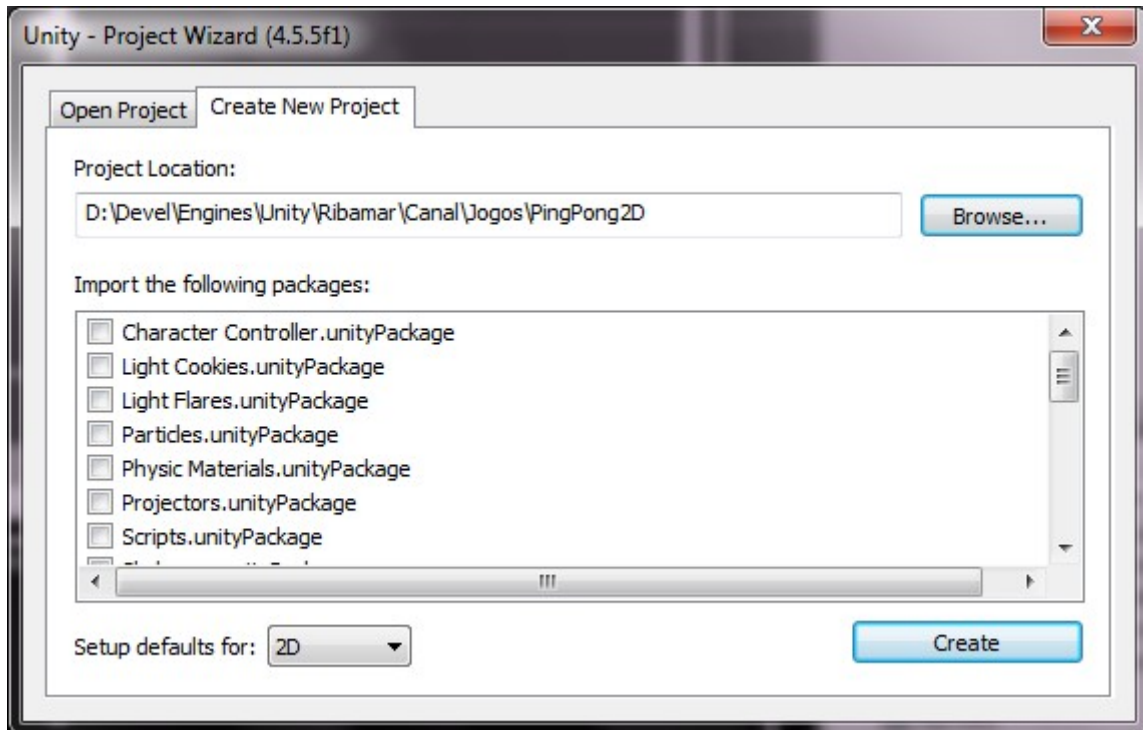


## Criando um Jogo tipo Pong no Unity 4 tipo 2D

### Criação do Jogo (criado na versão 4 mas facilmente adaptável para a versão 5)

Crie um novo projeto do tipo 2D com nome PingPong2D.

File – New Project. Escolha o diretório e nome para salvar e como do tipo 2D. Veja abaixo:



### Criação das Pastas

Criaremos as seguintes pastas:

Audio  
Cenas  
Materiais  
Scripts  
Sprites

Clique abaixo sobre a pasta **Assets** com o botão direito – Create – Folder

Entre com o nome Audio para a pasta.

Deixe a pasta Audio selecionada e tecele Ctrl+D e F2 e renomeie para Cenas.

Crie as demais pastas da mesma forma.

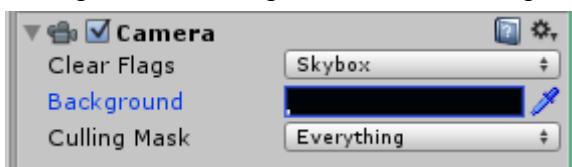
### Salvando a Cena Atual

Clique no painel da cena e tecele Ctrl+S então entre com o nome **Fase1** na pasta Cenas.

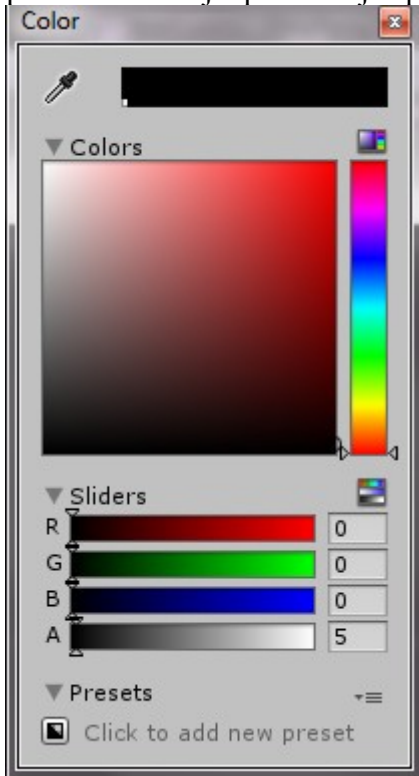
### Mudando a cor do Fundo

Selecione a Main Camera

No Inspector, no componente Camera clique na caixa Background



Na janela que abrir à esquerda clique no canto inferior direito do quadrado Colors para ter a cor preta acima. Veja que a cor já aparece na cena e poderá fechar a pequena janela:



Veja que o RGB ficou 000, que é preto.

