

PING PONG GAME USING AI

by

CHAITANYA KAMASANI 20BRS1037
SATYA NARAYANA REDDY 20BRS1263

A project report submitted to

Dr. Ummity Srinivasa Rao

SCHOOL OF COMPUTER SCIENCE ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2039 – FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE

in

**B. Tech. COMPUTER SCIENCE & ENGINEERING with
SPECIALIZATION in ARTIFICIAL INTELLIGENCE and ROBOTICS**



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

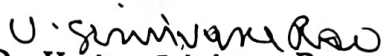
Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2022

BONAFIDE CERTIFICATE

Certified that this project report entitled "PING PONG GAME USING AI" is a bonafide work of –"CHAITANYA(20BRS1037), SATYA NARAYANA REDDY(20BRS1263)" who carried out the Project work under my supervision and guidance for CSE2039 – FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE


Dr. Umitty Srinivasa Rao

Assistant Professor (Senior Grade)

School of Computer Science and Engineering (SCOPE),

VIT University, Chennai

Chennai – 600 127.

ACKNOWLEDGEMENT

I wish to express my sincere thanks and deep sense of gratitude to my project guide, **Dr. Umitty Srinivasa Rao**, Assistant Professor (Senior Grade), School of Computer Science and Engineering, for his consistent encouragement and valuable guidance offered to me in a pleasant manner throughout the course of the project work.

I am extremely grateful to the Dean of School of Computer Science and Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

I express my thanks to the Head of the Department for her support throughout the course of this project.

I also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

I thank my parents, family, and friends for bearing with me throughout the course of our project and for the opportunity they provided me in undergoing this course in such a prestigious institution.

CHAITANYA

SATYA NARAYANA

ABSTRACT

My project name is “Ping Pong Game with AI Using Unity ” . It is a brain storming game to make our brain sharper and it's really enjoyable. In my project, game type is 2D. Here I used 2D type's tools and materials from unity game engine and also use game logic by using java scripts language. In my game I include one ball, 2 bat, midline side wall scouring timing. Hence i use different types of logics for each objects for run my game smoothly. My experimental results show that the games run properly without any intervention. This report provides detailed description on how can I created the ping pong game in unity game engine.

INTRODUCTION

Pong is one of the earliest arcade video games. It is a table-tennis inspired game featuring simple two dimensional graphics. In it, the player controls the paddle by moving it vertically across the left or the right side of the screen.

Each player earns points when the opponent fails to return the ball. The aim of this project is to create a C# application for Pong using the UNITY HUB SOFTWARE. It also involves creating a few AI models which the user can play against.

ADVANTAGES AND DISADVANTAGES

- Playing improves hand-eye coordination and it stimulates mental alertness, concentration and tactical strategy. ...
- Develops mental acuity.
- Improves reflexes.
- It's easy on the joints.
- Burns calories.
- Offers a social outlet.
- Keeps your brain sharp.
- Improves coordination.
- There are absolutely no cons of ping pong. Ping pong is a game of reflexes, agility and flexibility. It improves your reflexes by the most.

DESIGN OF THE GAME

The full project contains:

