

# Movimentação Básica

Flávio Roberto Dias Silva

Encontro 2  
**Desenvolvimento de Jogos Eletrônicos com Unity 3D.**

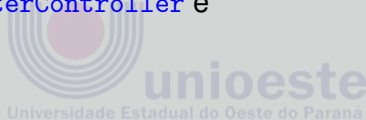
Universidade Estadual do Oeste do Paraná



## Introdução

**Introdução:** Até o momento trabalhamos com movimentação baseada em `transform`, mas esse tipo de movimentação não leva em consideração a física do ambiente 3D, isso é, não leva em consideração colisões com outros elementos.

Para refinar a movimentação dos nossos personagens em Unity devemos utilizar Componentes que levam em consideração a física, são eles `CharacterController` e `Rigidbody`.



## Movimentação Baseada em Rigidbody Vs CharacterController

**Rigidbody:** A tradução de Rigidbody é corpo rígido, esse componente é utilizado para simular uma física o mais realista possível dentro das capacidades da Unity.

Ele é utilizado para que os elementos do jogo se movimentem com física realista baseada em forças aplicadas.

Para utilizar corretamente as funcionalidades do componente Rigidbody é necessário adicionar também um componente do tipo **Collider**, que pode ser **CapsuleCollider**, **BoxCollider**, **SphereCollider** etc...

No jogo **Angry Birds** tanto os personagens quanto os elementos do cenário tem movimentações que podem ser associadas ao Rigidbody.



**CharacterController:** A tradução é controlador de Caractere(Personagem), esse componente tem suporte a movimentação e detecção de colisões mas com uma física menos realista mais voltada a personagens padrão de jogos de ação e aventura.

Esse componente é utilizado para fazer movimentações padrões estilo Skyrim, Dark Souls, Zelda 3D, etc...

O Controlador de Caracter já vem incorporado com um **Collider** do tipo **CapsuleCollider**, esse tipo de **Collider** é indicado para personagens humanoides, mas se torna bem incomodo quando seu personagem é um animal quadrupede, serpente ou outro personagem que não se adapte

A maioria dos jogos tem um tipo de movimentação que pode ser associada a um CharacterController.



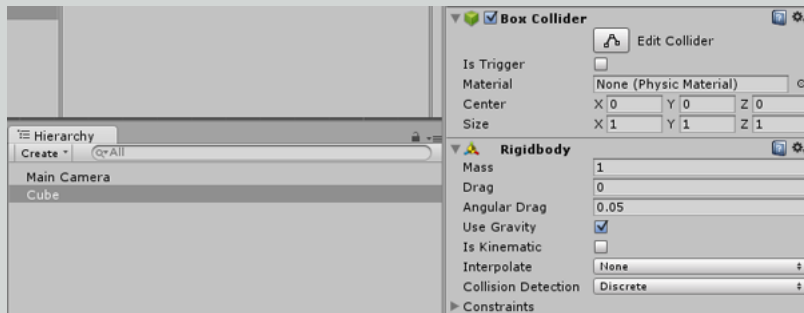
## Movimentação baseada em Rigidbody

Para criar uma movimentação baseada em Rigidbody precisamos adicionar o componente Rigidbody ao objeto 3D que irá se movimentar.

Para adicionar o componente você pode clicar no nome do objeto que consta na aba Hierarchy ou no objeto na aba scene para seleciona-lo e logo depois clicar no add componente da aba Inspector.

Outra maneira de adicionar o Rigidbody é: com o objeto selecionado utilizar o menu suspenso Component -> Physics -> Rigidbody.

**Observação** O objeto que carregará o Rigidbody precisa de um Collider, caso contrario, ao iniciar o jogo o objeto atravessará o chão pois o Rigidbody já é influenciado pela gravidade.





## Movimentação baseada em Rigidbody

Para fazer a movimentação do nosso objeto 3D é recomendado que o script da movimentação esteja anexado a ele.

**Exercício** Crie o script da movimentação com rigidbody. Nesse script será necessária uma variável do tipo `Rigidbody`.



## Observações sobre Orientação a objetos e

**MonoBehaviour Scripts:** Em programação Orientada a objetos as classes utilizam as palavras **public** e **private** para declarar a acessibilidade de métodos, propriedades e campos de uma classe para outra.

Em Unity 3D a palavra **public** de uma classe herdeira de MonoBehaviour tem uma função extra, uma variável pública aparece para edição na aba Inspector do editor do Unity. Isso possibilita a edição do valor de uma variável no editor sem a necessidade de mexer no script.

## Movimentação baseada em RigidBody

Ao lado temos um script com algumas variáveis declaradas como publicas outras como privadas e algumas sem nenhuma palavra reservada a antecedendo.

Abaixo da imagem do script está a imagem do Inspector desse script adicionado a um objeto como componente.

