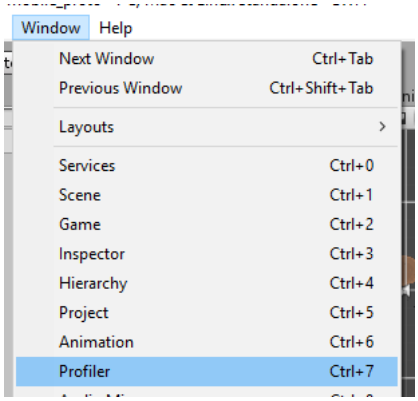
05

Utilizando o profiler

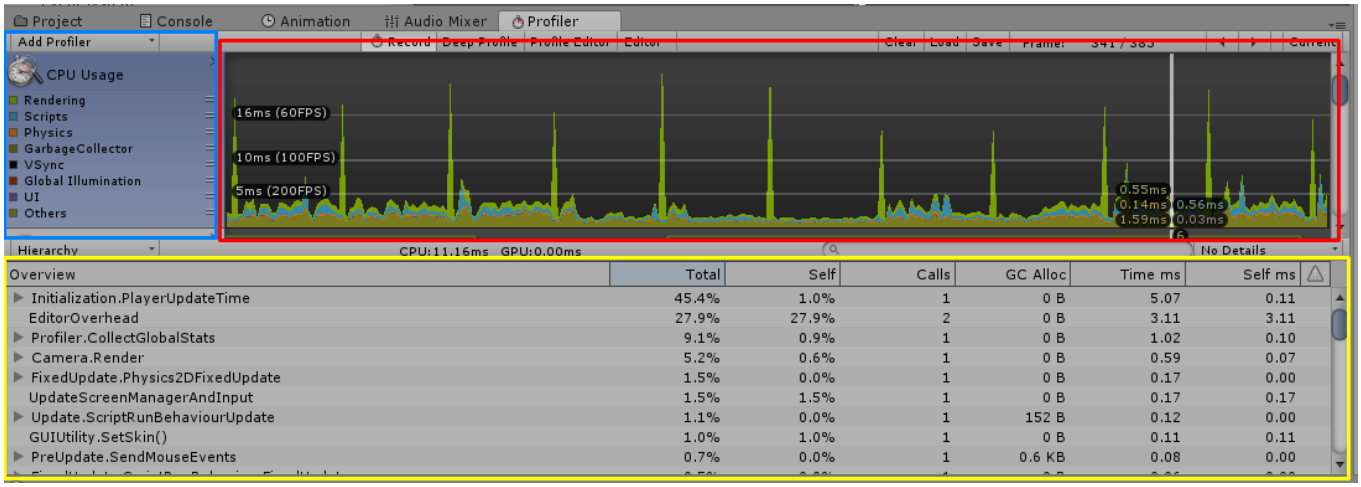
É muito comum durante o desenvolvimento de um jogo termos problemas com performance. E em muitos desses casos é difícil achar onde estão os gargalos dentro do projeto, por isso que utilizamos o profiler da unity para achar problemas de performance do jogo.

Para abrir a janela do profiler vá em **window->profiler**.



