

# Projectile Trajectory Server Documentation

## Overview

This C program simulates the trajectory of a projectile and provides a TCP server that calculates and sends the projectile coordinates to a client. The server receives the initial velocity, angle, and gravitational constant from the client, computes the trajectory, and returns the coordinates.

## Files

- **server.c:** Contains the main server logic and trajectory calculation functions.

## Dependencies

The program requires the following headers:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <math.h>
```

## Functions

```
double calculateX(double v0, double theta, double t)
```

Calculates the horizontal position  $x(t)$  of the projectile at time  $t$ .

### Parameters:

- **v0:** Initial velocity of the projectile.
- **theta:** Launch angle in radians.
- **t:** Time at which to calculate the position.

### Returns:

- The horizontal position  $x(t)$ .

```
double calculateY(double v0, double theta, double t, double g)
```

Calculates the vertical position  $y(t)$  of the projectile at time  $t$ .

### Parameters:

- $v_0$ : Initial velocity of the projectile.
- $\theta$ : Launch angle in radians.
- $t$ : Time at which to calculate the position.
- $g$ : Gravitational constant.

#### Returns:

- The vertical position  $y(t)$ .

```
int CalculateProjectileTrajectory(double *BufferCoordinatesX, double
*BufferCoordinatesY, double Velocity, double PTheta, double ConstG)
```

Calculates the projectile's trajectory and stores the coordinates in the provided buffers.

#### Parameters:

- `BufferCoordinatesX`: Buffer to store the X coordinates.
- `BufferCoordinatesY`: Buffer to store the Y coordinates.
- `Velocity`: Initial velocity of the projectile.
- `PTheta`: Launch angle in degrees.
- `ConstG`: Gravitational constant.

#### Returns:

- The number of calculated trajectory points.

## Main Function

```
int main()
```

Sets up the TCP server, receives parameters from the client, computes the projectile's trajectory, and sends the results back to the client.

#### Steps:

1. Set up the server socket.
2. Bind the socket to the specified IP and port.
3. Listen for incoming connections.
4. Accept a client connection.
5. Receive the initial velocity, angle, and gravitational constant from the client.
6. Calculate the projectile trajectory using `CalculateProjectileTrajectory`.
7. Send the trajectory coordinates back to the client.
8. Close the client connection and continue listening for new connections.

## Usage

**1. Compile the program:**

```
gcc -o projectile_server server.c -lm
```

**2. Run the server:**

```
./projectile_server
```

**3. Client sends the parameters:**

- Initial velocity (double)
- Launch angle in degrees (double)
- Gravitational constant (double)

**4. Server computes the trajectory and returns the coordinates to the client.**

## Example Client Interaction

1. Client connects to the server at IP `127.0.0.1` and port `5566`.
2. Client sends the parameters `[v0, theta, g]` as a double array.
3. Server computes the trajectory and sends back the X and Y coordinates.
4. Client receives and processes the trajectory data.

## Notes

- The server calculates the trajectory in real-time, simulating the projectile motion with a time step (`dt`) of 0.12 seconds.
- The server uses `usleep` to simulate real-time updates.
- The program ensures that the trajectory calculation stops when the projectile hits the ground ( $y < 0$ ).