## Criação das Quatro Paredes

### *Parede da esquerda*

GameObject – Create Other – Cube

Renomear para **ParedeEsq**

Com ele selecionado ajuste os valores do Inspector:

Position: X= -8, Y= 0 e Z=0

Scale: X= 0.1, Y= 9.5 e Z=1

### *Parede da direita*

Selecione na Hierarquia o objeto **ParedeEsq**

Tecle Ctrl+D para duplicar

Selecione o objeto duplicado e tecle F2 renomeie para **ParedeDir**

Mantenha **ParedeDir** selecionado

Agora altere no Inspector apenas Position X para 8

O restante mantenha

### *Parede superior*

Selecione na Hierarquia o objeto **ParedeDir**

Tecle Ctrl+D para duplicar

Selecione o objeto duplicado e tecle F2 renomeie para **ParedeSup**

Mantenha **ParedeSup** selecionado

Agora altere no Inspector apenas Position X para 0, Y para 4.7

Scale X=16, Y=0.1 e Z=1

### ***Parede inferior***

Selecione na Hierarquia o objeto **ParedeSup**

Tecla Ctrl+D para duplicar

Selecione o objeto duplicado e tecla F2 renomeie para **ParedeInf**

Mantenha **ParedeInf** selecionado

Agora altere no Inspector apenas Position Y para -4.7

O restante mantenha

### **Observação**

Caso estas medidas não se ajustem ao seu monitor, faça ajustes a vontade.

### **Adicionando a Luz**

GameObject – Create Other – Directional Light

Apenas mova um pouco para cima e esquerda para tirar do centro.

### **Adicionando as Raquetes**

#### ***Adicionando a Raquete maquina***

GameObject – Create Other – Cube

Selecione o cubo, tecla F2 e renomeie para **RaqueteMaquina**

Ainda com RaqueteMaquina selecionado altere no Inspector:

Position: X= -7.6, Y= 0 e Z=0

Scale: X= 0.25, Y=2.5 e Z=1

#### ***Adicionando a Raquete humano***

Selecione RaqueteMaquina na hierarquia

Tecla Ctrl+D e F2 para renomear para RaqueteHumano

Com RaqueteHumano selecionado altere no Inspector apenas X para 7.6

### **Adicionando a Bola**

GameObject – Create Other – Sphere

Selecione o cubo, tecla F2 e renomeie para **Bola**

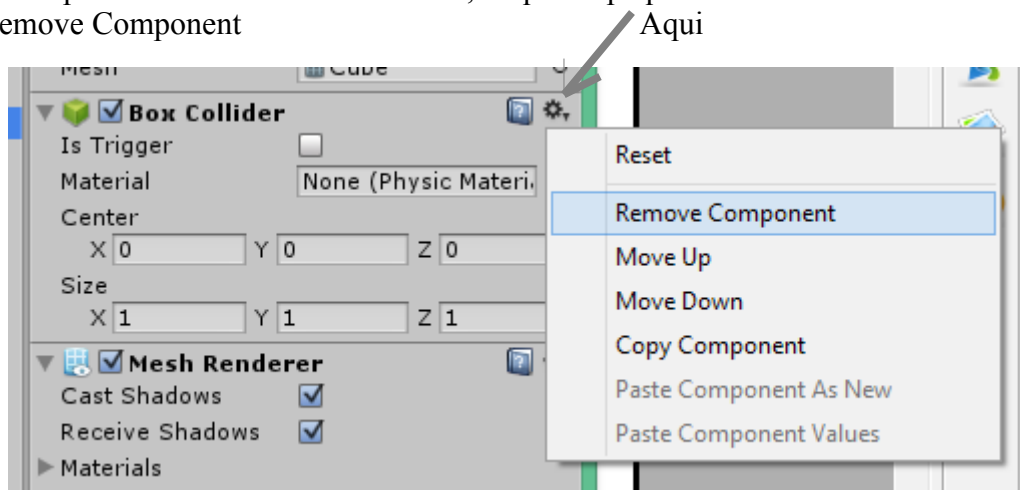
Ainda com bola selecionado altere no Inspector somente:

Scale: X= 0.5, Y=0.5 e Z=1

### **Removendo o Componente Box Collider**

Selecione as 4 paredes e as duas raquetes (segure a tecla Shift para selecionar todos)

No Inspector à direita de Box Collider, clique na pequena roldana acima e à direita e então em Remove Component



### **Adicionando Box Collider 2D**

Selecione as 4 paredes e as 2 raquetes

Clique em Inspector – Add Component – Physics 2D – Box Collider 2D

Então clique em Add Component – Physics 2D

### **Remover o Componente Sphere Collider**

Selecione a bola

Remova o componente Sphere Collider

### **Executando**

Ao executar (clique no botão Play acima) não acontecerá nenhum movimento.

### **Adicionando Circle Collider 2D e o Rigidbody 2D**

Selecione a bola

Clique em Inspector – Add Component – Physics 2D – Circle Collider 2D

Então clique em Add Component – Physics 2D

Adicione também o componente Rigidbody 2D

### **Executando**

Clique no botão Play para testar o jogo novamente. Veja que a bola cai (por conta da gravidade do rigidbody) e colidirá com a parede inferior (devido ao box collider) e para, fica quieta. Clique novamente no botão Play para voltar para a cena.

### **Adicionar Physics2D Material**

Para fazer a bolinha rebater nas paredes.

Clique sobre a pasta Materiais com o botão direito – Create - Physics2D Material

Renomeie para **BounceMaterial**

### **Configurando o material criado**

Selecione BounceMaterial

No Inspector mude Friction para 0 e Bounciness para 1.