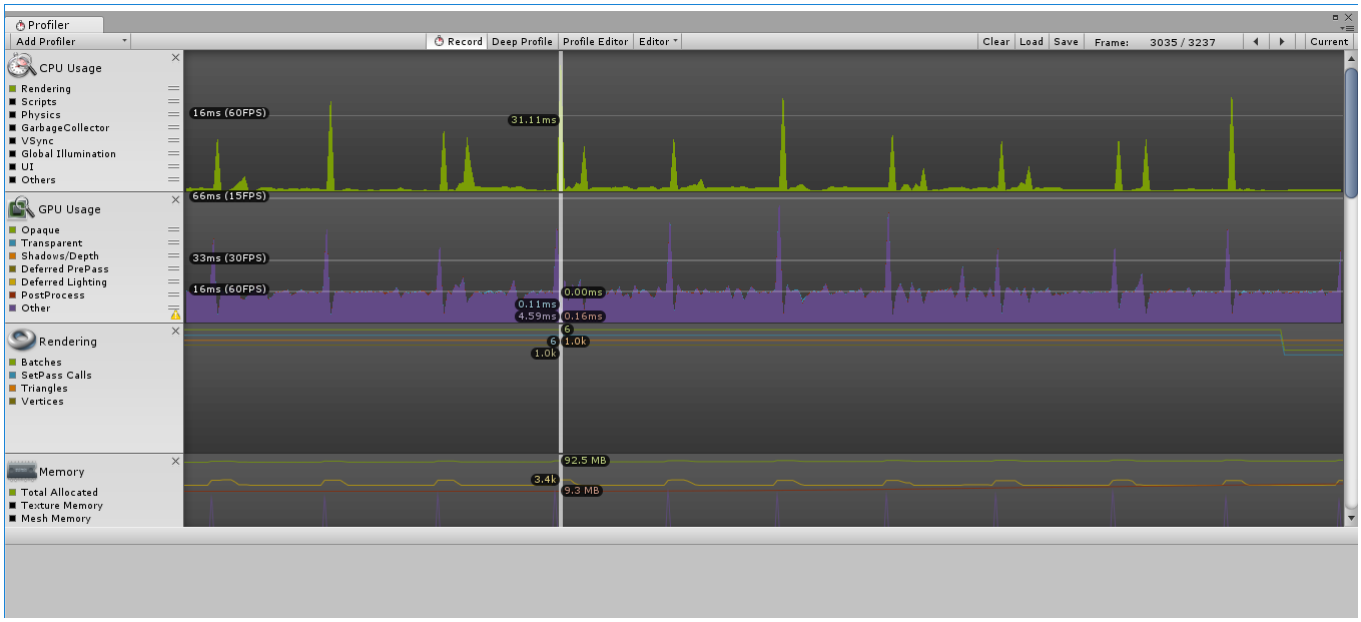
Ao fazer isso você verá a janela do profiler abrir. Com essa janela aberta veremos que um gráfico é formado enquanto estamos jogando o jogo (marcado em vermelho na imagem), esse gráfico nos mostra o uso de CPU por categoria do em cada instante do jogo. no canto esquerdo desse gráfico conseguimos ver quantos frames por segundo estamos alcançando em cada frame.

Em azul no a imagem, vemos as categorias do que queremos que apareça dentro do gráfico. Aqui podemos excluir as categorias que tem menos importância para o debug como por exemplo o VSync (Opção que mostra o quanto a CPU está esperando para calcular o próximo frame)



Além disso podemos clicar em qualquer momento desse gráfico e ver quais funções foram executadas nesse frame. Aqui temos opções para organizar essas funções por quanto tempo elas demoraram para ser executadas, número de vezes que elas foram chamadas, quando lixo elas geraram para o computador e assim por diante.

Com essas 3 abas já temos opções de sobra para achar a maior parte dos problemas de performance do jogo. Porém o profiler da Unity possui mais opções.



Se abaixarmos a barra de rolagem da parte superior do profiler, veremos que o profiler possui estatísticas sobre quanto tempo a placa de vídeo está trabalhando, como está a memória do computador, quanto estamos enviando de mensagens para a rede, etc...

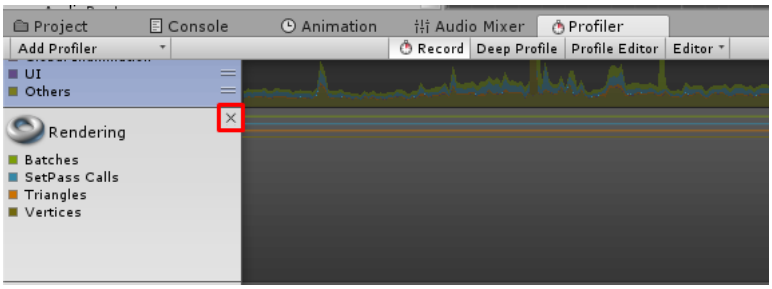
Veja que no nosso caso os picos de uso da CPU estão alinhados com os picos de uso da GPU e se selecionamos um desses frames veremos que na CPU o método que mais demorou para ser executado foi o `Gfx.WaitForPresent`.

