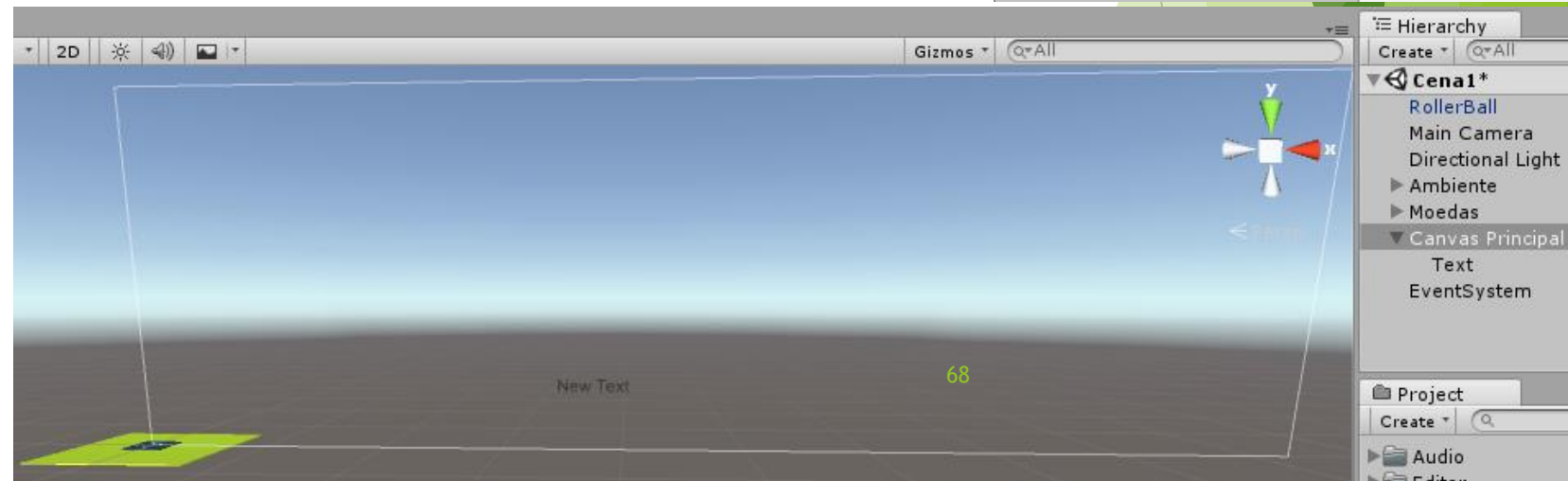
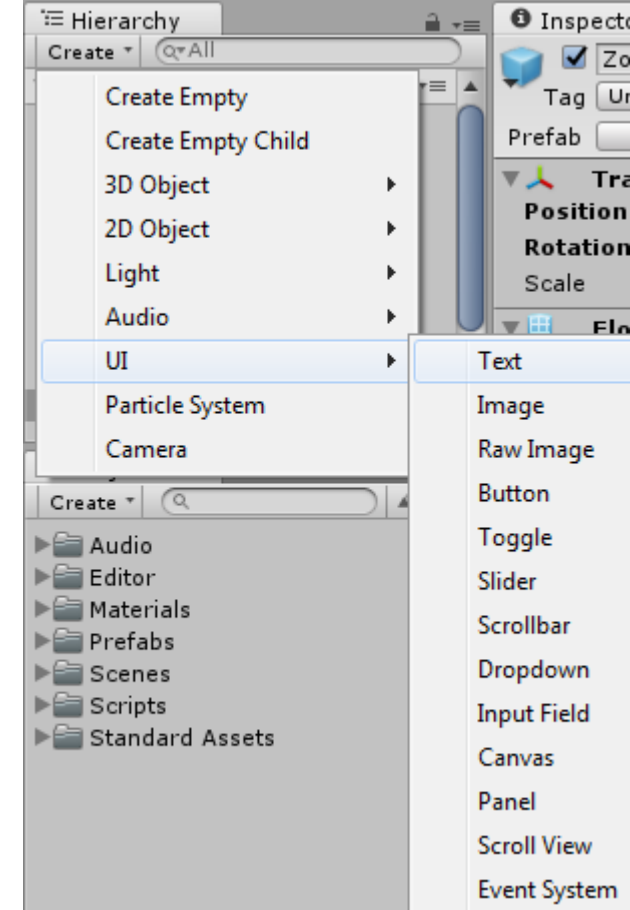


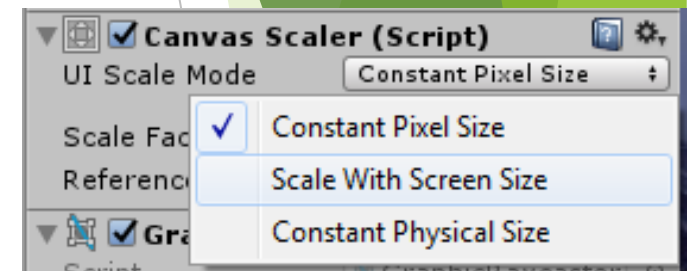
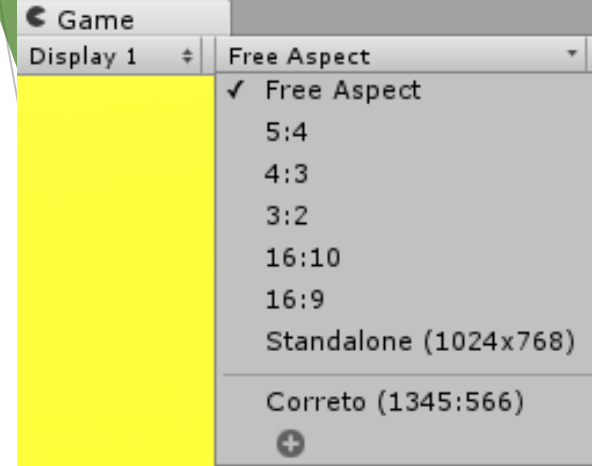
Roller Madness - UI Básica