### **Adicionando Material para a Bola**

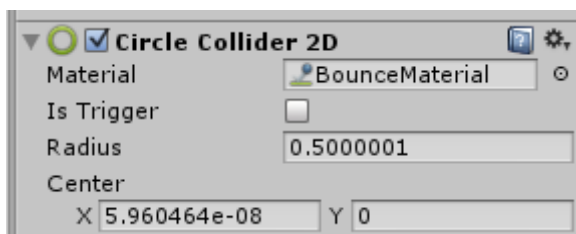
Mantenha a pasta Materiais aberta

Selecione a bola na hierarquia

No Inspector

Arraste o BounceMaterial para a caixa Material no componente Circle Collider 2D

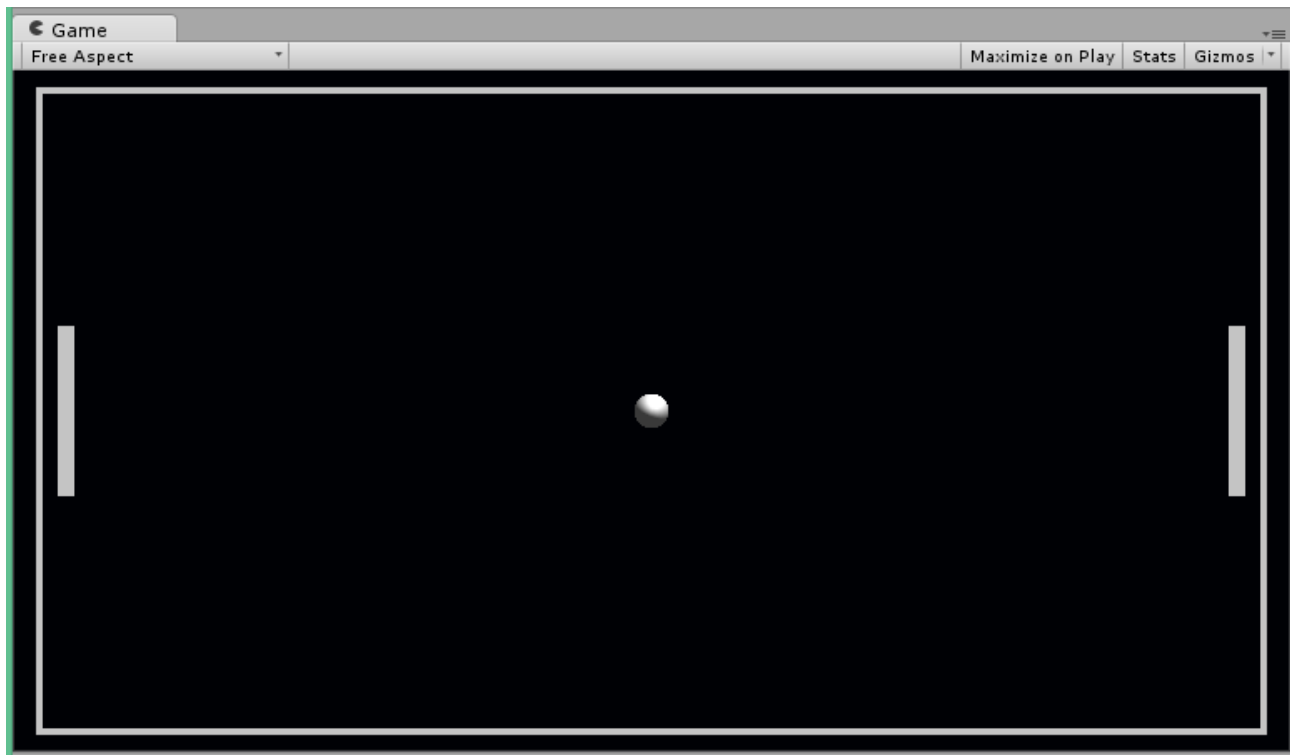
Focará assim:



### **Execute novamente**

Agora a bola cai e rebate no fundo e fica indeterminadamente assim, descendo e subindo.

A tela do jogo, nesta fase, ficará assim:



### Vida ao Jogo

Agora Vamos dar mais Vida e interatividade ao Jogo. Isso consegue-se especialmente com scripts.

#### **Criar um script para a bola**

Clique na pasta Script com o botão direito – Create – C# Script

Renomeie teclando F2 para **BolaScript**

Com o script bola selecionado tecle Enter para abrir o Editor Monodevelop

Entre este conteúdo no script (remova o conteúdo existente):

```
using UnityEngine;
using System.Collections;

public class BolaScript : MonoBehaviour {

    // Permite que a velocidade seja alterada pelo Inspector no Unity, por ser public
    public float velocidade = 15.0f;

    void Start() {
        rigidbody2D.velocity = Vector2.right * velocidade;
    }
}
```

#### **Anexando o script para o objeto bola**

Volte ao Unity

Selecione o Objeto bola na Hierarquia

No Inspector, abaixo clique em Add Component e selecione o script bolaScript.

Outra forma é arrastar o bolaScript e soltar sobre o objeto bola na Hierarquia.

Todo script que tem variável tipo public mostrará esta variável no Inspector, no Unity.

## **Criando o script para a raqueteHumScript**

Crie um novo script chamado raqueteHumScript tipo C#, contendo

```
using UnityEngine;
using System.Collections;

public class RaqueteHumScript : MonoBehaviour {

    public KeyCode teclaCima = KeyCode.UpArrow;
    public KeyCode teclaBaixo = KeyCode.DownArrow;

    void FixedUpdate() {
        Vector3 posicao = transform.position;

        if (Input.GetKey(teclaCima) && posicao.y < 3.4f) {
            transform.position = new Vector3(posicao.x, posicao.y + 0.2f, posicao.z);
        }
        else if (Input.GetKey(teclaBaixo) && posicao.y > -3.4f) {
            transform.position = new Vector3(posicao.x, posicao.y - 0.2f, posicao.z);
        }
    }
}
```

## **Anexando este script para o objeto RaqueteHumano**

Volte para o Unity

Anexe este script para o objeto RaqueteHumano, de forma semelhante a feita anteriormente. Como tem duas variáveis public para as teclas podemos alterar pelo Inspector.

## **Executando Novamente**

Agora a bola ficará rebatendo automaticamente nas paredes esquerda e direita, como também nas superior e inferior e você poderá mover a raquete da direita com as setas para cima e para baixo.

Observe a força da bola e se achar melhor aumente ou reduza. Selecione a bola na Hierarquia e altere o valor da propriedade Velocidade do script Bola Script.

