

# Project ILIA

OpenPlay

Isac Cruz

# Project ILIA

Game Development  
OpenPlay

Dezembro 2019

## Resumo

## Agradecimientos

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetos no unity . . . . .	1
1.2	Player . . . . .	1
1.3	Game System . . . . .	2
<b>2</b>	<b>Payer</b>	<b>3</b>
2.1	Componentes . . . . .	3
2.2	Children . . . . .	3
<b>3</b>	<b>Base</b>	<b>4</b>
3.1	Componentes . . . . .	4
3.2	Children . . . . .	4
<b>4</b>	<b>Game Master</b>	<b>5</b>
4.1	Conponentes . . . . .	5
4.2	Children . . . . .	5
<b>5</b>	<b>Componentes</b>	<b>6</b>
5.1	Player Movement . . . . .	6
5.1.1	Variaveis públicas . . . . .	6
5.2	Player Interactor . . . . .	6
5.2.1	Funções: . . . . .	6
5.3	Compass Changer . . . . .	7
5.3.1	Funções . . . . .	7
5.3.2	Variaveis públicas . . . . .	7
5.4	Health . . . . .	7
5.4.1	Funcões . . . . .	7
5.4.2	Variaveis publicas . . . . .	7
5.5	Mouse Look . . . . .	7
5.5.1	Variaveis públicas . . . . .	7
5.6	Weapon Switching . . . . .	7
5.6.1	Funcões . . . . .	8
5.6.2	Variaveis publicas . . . . .	8
5.7	Gun . . . . .	8

5.7.1	Variaveis publicas . . . . .	8
5.8	Base Parent . . . . .	8
5.8.1	Funcões . . . . .	8
5.8.2	Variaveis publicas . . . . .	8
5.9	Computer . . . . .	9
5.9.1	Funcões . . . . .	9
5.9.2	Variaveis publicas . . . . .	9
5.10	Door . . . . .	9
5.10.1	Funcões . . . . .	9
5.10.2	Variaveis publicas . . . . .	9
5.11	Interactable Trigger . . . . .	9
5.11.1	Variaveis publicas . . . . .	9
5.12	Base Trigger . . . . .	9
5.12.1	Variaveis publicas . . . . .	10
5.13	Compass . . . . .	10
5.13.1	Funcões . . . . .	10
5.13.2	Variaveis publicas . . . . .	10
5.14	Game Master . . . . .	10
5.14.1	Funcões . . . . .	10
5.14.2	Variaveis publicas . . . . .	10
5.15	Tool . . . . .	10
5.15.1	Funcões . . . . .	10
5.16	Ammo UI . . . . .	10
5.16.1	Variaveis publicas . . . . .	11
5.17	Health UI . . . . .	11
5.17.1	Variaveis publicas . . . . .	11
5.18	UI Widget . . . . .	11
5.18.1	Funcões . . . . .	11
<b>6</b>	<b>Prefabs</b>	<b>12</b>
6.1	Main Camera . . . . .	12
6.1.1	Componentes . . . . .	12
6.1.2	Childrens . . . . .	12
6.2	Weapon Holder . . . . .	12
6.2.1	Componentes . . . . .	12
6.2.2	Childrens . . . . .	12
6.3	Guns . . . . .	12
6.3.1	Componentes . . . . .	13
6.4	Computer . . . . .	13
6.4.1	Componentes . . . . .	13
6.4.2	Childrens . . . . .	13
6.5	Door . . . . .	13
6.5.1	Componentes . . . . .	13
6.5.2	Childrens . . . . .	13
6.6	Hitbox(Base) . . . . .	13
6.6.1	Componentes . . . . .	13

6.7	Hitbox(interactable) . . . . .	13
6.7.1	Componentes . . . . .	14
<b>7</b>	<b>UI</b>	<b>15</b>
7.1	UI Manager . . . . .	15
7.2	Ammo Counter . . . . .	15
7.2.1	Componentes . . . . .	15
7.3	Health Bar . . . . .	15
7.3.1	Componentes . . . . .	15
7.3.2	Children . . . . .	15
7.4	Compass . . . . .	16
<b>8</b>	<b>IA</b>	<b>17</b>
8.1	AI Behavior . . . . .	17
<b>9</b>	<b>Interactable's Data System</b>	<b>18</b>
9.1	Tool . . . . .	18

# Capítulo 1

## Introdução

### 1.1 Objetos no unity

No unity todos os objectos usados no jogo são compostos por componentes.

