CS 251: Lab 03: [Code Warrior] Box 2D

- Handed out: 07/Aug Due: 10/Aug Mon 11pm
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Overview

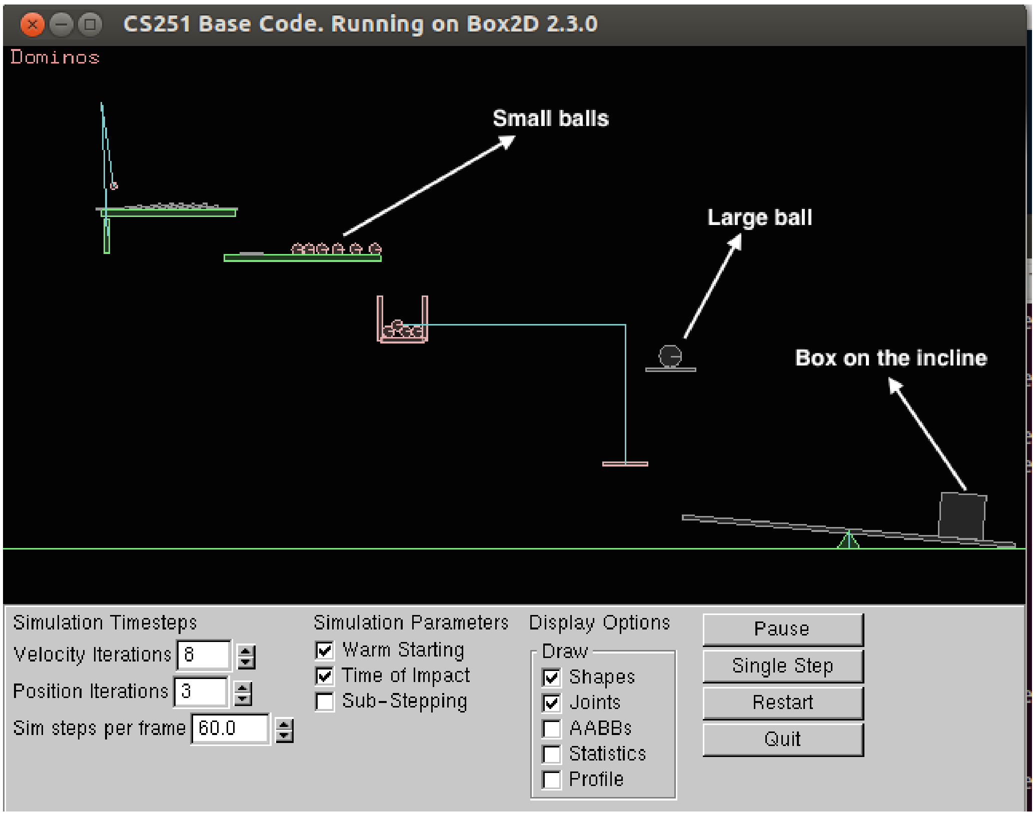
The purpose of this task is to improve your coding experience building upon on what you did in CS101 and CS154. In particular, we will take a industry-robust piece of software, learn how to build a library using the command line, and understand some parts of the source code.

Non-goals: The goal of this lab is NOT to fully understand Box2D but just have the ability to tinker with it ("some parts"). Later on you will use this software to create your course project.

If you have difficulty in building Box2D, you can still learn how to do it in the inlab, and then be in a position to complete the outlab.

The Tasks

1. Download the pre-packaged Box2D code from the lab03 assignment directory.
2. (Nothing to submit) Follow the instructions in the file `Instructions.txt` carefully. I cannot overemphasize this part: "carefully". At the end of this you should be able to get this picture



CS251 Base Code. Running on Box2D 2.3.0

Dominos

Simulation Timesteps

Velocity Iterations 8

Position Iterations 3

Sim steps per frame 60.0

Simulation Parameters

☑ Warm Starting

☑ Time of Impact

□ Sub-Stepping

Display Options

Draw

☑ Shapes

☑ Joints

□ AABBs

□ Statistics

□ Profile

3. (To Submit) You have to accomplish the following (run the simulation first):
   (a) Disturb the square box on the inclined plank with the minimum number of (small) balls on the horizontal plank.
   (b) Make the box on the inclined plank and the larger ball move to the right instead of left.

Important: Make sure you stick to the following constraints:

(a) You are not allowed to change the properties of the small balls.

(b) You are allowed to add objects, but every instance of the new addition (if any) should play a role in the simulation.

(c) Do "minimal" changes to the code. (Minimal is hard to define, but basically this is a hint – you can achieve these tasks with very small changes.)

4. Challenge Question: Remember, you can get full points if you do NOT complete the challenge question. We will not think less of you if you do not complete the challenge question. Box2D allows you to use the mouse to "control" the objects in the simulation. So you can drag and move the objects around with your mouse. Pick one object you like and make it draggable. You need to override `mouse_up`, `mouse_down` and `mouse_move`. These are three virtual functions declared in `cs251_base.hpp`. Using `b2MouseJoint` would make your life easier. Here is a link to a javascript example `http://www.binarytides.com/mouse-joint-box2d-javascript/`

   Note: A `b2MouseJoint` object has nothing to do with the mouse. It can help move an object from one place to another.

What to Submit

For this assignment, we want to continue to teach you how to be parsimonious. Therefore, we will be extra careful, and we will deduct points if you submit anything that we don't care for. In particular, we do not want you to submit anything that we cannot generate by ourselves.

Submit, for each of the three tasks above (including the challenge question, if you have attempted it), only those file(s) that you have modified. The directory structure should be as follows

```
lab03_groupXY_final.
  |- part1.
  |    |- All modified source files.
  |- part2.
  |    |- All modified source files.
  |- challenge.
  |    |- All modified source files.
  |-  readme.txt
```

The folder and its compressed version should both be named `lab03_groupXY_final`. Hence, you submit a `tar.gz` named `lab03_group07_final.tar.gz`

How we will score you

Scoring: In this lab, we will assume that you are perfect, and count down from 100. Every time you do something poorly or incorrectly, we will start deducting marks.

- First, we will look at the semantic content of the submission, i.e., the change. We will re-compile the program and run to see the changes.
- Next, we will do a syntactic check of your submission. If you sent anything extra, you will lose a significant percent of the marks.
- The distribution of marks among the tasks is as below:
  - Task 3(a): 30
  - Task 3(b): 30

  Score for the challenge question is not mentioned in this first release.