# Módulo 26 - Otimizando seu Projeto

Olá alunos, a seguir teremos 10 questões de múltipla-escolha relacionadas as dicas passadas nas aulas do módulo. O questionário tem como objetivo reforçar o aprendizado.

**Todas as questões possuem apenas 1 resposta correta.**

### Questão 1

Levando em consideração que fazer o Profiling e otimizar um projeto pode levar bastante tempo. **Qual seria o momento recomendado para se fazer o Profiling e verificar se seu jogo está rodando sem problemas?**

* A - A todo instante
* B - No final do projeto
* C - Sempre que fazer alguma mudança significativa (Ex.: Implementar uma mecânica ou fase nova)
* D - Uma vez por mês

### Questão 2

Quando falamos de otimização da CPU, é importante entendermos o que é cada parte do frame selecionado, para conseguirmos saber o que está acontecendo na execução do jogo. **No Profiler da CPU, em modo Timeline, qual é o item que representa o ciclo principal de execução do meu jogo?**

* A - WaitForTargetFPS
* B - Gfx.WaitForPresentOnGfxThread
* C - Gfx.WaitForCommands
* D - PlayerLoop

### Questão 3

Garbage Collector é um gerenciador de memória automático do C# que executa quando o sistema verifica que possui muito lixo na memória que precisa ser coletado. **No Profiler da CPU, em modo Hierarchy, qual função aponta que o Garbage Collector limpou a memória naquele frame?**

* A - GC.Alloc
* B - GC.Collect
* C - GC.Execute
* D - GC.Run

### Questão 4

Na Modelagem 3D, **quando falamos da quantidade de triângulos que um objeto deve ter, o que devemos levar em consideração?**

* A - Poder de processamento gráfico do Dispositivo Alvo (Target Device)
* B - Prioridade do objeto 3D na cena (Ex.: Personagem Principal, NPC, Prop, etc)
* C - Tamanho de espaço de tela que o modelo vai ocupar na maior parte do tempo.
* D - Todos os itens anteriores.

### Questão 5

Quando falamos de Textura um dos principais pontos de otimização é a compressão utilizada. **Quando estamos trabalhando com jogos para PC, quais as compressões recomendadas?**

* A - DXT1, ASTC, DXT5
* B - DXT1, BC7, DTX5
* C - ETC, ETC2, DXT5
* D - PVRTC, BC7

### Questão 6

Quando lidamos com Textura, incluindo Sprites, é importante manter o tamanho sempre em potência de 2 (64, 128, 512, 1024, etc) pois precisamos disso para poder usar determinadas compressões.  
Porém, **quando trabalhamos com Sprites, podemos desconsiderar o tamanho em potência de 2 por Sprite ao usar um:**

* A - Sprite Atlas
* B - Sprite Editor
* C - Sprite Mesh
* D - Sprite Material

### 

### Questão 7

Batching é um processo onde a Unity agrupa objetos com semelhanças para serem desenhados em apenas um comando (um draw call). **Qual é a principal semelhança que os objetos 3D precisam ter para serem selecionados para um Batching?**

* A - Mesma Mesh (Malha 3D)
* B - Mesmo Material
* C - Mesmo Material e Textura
* D - Mesmo Material e Mesh (Malha 3D)

### Questão 8

Arquivos de Áudio, em geral, antes de serem executados precisam ser descomprimidos, cujo precisa de um processamento da CPU e pode ser um problema para áudios muito frequentes, mesmo que sejam pequenos. **Qual opção deve ser usada para evitar esse processamento constante de descompressão do áudio na execução?**

* A - Force to Mono
* B - Compressão Vorbis
* C - Decompress on Load
* D - Compressed on Memory

### Questão 9

Quando trabalhamos com UI (Interface do Usuário) na Unity, utilizamos o Canvas. O Canvas é redesenhado sempre que uma mudança é feita em algum elemento dentro dele. **O que devemos fazer para diminuir a frequência com que o Canvas é atualizado?**

* A - Ter um Canvas para cada contexto de UI, com uma distribuição eficiente dos elementos de UI.
* B - Criar um Canvas para cada elemento de UI
* C - Usar apenas um Canvas para toda a UI
* D - Usar um Canvas com Render Mode configurado para Screen Space - Overlay

### 

### Questão 10

O Animator é uma ferramenta muito poderosa que facilita o controle dos estados de animação e possibilita o reaproveitamento dessa máquina de estados em outros objetos. Porém ele possui um custo padrão de processamento maior.

Além disso, quando utilizamos o Animator na UI, mesmo com uma animação vazia, ele força o Canvas a ser redesenhado a todo instante.

**Com isso em mente, o que é visto como uma boa prática para diminuir o uso da CPU nas animações?**

* A - Não usar o Animator. Usar somente o Animation para animações.
* B - Usar um Animator com Apply Root Motion ativado
* C - Separar os Estados em diversas Layers dentro do Animator
* D - Usar o Animation para animações simples