Observe que as variáveis declaradas como publicas aparecem no Inspector as demais não.

```
using UnityEngine;
using System.Collections;

public class PrimeiroScript : MonoBehaviour {

    public int velocidade = 10;
    public int velInicial;
    public string nome = "meu nome";

    public GameObject umObjeto;
    public Rigidbody R;

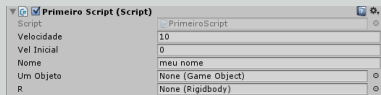
    private string info = "variaveis privadas nao aparecem no inspector";
    string info2 = "Se nenhuma palavra for usada a variavel é privada";
    Transform T;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        transform.position += new Vector3(1, 0, 0) * 0.01f;
    }

}
```



## Movimentação baseada em Rigidbody

A variável do tipo **Rigidbody** tem alguns métodos dedicados a movimentação que leva em consideração a física do ambiente 3D. Entre elas as mais utilizadas são:

```
Rigidbody R = GetComponent<Rigidbody>();  
R.AddForce(variavelDoTipoVector3);  
R.AddTorque(variavelDoTipoVector3);
```

**AddForce** adiciona uma força no sentido do vetor de 3 dimensões passado como parâmetro sendo responsável pelo deslocamento do objeto 3D nessa direção.

**AddTorque** adiciona um torque ao objeto 3D que é responsável pela rotação do objeto, o parâmetro da função é um vetor de 3 dimensões que é simultaneamente ortogonal ao vetor velocidade do movimento e ao vetor aceleração do movimento.

**Exercício:** Crie para seu objeto 3D uma movimentação baseada em Rigidbody utilizando as teclas AWSDF ou as setas do teclado. Utilize para a velocidade de movimentação uma variável publica que pode ser editada no Inspector.



## Movimentação baseada em RigidBody

Talvez nesse momento estejamos tendo dificuldades com a velocidade máxima do **RigidBody** que faz com que o controle do personagem seja perdido após alguns segundos segurando o botão de se deslocar.

Para evitar uma velocidade exagerada podemos inserir uma velocidade máxima para o **RigidBody** utilizando a propriedade `velocity`.



## Movimentação baseada em Rigidbody

Da forma:

```
Rigidbody R = GetComponent<Rigidbody>();  
if(R.velocity.magnitude < velMax)  
{  
    R.AddForce(velocidadeInserida);  
}
```

## Movimentação baseada em Rigidbody

- [1] Barnes, D. J., Kölling, M.(2009), Programação Orientada a Objetos com Java. Uma introdução prática usando BLUEJ, 4ªed., Pearson Prentice Hall.
- [2] Battaiola, A. L. (2000). Jogos por Computador ? Histórico, Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação In: XIX Jornada de Atualização em Informática. Curitiba: SBC, Julho/2000, v. 2. pp. 83 - 122
- [3] Battaiola, A. L.; Elias, N. C.; Domingues, R.G. et al (2002). Desenvolvimento de um Software Educacional com Base em Conceitos de Jogos de Computador In: XIII Simpósio Brasileiro de Informática na Educação. São Leopoldo: SBC, 2002, pp. 282-290.



## Movimentação baseada em RigidBody

- [4] Crua, E. W. G.; Bittencourte, J. R.(2005) Desenvolvimento de Jogos 3D: Concepção, Design e Programação. Anais da XXIV Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, pp.1313-1356, São Leopoldo, Brasil, Julho de 2005.
- Deitel, H. M., Deitel, P. J.(2010), Java: Como programar, 8ªed., Pearson Prentice Hall. Rio de Janeiro IMPA.
- [5] Freeman, E., Freeman, E.(2007), Use a Cabeça Padrões de Projetos, 2ªed., Rio de Janeiro Altabooks.
- [6] Sintes, A.(2002), Aprenda Programação Orientada a Objetos em 21 dias, São Paulo Makron Books.
- [7] Stellman, A.; Greene, J.(2011), Use a Cabeça! C#, Rio de Janeiro, AltaBooks.

## Movimentação baseada em RigidBody

- [8] Unity Technologies (2016)(A). Unity 3D User Manual [online]. Disponível em: [\[http://docs.unity3d.com/Manual/index.html\]](http://docs.unity3d.com/Manual/index.html) [[Acesso em 18/04/2016]
- [9] Unity Technologies (2016)(B). Unity 3D Community Forum [online]. Disponível em: [\[http://forum.unity3d.com/\]](http://forum.unity3d.com/) [[Acesso em 18/04/2016]
- [10] Unity Technologies (2016)(C). Unity 3D Online Tutorials [online]. Disponível em: [\[https://unity3d.com/pt/learn/tutorials\]](https://unity3d.com/pt/learn/tutorials) [[Acesso em 18/04/2016]
- [11] Unity Technologies (2016)(D). Unity 3D Community Wiki [online]. Disponível em:

## Movimentação baseada em RigidBody

[[http://wiki.unity3d.com/index.php/Main\\_Page](http://wiki.unity3d.com/index.php/Main_Page) ][Acesso em 18/04/2016]

