



**Certified**

**Expert**

**Technical Artist:  
Rigging &  
Animation**

# Objetivos do Exame

Especialista Certificado Unity  
Artista Técnico:  
Rigging & Animação

# Função

O Artista Técnico de Rigging e Animação é o elo vital entre game artists e programadores, entregando assets para integração sistemática em jogos, respeitando a visão artística e as limitações da plataforma. Os assets do Artista Técnico são flexíveis o suficiente para se adaptarem às mudanças nas necessidades de uso e nas experiências de jogo, além de responderem às necessidades da evolução do usuário. As habilidades aprofundadas de Rigging e Animação dos Artistas ajudam a melhorar a pipeline de produção do jogo ao fornecer auxílio com animação, desenvolvimento do jogo e criação de personagens.

Além disso, a equipe de desenvolvimento do jogo depende das habilidades essenciais dos Artistas Técnicos de Rigging e Animação para criar animações complexas e GameObjects, incluindo áudio e animação. Os artistas também criam e dão suporte a ferramentas da pipeline, otimizando assets para várias plataformas, e garantindo que estejam de acordo com o Technical Design Document (TDD) do jogo.

## Títulos de trabalho para esta função

- Artista Técnico
- Rigger
- Animador Técnico
- Designer Técnico de Personagens

# Pré-requisitos

Esta certificação de especialista é recomendada para pessoas trabalhando há muitos anos neste ramo, com grande experiência em aplicação prática avançada, por exemplo:

- Experiência em um estúdio de desenvolvimento de jogos, com pelo menos dois títulos lançados
- Conhecimento de scripting e programação em linguagens como C++, C# ou Unityscript
- Experiência com o ciclo completo de produção de um jogo, desde o conceito inicial até o lançamento
- Experiência com rigging/configuração de personagens e animação
- Familiaridade com linguagens de script para criação de conteúdo digital (CCD), como Python, MEL e MaxScript
- Compreensão das pipelines de animação de um jogo, incluindo configuração de personagens e ambientes
- Grande capacidade de organização ao lidar com estruturas de arquivos, convenções de nomenclatura e protocolos estabelecidos
- Domínio de ferramentas de criação de assets, como Adobe Creative Suite, Substance Designer, Substance Painter, Quixel Suite, Autodesk Maya e 3ds Max, Pixologic ZBrush, Motion Builder, etc.

# Habilidades essenciais

A certificação de Artista Técnico Especialista: Rigging & Animação verifica que os candidatos tenham as habilidades necessárias para integrar de forma eficiente os assets de rigging e animação a um jogo. Os candidatos bem sucedidos têm conhecimento avançado nas seguintes áreas.

## Prototipagem

- Avaliar um Game Design Document (GDD) para determinar as ferramentas do Editor voltadas para animação que permitirão à equipe de design construir o jogo e "manter o design"
- Criar e avaliar protótipos para desenvolver melhores práticas e refinar o Technical Design Document (TDD) de acordo com as especificações de desempenho da plataforma
- Avaliar e recomendar soluções técnicas para problemas de rigging e animação

## Pipelining

- Personalizar e automatizar a importação de assets
- Modificar proceduralmente e variar os atributos de GameObjects
- Controlar os parâmetros de vários GameObjects
- Implementar comportamentos e animações proceduralmente

## Preparar GameObjects

- Preparar Prefabs de assets com level of details (LOD) para implementação no jogo
- Implementar e mapear nas opções de Rig os tipos de animação Humanoid e Generic
- Fazer rig e script de montagens complexas usando Joints, Cloth, Rigidbodies e componentes de física para atribuição como Prefabs
- Criar e testar Physic materials personalizados para facilitar a jogabilidade
- Avaliar e otimizar Physic Materials, Mesh e Compound Colliders

## Preparar Animações

- Criar Blend Trees em Máquinas de Estados
- Criar scripts complexos de alto desempenho (múltiplas camadas com estados ativos) em Máquinas de Estados
- Criar camadas de animação na Máquina de Estados
- Criar estados e comportamentos para fazer avaliação e transição de Blend Shapes
- Configurar áudio/FX frame-based em clipes de animação

## Desempenho e Otimização

- Entender as especificações e limitações da plataforma alvo
- Entender as diferenças entre um rig IK e FK e seus impactos no desempenho
- Testar e otimizar montagens complexas de acordo com os requisitos da plataforma
- Avaliar o desempenho da cena e identificar gargalos usando o profiler
- Avaliar otimização de CPU e GPU no que diz respeito à complexidade do rig, batching e métodos de vertex shader

# Tópicos do Exame de Certificação

---

## Tooling e Pipeline

- Personalização do Editor
  - Personalização de asset
  - Automação de processos com ferramentas personalizadas
- 

## Prototipagem

- Prototipagem de rigging e animação
- 

## Animação e Rigging

- Configuração de máquinas de estados para sistemas de animação e eventos de animação
  - Configuração de rig e animação
  - Componentes de física para animação dinâmica
- 

## Desempenho

- Otimização de cena

# Exemplos de questões

## Questão 1

Um Programador de Jogabilidade está criando um protótipo de um jogo de tiro side scrolling. O personagem principal é um avião e os inimigos são tipos diferentes de Óvnis. Quando o jogador destrói um Óvni, um efeito de explosão é exibido no lugar do Óvni. Quando o jogador morre, um efeito de explosão especial da nave do jogador é exibido.

