

Partículas, modelos 3D e animações 3D

Flávio Roberto Dias Silva

Encontro 6

Desenvolvimento de Jogos Eletrônicos com Unity 3D.

Universidade Estadual do Oeste do Paraná

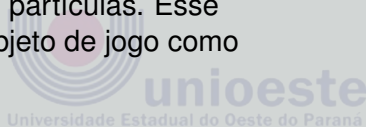


Sistema de Partículas

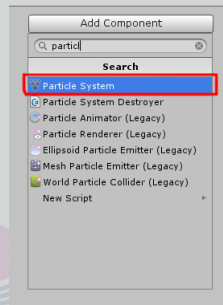
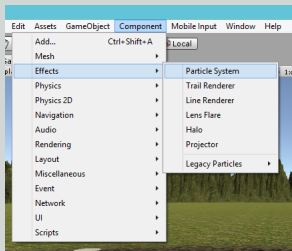
Sistema de Partículas

No desenvolvimento de um jogo muitas vezes sentimos a necessidade de efeitos de fumaça, areia, água, fogo, entre outros. Na maioria dos jogos modernos esses efeitos são feitos com sistemas de partículas.

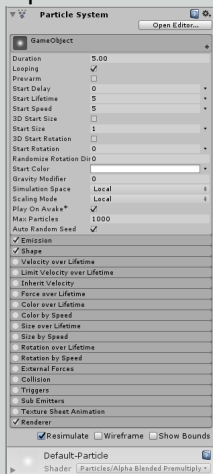
Em Unity temos uma excelente ferramenta de desenvolvimento voltada ao sistema de partículas. Esse sistema de partículas é inserido num objeto de jogo como componente.



Para adicionar um sistema de partículas utilizamos os mesmos procedimentos de adicionar qualquer outro componente.



O componente sistema de partículas é dotado de várias propriedades das quais destacaremos algumas delas.

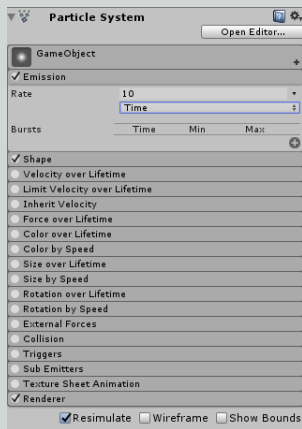


Ao lado podemos ver o inspetor do sistema de partículas. Nele podemos ver várias propriedades auto descritivas como: duração, looping, delay inicial, tempo de vida inicial, velocidade inicial, tamanho inicial, rotação inicial entre várias.

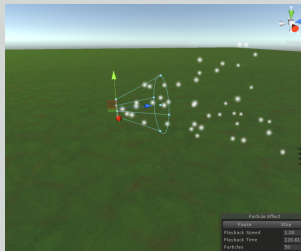
Podemos ver alguns checkbox onde temos marcados: Emissão, shape e render e outros tantos desmarcados.

Destacamos aqui o **Max Particles** que obviamente quer dizer número máximo de partículas mas que merece observações no sentido de que, esse número depende do loop, em cada loop temos o número max de partículas e esse número depende da emissão, essa última é a propriedade que diz quantas partículas serão emitidas por segundo.

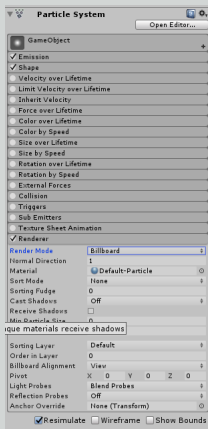




A emissão trata da quantidade de partículas emitidas por segundo limitada pelo max particles mencionado anteriormente.



O shape trata da forma geométrica que estará disparando as partículas. No caso da imagem um cone que tem base de raio 1 e abertura com 25 graus.



Por fim o renderer trata da imagem que será utilizada pelo sistema de partículas, além de inclusões recentes da Unity da possibilidade de utilizar mesh's (malhas 3D) como objeto de emissão propriedades de luz, reflexão entre muitas que são novidades.

Exercício: Aplique sistemas de partículas aos projeteis construídos no encontro anterior, tanto a um rastro do projetil quanto ao impacto do projetil.



Modelos 3D

Sendo a Unity um motor de jogos criado inicialmente para produção de jogos 3D, temos a disposição algumas ferramentas para manipulação de modelos 3D.