## **Novos Recursos**

Nesta parte do tutorial iremos adicionar ao nosso jogo os recursos:

- Adicionar nova cena Menu
- Pontos
- Recorde
- Jogar contra o computador (apenas um jogador)
- Som na batida da raquete com a bola
- Menu inicial para Iniciar e Sair do jogo
- Agrupar objetos dentro de GameObject Emptys para melhor organização da Hierarquia
- Pequena mudança nos nomes de objetos e scripts para maior coerência
- Entre outras

## Adicionar a Cena Menu

Clique em File – New Scene (caso perguntado se deseja salvar as alterações, escolha Sim)  
Tecle Ctrl+S e acesse a pasta Cenas e dê o nome **Menu** para a nova cena

## Completando a cena

GameObject – Create Other – 3D Text e renomeie para **Pong**

Mude a propriedade Text no Inspector para **Tutorial de Ping Pong 2D** e

Position X = -5 e Y = 5

Font Size = 12

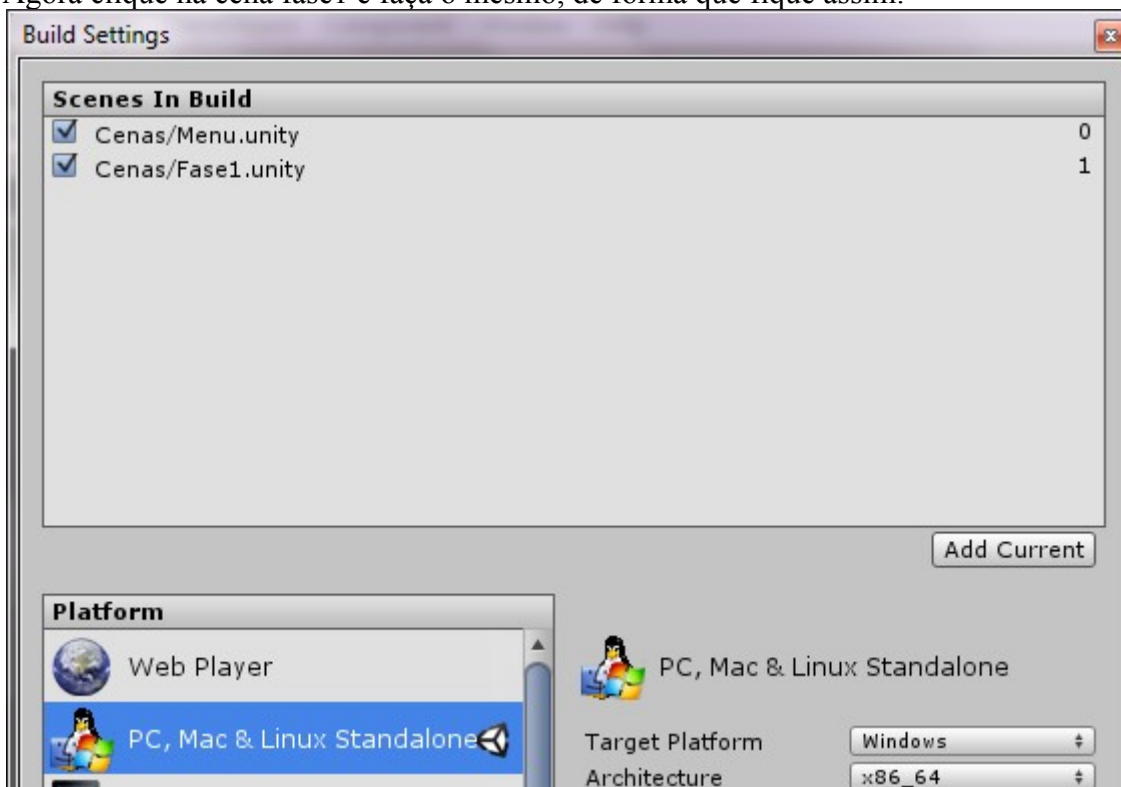
## Adicionando as Cenas ao Build Settings...

Abra a pasta Cenas

File – Build Settings...

Clique no nome da cena menu e arraste para o Build Settings e solte na região Scenes In Build

Agora clique na cena fase1 e faça o mesmo, de forma que fique assim:



Feche a janela do Build Settings.

## **Contando e Mostrando os Pontos dos Jogadores**

Vamos usar os objetos do tipo Gui Text para mostrar os pontos na tela.

Observe que chamei de RaqueteMaquina o jogador que usa a raquete esquerda, que é controlada pelo computador, enquanto que a raquete da direita chamei de RaqueteHumano.

Estou usando o tamanho default de fonte mas fique à vontade para alterar em Font Size.

### ***Criando PontosMaquina***

Abra a cena fase1 com duplo clique em Cenas/fase1

GameObject – CreateOther – GUI Text e renomear para PontosMaquina

Altere as propriedades do Inspector:

Text = Maquina: 0

Position X = 0.04, Y = 0.96 e Z = 0

### ***Criando PontosHumano***

GameObject – CreateOther – GUI Text e renomear para PontosHumano

Altere as propriedades do Inspector:

Text = Humano: 0

Position X = 0.8, Y = 0.96 e Z = 0

Obs.: também poderia duplicar PontosMaquina

### ***Criando Recorde***

GameObject – CreateOther – GUI Text e renomear para Recorde

Altere as propriedades do Inspector:

Text = Recorde: 0

Position X = 0.45, Y = 0.96 e Z = 0

## **Criar o MenuScript**

Crie um novo script chamado MenuScript do tipo C#, contendo:

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class MenuScript : MonoBehaviour {
```

```
    void Update() {
```

```
        // Teclando ESC sairá do Jogo, mas somente no executável e não dentro do editor
```

```
        if (Input.GetKey ("escape")) {
```

```
            Application.Quit();
```

```
        }
```

```
    }
```

```
    void OnGUI(){
```

```
        const int botaoLargura = 120;
```

```
        const int botaoAltura = 30;
```

```
        // Definindo as informações para os botões e no centro da tela
```

```
        Rect buttonIniciar = new Rect(Screen.width/2 - 50, Screen.height/2 +
```

```
10, botaoLargura, botaoAltura);
```

```
        Rect buttonSair = new Rect(Screen.width/2 - 50, Screen.height/2 + 50, botaoLargura,
```