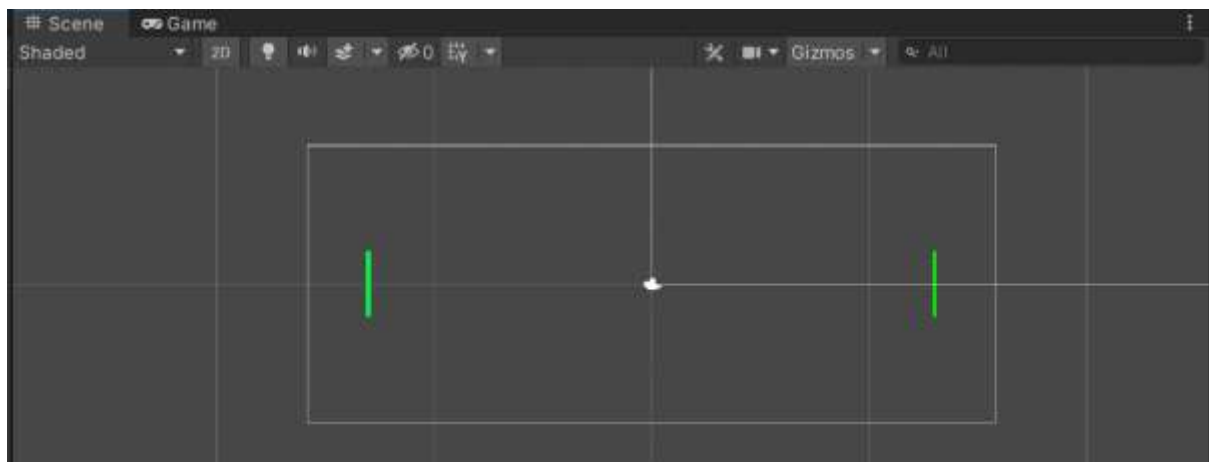
- ♣ A full game world which is played by the user.
- ♣ The ball can move anywhere with touching .
- ♣ Two player have two PADDLE but it is fixed by the midline .
- ♣ If the ball touches the opposite sides player's wall then it counts the point.
- ♣ Game score show the both player of their upper side of the screen.
- ♣ Score when the ball touches opposite side player's wall. ♣ Game over when the time is over.
- ♣ Pressing the replay button they can play again.

FEATURES OF THE GAME

The features of this game are:

- ♣ Ping pong game which is played by the user.
- ♣ Touching the ball both of the player's target of their opposite side.
- ♣ Game score show the both player of their upper side of the screen.
- ♣ The collision between ball and opposite side's wall.
- ♣ Scoring when ball touch the opposite sides wall.

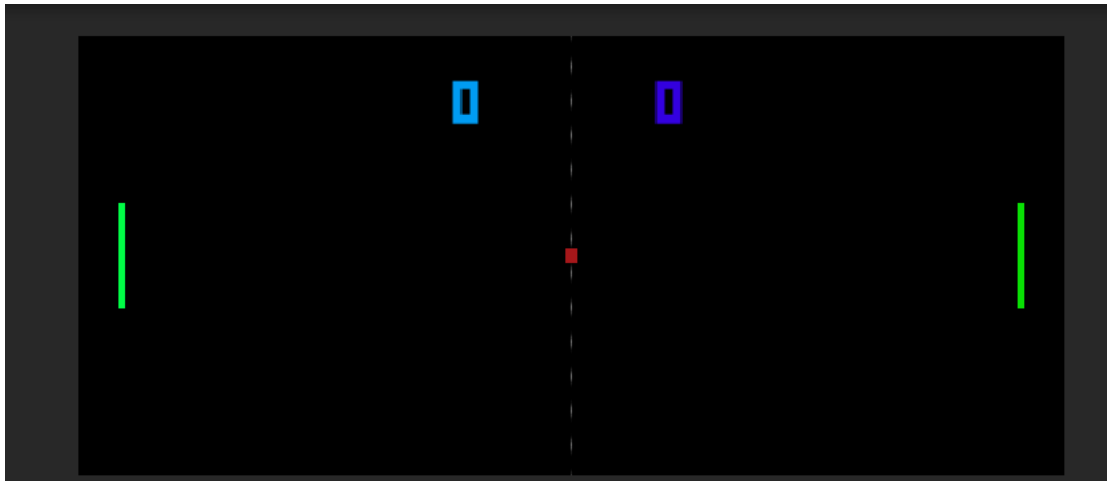
GAME SCENE



IMPLEMENTATION

GAME SCREEN

In this game screen user or player can easily sound off/on by touching the sound icon, then they can start the game pressing the start button.



BALL

The radius of the ball size is 0.37. Here i also use only blue colors ball for the game. In ball logic I also use 2D physics reject body which can push power and also use 2D bounce which can push the ball along with the axis.



- The ball has no acceleration due to no gravity.
- Vertical velocity is never Attraction to the downwards.
- Ball always movement anywhere by touching the wall along with the axis or player push the ball touching their bat on that axis.

PADDLE



The length and breadth of the paddle is 3 and 0.2. Here i also used green colour for the paddle. In paddle logic I also use 2D physics collider which can press power to the ball along with the axis.