Por exemplo um cubo que é colocado no mapa possui varios componentes, cada um responsavel por uma caraterística. Neste caso específico os componentes são:

- *Box Collider* : responsavel pela colisão;
- *Mesh Render* : responsavel pela renderização do objeto na tela;
- *Rigid Body* : responsavel pela simulação da fisica do objeto(peso, queficiente de atrito, gravidade, ect..)
- *Transform* : responsavel por guardar e modificar as informações de *Location*, *Rotation* e *Scale*

Estes são os componentes que um cubo tem por defeito, mas é possivel adicionar mais componentes quer sejam eles parte do unity o criados à parte em C#.

Por exemplo: criar um componente Health que adiciona um valor de vida ao cubo.

### 1.2 Player

O *First Person Player* é um *Charater Controller* que possui varios componentes, para interagir com outros objetos, ter um sistema de vida e dano, simular gravidade entre outros. Capítulo 2.

Além dos Componentes tambem possui 3 childrens: a main Camera, o groundCheck, e o cilinder. Capítulo 2



## 1.3 Game System

Game System possui um conjunto de bases espalhadas no mapa e um *Game-Manager* também no mapa, porém só pode haver um *GameManager*.

# Capítulo 2

## Player

### 2.1 Componentes

Os Componentes programados à parte tem funções bem específicas:

- *Player Movement* : físicas, Seção 5.1;
- *Interactor* : interação com objetos ao redor, Seção 5.2;
- *Compass Changer* : orientação da bussola, Seção 5.3;
- *Health* : vida, Seção 5.4.

### 2.2 Children

- Main Camera: camera first person Seção 6.1;
- Cylinder: ocupa o espaço da skin;
- GroundCheck: usado no player movement (Seção 5.1), para verificar se o player não está em queda livre.

# Capítulo 3

## Base

### 3.1 Componentes

Os Componentes programados à parte tem funções bem específicas:

- *Base Parent* : Responsavel pelo funcionamento da base Seção 5.8.

### 3.2 Children

- estrutura da base: ocupa o espaço da mesh e da skin;
- HitBox: trigger que detecta se o player está dentro da base Seção 6.6.
- computer: permite ao player interagir com a base Seção 6.4;
- Door: Porta da base Seção 6.5.

## Capítulo 4

# Game Master

Responsavel pelo funcionamento geral do jogo;

### 4.1 Componentes

- Game MasterSeção 5.14;

### 4.2 Children

# Capítulo 5

## Componentes

### 5.1 Player Movement

Responsavel pela simulação de fisicas.

#### 5.1.1 Variaveis públicas

- *FLOAT* speed: velocidade de movimento(andar);
- *FLOAT* gravity: força da gravidade sobre o player;
- *FLOAT* ground Distance: distancia minima do chão para reconhecer que o player está sobre uma superficie;
- *FLOAT* jump Height: altura de salto maxima;
- *FLOAT* max Fall Height: altura maxima de queda sem levar dano;
- *FLOAT* multiplayer: multiplicador de variaveis fisicas, multiplica tudo de forma proporcional, se estiver a 1 não faz efeito;

### 5.2 Player Interactor

Guarda a referencia do objeto proximo com o qual se possa interagir.

#### 5.2.1 Funções:

- *setInteract(Tool item)* que aceita como argumento, um objeto do tipo *Tool*(Capítulo 9) e muda a referencia do objeto de interção(ex. ammo, gun, usado para interagir com trigger boxes de items, e objectos iterativos);
- *resetInteract()* que muda a referencia para null, não aceita argumentos.

## 5.3 Compass Changer

Muda a orientação da bússola do player, através do próprio Player.

### 5.3.1 Funções

- *PointTo(Transform nextMission)*: Permite mudar a direção para onde a bússola aponta, aceita como argumento um *Transform*. A bússola passará a apontar para a localização do que for passado como argumento.

### 5.3.2 Variáveis públicas

- *Compass* compass: referência da bússola do player;

## 5.4 Health

Responsável pela vida.

