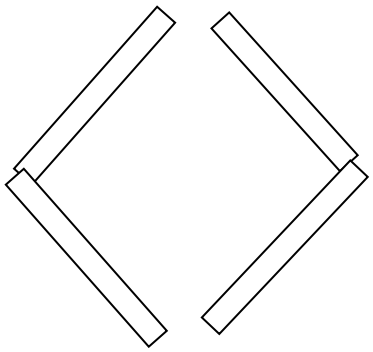
Use of physics effect in game can bring more variations in the outcome as well as game play. In this assignment, we are going to implement a virtual UFO Catcher with the physics effect in a game engine. The game is enjoyed by many people in any gaming arcades. The game played by placing many products, typically puppet dolls in a transparent box. And on the top of the box, there is a robot arm which can grasp things using its claw. The player has to control the arm using skills to grasp his/her own desired puppets. The game's attraction lies in getting some costly merchandise with just a little money.

You can complete this assignment with either Unreal or Unity engine, though Unity will be a preferred choice considering our tutorial content.

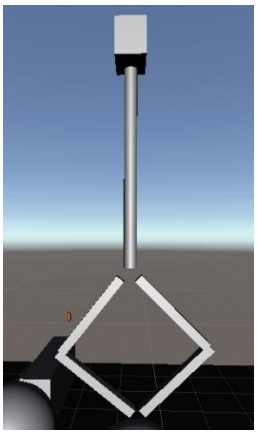
Requirement

You are required to write a program to simulate the clawing game. The program should have the following features:

1. The recommended claw shape must be like this:



Screenshot for the whole set up is as below. The top box is supposed to be clamped to the moving platform. The long rod simulates the rope hanging the claw (we don’t recommend simulate the rope effect here due to the complexity)



2. You are suggested to use the simple 3D primitives eg cubes as the objects to be clawed, though you may consider to use other models as long as your program can show to be able to grab it.
3. The claw should be able to do the following actions:
   I. Descent/ascent
   II. Relax/grab the claw
   III. The whole set up should be able to move along an axis eg horizontally,

For simplicity, the clawer(UFO) should be able to move along one of planar axis i.e. any of x, z direction assume y is the height in 3D space. The workflow should be as follow.

Once the user press the "E/e' key,

I. the clawer should open the claws, and start to descent to a fix distance above the ground,

II. At the bottom position, the clawer then close the claw, grabbing anything in between,

III. The UFO should then ascent to its original top position, wait for a second,

IV. the UFO will then move to some distance away from original position, release the claw (drop the clawed object),

V. UFO then return to its original position, waiting for next command. The program need not respond to any input during this time.

You may also watch the sample video in assignment page for the designated effect.
Requirements:

1. You need not handle the camera in this assignment.
2. As it is a physics simulation, our experiment proved that the whole set up can sometimes go weird. So relax if your setup sometimes deviate from its original position after grabbing an object or hit another object in the middle of a translational motion. All we want is most of the time the whole UFO can do what we required for steps I to V above,
3. We won't accept your project using those robot arms from Unreal/Unity asset stores; just the plain simple rectangular block will do.

The marking scheme will be based on whether you can achieve the above mentioned requirements.

Submissions

Submit your completed project using either Unreal or Unity in a single archive to Blackboard assignment submission page.
http://blackboard.cuhk.edu.hk

Hints

1. If you are working in Unity, making your control scripts under one single file should be better.
2. Unreal also have similar joint system called Physics constraint
   https://docs.unrealengine.com/en-US/Resources/ContentExamples/Physics/1_5/index.html

3. Be careful when a number of objects are articulated together, if their properties are different eg. An object hook up through a hinge joint to another object which is not affected by gravity, altering one object may generate unwanted effect on the other,
4. No matter which physics engine(game engine) you are using, bear in mind that physics engine try their best to simulate real world physics, but NOT EXACTLY the same.