WALLS



They make sure the ball doesn't go outside boundaries and make the ball bounce. The length and breadth of the wall are 30 and 1.

SCORING



In this game the score a point up or plus when the ball touch the opposite sides wall. Game score show the both player of their upper side of the screen. The collision between ball and opposite side's wall.

- The AI has a position where it wants to be, and moves towards that position at a constant velocity, using the standard “arrive” behavior (look into steering behaviors for AI - the trick with “arrive” is that you have an arrive radius, and scale your velocity according to it, so that your velocity is maximum at the radius and outside, but becomes zero in the middle).
- The desired position is calculated by casting a ray from the current ball position, along its velocity vector, and seeing if it intersects the line where the AI is guarding the goal. If it intersects a wall, it is reflected from there, and tested again. I only calculate one reflection, seeing how a human would also have problems predicting more than one.
- When the ball hits something, a delay is started which prevents the AI from immediately recalculating the desired position, for around 200 milliseconds. This simulates reaction time, and is based on actual data about human reaction times (around 215 ms on average).
- Later I added an “enrage” mechanic, where when the player scored a goal the AIs speed would be increased a bit and the reaction time reduced a bit. When the AI scored, the opposite would happen. This turned out to be a very good idea, since the AI would automatically scale its difficulty to any player skill level.

CODE SCRIPTS

BALL:

 Ball (Mono Script)

Assembly Information	
Filename	Assembly-CSharp.dll

```
using UnityEngine;

public class Ball : MonoBehaviour
{
    public float speed = 200.0f;
    private Rigidbody2D _rigidbody;
    private void Awake()
    {
        _rigidbody = GetComponent<Rigidbody2D>();
    }
    private void Start()
    {
        ResetPosition();
        AddStartingForce();
    }

    public void ResetPosition()
    {
        _rigidbody.position = Vector3.zero;
        _rigidbody.velocity = Vector2.zero;
    }

    public void AddStartingForce()
    {
        float x = Random.value < 0.5f ? -1.0f : 1.0f;
        float y = Random.value < 0.5f ? Random.Range(-1.0f, -0.5f) :
            Random.Range(0.5f, 1.0f);

        Vector2 direction = new Vector2(x,y);
        _rigidbody.AddForce(direction * this.speed);
    }
    public void AddForce(Vector2 force)
    {
        _rigidbody.AddForce(force);
    }
}
```

PADDLE:

Paddle (Mono Script)

Assembly Information	
Filename	Assembly-CSharp.dll

```
using UnityEngine;

public class Paddle : MonoBehaviour
{
    public float speed = 10.0f;
    protected Rigidbody2D _rigidbody;

    private void Awake()
    {
        _rigidbody = GetComponent<Rigidbody2D>();
    }

    public void ResetPosition()
    {
        _rigidbody.position = new Vector2(_rigidbody.position.x, 0.0f);
        _rigidbody.velocity = Vector2.zero;
    }
}
```

PLAYER PADDLE:

Player Paddle (Mono Script)

Assembly Information	
Filename	Assembly-CSharp.dll

```
using UnityEngine;

public class PlayerPaddle : Paddle
{
    private Vector2 _direction;

    private void Update()
    {
        if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.UpArrow)) {
            _direction = Vector2.up;
        }
        else if (Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow)) {
            _direction = Vector2.down;
        }
        else {
            _direction = Vector2.zero;
        }
    }

    private void FixedUpdate()
    {
        if (_direction.sqrMagnitude != 0) {
            _rigidbody.AddForce(_direction * this.speed);
        }
    }
}
```

COMPUTER PADDLE:

Computer Paddle (Mono Script)

Assembly Information	
Filename	Assembly-CSharp.dll

```
using UnityEngine;

public class ComputerPaddle : Paddle
{
    public Rigidbody2D ball;
    private void FixedUpdate()
    {
        if (this.ball.velocity.x > 0.0f)
        {
            if (this.ball.position.y > this.transform.position.y) {
                _rigidbody.AddForce(Vector2.up * this.speed);
            } else if (this.ball.position.y < this.transform.position.y) {
                _rigidbody.AddForce(Vector2.down * this.speed);
            }
        }
        else
        {
            if(this.transform.position.y > 0.0f) {
                _rigidbody.AddForce(Vector2.down * this.speed);
            } else if (this.transform.position.y <0.0f) {
                _rigidbody.AddForce(Vector2.up * this.speed);
            }
        }
    }
}
```