- ▶ Agora que temos um sistema de coletas, precisamos de uma maneira de exibir os pontos obtidos pelo jogador.
- ▶ Também é aconselhável ter uma mensagem de Game Over quando ocorrer a morte do jogador ao invés de simplesmente reiniciar a cena.
- ▶ Faremos as duas coisas criando uma interface com usuário básica para nosso jogo.
- ▶ Para criar uma interface com usuário (UI) no jogo, acesse o botão [+] da Hierarquia, selecione UI e Text, para criarmos uma exibição de texto para nossas moedas.
- ▶ Após isso ter feito, Unity criou um objeto Canvas e um item filho chamado Text, além de um objeto EventSystem.
- ▶ Canvas define o espaço 2D onde os elementos de UI são gerados.
- ▶ Mude o nome do Canvas para Canvas Principal.



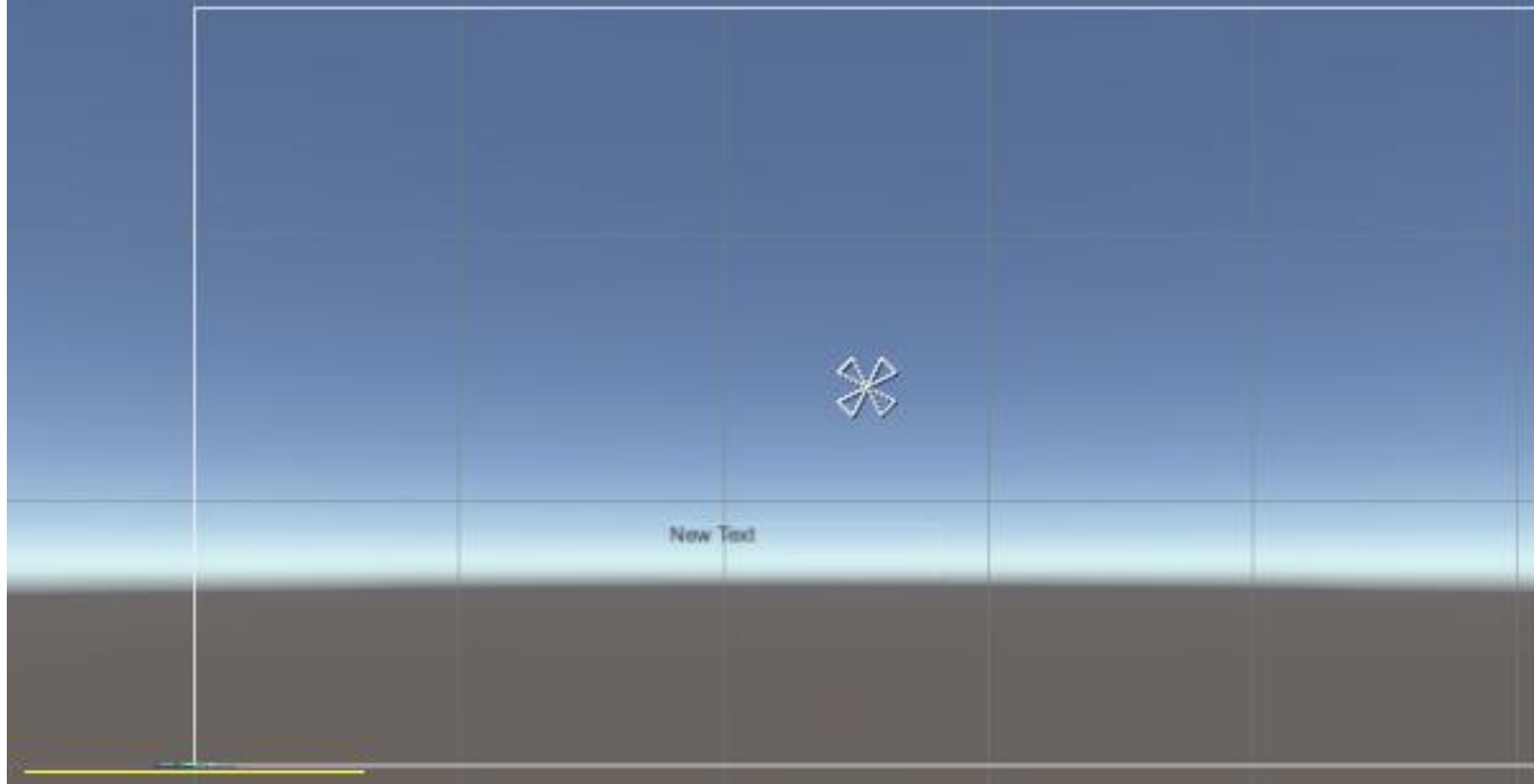
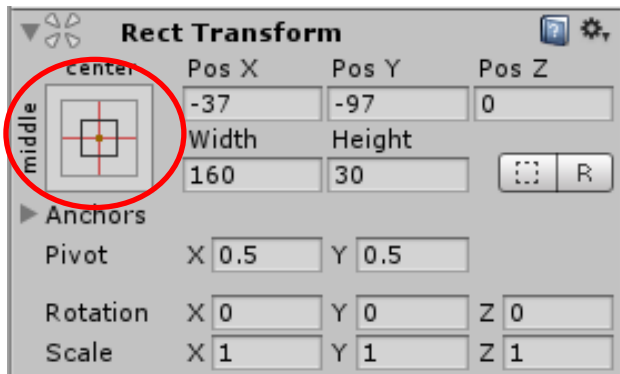
Roller Madness - UI Básica

