

BASIC 3D SHOOTING GAME

Final Project

Basic 3D Shooting Game

Yuji Shimojo

CMSC 325

Instructor: Dr. Amitava Karmaker

March 10, 2013

BASIC 3D SHOOTING GAME

Introduction

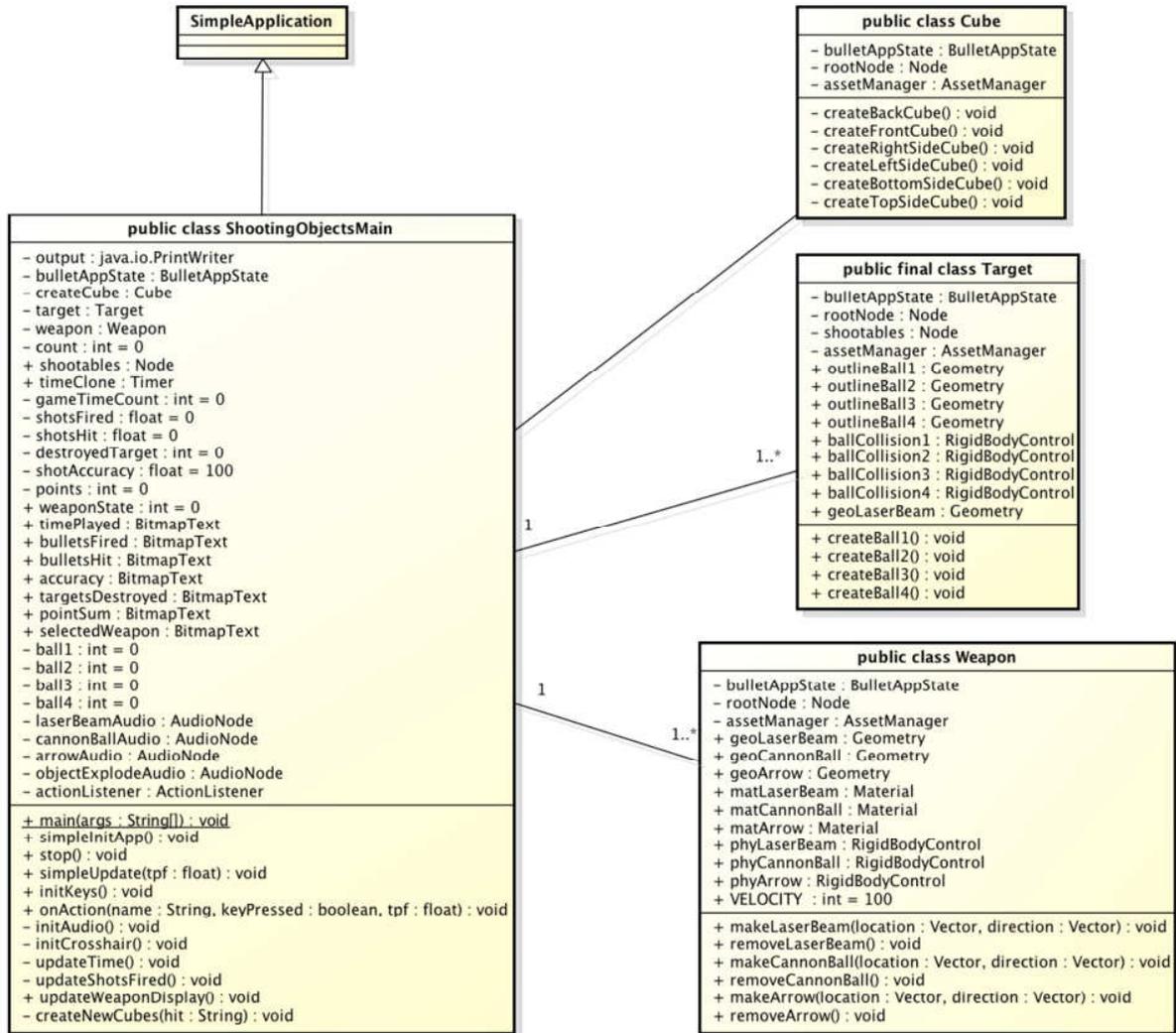
I created a basic Java shooting game with jMonkeyEngine (jME) framework. To be precise, my development environment is jME 3 SDK RC2 on Mac OS X Snow Leopard (10.6). My project is mainly based on a sample code given by the professor in the class. I redesigned the program including dividing and adding classes, and enhanced the functions.

