COP 290 - Assignment 1
Changes and added functionalities

Akshay Kumar Gupta 2013CS50275
Barun Patra 2013CS10773
J. Shikhar Murty 2013EE10462

Abstract

The document enumerates and expands on the differences and/or additions to the final project were previously not mentioned in the design document.

1 One-to-One thread Communication Model

Communication between threads has been made one-to-one. Each thread has access to the data of the `Ball(s)` it controls and the `MessageQueue` of the other threads. The `MessageQueue` of a thread contains the position of all other balls, which is processed by the thread, once the `MessageQueue` is full. In this way, a thread cannot change the data members of any `Ball` that it does not control. This is the model of communication adapted.
The `MessageQueue` class has its own `Mutex` and `Conditional variable` that allows safe, race free insertions and deletions.

2 GUI

1. Buttons

   All `Buttons` have been created manually. The `Button` Class defines the properties of a `Button`. It has the following members:

   - `xTopLeft` – The x coordinate of the top left corner of the button.
   - `yTopLeft` – The y coordinate of the top left corner of the button.
   - `width` – The width of the Button
   - `height` –The height of the Button
   - `text` – The Label on the Button
   - `textLength` –Length of the text, used to centre the font.
   - `click_color` –The color of button when clicked.
   - `back_color` – The general background color of the Button.

- `pressed` – Whether the Button is pressed or not.

The objects of the `Button` are increaseSpeed, decreaseSpeed, Enable/Disable Gravity and Pause/Play. They are discussed below:

- `increaseSpeed` – Increases the speed of the selected `Ball`, upto a defined `Max Velocity`
- `decreaseSpeed` – Decreases the speed of the selected `Ball`
- `Enable/Disable Gravity` – Toggles between enabling the effects of gravity inside the `Box`.
- `Pause/Play` – Used to pause the simulation, so as to allow easier selection.

2. KeyBoard Controls

   The following is a list of the `KeyBoard functions` available:

   - `Increase/Decrease Speed`: `w` and `s` allow the user to increase and decrease the speed of a selected `Ball` respectively.
   - `Toggle Selection`: `a` and `d` allow the user to toggle between selected `Balls`.
   - `Enable/Disable Gravity`: `g` allows the user to enable/disable the effects of gravity within the `Box`.
   - `Pause/Play`: `Space_Bar` allows the user to pause the simulation, and resume the same, as per his convenience.
   - `View Selection`: The `Up`, `Down`, `Left` and `Right` arrow keys allow the user to change the view of the camera, allowing the user to view the `Box`, and the `Balls` from different angles,while facilitating the selection of `Balls`.
   - `3D-2D-3D`: The `2` key is used to toggle between 2D and 3D modes. However, in the 2D mode, the rotation buttons are disabled.

3. Mouse Controls

   - `Selection`: Clicking on a `Ball` would select the `Ball`. In case a ball to be selected is hidden behind another, rotating the `box` and then selecting the `Ball` would achieve the same.

3 Additional Features

1. `Single Thread controlling multiple Balls`: The number of `Balls` and the number of `Threads` are command line inputs. The program functions smoothly for cases wherein num_Balls $\geq$ num_Threads. Given num_Balls = m and num_threads = n (m $\geq$ n), each thread controls $\lfloor \frac{m}{n} \rfloor$ `Balls`.

2. `Realistic Collisions`: The *Coefficient of Restitution* C_{R} is a command line argument, and by default is set to 1.0 (Elastic collision). Any other value of C_{R} ($0.0 \leq C_{\mathrm{R}} \leq 1.0$) adjusts the `updateBalls()` (mentioned in the Design Document) to handle the collisions appropriately, simulation collisions somewhat similar to what happens in real life.

4 Debugging

All debugging has been done on the LLDB debugger on Xcode version 6.0.1