Modern Robots: Evolutionary Robotics

Programming Assignment 4 of 10*

Description

In this assignment you will download and install the Bullet Physics Library, the open source physics engine we will be using for the remainder of this class. After installation and compilation, you will run a test application that comes with Bullet which simulates ragdolls (Fig. 1a). You will then compile the source file provided with this assignment, which will become the basis for all future assignments.

Important: If you get stuck installing Bullet, get on the Bullet forum and ask your question; you will usually get a response in a couple of hours. **This means of course that you will have to start the assignment before the night before.** If you are still stuck, go through the steps for getting unstuck on the syllabus.

Tasks

1. Back up your code from last week. If you lose your laptop tomorrow, will you still have your code from assignments 1-3?

2. Download Bullet version 2.81 (bullet version 2.82 may work as well, but this has not been tested). Download instructions can be found on Bullet's main page. Make sure to download from code.google.com (https://code.google.com/p/bullet/downloads/list as of the time of writing).

3. Unzip the archive.

4. Follow the installation instructions found in INSTALL.

If you are a Mac user, there have been problems in the past where any object will just start to slide across the environment. This may be fixed by providing the option `USE_DOUBLE_PRECISION` to the bullet installer. If you do so, you'll have to compile all future programs with: `-DBT_USE_DOUBLE_PRECISION`.

5. For windows users you should now find executables of the demo programs in
`bullet-2.81/src/.libs/libBulletDynamics.dll`. For Linux and Mac users, you should find the executables in
`bullet-2.81/Demos/RagdollDemo`. Run the RagdollDemo executable. The executables are in the home directory of bullet, if made with visual studio, and in the same place as Mac and Linux if using cmake (although some might not work).

*Original material was graciously provided by Josh Bongard. Jeff Clune slightly modified it. Joost Huizinga heavily modified it.

6. You should see something like in Fig. 2. Take a screen capture of the simulation window, and include the result into your Latex PDF document.

7. Try familiarizing yourself with the commands given in Figure 1.

Key	Description
l,r,f,b	Move camera left, right, front, or back
z,x	Zoom in or out
e	Add another ragdoll to the simulation
g,u	Toggle shadows, and textures
h	Toggle debug text
w	Draw wireframes and body reference frames
a	Draw Axis Aligned Bounding Box (AABB)
c	Draw contact normals (black lines)
C	Draw contraints
L	Draw contraint limits
d	Disable body deactivation
space	Reset scene
.	Shoot box (or right-click)
+,-	Increase or decrease box speed
left-click-and-drag	Drag a body part
q	Quit application

Figure 1: Brief list of key commands for Bullet demo applications.

8. Extract `hw_4_code.zip` and compile the supplied `main.cpp`. To compile `main.cpp` you will need to include the following folders, and link against the following libraries (note, replace all instances of `<bullet_folder>` with the actual bullet folder on your machine. Also, do not type the '<' and '>' symbols.):

1. Include: `<bullet_folder>/Demos/OpenGL`
2. Include: `<bullet_folder>/src`
3. Add library path: `<bullet_folder>/lib`
4. Link: `OpenGLSupport`
5. Link: `BulletDynamics`
6. Link: `BulletCollision`
7. Link: `LinearMath`

You will also need to link against the appropriate GLUT, and OpenGL libraries. The exact commands necessary will differ between systems and operating systems, here are some example commands for MacOS and Linux.

MacOS
Navigate the the folder containing `main.cpp`.

Use the following two command to compile and link `main.cpp`:

```
g++ -I"<bullet_folder>/Demos/OpenGL" -I"<bullet_folder>/src" -c -o main.o main.cpp
g++ -L"<bullet_folder>/lib" -framework GLUT -framework OpenGL -o hw4 main.o
-lOpenGLSupport -lBulletDynamics -lBulletCollision -lLinearMath
```

Repeated in small font for easier copy-pasting (do not forget to replace `<bullet_folder>` though).

```
g++ -I"<bullet_folder>/Demos/OpenGL" -I"<bullet_folder>/src" -c -o main.o main.cpp
g++ -L"<bullet_folder>/lib" -framework GLUT -framework OpenGL -o hw4 main.o -lOpenGLSupport -lBulletDynamics -lBulletCollision -lLinearMath
```

Linux
Navigate the the folder containing `main.cpp`.

Use the following two command to compile and link `main.cpp`:

```
g++ -I"<bullet_folder>/Demos/OpenGL" -I"<bullet_folder>/src" -c -o main.o main.cpp
g++ -L"<bullet_folder>/lib" -o hw4 main.o -lOpenGLSupport -lBulletDynamics
-lBulletCollision -lLinearMath -lGL -lGLU -lglut
```

Repeated in small font for easier copy-pasting (do not forget to replace `<bullet_folder>` though).

```
g++ -I"<bullet_folder>/Demos/OpenGL" -I"<bullet_folder>/src" -c -o main.o main.cpp
g++ -L"<bullet_folder>/lib" -o hw4 main.o -lOpenGLSupport -lBulletDynamics -lBulletCollision -lLinearMath -lGL -lGLU -lglut
```

9. Run the compiled version of `main.cpp`. You should now see a window that looks like figure 3. Screen-capture the simulation window and paste the result into your document.

10. The simulation is currently paused. Press 'p' to unpause the simulation and turn the camera to produce an image that looks like figure 4. Add the image to your document.

Deliverables

A pdf document containing the figures resembling figures 2, 3, and 4.