BOUNCY SURFACE:

Bouncy Surface (Mono Script)

Assembly Information	
Filename	Assembly-CSharp.dll

```
using UnityEngine;

public class BouncySurface : MonoBehaviour
{
    public float bounceStrength;
    private void OnCollisionEnter2D(Collision2D collision)
    {
        Ball ball = collision.gameObject.GetComponent<Ball>();

        if (ball != null)
        {
            Vector2 normal = collision.GetContact(0).normal;
            ball.AddForce(-normal * this.bounceStrength);
        }
    }
}
```

GAME MANAGER



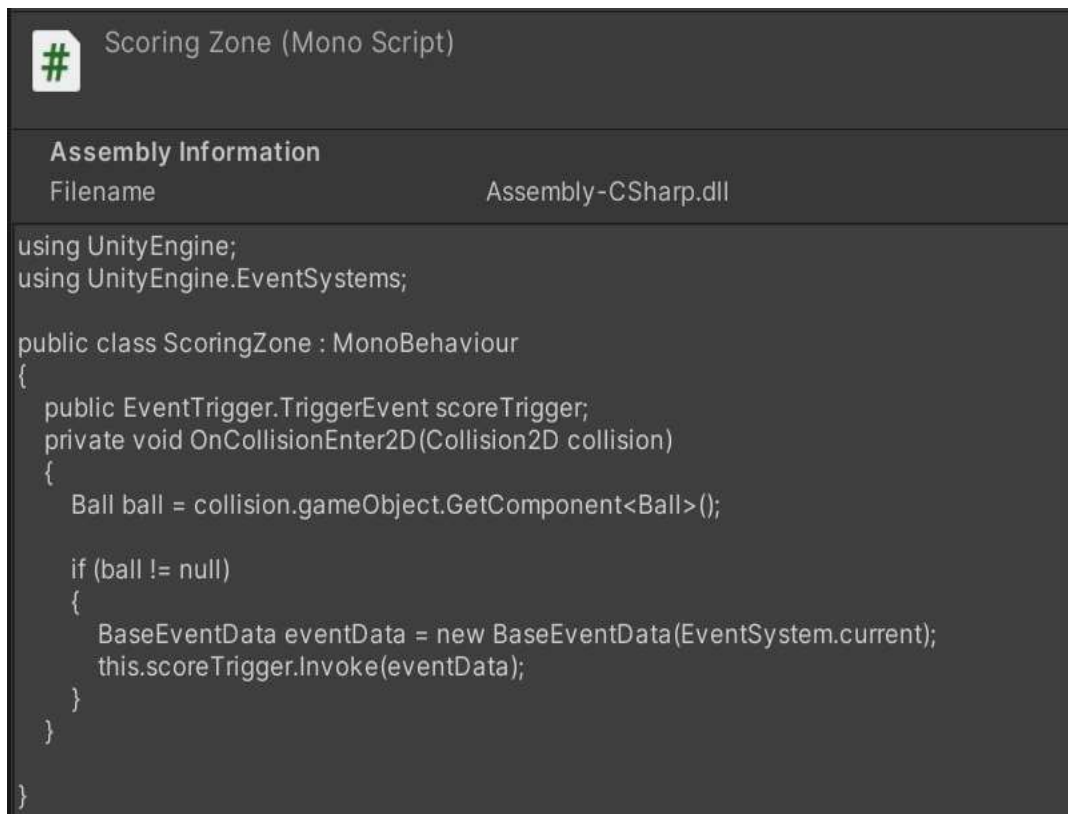
Game Manager (Mono Script)

```
using UnityEngine;
using UnityEngine.UI;

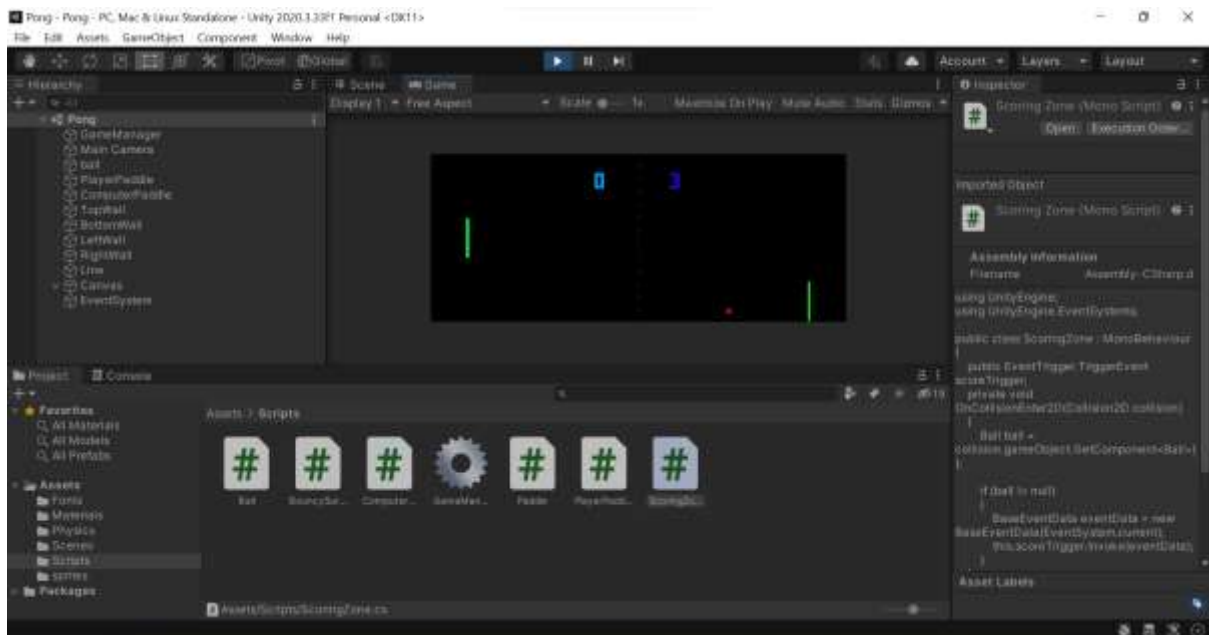
public class GameManager : MonoBehaviour
{
    public Ball ball;
    public Paddle PlayerPaddle;
    public Paddle computerPaddle;
    public Text playerScoreText;
    public Text computerScoreText;
    private int _playerScore;
    private int _computerScore;
    public void PlayerScores()
    {
        _playerScore++;
        this.playerScoreText.text = _playerScore.ToString();
        ResetRound();
    }
    public void ComputerScores()
    {
        _computerScore++;
        this.computerScoreText.text = _computerScore.ToString();
        ResetRound();
    }

    private void ResetRound()
    {
        this.PlayerPaddle.ResetPosition();
        this.computerPaddle.ResetPosition();
        this.ball.ResetPosition();
        this.ball.AddStartingForce();
    }
}
```

SCORING ZONE



OUTPUT SCREENSHOTS



CONCLUSION

Unity 2020 game software is the best game engine in windows and it is the most popular game engine. This project has successfully demonstrated the game. games are developed in Unity 2020 and use logics on C#Script, but Unity Game Engine are not a Complete C#Script implementation. Many of the packages that I have used for OpenGL and other graphics are included in the software development kit. Some. But most of the important 2D tools, physics and logics are available in Unity game engine for this game, because I have developed a 2-D game.. I think this game help me to develop an application and learn a lot about Unity game engine. I have made my game keeping some space for further development. This game will inspire new developers to create new games with different dimension.

FUTURE WORKS

- I want to add many types of attractive types of new balls for the new levels.
- Modify new game levels which was locked but every step open the next new game level.
- I want to add many levels in this game (Easy, Medium, and Hard).
- I want to make an iOS version of this application. Because besides android, iPhone is also widely used smart phone across the world