

School of Interactive & Digital Media
Digital Entertainment Technology
2007 S1

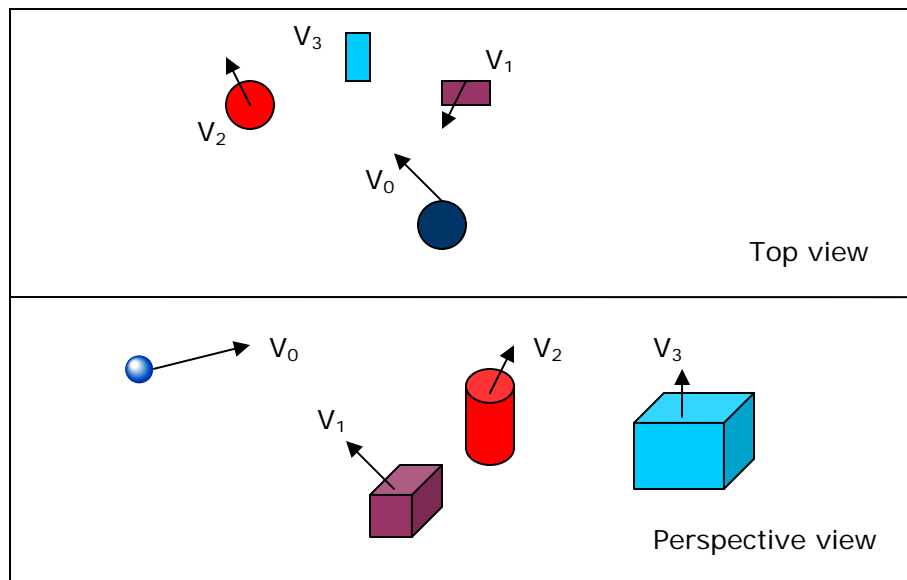
DM2212 – Programming Physics Assignment 1: 3D kinematics

Value: 15%
Due date: Friday, 15/June 9am - part a → CD submitted in cabinet BIK M level 5
Friday, 22/June 9am – all → CD submitted in cabinet BIK M level 5
Extra credit for early submission
Teams: 3 people per team

Description

Create an interactive application that realistically simulates 3D projectile motion of a ball. The ball will be thrown with an initial velocity V_0 into an area populated with static objects that will “respond” with velocities V_i and make the object “bounce” (note: no dynamics are involved; calculate movement using gravity/velocity)

Show a top and perspective view of the environment, i.e. your application window should look like:



Software to use:
C/C++, OpenGL, GLUT

Conditions/restrictions:

- There must be at least 3 objects to interact.
- Render an arrow to represent velocities (V_i)
- The program will have 2 options:
 - a) Let the user specify initial position (x,y,z) and velocity of the ball (using arrow keys for direction, space bar for initial speed) and then calculate/animate the resulting movement.
 - b) Let the user specify the initial position and an object i , and then the program calculates the initial velocity V_0 that the ball should have in order to hit object i .
- The user must be able to modify the position (x,y,z) and “response” velocity of any object i .
- In order to fire the ball, the user must press the space bar. The longer the bar is pressed, the bigger the initial speed.
- After the object hits the ground, stop the animation and print the final velocity and total horizontal distance moved.
- You must use your own vector class and any other classes/libraries to calculate the movement. No library other than `<math>` can be used.
- The application must reset after the object hits the ground and prints results.

Grading:

Previous exercises (vector class, proofs, etc)	3%
Top and perspective views, UI	2%
Movement animation given V_i and initial pos.	5%
Calculation/animation of V_0 in order to hit object i	5%

Extra credit (max 5%):

1. Show an speedometer (1%)
2. Add the ability to move objects after the ball is fired. (1%)
3. Let the user specify initial position of ball (V_0) and positions of all objects, then calculate all velocities in order for the ball to bounce on every object at least once. (3%)

Suggested project schedule:

Week 4 – Window and viewports creation, object creation (or model loading) and UI

Week 5 – Constant velocity movement, addition of velocities

Week 6 – Gravity

Week 7 – Calculation of V_0 given initial position and object i