### 5.4.1 Funções

- *TakeDamage(float damage)*: retira vida, na quantidade passada como argumento;
- *Die()*: destrói o objeto;

### 5.4.2 Variáveis públicas

- *float health* vida atual;
- *float maxhealth* vida máxima;

## 5.5 Mouse Look

Responsável pela rotação da câmera. Faz com que a câmera seja controlada pelo rato.

### 5.5.1 Variáveis públicas

- (*FLOAT*) Mouse Sensivity: Sensibilidade do rato (valor base 100);
- (*TRANSFORM*) Player Body: Referência do Player.

## 5.6 Weapon Switching

Responsável pela troca das armas a para usar(ex. trocar da pistola para a shotgun).

### 5.6.1 Funções

- *GetActualGun()*: Retorna a arma atual a ser usada;

### 5.6.2 Variaveis publicas

- *(Transform)*Gun List: Weapon Holder;

## 5.7 Gun

Funcionamento da arma.

### 5.7.1 Variaveis publicas

- *(FLOAT)*Damage: Dano da arma;
- *(FLOAT)*Impact force: força de impacto da bala;
- *(FLOAT)*Range: Alcance da arma;
- *(FLOAT)*Fire Rate: Numero de balas por segundo;
- *(INT)*Ammo: Numero de balas no pente da arma;
- *(INT)*Max Ammo: Numero maximo de balas no pente da arma;

## 5.8 Base Parent

Responsavel por comunicar ao game master as ações do jogador.

### 5.8.1 Funções

- *ComputerActivated()*: Activa a base;
- *ResetComputer()*: desativa o computador permitindo que este seja reativado;
- *Unlock()*: Destranca a porta;
- *Sequence()*: Pede ao game master para avançar com o jogo;

### 5.8.2 Variaveis publicas

- *(STRING)*id: Nome de identificação da base;
- *(FLOAR)*health: vida da base;
- *(DOOR)*Door: porta da base;
- *(FLOAT)*R: ;

## 5.9 Computer

Responsavel por comunicar ao game master as ações do jogador. Descende de *TOOL* (Seção 5.15).

### 5.9.1 Funções

- *Action()*: executa a ação do Computer(HardCoded).

### 5.9.2 Variaveis publicas

- *(BASE PARENT)*baseParent: a base a que está associado.

## 5.10 Door

Responsavel abrir e fecha a porta. Descende de *Tool* (Seção 5.15).

### 5.10.1 Funções

- *Action()*: executa a ação do Porta(HardCoded);
- *open()*: abre a porta;
- *close()*: fecha a porta.

### 5.10.2 Variaveis publicas

- *(BOOL)*Lock: referente ao facto da porta estar trancada;
- *(Animation)*: Animações de abrir e fechar.

## 5.11 Interactable Trigger

Detecta a colisão com o player e conecta o à *Tool* (Seção 5.15) à qual está associado.

### 5.11.1 Variaveis publicas

- *(Tool)* Item: referente referente ao objecto o parent está associado;

## 5.12 Base Trigger

Detecta a colisão com o player reconhece se ele está dentro da base.



### 5.12.1 Variaveis publicas

- (*Base Parent*) Base: referente ao objecto o parent está associado;

## 5.13 Compass

Calcula qual a direção para onde está o norte, e a missão.

### 5.13.1 Funções

- *ChangeMission(Transform newMission)*: Muda o local para onde a bússola aponta.

### 5.13.2 Variaveis publicas

- (*Transform*) Player: referencia da localização do player;
- (*Rect Transform*) North Layer: camada do ponteiro do norte;
- (*Rect Transform*) Mission Layer: camada do ponteiro da missão;

## 5.14 Game Master

### 5.14.1 Funções

- *sequence(string id)*: executa a sequencia de ações da base(ex. defence, run, etc), recebe como argumento o id da base pretendida.

### 5.14.2 Variaveis publicas

- (*Transform*)Player: referencia do player.

## 5.15 Tool

Class Mãe de todos os objetos interativos do jogo.

### 5.15.1 Funções

- *Action()*: função onde fica a ação do objecto;

## 5.16 Ammo UI

Responsavel por atualizar o numero de balas que aparece no ecrã.

