

Assignment 1

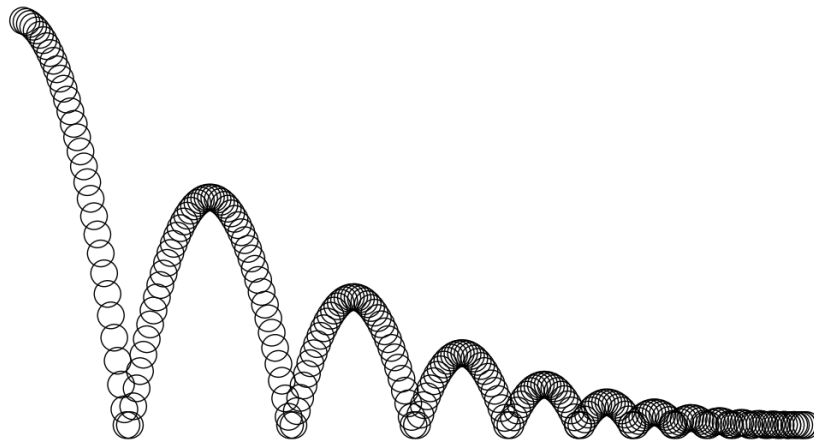
Computer Graphics, Spring'23

IIT Kharagpur

Posted: 26th January 2023
Due: 9th February 2023, 11:55 PM

Description

The purpose of this assignment is to apply Bresenham's circle drawing algorithm to create the trajectory of a bouncing ball. Suppose that a ball is thrown horizontally at some elevation above ground. The ball starts bouncing, and eventually after some time it stops. You may use standard equations of motion to model the trajectory. Let us consider that different frames are drawn into a single image, in which the trajectory looks like the following:



Your task is to implement this in OpenGL using C/C++. Your code should produce the trajectory similar to the above. Feel free to use assumptions like the height from which the ball is thrown, velocity, condition when the bouncing will be stopped, etc. Your trajectory need not be exactly the same as what is shown in the figure, but you have to make sure that it looks relevant.

Bonus mark

If you can make the ball to squish into an ellipse when it hits the ground (considering the ball to be elastic), that will give you 3% bonus mark (will be adjusted either with this assignment, or with the final grade).

Weightage

This assignment carries 10% of the total mark.

What to submit?

Submit the program file(s) you have implemented. You must use OpenGL with C/C++ to implement the assignment. Put all the file(s) into a zip and submit in Moodle (no files will be accepted by email). Please do not submit any unnecessary files (such as the whole project).

Plagiarism

Copying the code is a serious academic offence, which will be treated with zero tolerance. Remind that changing variable names and white spaces do not make your code unique, it's very easy to detect these cases using softwares.

General marking scheme

The marks will be distributed as follows:

- Working program: 70%
 - Properly drawing the circles: 20%
 - Making the proper trajectory: 30%
 - Displaying properly within the viewport: 10%
 - Aesthetic aspect of the output: 10%
- Documentation: 10%
 - Main comment block identifying the student (name, roll number, email address): 4%
 - Defining input and output parameters for a function: 3%
 - Purpose of functions/blocks of code: 3%
- Program style: 10%
 - Meaningful variable names: 3%
 - Constants instead of “magic numbers”: 2%
 - Readability (complete sentences, indentation, white spaces, etc): 2%
 - Code flows “nicely”: 3%
- Program structure: 10%
 - Modular code: 4%
 - Uses appropriate data structure: 3%
 - Loops when needed/no loops when not needed: 3%