- ▶ No Inspector do Canvas Principal há vários itens, uns dos quais é o script CanvasScaler, que define como o Canvas é escalado em função do tamanho da tela de jogo.
- ▶ Atualmente estamos usando a proporção Free Aspect mas talvez você queira usar a proporção 16:9, que exibirá mais coisas na tela.
- ▶ Além disso, vamos mudar o UI Scale Mode de “Constant Pixel Size” para “Scale with Screen Size”.
- ▶ Renomeie o objeto Text para txtScore e pressione [F] ou clique duas vezes sobre ele para aproximar a visualização desse objeto.
- ▶ Observe que ele está na cena, mas distante do objeto Chao, onde nossa ação ocorre. Também possui a string “New Text” sendo exibida. Estamos vendo esse objeto de texto no espaço 3D. No entanto, ele é um objeto bidimensional.
- ▶ No topo da janela Scene, clique a opção 2D. Faça zoom-out e note que o txtScore não está alinhado corretamente no Canvas.
- ▶ Temos que alinhá-los de forma apropriada.



Roller Madness - UI Básica

- No Inspector de txtScore, note no topo um pequeno diagrama na propriedade Rect Transform:

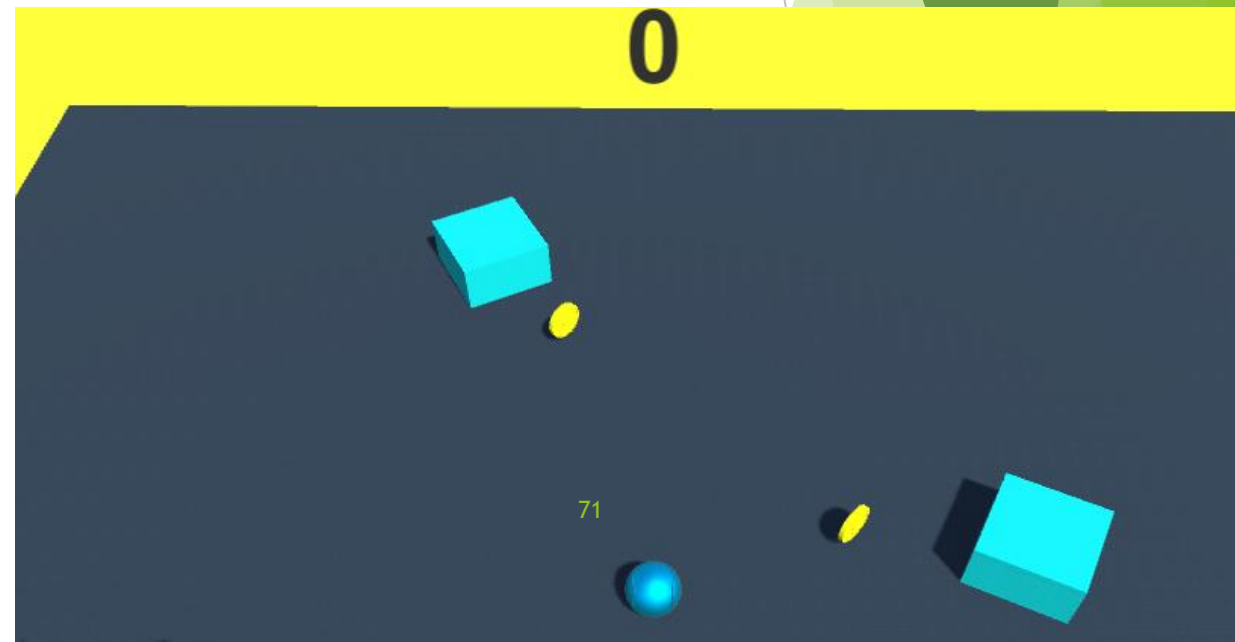


- Clicar nesse quadrado abre as opções de ancoragem.
- No topo, essa janela informa :
 - Shift : Also set pivot
 - Alt : Also set position.
- Isso significa que, se pressionar [Shift] e [Alt], poderá realizar várias coisas enquanto posiciona o txtScore.