O avião pode disparar até 64 tiros na tela. O número máximo de Óvnis de qualquer tipo exibido na tela é 128. O número máximo de tiros de Óvnis na tela é 1024. O número máximo de efeitos de explosão na tela é 128.

O Programador precisa determinar quais GameObjects devem ser colocados no editor e quais devem ser gerados em tempo de execução (por exemplo, de um pool de objetos).

### Como isso deve ser feito?

- A** O jogador, os Óvnis, o efeito de explosão do jogador e dos Óvnis devem ser colocados no editor. Os tiros do jogador e dos Óvnis devem ser gerados em tempo de execução.
- B** O jogador, os tiros do jogador, os Óvnis e os tiros dos Óvnis devem ser colocados no editor. O efeito de explosão do jogador e dos Óvnis devem ser gerados em tempo de execução.
- C** O jogador e os Óvnis devem ser colocados no editor. Os tiros do jogador e dos Óvnis e os efeitos de explosão do jogador e dos Óvnis devem ser gerados em tempo de execução.
- D** O jogador e o efeito de explosão do jogador devem ser colocados no editor. Os Óvnis, os tiros dos Óvnis e do jogador e o efeito de explosão dos Óvnis devem ser gerados em tempo de execução.

# Questão 2

O Game Design Document (GDD) é um jogo de mundo aberto em terceira pessoa. O GDD especifica dois tipos de jogabilidade para o jogador principal:

1. Andar pelo mundo a pé
2. Dirigir uma motocicleta

A altura relativa de cada modo é similar, mas o jogador vai muito mais rápido com a motocicleta do que a pé.

O Programador de Jogabilidade determina diversas áreas do jogo que devem ser ajustadas durante a troca de andar para pilotar.

- Field of View (FOV) da câmera
- Distâncias de Level of Details (LOD)
- Level streaming

**Quando o jogador está na motocicleta, qual estratégia o Programador deve usar para lidar com cada área neste GDD?**

- A** O FOV da câmera deve ser um valor menor.  
O level streaming deve ser muito mais rápido.
- A** O FOV da câmera deve ser um valor maior.  
O level streaming deve ser muito mais lento.
- C** As distâncias de LOD devem ser um valor maior.  
O level streaming deve ser muito mais lento.
- D** O FOV da câmera deve ser um valor maior.  
O level streaming deve ser muito mais rápido.



# Questão 3

O GDD se passa em uma grande cidade. O jogador pode andar por ela e interagir com qualquer um. O jogador pode concluir várias missões aleatórias para os cidadãos. O jogador pode ter apenas cinco missões ativas por vez. Para andar de uma ponta da cidade para a outra, o jogador precisa de aproximadamente quatro horas.

O jogo contém um minimapa visto de cima no Heads-Up Display (HUD) que mostra um ícone 2D de cada um dos seguintes:

- localização do jogador
- todas as missões ativas
- todos os cidadãos andando pela cidade

É possível aumentar e diminuir o zoom no minimapa. No zoom máximo, apenas o ícone do jogador é visível. No zoom mínimo, 25% da cidade é visível. Há uma lentidão que acontece apenas quando o jogador aumenta o zoom no minimapa ao máximo.

## Qual é uma possível causa para essa lentidão?

- A** Há muitas requisições para descarregar todas as texturas de ícones dos cidadãos da memória.
- B** Os ícones de missões ativas estão sendo consultados com muita frequência para saber quantos estão ativos.
- C** O algoritmo de culling do minimapa está levando muito tempo para remover grande parte da cidade.
- D** O HUD é muito complicado e está sendo recriado a cada quadro.

# Questão 4

O GDD é um jogo para dispositivos móveis com pouca memória disponível. O jogo tem seções carregadas em AssetBundles para garantir um bom desempenho. Cada seção é independente das demais, por isso AssetBundles podem ser carregados e descarregados sem afetar outra seção.

No entanto, texturas rosas aparecem às vezes.

## Qual pode ser a causa desse problema?

- A** AssetBundle.mainAsset está quebrado.
- B** AssetBundle.Unload() está sendo chamado muito cedo.
- C** AssetBundle.LoadAllAssets() está sendo chamado com o Tipo errado.
- D** AssetBundle.LoadAssetWithSubAssets() está sendo chamado com Tipo e string errados.

# Questão 5

O GDD é um jogo de quebra-cabeça de combinação 3D para dispositivos móveis com mais de 300 níveis. O jogo contém os seguintes eventos personalizados de Analytics para cada nível:

1. "levelStarted": Desencadeado quando um jogador começa o nível
2. "levelCompleted": Desencadeado quando um jogador conclui o nível

A equipe quer que você determine quais são os cinco níveis mais difíceis do jogo. Os níveis têm limite de tempo.

**Qual evento personalizado adicional seria necessário para determinar os cinco mais difíceis?**

- A** Adicionar um "levelTime" para enviar o tempo gasto no nível.
- B** Adicionar um "levelRestarted" para ser desencadeado quando o aplicativo móvel é fechado pelo sistema operacional.
- C** Adicionar um "levelFailed" para ser desencadeado quando um jogador falha ou sai do nível mais cedo.
- D** Adicionar um "levelResumed" para ser desencadeado quando o aplicativo móvel for retomado dos processos em segundo plano.

---

Respostas corretas: C, D, C, B, C