### 5.16.1 Variaveis publicas

- (*Text*) Ammo Display : referente texto que mostra o numero de balas;
- (*WeaponSwitching*) Seção 5.6 Weapon list: é o Weapon holder do player Seção 6.2.

## 5.17 Health UI

Responsavel por decidir quantas mini barras de vida aparecem.

### 5.17.1 Variaveis publicas

- (*Transform*) Display : referente ao objeto que contem todas as mini barras;
- (*Health*) Seção 5.4 Player health: referente á vida do player.

## 5.18 UI Widget

Classe mãe de todos os scripts usados na UI.

### 5.18.1 Funções

*Get UI Manager()* : Retorna o UI manager.

## Capítulo 6

# Prefabs

### 6.1 Main Camera

Camera do Player.

#### 6.1.1 Componentes

- Mouse Look : rotação ad camera Seção 5.5.

#### 6.1.2 Childrens

- Weapon Holder: sistema de armas Seção 6.2.

### 6.2 Weapon Holder

Responsavel por todas as armas que o player usa, e pela capacidade de trocar entre elas.

#### 6.2.1 Componentes

- Weapon Switching : trocar de armas Seção 5.6.

#### 6.2.2 Childrens

- Todas as armas que o player possui.

### 6.3 Guns

Responsavel por todas as armas que o player usa, e pela capacidade de trocar entre elas.

### 6.3.1 Componentes

- Gun : funcionamento da arma Seção 5.7.

## 6.4 Computer

Responsavel pela interação player/base.

### 6.4.1 Componentes

- computer : Permite ao jogador interagir com a base Seção 5.9.

### 6.4.2 Childrens

- HitBox: Detecta se o player está proximo, para interagir, Seção 6.7.

## 6.5 Door

Porta da base.

### 6.5.1 Componentes

- Door : Controla a porta Seção 5.10.

### 6.5.2 Childrens

- HitBox(interactable): Detecta se o player está proximo, para interagir, Seção 6.7.
- DoorMesh : a skin e o mesh da porta.

## 6.6 Hitbox(Base)

Uma hitbox especifica para Bases.

### 6.6.1 Componentes

- Base Trigger : interage com o interactor do player e para que este possa interagir com uma base associada. associado.Seção 5.12.

## 6.7 Hitbox(interactable)

Uma hibox para todo o tipo de *Tools* Seção 5.15.

### 6.7.1 Componentes

- Intractable Trigger : interage com o iterator do player e para que este possa interagir com o objeto associado.Seção 5.11.

# Capítulo 7

## UI

### 7.1 UI Manager

O UI Manager é o script mais importante de todo o UI. Nele são guardadas e alteradas todas as variáveis usadas pelos UI Widgets (Seção 5.18), os UI Widgets usam apenas e somente apenas variáveis fornecidas pelo UI Manager.

### 7.2 Ammo Counter

Responsavel por mostrar o numero de balas da arma.

#### 7.2.1 Componentes

Ammo Counter: responsavel por conseguir os valores certos para mostrar no ecrã Seção 5.16.

### 7.3 Health Bar

Responsavel por mostrar a barra de vida, existem varias mini barras de vida no ecrã, conforme a vida decresce elas vão desaparecendo, quando a vida chega a zero as barras desaparecem, e o player morre.

#### 7.3.1 Componentes

Health UI : controlas as barras que aparecem no ecrã Seção 5.17.

#### 7.3.2 Children

Todas as mini barras.

## 7.4 Compass

São duas 3 Camadas de imagens: o fundo, o ponteiro do norte e o ponteiro da missão.

Os Ponteiros são controlados pelo Compass Seção 5.13.

# Capítulo 8

## IA

### 8.1 AI Behavior

Responsavel pelo comportamento da AI de forma geral, controlando a for como o player é detectado.



## Capítulo 9

# Interactable's Data System

### 9.1 Tool

A Tool é a classe mãe de todos os objectos interactivos, tais como items, baús, computers (Seção 6.4), etc. As construções como as bases (Capítulo 3) ficam de fora desta família.

a sua característica principal é o facto de terem uma função chamada `Action()` (Seção 5.15) que contém todos os efeitos do objecto quando ativado. Como é uma função pública, pode ser chamada por qualquer script, que se referencie ao objecto interactivo como Tool.