```
        botaoAltura);
```

```
        // Desenha na tela o botão Iniciar
```



```

        if(GUI.Button(buttonIniciar,"Iniciar")){
            // Ao clicar carregar a primeira fase, Fase1
            Application.LoadLevel("Fase1");
        }

        // Desenha na tela o botao Sair
        if(GUI.Button(buttonSair,"Sair")){// Sair e o rotulo do botao
            // Quando clicado ele sai do jogo. Lembre que estando dentro do Unity nao
saira, apenas no EXE
            Application.Quit();
        }
    }
}

```

Observe que ele está bem comentado, especialmente quando pode haver dúvida.

### **Anexar Este Script à Main Camera da cena menu**

Selecione a cena menu abaixo nos Assets e efetue um duplo clique nela

Arraste o script MenuScript e solte sobre a Main Camera da cena menu

Como é um jogo simples deixaremos na Main Camera. Para um jogo maior criaríamos um objeto exclusivamente para ele e soltaríamos na Main Camera, pois esta está disponível todo o tempo

### **Criando PontosScript**

Criar novo script chamado PontosScript, do tipo C# e contendo:

```

using UnityEngine;
using System.Collections;

// Este script foi anexado a um objeto GameObject Empty arrastado para dentro da Main Camera
// Este script precisa ficar anexado a um objeto que esteja o tempo todo valendo, como a main
camera ou o background
public class PontosScript : MonoBehaviour {

    // Da hierarquia arrastar pontosMaquina para a variavel publica PontosGuiMaq de
Pontuacao em Main Camera
    public GUIText pontosGuiMaq;
    // Da hierarquia arrastar pontosHumano para a variavel publica PontosGuiHum de
Pontuacao em Main Camera
    public GUIText pontosGuiHum;
    // Da hierarquia arrastar recorde para a variavel publica recordeGui de Pontuacao em Main
Camera
    public GUIText recordeGui;
    public int pontosHum = 0;// Dar acesso ao script nave, por isso public
    public int pontosMaq = 0;// Dar acesso ao script nave, por isso public

    void Update () {
        // Se os pontos do humano da partida atual for maior que o recorde armazenado em
PlayerPrefs
        if(pontosHum > PlayerPrefs.GetInt("Recorde")){
            // Guarde em PlayerPrefs os pontosHum
            PlayerPrefs.SetInt("Recorde", pontosHum);
            // Se os pontos da maquina da partida atual for maior que o recorde armazenado em

```

PlayerPrefs

```
} else if(pontosMaq > PlayerPrefs.GetInt("Recorde")){
    // Guarde em PlayerPrefs os pontosMaq
    PlayerPrefs.SetInt("Recorde", pontosMaq);
}

//pontosGui.fontStyle = FontStyle.Bold;
//pontosGui.fontSize = 18;
if(pontosMaq < 20){
    // Mostrar na tela Maquina mais pontosMaq
    pontosGuiMaq.text = "Maquina " + pontosMaq;
} else {
    // Caso seja 20 mostre na tela Maquina Venceu: mais pontosMaq;
    pontosGuiMaq.text = "Maquina Venceu: " + pontosMaq;
}
if(pontosHum < 20){
    pontosGuiHum.text = "Humano " + pontosHum;
} else {
    pontosGuiHum.text = "Humano Venceu: " + pontosHum;
    Application.Quit();
}
if (pontosHum > pontosMaq) {
    recordeGui.text = "Recorde Humano: " + PlayerPrefs.GetInt ("Recorde");
} else {
    recordeGui.text = "Recorde Maquina: " + PlayerPrefs.GetInt("Recorde");
    Application.Quit();
}