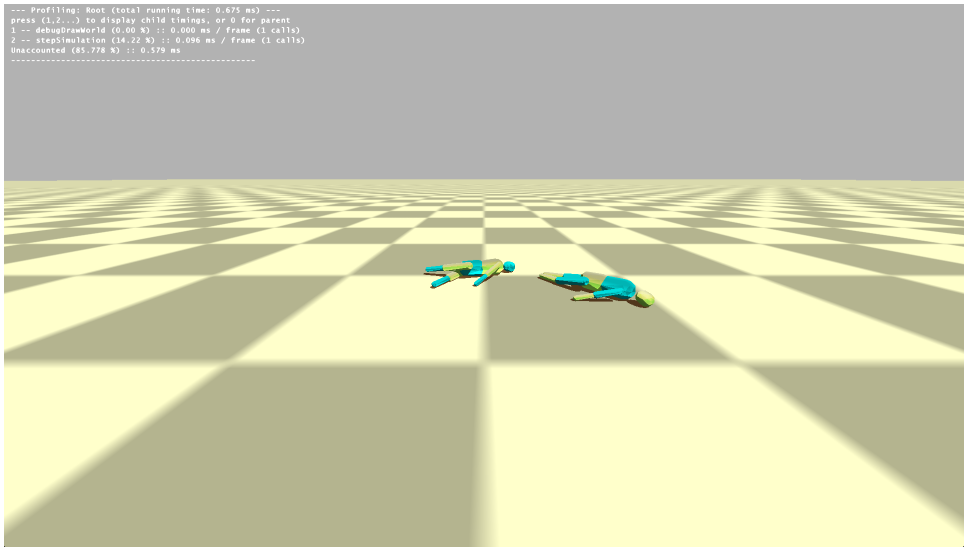

Figure 2: Successful compilation and execution of the RagdollDemo program should produce this figure.

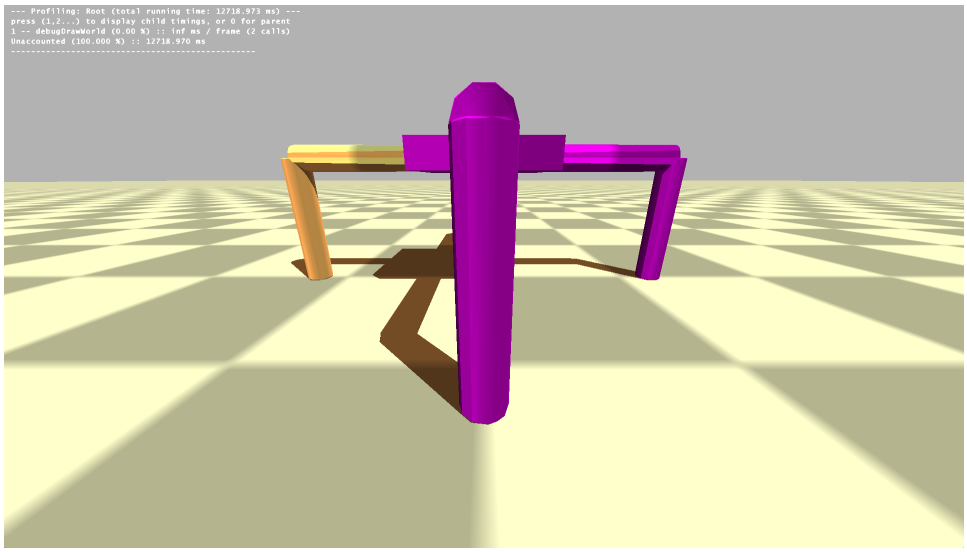

Figure 3: Successful compilation and running of the supplied `main.cpp` should produce this image.

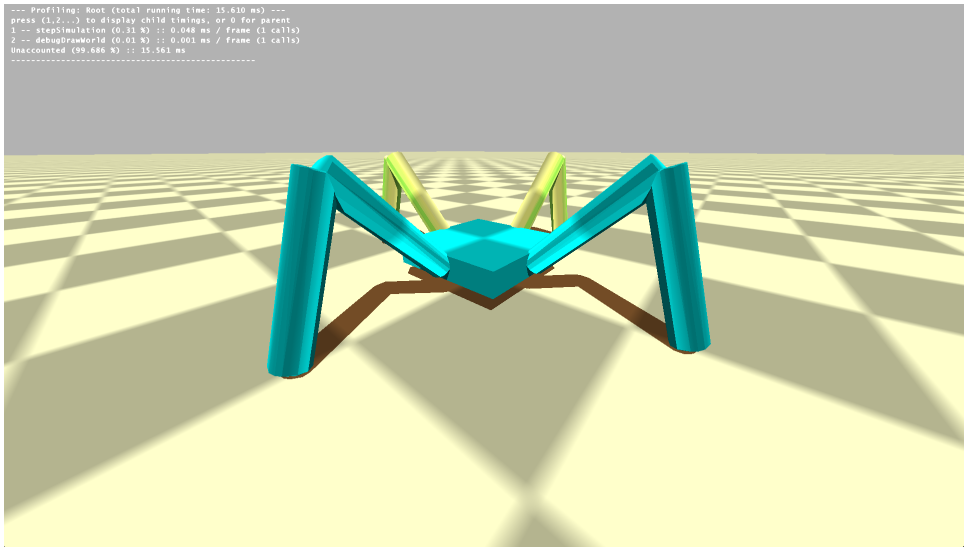

Figure 4: Unpausing the simulation and turning the camera should produce this image.