Inicialmente focaremos na importação e utilização de modelos 3D.



Como já ferramenta rotineira na manipulação de elementos da Unity, temos a disposição geometrias primitivas nativas na engine como: cubos, esferas, capsulas, planos entre outros. Porém, a maioria dos jogos não é restrita a essas geometrias primitivas, temos modelos customizados que dão particularidade ao jogo.

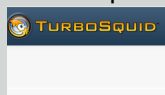
A unity tem suporte a várias extensões de objetos 3D como os .OBJ, .BLEND, .3DS, .FBX, entre outros. Para adicionar um novo modelo 3D no nosso projeto devemos simplesmente ter um objeto compatível e coloca-lo na pasta *Assets* do projeto, dessa forma ele fica disponível para ser adicionado a cena.

Existem alguns sites dedicados a publicação e distribuição de modelos 3D. Entre esses estão:

TF3DM



TurboSquid



Archive3D



Para criar seus próprios modelos 3D você deve utilizar um software de modelagem 3D(de preferencia poligonal). Por exemplo

Blender



3D studio Max



Maia



Existem software de criação 3D artística, baseada em esculturas 3D que podem ser utilizados também. Entre eles:

ZBrush



Sculptris



Nesse curso não serão abordados conceitos de modelagem 3D, nos reservaremos aqui a adicionar e manipular os objetos 3D.



Exercício: Importe um novo modelo 3D para o seu projeto de jogo. [Obs: De preferencia para .OBJ ou .FBX, tente adicionar alguma construção(prédio, casa, palácios, castelos)]



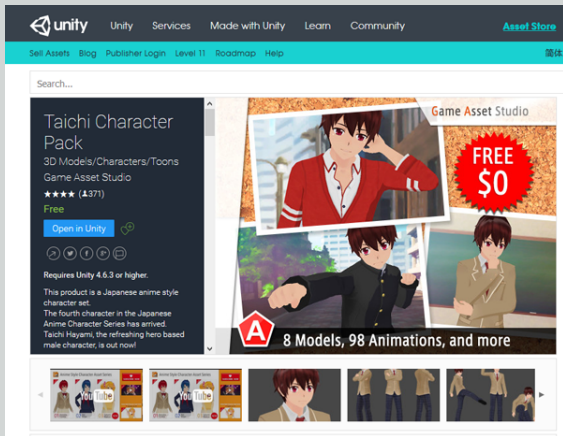
Animações 3D

Um elemento quase indispensável em qualquer jogo é a animação, a Unity usa como ferramenta de animação um componente chamado Animator.

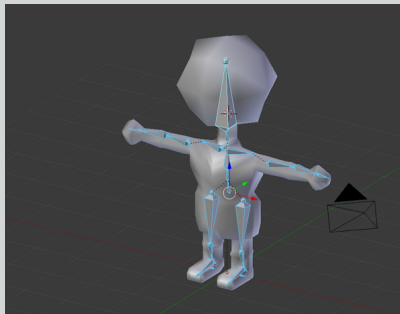
É importante destacar que não criaremos animações, apenas trabalharemos em como programar a animação no jogo a partir de animações prontas.



Para trabalhar animações importaremos um pacote da asset store que contém um número vasto de animações humanoides.



Para utilizar essas animações precisamos de um modelo 3D com um esqueleto compatível com animações humanoide.



Deve ter três ossos na coluna, dois ossos para cada perna e cada braços, pelo menos um osso nas mãos e pés, um osso no pescoço, um osso na cabeça e um osso para cada ombro.

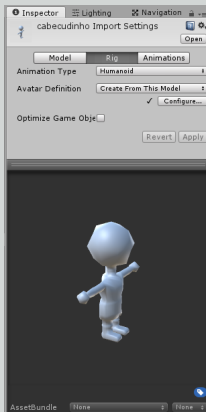
O modelo da imagem anterior está disponível no Asana para download

Nosso objetivo daqui em diante é substituir nosso personagem da aula anterior (que é uma geometria primitiva) por um personagem animado.

Nesse caminho adicionemos um personagem com esqueleto humanoide a cena da Unity.

