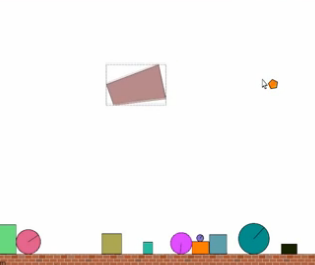
Ian Myers

2D Physics Simulator

The goal is to build a 2D physics simulator, which is essentially a physics engine combined with a GUI that allows the user to create objects with various properties. The physics simulator will be used by people who are interested in learning and/or experimenting with 2D physics. Users will be able to simulate the ballistics of objects falling through Earth’s atmosphere, bullets, cannon balls and rockets by adjusting their mass, velocity, the forces acting upon them, etc... The simulator is not meant to be highly graphical. For example, if you want to simulate the ballistics of a rocket you just add an object, like a circle, and add an upwards force to it.

The heart of the physics simulator is its integration methods. These are the methods that calculate the location of all objects based on variables such as velocity and acceleration. The GUI will allow you to choose one from a few types of integration methods that will vary in sophistication, precision and processor intensity. They will account for linear velocity and forces as well as angular velocity and forces. The collision detection between objects will also consider angular velocity and forces, so that if you drop a circle on another object it will roll off the side. The types of physical bodies that are being simulated are known as 2D oriented rigid bodies, which are depicted in the image to the right.

# Other 2D Physics Simulators

There are a handful of good quality 2D physics simulators already available. Many of these have been around for a long time and are very sophisticated. The purpose of this project is to create something that is technically challenging for me, rather than to create something that is competitive with similar products.

* Other Physics Simulators
  1. Physion (physion.net)
  2. Algodoo (algodoo.com)
  3. The Powder Toy (powdertoy.co.uk)
  4. OE-Cake (no website)
  5. OSP EJS (opensourcephysics.org)

# The Basic Structure

The physics simulator will be composed of two main parts: a single physical world and physical objects that you can add to the world. Both of these will have their own set of variables / properties that you can change, such as mass, velocity and gravitational acceleration.

Properties of the physical world

* Vectors
  + Gravitational acceleration
* Scalars
  + Width
  + Height
  + Atmosphere’s density
  + Elasticity of the ground

Properties of physical objects

* Polar Vectors
  + Velocity (x, y)
  + Acceleration (x, y)
  + Location (x, y)
* Axial Vectors
  + Angular Velocity
  + Angular Acceleration
* Scalars
  + Mass
  + Width
  + Height
* Shapes
  + Circle
  + Square

The physical objects can be circles or polygons. The world will be in charge of keeping a list of objects, collision detection and integration. The simulation can be started, paused, reset and rewound.

The interface will allow you to test all the features available in the physics engine. Every variable will be shown in the interface and can be edited. For convenience, I will supply at least 8 templates for simulating various scenarios, such as launching a rocket or firing a bullet or cannon ball.

# User Stories

1. As a user, I want every object I create to be influenced by gravity.
2. As a user, I want there to be a ground at the bottom with collision detection.
3. As a user, I want to simulate projectiles, such as bullets and cannon balls.
4. As a user, I want the option to trace the entire motion of simulated projectiles.
5. As a user, I want to simulate objects that propel themselves, such as rockets and missiles.
6. As a user, I want premade templates for objects such as bullets, cannon balls and rockets for convenience.
7. As a user, I want to be able to switch from a side-view with the ground at the bottom to a top-view to simulate other things such as a pool table.
8. As a user, I want to simulate an object falling through Earth’s atmosphere.
9. As a user, I want premade templates for the 8 planets plus Pluto and the Moon so I can see how gravity on other planets affects things.
10. As a user, I want to be able to create squares that interact with each other naturally, such as if I drop a few on top of each other I want them to tumble/rotate.

# Timeline

* Week 1
  1. Create a physical world that physical objects can be added to
  2. The physical world will have the following properties:
     + Width
     + Height
     + Gravitational acceleration
  3. The physical world will have a ground
  4. You can add borders to the world
  5. The ground and borders will have an elasticity property
  6. Euler integration method
* Week 2
  1. The physical objects will have the following properties:
     + All polar vectors
     + All scalars
  2. Trace option for physical objects
  3. Start, pause and clear simulation
* Week 3
  1. Collision detection
  2. Templates for physical objects
  3. Templates for the world (8 planets and Pluto)
  4. Templates for a complete setup such as a pool table
* Week 5
  1. Add more integration methods:
     + Velocity Verlet
     + Störmer–Verlet
  2. Space option for the physical world
* Week 6 and 7
  1. Axial vectors for physical objects
  2. Collision detection will account for axial vectors
  3. Integration methods will account for axial vectors
* Week 8
  1. Runge-Kutta 4 (RK4) integration method
  2. Move backward in time
* Week 9 and 10
  1. Oriented rigid body physics with squares
* Extras
  1. Pin objects so they don’t move
  2. Ability to right-click on objects to bring up their properties and edit them
  3. Springs
  4. Ropes
  5. Fluids
  6. Explosives