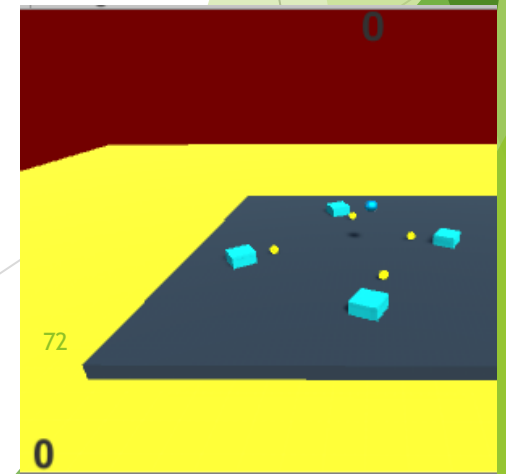
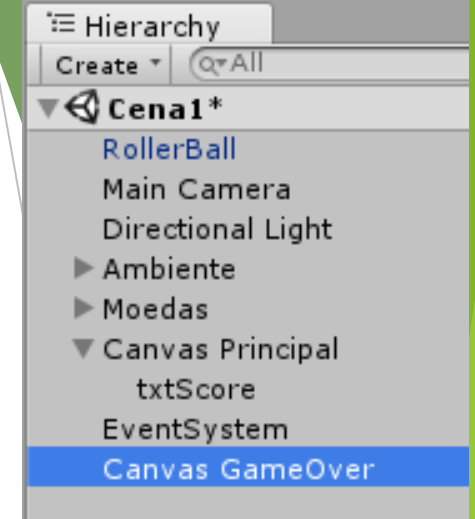
Roller Madness - UI Básica

- ▶ Pressione [Shift], [Alt] e clique no quadro central do topo da janela.
- ▶ Isso fará com que txtScore fique posicionado na parte superior e central do Canvas.
- ▶ Mude o texto para 0 (zero), o tamanho do fonte para 40 e deixe-o como bold.
- ▶ Ele desapareceu da tela porque ficou muito grande para o espaço disponível no txtScore.
- ▶ Uma forma rápida de resolver isso é mudar as propriedades Horizontal Overflow e Vertical Overflow de Wrap para Overflow.
- ▶ Se o txtScore não aparecer, clique na Camera e desligue o script SmoothFollow, ajuste a Camera e ligue o script novamente. Você também pode mudar Scale na janela Game.



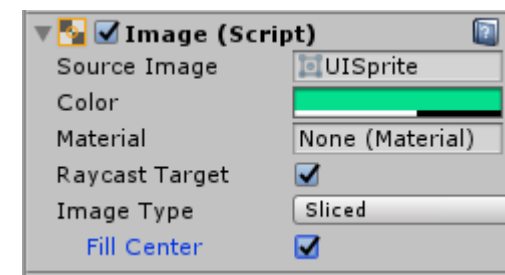
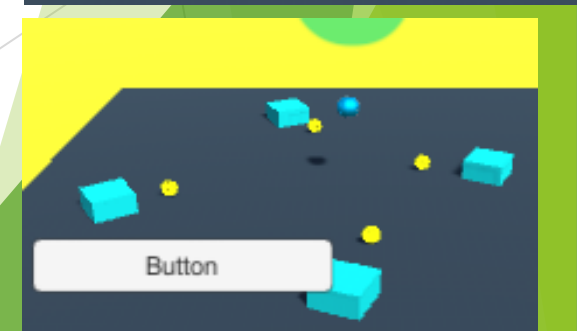
Roller Madness - UI Básica

- ▶ Vamos também criar uma mensagem de Game Over. Para isso, na janela Hierarquia, clique em [+] | UI | Canvas e criará um novo Canvas.
- ▶ Mude o nome desse novo objeto para Canvas GameOver.
- ▶ No Script Canvas Scaler mude a variável UI Scale Mode para “Scale with Screen Size”.
- ▶ Queremos colocar um texto nesse Canvas, portanto com o Canvas GameOver selecionado, clique em [+] | UI | Text e um novo campo de texto será criado como filho do Canvas GameOver.
- ▶ Ao invés de configurá-lo do início, podemos copiar as propriedades do txtScore para esse novo campo de texto.
- ▶ Para isso, abra o Canvas Principal. Clique no txtScore e nos três pontinhos verticais. Selecione a opção Copy Component.
- ▶ Agora, no campo de texto recém-criado, clique na engrenagem e, em seguida, selecione a opção “Paste Component Values”. Os valores do txtScore e suas configurações serão copiados nesse outro texto.
- ▶ Mude o nome desse objeto para txtEndGameScore.