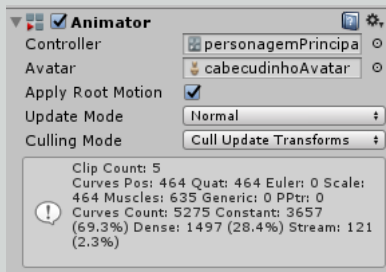


Com o personagem selecionado na aba project alteremos o inspector do modelo para humanoide.

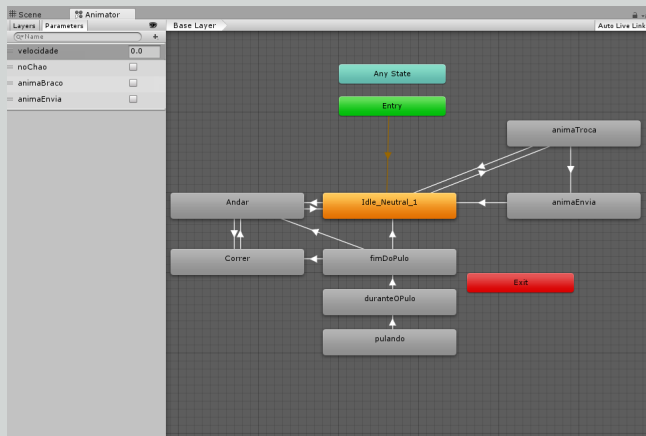


Agora com o personagem adicionado a cena na aba Hierarchy adicionemos a ele um componente Animator.

Para que o Animator funcione precisamos preencher o inspector dele com um AnimatorController. Criemos um AnimatorController na aba project.



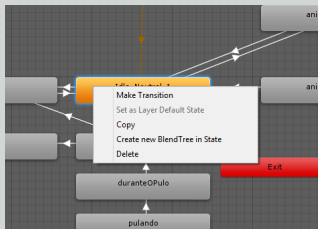
Precisaremos agora adicionar ao AnimatorController na aba Animator as animações desejadas e configurar as transições.



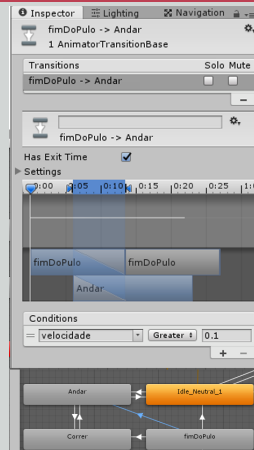
Arrastando uma animação para a aba animator criamos um estado de animação. Clicando com o botão direito em cima do estado de animação podemos criar uma transição entre duas animações com a opção make transition.

O make transition criará uma seta que deverá ligar duas animações.

Clicando sobre a seta de transição podemos adicionar condições de transição que serão alteradas via código.

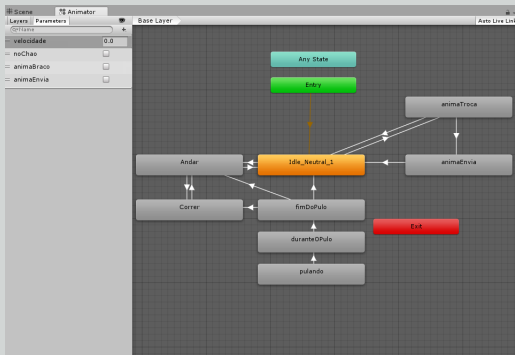


criando uma transição



editando as condições de
transição

Para mudar de uma animação para a outra utilizamos parametros dentro do animator, para criar esses parâmetros, podemos acessar a aba parametros dentro do animator e clicar no sinal de mais “+” e criar parametros com alguns tipos bem conhecidos nossos.



Com um animator corretamente configurado podemos acessa-lo via script através de um componente do tipo **Animator**, fazendo:

```
Animator animator = GetComponent<Animator>();
```

E para alternar as animações modificamos os parametros atraves de funções específicas para isso:

```
animator.SetBool(“nomeDoParametroBool”, true);
```

```
animator.SetInt(“nomeDoParametroInt”, 1);
```

```
animator.SetFloat(“nomeDoParametroFloat”, 0.25f);
```

entre alguns.

Exercício:

Animar nosso personagem humanoide no lugar do nosso personagem geometria primitiva.



- [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.

- [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.

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

[http://wiki.unity3d.com/index.php/Main_Page][Acesso em 18/04/2016]