// Para o caso de querer zerar manualmente o recorde
if(Input.GetKeyDown(KeyCode.R)) PlayerPrefs.SetInt("Recorde", 0);
}
}
```

Também está comentado para melhor compreensão, por quem ainda não está familiarizado com o C# e Unity.

### **Criar objeto Pontuacao**

Criar objeto que será anexado ao PontosScript e que ficará sob a Main Camera da cena fase1.

GameObject – Create Empty e renomeie para **Pontuacao**

Arraste o objeto Pontuacao e solte sobre a Main Camera

### **Anexar este Script PontosScript ao objeto Pontuacao**

Na cena Fase1

Expand a Main Camera para ver **Pontuacao**

Arraste este script PontosScript e solte sobre o objeto Pontuacao

## Atribuindo Valor para as Variáveis tipo public do Script em Pontuacao

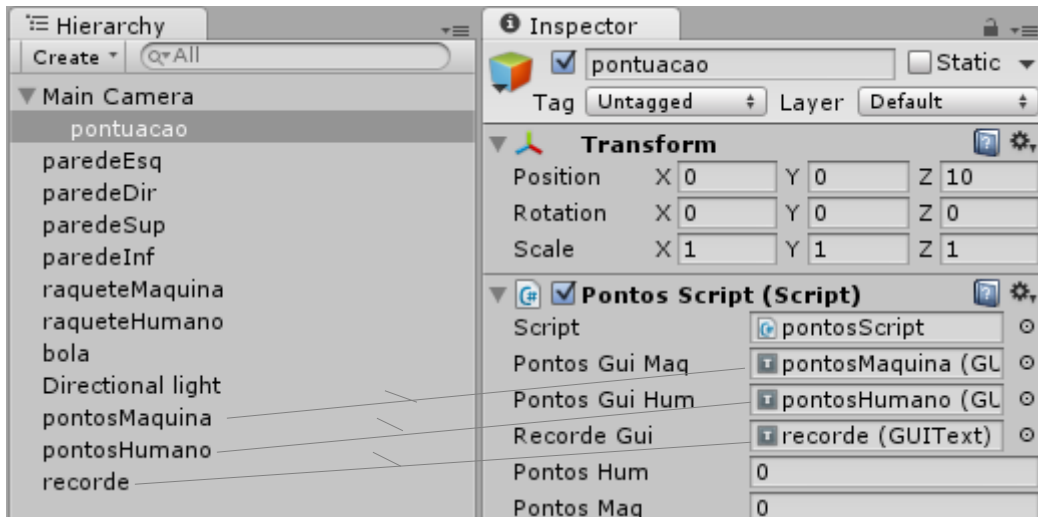
Expanda Pontos na Hierarquia

Selecione Pontuacao para que veja as variáveis public no Inspector

Arraste da esquerda e solte sobre a respectiva caixa na direita, como indicado abaixo.

Arraste PontosMaquina e solte na caixa de texto à direita Pontos Gui Maq

Faça o mesmo com PontosHumano e com recorde, para que fique assim:



## Criar Agrupamentos para Objetos

Para melhor organizar a hierarquia, vamos criar GameObjects Empty e mover alguns objetos para eles.

Vamos criar os objetos e agrupar para melhorar a organização:

GameObject – Create Empty e renomear para Paredes

Selecione os 4 objetos paredes, basta clicar no primeiro, segurar o Shift e clicar no último, ou arrastar de um em um

Arraste os 4 objetos parede e solte sobre o novo objeto Paredes

Agora crie um para as raquetes

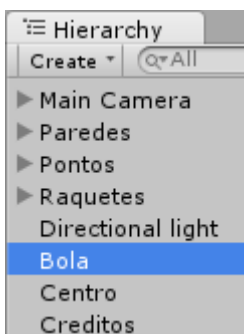
GameObject – Create Empty e renomear para Raquetes

Arraste as raquetes e solte no objeto Raquetes

Agora crie mais um GameObject Empty chamado Pontos

Arraste os objetos PontosMaquina, PontosHumano e recorde para Pontos

Veja como ficou com agrupamentos:



### Adicionar Sprite no Centro

Abra a pasta Sprites

Baixe este arquivo e renomeie para Centro.png

<http://noobtuts.com/content/unity/2d-pong-game/Middle.png>

Importe o script Centro.png para a pasta Sprites

Arraste o sprite Centro e solte na hierarquia

Position (0,0,0) e Scale (1, 2, 1)

### Edite o script BolaScript

E deixe assim:

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class BolaScript : MonoBehaviour {
```

```
    private PontosScript ptScript; // Para se comunicar com o script PontosScript
```

```
    public float velocidade = 15.0f;
```

```
    void Start() {
```

```
        // Sempre que o jogo começar a bolinha sera impulsionada para a direita e com a  
força 15
```

```
        rigidbody2D.velocity = Vector2.right * velocidade;
```

```
        //Procure o script pontosScript anexado ao objeto Pontuacao
```

```
        ptScript = GameObject.Find ("Pontuacao").GetComponent<PontosScript> ();
```

```
    }
```

```
    // Quando ocorrer colisao
```

```
    void OnCollisionEnter2D (Collision2D outro){
```

```
        // Da bola com a ParedeEsq
```

```
        if(outro.gameObject.name == "ParedeEsq"){
```

```
            // Some um ponto para o humano
```

```
            ptScript.pontosHum ++;
```

```
        // Colisao da bola com a ParedeDir
```

```
        }else if(outro.gameObject.name == "ParedeDir"){
```

```
            // Ponto para a maquina
```

```
            ptScript.pontosMaq ++;
```

```
        }
```

```
    }
```

```
    void OnGUI() {
```

```
        // Sempre que maquina ou humano facam 20 pontos
```

```
        if (ptScript.pontosMaq == 20 || ptScript.pontosHum == 20) {
```

```
            // Pare a bola
```

```
            rigidbody2D.velocity = Vector2.right * 0.0f;
```

```
            rigidbody2D.gravityScale = 0.0f;
```

```
            // Crie um botao com rotulo Menu
```

```
            if (GUI.Button(new Rect(220, 50, 50, 30), "Menu"))
```

```
                // Quando clicado abra o menu
```

```

        Application.LoadLevel("Menu");
    }
}

```

### Adicionar um arquivo de Áudio

Criar um arquivo de áudio para a colisão da bola com as raquetes.

Usaremos a ferramenta online abaixo para criar o arquivo:

<http://www.superflashbros.net/as3sfxr/>

Clique em PICKUP/COIN e em EXPORT .WAV

Salve com o nome Pong.wav

Abra a pasta Audio e clique com o botão direito – Inport New Asset e importe o pong.wav

O Unity também pode usar áudio pela interface mas usaremos o arquivo via script..

### Crie o script RaqueteMaqScript

Adicione o conteúdo:

```

using UnityEngine;
using System.Collections;

```

// Uma dica: para receber help sobre qualquer comando, apenas clique sobre o nome do comando e tecle Ctrl+'

```

public class RaqueteMaqScript : MonoBehaviour {

    private float velocidade = 5.0f;
    private float altura = 3.5f;
    private PontosScript ptScript; // Para se comunicar com o script PontosScript
    public AudioClip bolaColisao; // Arrastar o audio laser para esta variavel

    void Start(){
        // Criando uma referencia para o script PontosScript
        ptScript = GameObject.Find ("Pontuacao").GetComponent<PontosScript> ();
    }

    void Update(){
        // Criando a variavel posicao do tipo Vector3 que armazena transform.position
        Vector3 posicao = transform.position;
        float posicaoY = Mathf.Sin(Time.time * velocidade) * altura;
        posicaoY += altura/2 - 1.3f;
        // Atribuindo a transform.position o Vector3
        transform.position = new Vector3(posicao.x, posicaoY, posicao.z);

        // Se os pontos da maquina ou do humano somarem 20
        if (ptScript.pontosMaq == 20 || ptScript.pontosHum == 20) {
            //pare a raquete da maquina
            transform.position = new Vector3(posicao.x, posicao.y, posicao.z);
        }
    }

    // Em caso de colisao da raqueteMaquina com a bola
    void OnCollisionEnter2D (Collision2D outro){

```

```

        if(outro.gameObject.name == "Bola"){
            // Toque o som pong que foi anexado ao script atual
            AudioSource.PlayClipAtPoint(bolaColisao, transform.position);
        }
    }
}