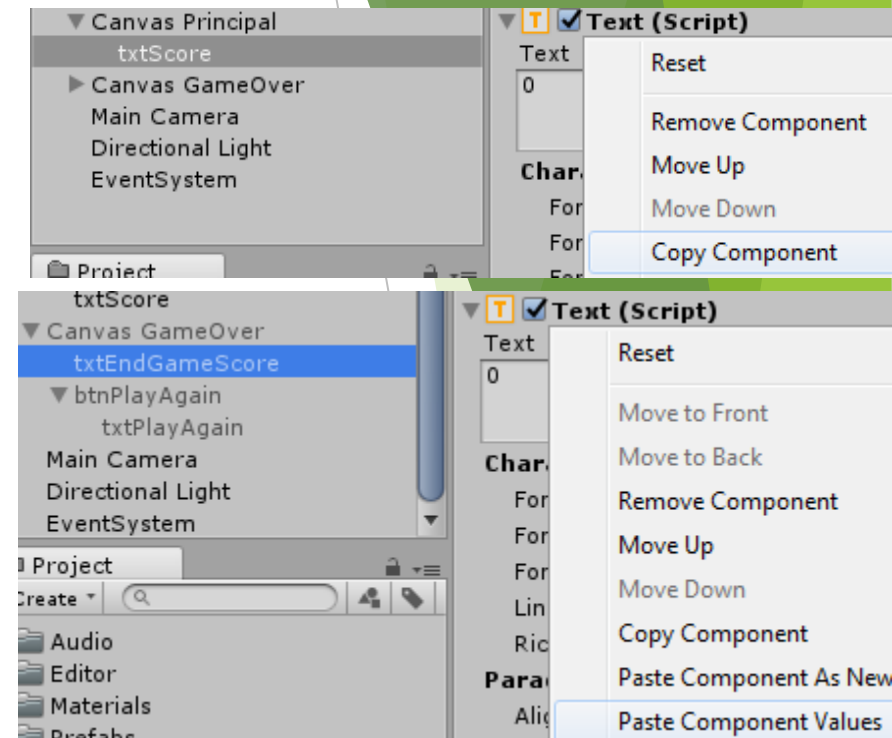
Roller Madness - UI Básica

- ▶ Centralize esse texto, clicando no quadro abaixo de Rect Transform no Inspector e, pressionando as teclas [Shift] e [Alt], clique no quadro central da ancoragem.
- ▶ Mude o tamanho do fonte para 150 e suba a coordenada y do transform do txtEndGameScore para 120. Mude Color para 04DF8D96.
- ▶ Vamos agora criar um botão para que o jogador possa escolher entre terminar o jogo ou reiniciá-lo.
- ▶ Assim, selecione o Canvas GameOver, clique em [+] da Hierarquia, selecione UI e Button.
- ▶ Um botão aparecerá no seu Canvas. Mude seu nome para btnPlayAgain.
- ▶ Vamos centralizar o botão, usando o quadro de ancoragem e as teclas [Shift] e [Alt], clicando na posição central de ancoragem.
- ▶ Se o botão se sobrepor ao txtGameOver desse Canvas, mude a coordenada y do botão para que ele fique abaixo do texto.
- ▶ Mude a cor para 04DF8D96, a mesma cor do texto.
- ▶ Fill Center faz com que o botão seja preenchido.



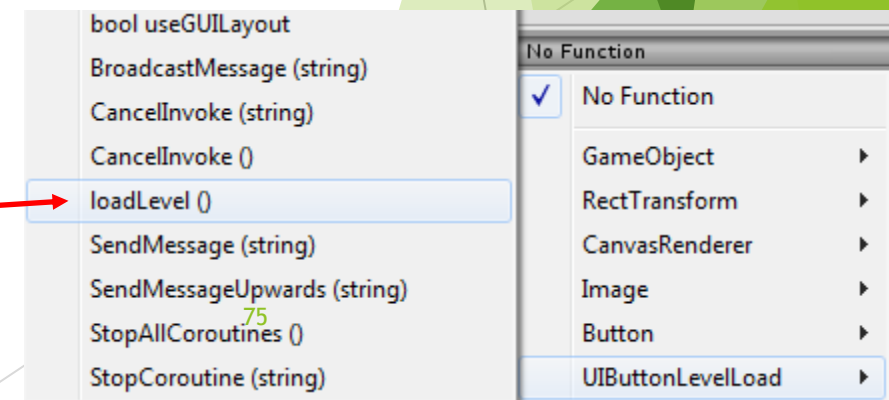
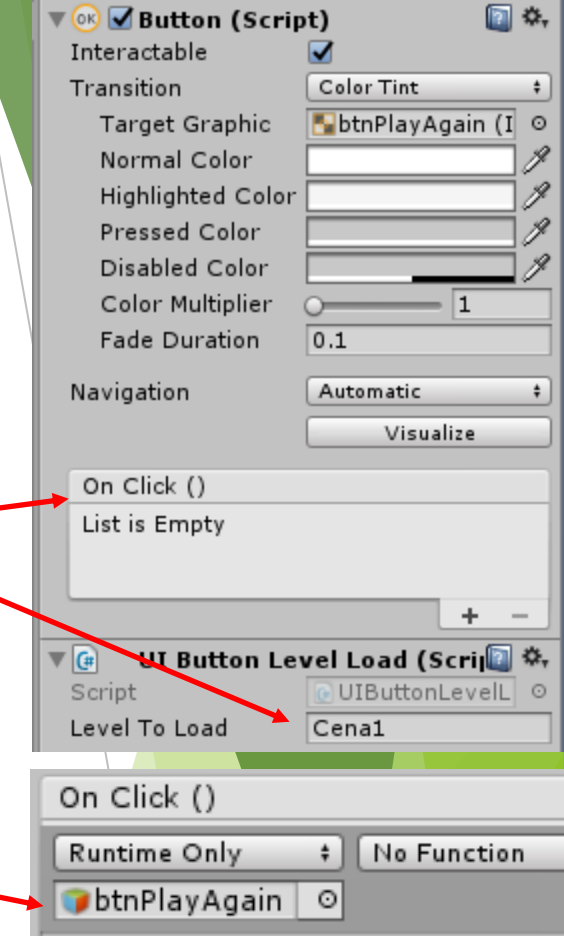
Roller Madness - UI Básica

- ▶ Vamos mudar o texto do botão. Existe um objeto Text subordinado ao botão na Hierarquia.
- ▶ Podemos copiar as propriedades de outro texto sobre esse texto e configurá-lo mais rapidamente.
- ▶ Clique no componente txtScore do Canvas Principal, clique na engrenagem do script Text e selecione Copy Component.
- ▶ Agora, selecione o txtPlayAgain do btnPlayAgain, clique na engrenagem do script Text e selecione Paste Component Values para que o texto fique igual ao do txtScore. Mude o texto exibido para “Play Again” e o tamanho do fonte para 20, por exemplo.
- ▶ Se necessário, mude os alinhamentos e ligue Align By Geometry.
- ▶ Você pode alterar as propriedades Width e Height do botão, na opção Rect Transform, caso o botão não esteja do tamanho adequado.