Operating Instructions

1. Double-click ShootingObjectsMain.jar in the attachment folder, and then the jME window will open.
2. Press 'OK' to start the application.
3. After you start the application, you will see four of moving balls in a scene.
4. You can shoot those balls by clicking the left mouse button or the space bar.
5. You can destroy: the biggest ball by shooting one time for getting 2 points; the second biggest ball by shooting two times for getting 6 points; the second smallest ball by shooting three times for getting 10 points; and the smallest ball by shooting four times for getting 16 points.
6. You can change the weapon: press 'L' for Laser Beam, 'C' for Cannon Ball, or 'A' for Arrow.
7. After two minutes, the game will stop and show your final score.

Class Diagram and Design Description

BASIC 3D SHOOTING GAME



First, Cube class creates cubic scene objects, made by dividing the professor's sample code. The reason I did is to simplify and encapsulate each class more.

Second, Target class creates balls which are shot, made by dividing the professor's sample code as well as the Cube class.

Next, Weapon class creates three types of weapons shot by the user. They are a laser beam, a cannon ball, and an arrow.

BASIC 3D SHOOTING GAME

Finally, ShootingObjectsMain is the main class of my final project including a main method, which creates instances of Cube, Target, and Weapon classes. Its main functions are text display on the screen, file export, and event handlers for key presses and collision detections.

Assets and Library

I downloaded the sounds of laser beam, stoning, and releasing an arrow from a bow from a sound archived Website for this project ("SoundJax," 2013). Also I added jme3-test-data library to use a stone material for the cannon ball.

Research Methods and References

As mentioned earlier, I enhanced the professor's sample code, and I took sort of code snippets from Hello Audio and Hello Physics in jME tutorials ("jMonkeyEngine Tutorials and Documentation," n.d.). Furthermore, some videos on YouTube explains usage demo of jME components and methods. For example, a video I saw shows how to use Arrow class clearly (Wesley Shillingford, 2012).

Test Data

No	Bullets Fired	Bullets Hit	Accuracy	Targets Destroyed	Points	Weapon	Targets
1	268.0	97.0	36.19403%	85	218	Laser Beams	Bigger balls
2	205.0	93.0	45.36586%	83	202	Cannon Balls	Bigger balls
3	209.0	112.0	53.58852%	93	262	Arrows	Bigger balls
4	264.0	79.0	29.92424%	28	276	Laser Beams	Smaller balls
5	262.0	101.0	38.54962%	32	352	Cannon Balls	Smaller balls
6	291.0	123.0	42.26804%	40	440	Arrows	Smaller balls
7	470.0	129.0	27.44681%	70	380	Laser Beams	All balls
8	428.0	141.0	32.94392%	75	430	Cannon Balls	All balls
9	427.0	163.0	38.17330%	80	514	Arrows	All balls

Results and Consideration

When I focus on bigger balls, the points are fairly low in whole. Because there are many of bullets fired when I don't have any aims, points are relatively high. In weapons, using arrows is expected to get high scores the most maybe because arrows are thinner and sharper so I could pinpoint the objects unintentionally.

References

- SoundJax. (2013). Retrieved March 10, 2013, from <http://soundjax.com/>
- Wesley Shillingford. (2012). [jME Tutorial 1-6] Rotations in the jMonkeyEngine (Part 2/2), Retrieved March 10, 2013 from <http://www.youtube.com/watch?v=tYZdqmsegFY&list=UUB59JrgCNoRddloox7ySkBQ&index=30>
- jMonkeyEngine Tutorials and Documentation. (n.d.). *jMonkeyEngine.org*, Retrieved <http://jmonkeyengine.org/wiki/doku.php/jme3>