```

### **Anexar ao objeto RaqueteMaquina**

Da mesma forma feita anteriormente

### **Atribuir o audio Pong ao campo BolaColisao**

Selecionar o objeto RaqueteMaquina na Hierarquia

Arraste o audio Pong e solte sobre o campo Bola Colisao no Inspector

### **Editar o script RaqueteHumScript**

E deixar assim:

```

using UnityEngine;
using System.Collections;

public class RaqueteHumScript : MonoBehaviour {

    // Criar variavel publica que recebera teclas
    public KeyCode teclaCima = KeyCode.UpArrow;
    public KeyCode teclaBaixo = KeyCode.DownArrow;
    // Criar variavel que recebera um arquivo de audio
    public AudioClip bolaColisao; // Arrastar o audio pong para esta variavel no Inspector ao
    selecionar raqueteHumScript

    void FixedUpdate() {
        // Criando a variavel posicao do tipo Vector3 que armazena transform.position
        Vector3 posicao = transform.position;

        // Caso a teclaCima seja pressionada e se o y de posicao for menor que 3.4
        if (Input.GetKey(teclaCima) && posicao.y < 3.4f) {
            // Faca a raquete se mover para cima
            transform.position = new Vector3(posicao.x, posicao.y + 0.2f, posicao.z);
        }
        // Caso a teclaBaixo seja pressionada e se o y de posicao for maior que -3.4
        }else if(Input.GetKey(teclaBaixo) && posicao.y > -3.4f){
            // Faca a raquete se mover para baixo
            transform.position = new Vector3(posicao.x, posicao.y - 0.2f, posicao.z);
        }
    }

    // Quando houver colisao
    void OnCollisionEnter2D (Collision2D outro){
        // Da raqueteMunano com a Bola
        if(outro.gameObject.name == "Bola"){
            // Tocar o som do arquivo pong
            AudioSource.PlayClipAtPoint(bolaColisao, transform.position);
        }
    }
}

```

```
}  
    }  
}
```

### Variável BolaColisão

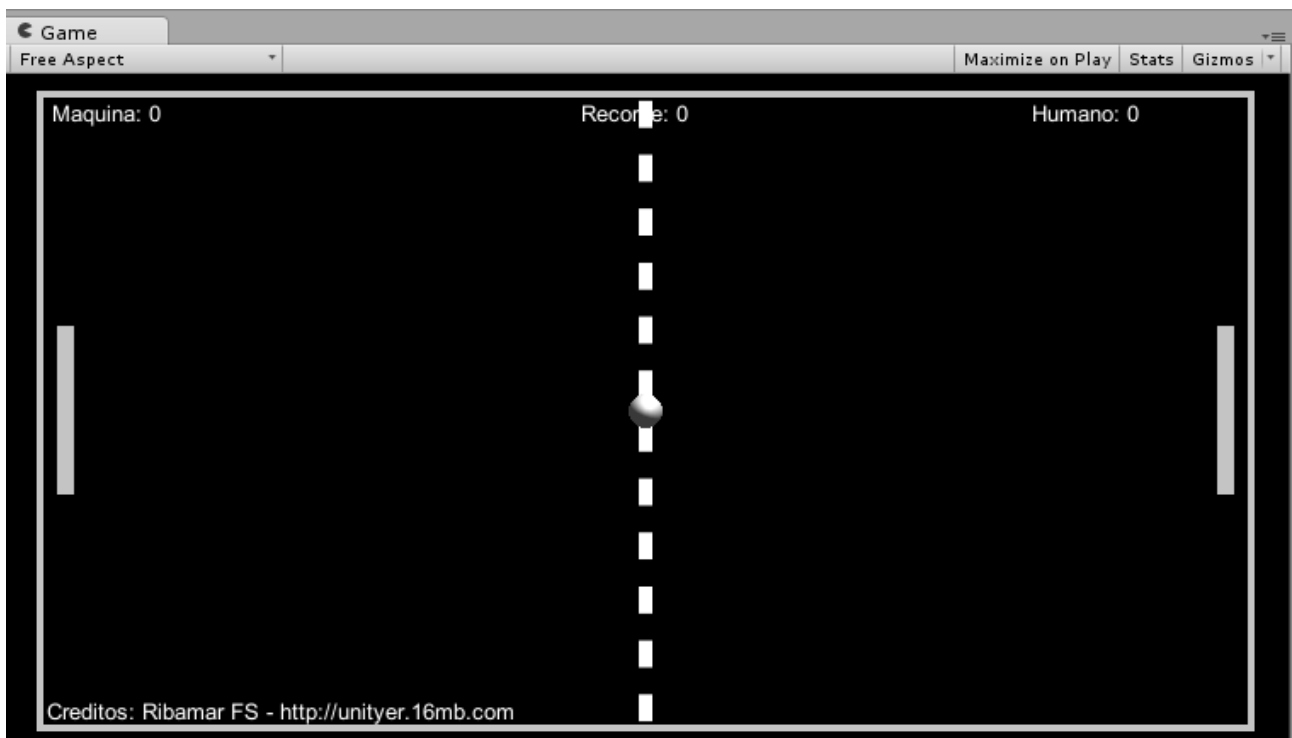
Selecionar o objeto RaqueteHumano na Hierarquia