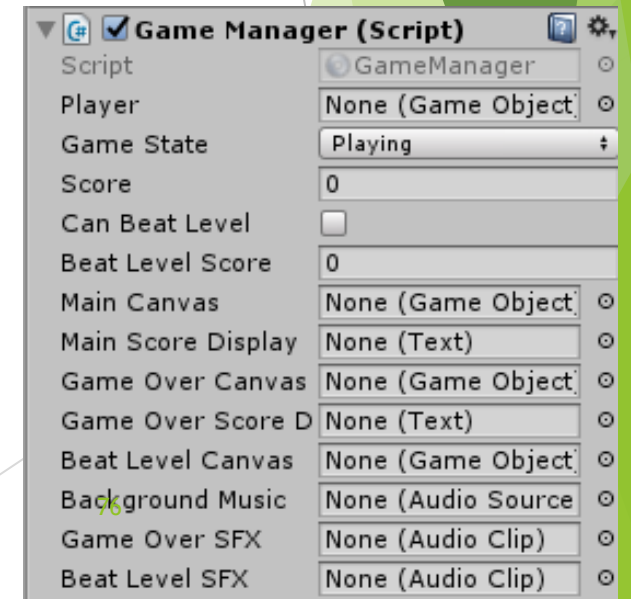
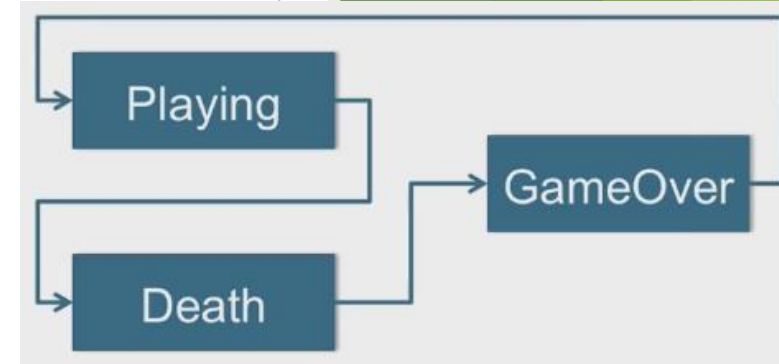
Roller Madness - UI Básica

- ▶ No botão btnPlayAgain, adicione o script UIButtonLevelLoad, que já está disponível em nosso projeto. Há também um script UIButtonQuitGame, para o caso de você desejar parar o jogo (só funcionará fora do Unity).
- ▶ Na variável LevelToLoad, digite o nome da cena atual (cena1) já que é a única que temos no momento.
- ▶ Agora, temos que fazer o botão realmente chamar esse script.
- ▶ Na figura ao lado vemos o editor do evento OnClick(), que é usado para definir que ação será realizada quando clicarmos no botão.
- ▶ Clique no botão [+] e arraste o botão btnPlayAgain da Hierarquia para a caixa abaixo da lista com as palavras Runtime Only.
- ▶ Isso trará o objeto com todas suas propriedades para esse componente e poderemos escolher o que fazer. Note que há uma lista com as palavras No Function, indicando que nenhuma ação foi determinada ainda.
- ▶ Clicando nessa lista, aparecerão os componentes.
- ▶ Escolha UIButtonLevelLoad e LoadLevel().
- ▶ Torne o Canvas GameOver inativo pelo Inspector para que ele não fique visível antes de o jogo terminar.



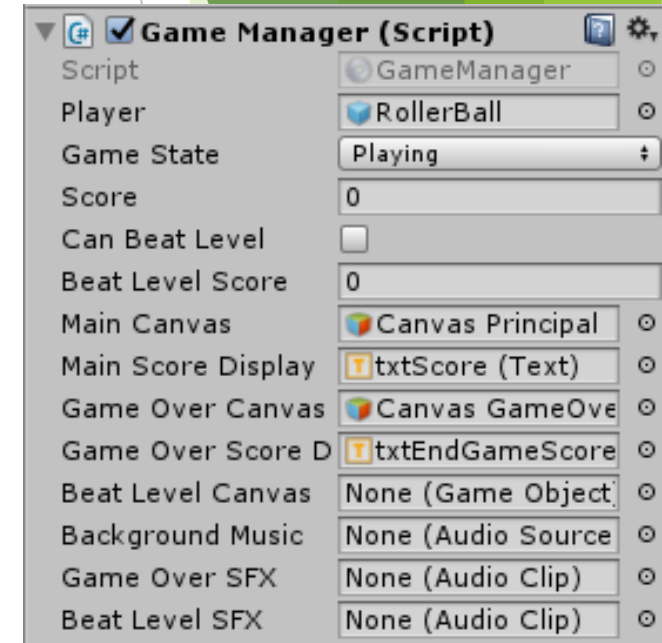
Roller Madness - Game Manager

- ▶ Vamos criar uma classe, chamada GameManager, que controlará os estados do nosso jogo e atualizará os elementos da UI de forma condizente.
- ▶ Este jogo é bastante simples, com apenas 3 estados:
 - ▶ Playing → o jogo começa nesse estado e nele fica até o player morrer.
 - ▶ Death → o jogo vai para esse estado quando o player morre e se alterna para o Canvas GameOver e a música diminui até parar.
 - ▶ GameOver → o jogo transita para esse estado e a música de GameOver toca. Fica nesse estado até que o jogador clica o botão [Play Again], o que leva à transição para o estado Playing.
- ▶ Vamos criar um game object vazio e nele incorporar o componente GameManager.
- ▶ Crie um objeto vazio na Hierarquia e o chame de Game Manager. Arraste-o para o topo da hierarquia e faça o reset de sua posição.
- ▶ Adicione o Componente de Script Game Manager no Inspector desse objeto.



