CIT2213 Game Engine Architecture

Lab Exercises: Simple Dynamics

Dr. Minsi Chen

Introduction:

In this lab session, you will be given a rigid body physics skeleton program written in C#. The source files (*SimpleDynamics.zip*) can be downloaded from Learning Resources on UniLearn. This program contains a small set of classes and methods for simulating simple rigid body physics in 2D.

There are currently three types of simulations in the framework:

A. Particle motions - this simulates the basic motion of a particle. You can spawn a particle with a randomly assigned initial velocity by right-clicking on the screen. The only force currently in the world is gravity so each particle simply falls.

B. Spring forces - this simulates a linear springs using the Hooke's law. You can left-click on the screen to create a pair of particles (one static and one dynamic) joined by a spring. The spring force is not currently calculated. The dynamic particle simply falls due to gravity.

C. Upon launching the program, you will also see an inverted pendulum mounted on top of a rectangular block. You can move the base cart using the left and right arrow key. However, the physics for the pendulum has not be implemented, so it simply moves with the cart in its upright pose.

The main focus of this tutorial is to complete the implementation of spring force and the inverted pendulum, i.e simulation B and C. You can find an example demo of the complete simulation under Study Materials on CourseResources.

Exercise 1:

Implement the spring force calculation based on the Hooke's law. A spring is encapsulated in the *SpringJoint* class. The TODO comments in the source will give you guidance on calculating the spring force using Hooke's law.

Exercise 2:

Complete the implementation of the inverted pendulum. The inverted pendulum is encapsulated in the *InvertedPendulum* class. In order to get the pendulum behave plausibly, you need to calculate the torque on the pendulum. In the current configuration, the only force that can cause the pendulum to swing is gravity, see Figure 1.

The *Step* method in the *InvertedPendulum* class has TODO comments signposting the required calculations. You can also to refer to the lecture slide for the formula for angular motions.

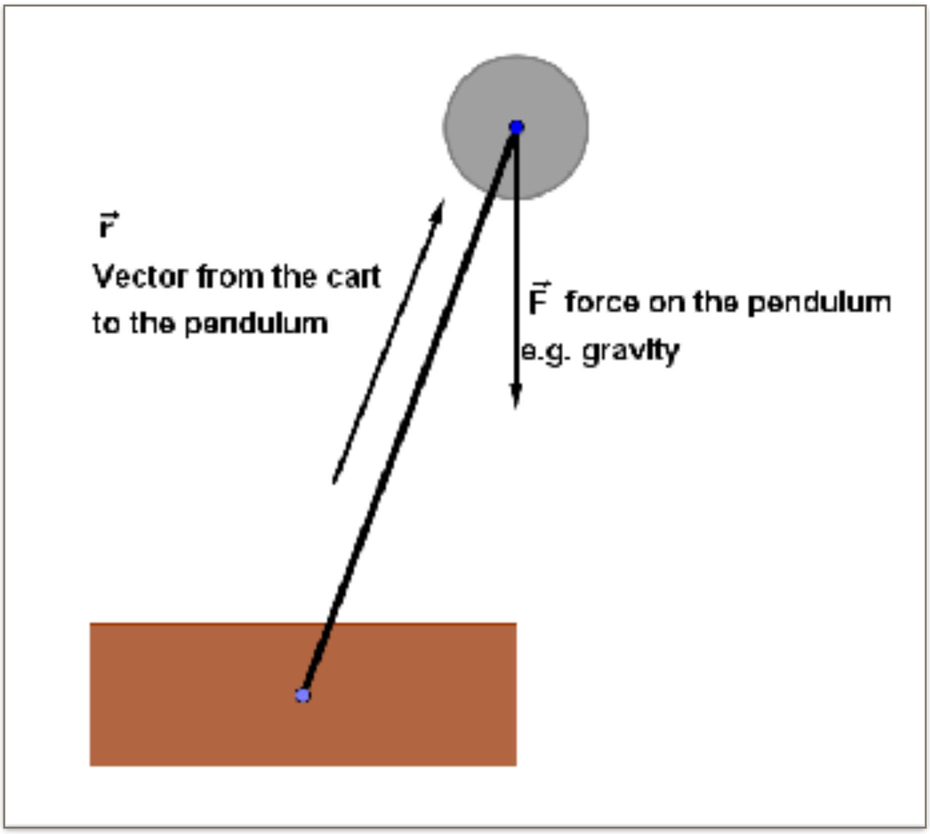

FIGURE 1: INVERTED PENDULUM ATTACHED TO A BOX CART.

Exercise 3:

Using the SimpleDynamics skeleton, implement a brute force n-body simulation using Newton's gravity. Wikipedia has a general definition of n-body simulation (see https://en.wikipedia.org/wiki/N-body_simulation). The essence is to model the gravitational pull amongst all the rigid bodies and observe the evolution of the formation.

Exercise 4 (Unity3D):

In Unity3D, 3D rigid bodies are encapsulated in the *RigidBody* class. You can find the design description of this class at (https://docs.unity3d.com/560/Documentation/Manual/RigidbodiesOverview.html). Its programming interface is described here (https://docs.unity3d.com/560/Documentation/ScriptReference/Rigidbody.html).

For this exercise, you will modify the Roll-a-ball tutorial (https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial) so that it uses torque to control the movement of the ball instead of force. In order for this to work, both the ball and the platform (play area) need to have friction. Friction can be set via physics materials which can be assigned to game objects, see https://docs.unity3d.com/560/Documentation/Manual/class-PhysicMaterial.html.