Curso Técnico

de Programação de Jogos Digitais

Aula 33

índice

[1. Cuidados na importação 3](#_Toc170111769)

[2. importando cenários para unity 4](#_Toc170111770)

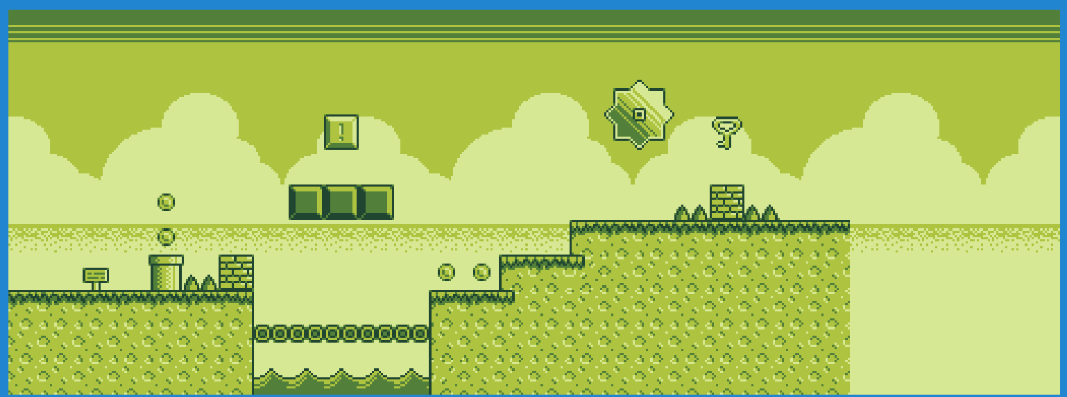
[3. Referencias 8](#_Toc170111771)

# Cuidados na importação

Após nosso cenário todo desenhado na ferramenta Sprite Fusion devemos então acomodá-lo ao nosso projeto na Unity, mas antes devemos ter a certeza se tudo está correto.

Para importar um cenário feito no Sprite Fusion para Unity, aqui estão alguns cuidados principais que você deve ter:

Formato de Arquivo: Verifique se os arquivos de imagem exportados do Sprite Fusion estão em formatos suportados pelo Unity, como PNG, JPG ou TIF. O Unity suporta uma variedade de formatos de imagem, mas é bom garantir que estão adequados para otimização e desempenho.



Configuração de Textura: Ao importar para o Unity, ajuste as configurações de importação de textura conforme necessário. Isso inclui definir a resolução correta, ajustar a compressão (se aplicável) para equilibrar entre qualidade visual e desempenho, e considerar a geração automática de mipmaps para melhorar o desempenho em diferentes distâncias de visualização.

Sistema de Coordenadas: Verifique se os cenários importados estão alinhados corretamente com o sistema de coordenadas do Unity. Isso é especialmente importante se você estiver integrando vários ativos ou se o cenário for parte de uma cena maior.

Organização de Camadas: Se o cenário foi criado com múltiplas camadas ou elementos separados no Sprite Fusion, organize-os adequadamente no Unity para facilitar o trabalho com eles. Isso pode envolver o uso de objetos vazios (empty game objects) como pais para cada camada ou grupo de elementos relacionados.

Performance: Considere o desempenho ao importar grandes cenários. O Unity possui ferramentas para otimização de desempenho, como a combinação de malhas e o uso eficiente de texturas, que podem ser aplicadas para melhorar o desempenho do jogo.



Como são duas ferramentas usadas para gerar o projeto tenha um cuidado maior quanto as dimensões e tamanhos de objetos pois isso pode interferir diretamente na performance do jogo.

Colisão e Física: Se o cenário requer interação física ou colisão com personagens ou outros objetos, configure adequadamente as colisões no Unity. Isso pode envolver a adição de colisores 2D ou 3D, dependendo do tipo de jogo e do cenário.

Ao seguir esses cuidados, você estará melhor preparado para integrar cenários criados no Sprite Fusion ao Unity de maneira eficiente e sem problemas significativos.

# importando cenários para unity

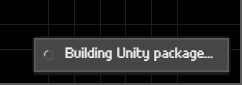
Primeiro vamos importar um cenário maior já criado através do sprite fusion.