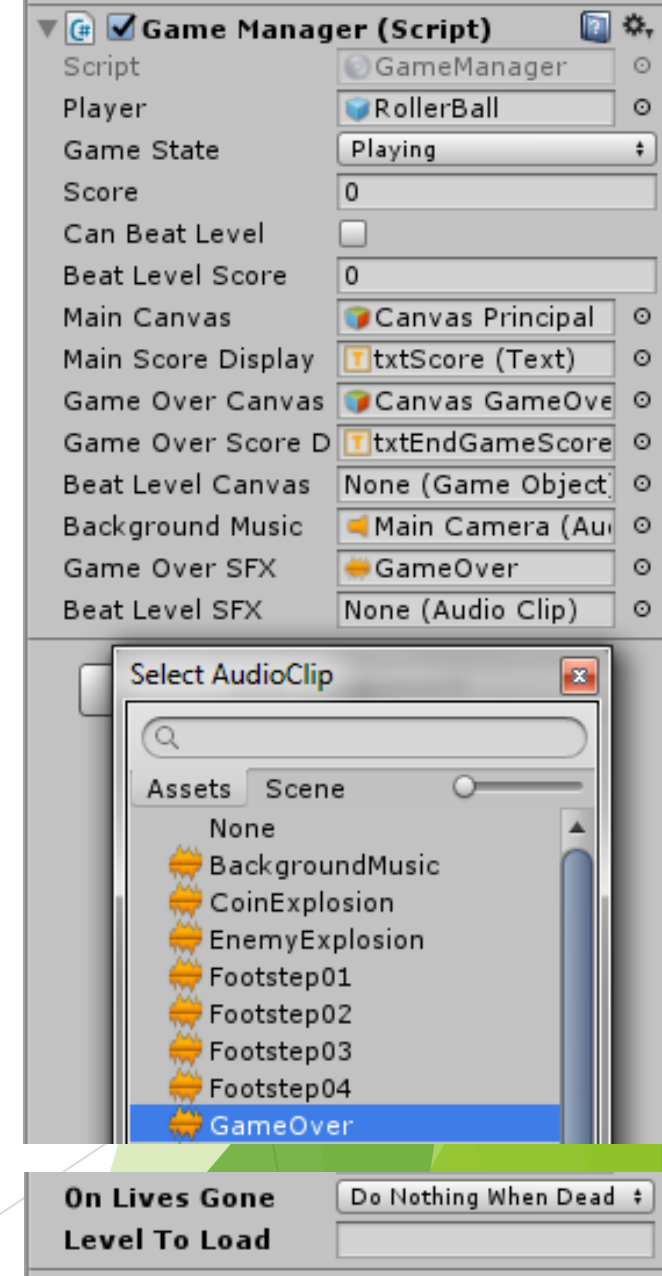
Roller Madness - Game Manager

- ▶ Este é um gerenciador de jogo básico mas que pode funcionar com vários tipos de jogos.
- ▶ Você pode ver as várias propriedades que teremos de configurar.
- ▶ Arraste RollerBall para a variável **Player** do Game Manager.
- ▶ A variável **Score** armazena o valor de pontuação do jogador.
- ▶ **Can Beat Level** igual a true permite que a fase seja ganha quando o jogador atingir a pontuação definida na variável **Beat Level Score**. Por exemplo, CanBeatLevel = true e BeatLevelScore = 5 fará com que mude a fase caso o jogador colete 5 moedas, pois cada moeda vale 1 ponto.
- ▶ **Main Canvas** deve receber nosso objeto **Canvas Principal**.
- ▶ **Main Score Display** deve receber o **txtScore**, para exibir a pontuação.
- ▶ **Game Over Canvas** deve receber nosso objeto **Canvas GameOver**.
- ▶ **Game Over Score Text** deve receber o **txtEndGameScore**.
- ▶ Se tivéssemos a possibilidade de vencer a fase, poderemos ter um Canvas específico para parabenizar o jogador e o indicariamos em **Beat Level Canvas** no Game Manager.



