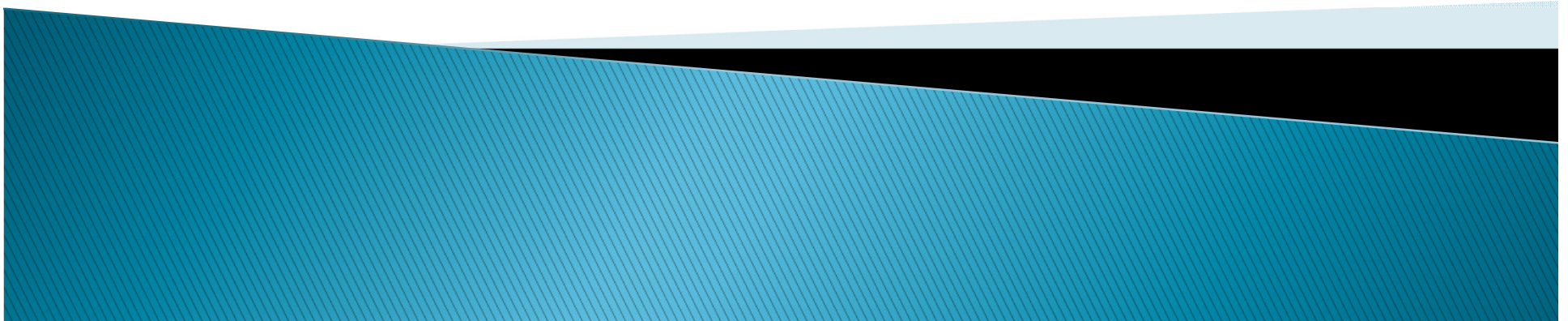


# Unity Scripting II

Esteban Walter Gonzalez Clua  
Instituto de Computação – UFF  
esteban@ic.uff.br



# Printing in Console mode

```
Debug.Log (“ bla bla bla ...”);  
Print ( “bla bla bla” + qualquerCoisa);
```

# Loops – for

```
for (valorInicial; Condição; passo)
{
    ...
}
```

```
for (i =0; i<5; i++)
{
    print (“i vale “+ i);
}
```

**Loops infinitos**

# Loops – While

```
while (enquanto uma condição for verdadeira)
{
    ...
}
```

**Fazer um contador igual  
ao for**

# Funções

Start ()  
Update()

**Fazer um contador começando pelo  
parametro 1 e terminando no parametro 2**

# Funções gerais

```
Function nomeFuncao (argumentos)
{
    ... // corpo da funcao
}
```

Exemplo: function conta (inicio: int, termino: int)

**Fazer um contador começando pelo  
parametro 1 e terminando no parametro 2**

# Funções – return

```
Var myGame = “ ”;  
myGame = MudaStatus (myGame);  
print (myGame);
```

```
myGame = mudaStatus (myGame);  
print (myGame);
```

```
function mudaStatus (status : String)  
{  
    if (status == “Loading”)  
    {  
        return “Correndo”;  
    }  
    else return “Loading”;  
}
```

# MonoBehavior Functions

Update

LateUpdate

FixedUpdate

Awake

Start



# Class Transform

# Método update

# Entendendo o update ()

- 1 – Criar um plano
- 2 – Criar uma esfera
- 3 – Criar e associar o script abaixo para a esfera:

```
function Update()  
{  
    transform.Translate(0, 0, 2);  
}
```

# Problema do Tempo Relativo

```
function Update() {  
    transform.Translate(0, 0, 2 *  
    Time.deltaTime);  
    print(transform.position.z);  
}
```

# Rotação

```
function Update() {  
    transform.Rotate ...  
}
```

**Fazer um planeta orbitando e verificar movimento do centro local**

# Transform attributes

```
function OnMouseDown() {  
    transform.position.z = 10;  
    transform.position.x = 5;  
}
```

**Fazer o planeta alternar entre mover e não mover ao clicar no mouse**

# Exemplo 1 – Controler de Player

1 – Criar uma cena vazia e contruir um pequeno terreno

## **Movement**

*Implementar o WASD*

O Edit → Project Settings → Input



# Input Class

GetAxis → Retorna qual Axis foi pressionado (faz o mapeamento) (de -1 a 1)