Com todo cenário pronto, siga para o export e escolha o formato Unity



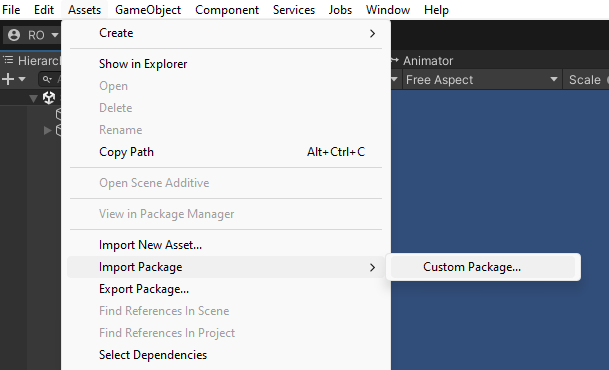
Uma mensagem de seu arquivo sendo gerado irá aparecer



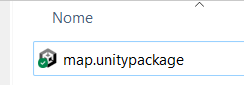
E teremos o arquivo pronto.



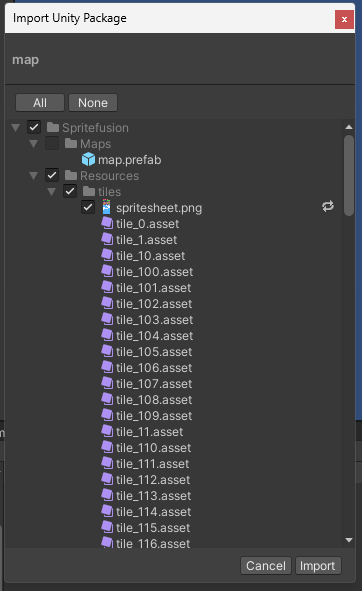
Abra seu projeto da Unity e escolha na barra de ferramentas o seguinte caminho : Assets/import package/custom package



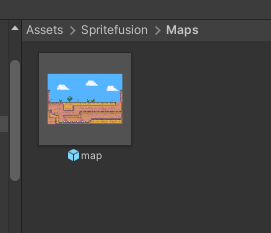
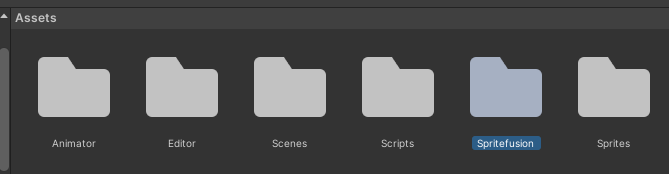
Selecione o arquivo gerado no sprite fusion



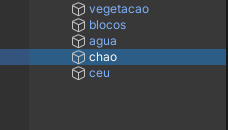
Na importação será gerado todas as pastas e arquivos conforme feito no Sprite fusion.

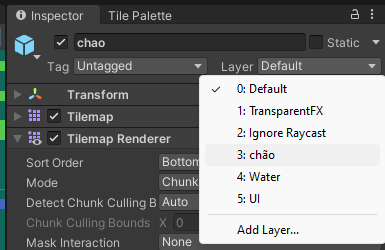


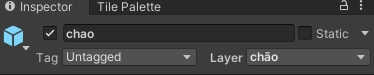
Será criada a pasta do Sprite Fusion com todo cenário.



Em nosso cenário, foi criado uma layer para ser a superfície do player. Selecione a layer de chão.







Temos um novo cenário com tudo gerado no Sprite Fusion e podemos agora utilizar em nosso jogo. Qualquer modificação pode ser feita diretamente na Unity ou refazendo o cenário no Sprite Fusion e o substituindo em nosso projeto de jogo.

Os códigos usados em nossas aulas estão disponíveis em nosso repositório no GitHub.



https://github.com/rildexter/pjd/tree/main/djm

# Referencias

1. Unity Technologies. (2020). **Unity User Manual.** Unity Technologies.
2. [C#, 2020] **Visual C# Developer Center**, Microsoft Docs.

Microsoft Corporation. (2020).

1. Geig, M. (2018). **Unity 2018 Game Development in 24 Hours**, Sams Teach Yourself. Sams Publishing.
2. Hocking, J. (2015). **Unity in Action: Multiplatform Game Development in C#.** Manning Publications.