Atribuir o audio Pong para o campo BolaColisao no Inspector

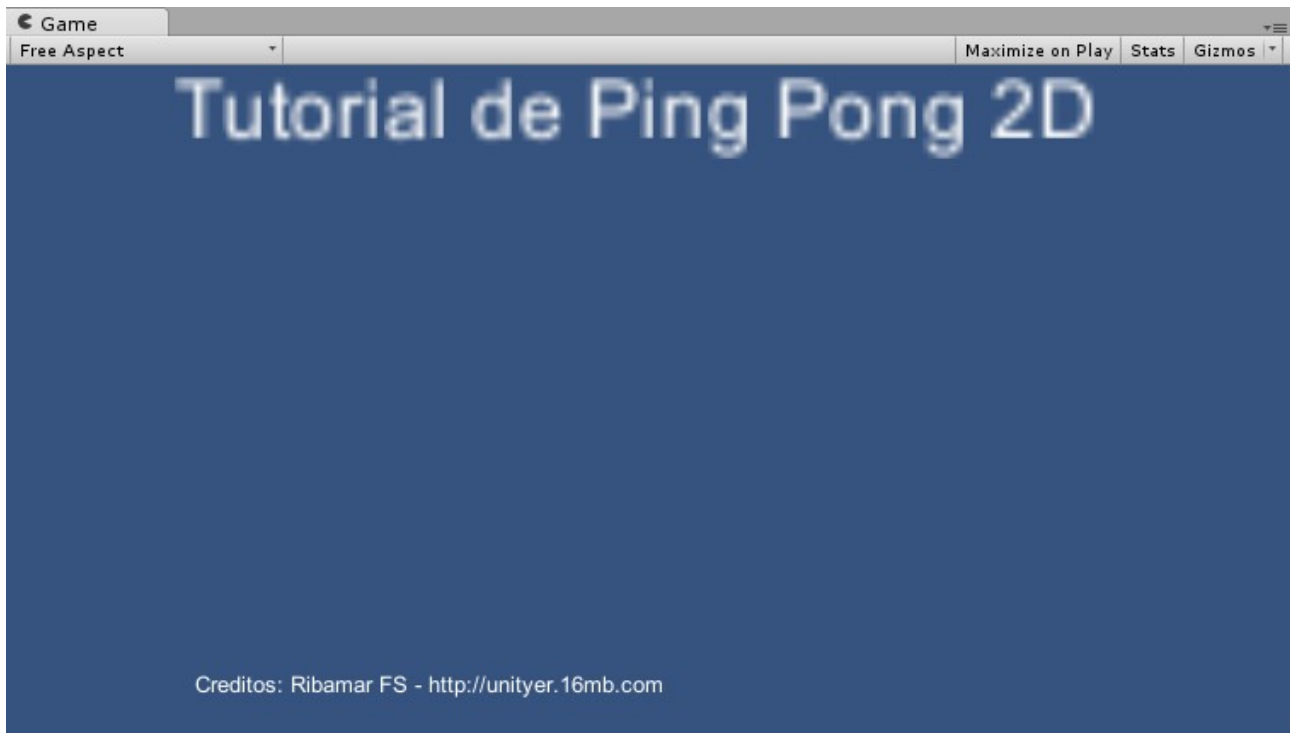
### Execute novamente

Agora verá o jogo completo, mova as setas para cima e para baixo para rebater a bola que será enviada de volta pelo computador.

### Veja como ficará a tela final do nosso jogo



## Aqui a tela do menu



## Gerando o EXE do jogo

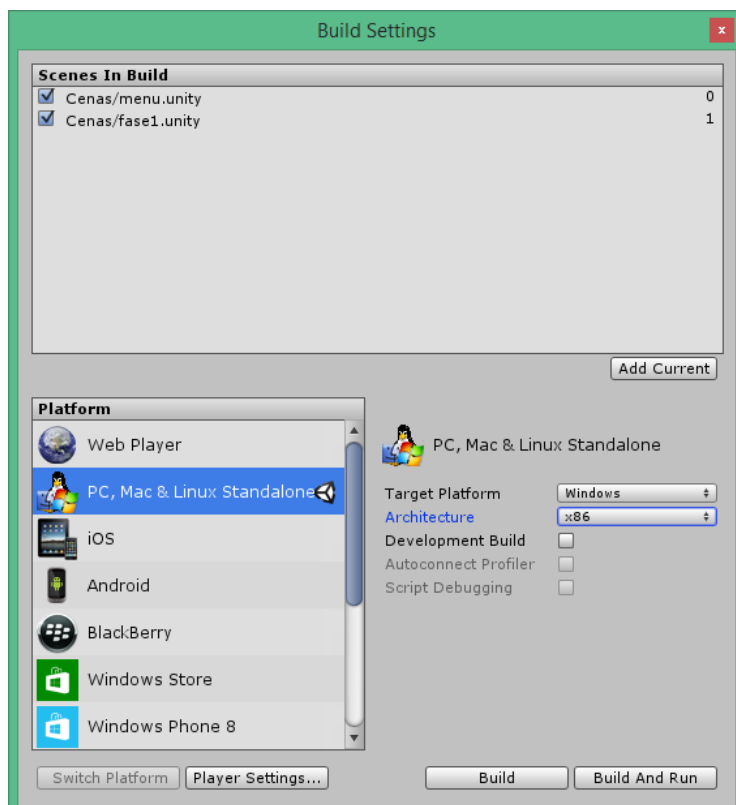
File – Build Settings

Clique em Platform - PC, Mac & Linux Standalone

À direita deixe selecionados Windows e X86.

Clique abaixo em Build e entre o nome PingPong e aguarde.

Quando ele terminar de gerar o EXE mostrará no Windows Explorer e já poderá executar.





**Referências:**

<http://unity3d.com>

<http://www.supercoderbrothers.com/unity-tutorial-our-first-game-pong-part-1/>

<http://www.supercoderbrothers.com/unity-tutorial-pong-part-2-adding-interaction/>

<https://www.youtube.com/watch?v=tnKcrboiG6Y>

[https://www.youtube.com/watch?v=DYLzBb\\_y0YA](https://www.youtube.com/watch?v=DYLzBb_y0YA)

<https://www.youtube.com/watch?v=Tb738KE4Xqc>

<http://noobtuts.com/unity/2d-pong-game>

<http://unityer.16mb.com>

**Mudando o Jogo para Dois Jogadores**

Para ter um jogo para dois jogadores o que precisamos mudar?

Remover o MaqHumScript ou apenas remover o componente do script no objeto raqueteMaruina.

Depois disso anexar o scriptHumScript ao objeto raqueteMaquina e mudas as teclas de movimento para usar W e S ou outra do seu gosto.