GetButton → True/False dependendo do botão pressionado

GetKey → True se a tecla for pressionada

GetMouseButton → True para tecla de mouse

ResetInputAxes → limpa todas as entradas

Atributos:

AnyKey → True se qualquer tecla for pressionada

mousePosition → coordenada do mouse na tela, como vetor 2D

# Player Controller Simple

```
function Update () {  
    // print (Input.GetAxis("Horizontal"));  
    // print (Input.GetAxis("Vertical"));  
    // print (Input.anyKey);  
    // print (Input.mousePosition);  
    print (Input.GetKey("r"));  
}
```

# Player Controller Simple

```
var vel = 10.0;
```

```
function Update () {  
    if (Input.GetAxis("Vertical"))  
    {  
        transform.Translate (vel*Time.deltaTime, 0, 0);  
    }  
}
```

# Player Controller II – direcao negativa

```
var vel = 10.0;

function Update () {
    if (Input.GetAxis("Vertical"))
    {
        transform.Translate (Input.GetAxis("Vertical")*vel*Time.deltaTime, 0, 0);
    }
}
```

# Player Controller II – Agora andando para os lados...

```
var vel = 10.0;

function Update () {
    if (Input.GetAxis("Vertical"))
    {
        transform.Translate (Input.GetAxis("Vertical")*vel*Time.deltaTime, 0, 0);
    }
    if (Input.GetAxis("Horizontal"))
    {
        transform.Translate (0, 0, Input.GetAxis("Horizontal")*vel*Time.deltaTime);
    }
}
```

# Player Controller II – girando com o mouse...

```
var vel = 10.0;
var roll = 10.0;
private var mouseX:int;
private var posMouse:int;

function Start ()
{
    posMouse = Input.mousePosition.x;
}
function Update () {

    mouseX = Input.mousePosition.x - posMouse;

    if (Input.GetAxis("Vertical"))
    {
        transform.Translate (Input.GetAxis("Vertical")*vel*Time.deltaTime, 0, 0);
    }
    if (Input.GetAxis("Horizontal"))
    {
        transform.Translate (0, 0, Input.GetAxis("Horizontal")*vel*Time.deltaTime);
    }
    transform.Rotate (0, mouseX*roll*Time.deltaTime, 0);
    posMouse = Input.mousePosition.x;
}
```

# Pulo – Voltar ao Script original

```
if (Input.GetButton("Jump"))  
{  
    moveDirection.y = jumpSpeed;  
}
```

Exercicio: fazer um FPS



# Exercicio: transformar em script de carro

**Criar um cenário simples para o carro,  
usando terreno**

# Exemplo 2 – Tiro

```
Var tiro:Rigidbody;  
Var velDisparo = 5.0;  
  
if (Input.GetButtonDown ("Fire1"))  
{  
    var tiroInstanciado : Rigidbody = Instantiate (tiro, transform.position,transform.rotation);  
    tiroInstanciado.velocity = transform.TransformDirection( Vector3 (0,0,speed));  
}
```

# Script 1: Player Controller

- 1 – Criar um script chamado PlayerController
- 2 – Criar um Asset Player
- 3 – Criar um Character Controller

# Player Controller com Fisica

```
var gravity = 20.0;
```

```
private var moveDirection = Vector3.zero;  
private var grounded : boolean = false;  
var isControllable : boolean = true;  
var controller : CharacterController ;  
controller = GetComponent(CharacterController);
```

```
function FixedUpdate() {  
    if(!isControllable)  
        Input.ResetInputAxes();  
    else{  
        if (grounded) {  
            moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));  
            moveDirection = transform.TransformDirection(moveDirection);  
  
        }  
        moveDirection.y -= gravity * Time.deltaTime;  
        var flags = controller.Move(moveDirection * Time.deltaTime);  
        grounded = ((flags & CollisionFlags.CollidedBelow) != 0 );  
    }  
}
```

# Player Controller

```
@script AddComponentMenu("Player/MiControler")
```

# Pulo – Voltar ao Script original

```
var jumpSpeed = 8.0;  
  
if (Input.GetButton("Jump"))  
{  
    moveDirection.y = jumpSpeed;  
}
```

