



ALÉM DE JOGAR,
EU FAÇO JOGOS!

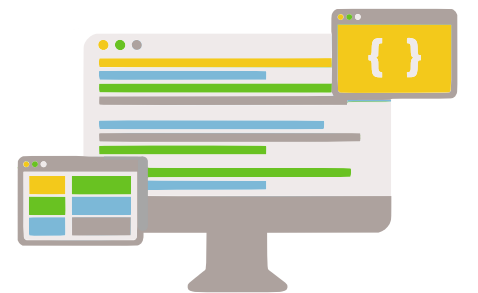


Material desenvolvido pela professora Juliana Oliveira para o projeto além de jogar eu faço jogos. Divulgação proibida.

O que vamos aprender hoje?



Sistema de partículas





O sistema de física LiquidFun permite simular torneiras, piscinas, ondas, riachos e outros efeitos incríveis baseados em física semelhantes aos apresentados em jogos movidos a Corona como Freeze! 2 – Irmãos ou outros jogos onde o mundo da física não é composto inteiramente de objetos rígidos.

Além dos efeitos líquidos básicos, podemos implementar objetos elásticos/gelatina semilíquidos, criar líquidos que se misturam quando entram em contato, detectar e manipular "regiões" dos líquidos e muito mais.

FREEZE! 2

BROTHERS



Sistema de partículas



No liquidFun, todos os líquidos são representados por um sistema de partículas, o que significa que uma corrente ou piscina é composta por muitas partículas que interagem entre si para simular o efeito desejado. Cada partícula é redonda e são a unidade mínima de matéria em um sistema de partículas. Por padrão, uma partícula se comporta como um líquido, mas podemos alterar o comportamento de partículas individuais ou grupos de partículas. Podemos também definir propriedades de partículas como posição, velocidade e cor.

Para criar um objeto de partículas utilizamos a função `physics.newParticleSystem()`. Nesta API, há uma tabela de parâmetros com apenas um elemento obrigatório: o nome do arquivo (`filename`) da imagem da partícula a ser renderizada para cada instância da partícula



No entanto, para fazermos o sistema de partículas se comportar melhor, devemos especificar algumas propriedades adicionais. No mínimo, devemos definir a partícula radius para controlar o tamanho das partículas que constituem o efeito líquido. Outro parâmetro útil é imageRadius que, se definido, instruirá o Corona a renderizar a imagem da partícula em um tamanho diferente do corpo físico definido por radius. Essencialmente, se imageRadius for ligeiramente maior que radius, as partículas se sobreporão um pouco, resultando em uma aparência líquida mais coesa.



Uma vez que temos um sistema de partículas estabelecido, novas partículas são geradas usando o método `object:createParticle()` . Isso aceita uma tabela de parâmetros opcionais que controlam o comportamento de cada partícula gerada.



Os comportamentos das partículas são definidos para grupos inteiros de partículas ou para partículas individuais.

- Sólido - Um grupo de partículas sólidas impede que outros corpos se alojem dentro dele. Caso algo penetre nele, o grupo de partículas sólidas empurra o corpo agressor de volta à sua superfície. Um grupo de partículas sólidas também possui uma força repulsiva especialmente forte. É útil, por exemplo, em um caso em que:
 - Deve-se esperar que algo salte com vigor incomum, como quando uma bola de squash atinge a parede de uma quadra

Comportamento de partículas



- Rígido - Grupos de partículas rígidas são aqueles cuja forma não muda, mesmo quando colidem com outros corpos. Trabalhar com grupos de partículas rígidas oferece algumas vantagens sobre simplesmente trabalhar com corpos rígidos: Com um grupo de partículas rígidas, você pode:
 - Exclua parte do grupo (ou seja, algumas de suas partículas).
 - Por exemplo, disparar uma bala que deixa um buraco em um grupo de partículas em forma de caixa.
 - Mescle-o com outros grupos.
 - Por exemplo, criando um boneco de neve a partir de três grupos de partículas redondas e, em seguida, mesclando-os em um único grupo de partículas.
- Elástico - Partículas elásticas se deformam e também podem saltar quando colidem com corpos rígidos.

Comportamento de partículas



- Mistura de cores - As partículas de mistura de cores assumem um pouco da cor de outras partículas com as quais colidem. Se apenas uma das duas partículas em colisão for uma mistura de cores, a outra partícula retém sua cor pré-colisão.
- Em pó - As partículas de pó produzem um efeito de dispersão, como você pode ver com areia ou poeira.

É importante observar que `object:createParticle()` cria apenas uma partícula no sistema geral, portanto, para criar um cenário útil, precisamos gerar várias partículas chamando essa função em um temporizador de repetição ou por meio de algum outro método de repetição.



Além de criar partículas únicas em um sistema, podemos criar grupos de partículas relacionadas usando a função `object:createGroup()`. Essencialmente, isso "preenche" uma região definida com várias partículas em um comando e o grupo pode receber várias propriedades comportamentais e visuais.



A força linear pode ser aplicada ao centro de todas as partículas em um sistema de partículas por meio da API `object:applyForce()`. Este método aceita dois valores numéricos que indicam a quantidade de força a ser aplicada nas direções x e y, respectivamente.