# Multi-Dimensional Pong

Cameron Cuzmanko

November 14, 2015

**Abstract**

The purpose of this project is to present a classic game, Pong, in a new and innovative way by building a three-dimensional projection of four-dimensional Pong using VPython. Briefly, for anyone who may not know, Pong is a two-dimensional game that involves two paddles virtually bouncing a ball back and forth until one of the paddles misses.

## 1 The Game

Pong was invented in 1972 by Allan Alcorn. The game, originally intended to be Alcorn's introduction to game coding, became an instant hit throughout the nation and is still regarded highly in the modern industry. Initially exclusive to arcades, Pong soon moved into households and launched Atari (the company behind the game) into instant success. Many credit Pong as the game that launched the video game industry to brand new heights.

## 2 The Mathematics

### 2.1 Collision Detection

As the game runs, the program constantly needs to check if the ball has come into contact with a wall or paddle. To do this, the program needs to constantly be checking the location of the ball relative to the locations of the walls and paddles. To make the most realistic/aesthetically pleasing scenario, the program needs to account for the radius of the ball when detecting collisions. If it is not accounted for, then half the ball will warp through the wall before a collision is recognized. Therefore, the ball's position +/- it's radius must be greater than the negative position components and less than the positive position components of the walls.If that condition is failed, then the program will understand that a collision has occurred and react accordingly.

### 2.2 Momentum

Before collisions can be understood, one must understand the basics of momentum as momentum is rudimentary to the idea of elastic collisions. Momentum

is representative of mass in motion. Momentum is represented by the equation

$$\vec{p} = m\vec{v} \tag{1}$$

where $\vec{p}$ is momentum of the ball
$m$ is the mass of the ball
and $\vec{v}$ is the velocity of the ball

The law of conservation of momentum is directly applicable to collisions. The law states that in a collision, the force exerted on object 1 is equal and opposite to the force exerted on object 2. This even applies when there is energy loss in the collisions. To derive this law, we need to apply the impulse-momentum theorem which is represented by

$$\vec{F}t = \vec{p} \tag{2}$$

or

$$\vec{F}t = m\vec{v} \tag{3}$$

In a collision, the forces that the two objects exert on each is equal and opposite, and the time the collision takes place is equal for both objects. Because of this,

$$\vec{F_1}t_1 = -\vec{F_2}t_2 \tag{4}$$

$$\vec{p_1} = -\vec{p_2} \tag{5}$$

$$m_1\vec{v_1} = -m_2\vec{v_2} \tag{6}$$

where $\vec{F}$ is the force
$t$ is the time the force acts on the object

This final equation shows that whatever momentum is lost by one object is gained by the other. Hence, momentum is conserved.

## 2.3 Collisions

In the case of my pong game, the only collisions occurring will happen between the ball and the walls or paddles. Without collisions between multiple balls, collisions can be handled quite simply. I will be focusing on only elastic collisions. An elastic collision is a collision in which kinetic energy and momentum are conserved. The significance of this is that the ball retains all of its initial velocity upon completion of a collision. Since the wall has no velocity, the only velocity that will be changed is that of the ball. To represent the collision in code, one can simply flip the sign of a single component of the velocity. This can be written as

$$\vec{v_{ci}} = -\vec{v_{cf}} \tag{7}$$

where $\vec{v_{ci}}$ is an initial (pre-collision) component of the velocity
and $\vec{v_{cf}}$ is a final (post-collision) component of the velocity

In order to understand which component to flip, the program will need to identify what the ball is colliding with. Since each wall and paddle is initiated separately, the program can differentiate between each. Therefore, if the ball collides with the wall in the yz-plane, then the program can identify that it needs to flip the x-component of the velocity.

# 3 Artificial Intelligence

Artificial intelligence, henceforth referred to as AI, is used to generate behaviors in non-player characters that exhibit human-like intelligence. In this Pong game, I will need to implement an AI for a player to play against. In order to make a successful AI, I will need to program an intentionally imperfect AI. An AI that always wins would be no fun for the player, so I must make sure the AI can lose. In order to accomplish this, I will use conditional statements that will check the position of the AI's paddle relative to the ball. Once the AI has determined which directions it needs to travel, it will move a slight bit in that direction. I will limit the AI's ability to reach well placed shots so that it is beatable.

# 4 Goals

- Build an aesthetically pleasing three-dimensional projection of a four-dimensional playing field for the Pong game
- Program smooth movement into the game for both the ball and paddles
- Create an intentionally imperfect AI that uses relative position to determine where to shift its paddle