# Exemplo 3 – Movimento de camera

# Movimento de Camera

```
var target : Transform;  
var distance = 10.0;  
var height = 5.0;
```

```
var heightDamping = 2.0;  
var rotationDamping = 3.0;  
var distanceDampingX = 0.5;  
var distanceDampingZ = 0.2;
```

```
var camSpeed = 2.0;  
var smoothed = true;
```



# Movimento de Camera

```
function LateUpdate () {  
  
    wantedRotationAngle = target.eulerAngles.y;  
    wantedHeight = target.position.y + height;  
    wantedDistanceZ = target.position.z - distance;  
    wantedDistanceX = target.position.x - distance;  
  
    currentRotationAngle = transform.eulerAngles.y;  
    currentHeight = transform.position.y;  
    currentDistanceZ = transform.position.z;  
    currentDistanceX = transform.position.x;  
}
```

# Movimento de Camera

```
currentRotationAngle = Mathf.LerpAngle (currentRotationAngle, wantedRotationAngle, rotationDamping * Time.deltaTime);
```

```
currentHeight = Mathf.Lerp (currentHeight, wantedHeight, heightDamping * Time.deltaTime);
```

```
currentDistanceZ = Mathf.Lerp(currentDistanceZ, wantedDistanceZ, distanceDampingZ * Time.deltaTime);
```

```
currentDistanceX = Mathf.Lerp(currentDistanceX, wantedDistanceX, distanceDampingX * Time.deltaTime);
```

```
currentRotation = Quaternion.Euler (0, currentRotationAngle, 0);
```

# Movimento de Camera

```
transform.position -= currentRotation * Vector3.forward * distance ;  
transform.position.x = currentDistanceX;  
transform.position.z = currentDistanceZ;  
transform.position.y = currentHeight;  
  
transform.LookAt(target);  
}
```

# Colocando Movimento Suave

// Substituir a última linha pela função LookAtMe:

```
function LookAtMe(){
    if(smoothed)
    {
        var camRotation = Quaternion.LookRotation(target.position - transform.position);
        transform.rotation = Quaternion.Slerp(transform.rotation, camRotation, Time.deltaTime *
camSpeed);
    }
    else{
        transform.LookAt(target);
    }
}
```

Exemplo 2 – Fazer um ambiente de jogo de plataforma, com visao ortogonal e com diversos niveis entre as plataformas. Ataptar o Character Controller para 2D

# Time.time

```
Print (Time.time);
```

# Gui Text

Inserir um Gui ->Text e posicionar num canto da tela  
Ver Parametros

# Gui Text

```
function Update () {  
    GetComponent (GUILayout).text = Time.time;  
}
```



# Gui Text

```
function Update () {  
    GetComponent (GUIText).text = Time.time.ToString();  
}
```

# Gui Text

```
function Update () {  
    GetComponent (GUILayout).text = parseInt (Time.time).ToString();  
}
```

# Maniuplando posicoes

```
var targetElement:Transform;
private var distancia:float;

function Start () {

}

function Update () {
    if (targetElement != null)
    {
        distancia = Vector3.Distance (targetElement.position, this.transform.position);
        print ("distancia = " + distancia);
    }
}
```

# Maniuplando posicoes

```
var targetElement:Transform;
private var distancia:float;

function Start () {
    targetElement = gameObject.Find("meu player").transform;
}

function Update () {
    if (targetElement != null)
    {
        distancia = Vector3.Distance (targetElement.position, this.transform.position);
        print ("distancia = " + distancia);
    }
}
}
```

# Exercicio

Colocar elevadores moveis no jogo de plataforma, com os objetos indo de um lado a outro, controlado por “waypoints”.

# Buscando por algo...

```
function Update () {  
    if (Physics.Raycast (transform.position, transform.forward, 10))  
        print ("vejo alguien");  
}
```

# Buscando alguém



Consultar...

```
var visao : RaycastHit;
```

```
function Start () {  
}
```

```
function Update () {  
    if (Physics.Raycast (transform.position, transform.forward, visao, 10))  
        print ("vejo a "+ visao.transform.name);  
}
```

# Instancias de objetos

```
var proyectil:GameObject;  
  
function Start () {  
  
}  
  
function Update () {  
    if (Input.GetButton("Fire1"))  
        var clonedProjetil : GameObject = Instantiate (projetil, transform.position,  
transform.rotation);  
  
}
```