Esse método não é algo que foi chamado pelo nosso jogo, porém o profiler mostra tudo que foi chamado pela própria Unity também e com isso podemos saber o que está acontecendo. Nesse caso a CPU está esperando a GPU terminar de desenhar uma cena.

É importante saber que ao mesmo tempo que estamos testando a performance do jogo e onde devemos otimizar ele, o próprio editor da Unity adicionar um *overhead* no nosso computador, ou seja, o jogo irá executar mais lento dentro do editor da Unity do que quando geramos a *build* dele.

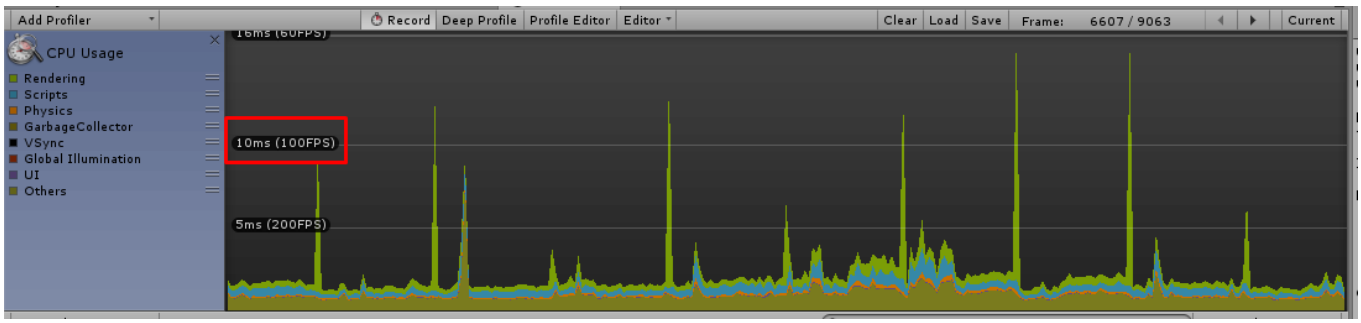
Essa lentidão é aumentada quando estamos utilizando o profiler, isso porque agora o editor da unity, além de executar o jogo está verificando como essa execução é feita, quais métodos estão sendo chamados e quando tempo cada um demora, além das estatísticas de memória, isso tudo adiciona mais coisas para o processador fazer ao mesmo tempo que estamos jogando.

Por isso é bom ter em mente que quanto mais dessas janelas do profiler nós tivermos abertas mais ele vai causar lentidão no jogo. Se olharmos de novo para o pico de processamento do nosso jogo, ele provavelmente está sendo causado pelo próprio editor da unity redesenhando a interface do profiler enquanto estamos jogando. Ou seja, esse não é um problema que teremos no jogo final.



Para não termos um pico tão grande podemos fechar as janelas do profiler que não estamos utilizando clicando no X no canto superior direito da categoria que não estamos utilizando.

Fazendo isso, se rodarmos de novo o jogo, veremos que ainda temos picos de processamento, porque a CPU está esperando a GPU. Porém agora esses picos do gráfico estão mais próximos dos 100 FPS do que dos 60 FPS, como tínhamos antes.



No geral o próprio profiler da Unity é uma ferramenta muito útil para acharmos problemas de perfomance no nosso jogo e principalmente para atacarmos problemas reais. E não ficarmos chutando onde está o erro e como poderíamos resolver ele.

Sempre que possível utilize ele para melhorar a performance dos seus jogos e não ter problemas durante o lançamento deles.