# Limitando os tiros...

```
var projetil:GameObject;  
  
function Start () {  
  
}  
  
function Update () {  
    if (Input.GetButtonDown("Fire1"))  
        var clonedProjetil : GameObject = Instantiate (projetil, transform.position,  
transform.rotation);  
  
}
```

# Limitando mais ainda os tiros...

```
var projetil:GameObject;  
var fireRate:float = 0.3;  
var timeToNext : float;  
  
function Start () {  
  
}  
  
function Update () {  
    if (Input.GetButtonDown("Fire1") && Time.time > timeToNext){  
        TimeToNext = Time.time + fireRate;  
        var clonedProjetil : GameObject = Instantiate (projetil, transform.position,  
transform.rotation);  
    }  
  
}
```

# Fisica no Tiro...

```
var projetil:GameObject;  
var vel:float = 30;  
  
function Start () {  
  
}  
  
function Update () {  
    if (Input.GetButtonDown("Fire1")){  
        var clonedProjetil : GameObject = Instantiate (projetil, transform.position,  
transform.rotation);  
        clonedProjetil.rigidbody.velocity = transform.TransformDirection(Vector3(0,0,vel));  
    }  
}
```

# Destruindo o Tiro...

```
var lifeBullet:float = 3;  
  
function Start () {  
    Destroy (gameObject, lifeBullet);  
}
```

# Collision Callback...

```
function OnCollisionEnter (collision:Collision)
{
    Destroy (gameObject);
}
```

# Collision Callback...

```
function OnCollisionEnter (collision:Collision)
{
    print (collision.transform.name);
    Destroy (gameObject);
}
```

# Tags...

// Criar um Tag de indestrutivel...

```
function OnCollisionEnter (collision:Collision)
{
    if (collision.transform.tag != "indestrutivel")
    {
        print (collision.transform.name);
        Destroy (gameObject);
    }
}
```

# Destruindo as coisas!!!

```
function OnCollisionEnter (collision:Collision)
{
    collision.transform.SendMessage("acertou", SendMessageOptions.DontRequireReceiver);
    Destroy (gameObject);
}
```

// Criar script para o objeto a ser destruido:

```
function acertou ()
{
    Destroy(gameObject);
}
```



# Variavel de vida

```
var vida:int=2;
```

```
function acertou ()
```

```
{
```

```
    vida--;
```

```
    if (vida ==0)
```

```
        Destroy(gameObject);
```

```
}
```

# “Master” Script

**// Exemplo: apagar mouse da tela e mudar resolucao:**

```
function Start () {  
    Screen.showCursor=false;  
    Screen.fullScreen = true;  
}
```

# AI Pathfinding

**Criar um pequeno cenário com rampas**

**Selecionar objetos do percurso e colocar como static (porque?)**

**Window -> Navegation -> Bake (analisar)**

# AI Pathfinding

**Colocar um personagem (ou Game Object)**

**Inserir componente Navigation -> NavMeshAgent**

# AI Pathfinding Scripting

```
var objetivo:Transform;  
  
function Start () {  
  
}  
  
function Update () {  
    GetComponent(NavMeshAgent).destination = objetivo.position;  
}
```

# AI Pathfinding Scripting

```
var objetivo:Transform;  
  
function Start () {  
  
}  
  
function Update () {  
    GetComponent(NavMeshAgent).destination = objetivo.position;  
    print (GetComponent(NavMeshAgent).remainingDistance);  
}
```

# AI Pathfinding – caminho complexo

```
var objetivo:Transform;  
var objetivo1:Transform;  
var objetivo2:Transform;  
  
function Update () {  
  
    GetComponent(NavMeshAgent).destination = objetivo.position;  
  
    var distancia: float = GetComponent(NavMeshAgent).remainingDistance;  
  
    if (distancia < .1 && distancia != 0){  
        if (objetivo == objetivo1){  
            GetComponent (NavMeshAgent).SetDestination(objetivo2.position);  
            objetivo = objetivo2;  
        }  
        else{  
            GetComponent(NavMeshAgent).SetDestination(objetivo1.position);  
            objetivo = objetivo1;  
        }  
    }  
}
```

# AI Pathfinding – perseguição...