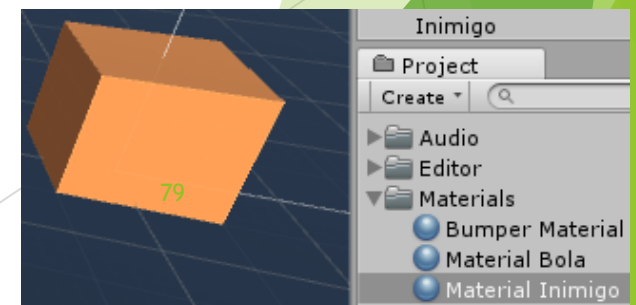
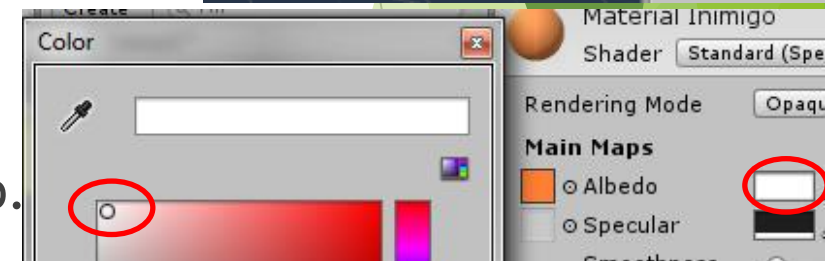
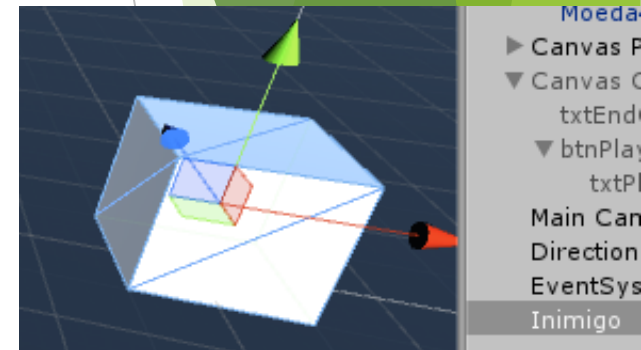
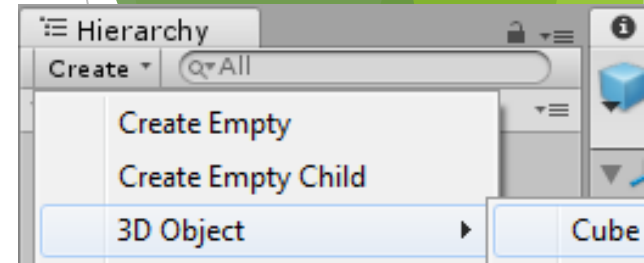
Roller Madness - Game Manager

- ▶ **Background Music** deve receber a fonte de música, ou seja, a **Main Camera**.
- ▶ Como Efeito Sonoro do Game Over (**Game Over SFX**) podemos clicar no pequeno círculo à direita dessa variável e selecionar a música **GameOver**.
- ▶ Agora temos que fazer uma pequena alteração em nosso player, ou seja, no objeto **Rollerball**.
- ▶ Quando adicionamos o script **Health** à **Rollerball**, informamos na variável **On Lives Gone** o que deveria ser feito, ou seja, “**Load Level when dead**” (carregar o nível quando morto) e indicamos **Cena1** como fase a ser carregada.
- ▶ Agora, será o **Game Manager** que cuidará disso e, portanto, vamos indicar em **On Lives Gone** que nada será feito pelo script **Health**, marcando “**Do nothing when dead**” nessa variável.
- ▶ O **Game Manager** detectará que o player morreu e exibirá a UI adequada.
- ▶ Execute e observe o resultado. Quando ocorre a morte, o jogo fala “**Game Over**” e podemos ver a pontuação obtida.



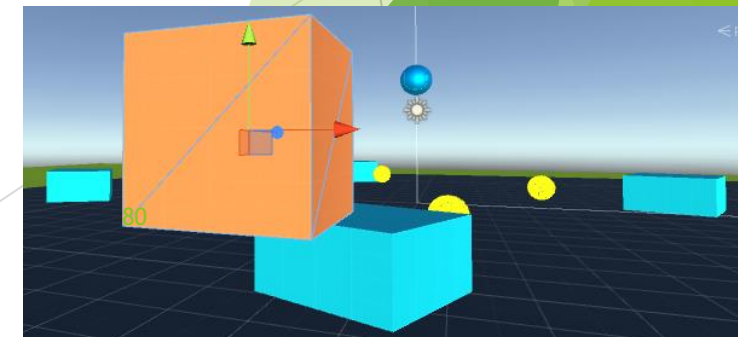
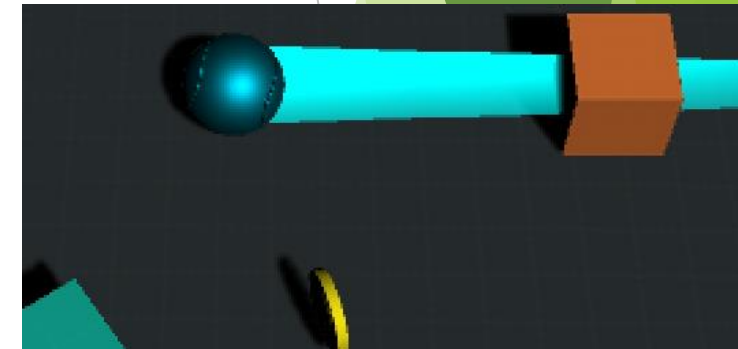
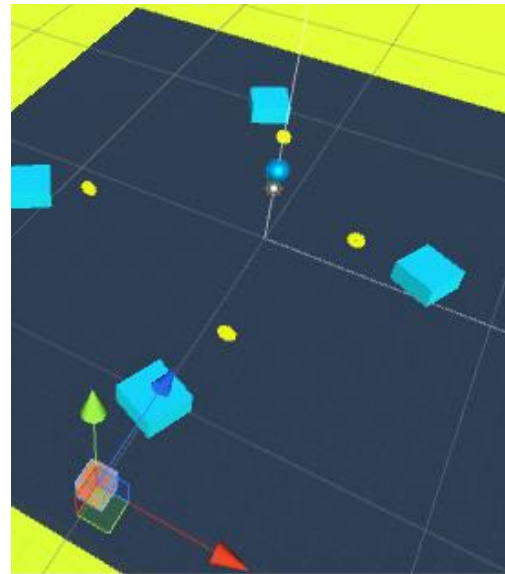
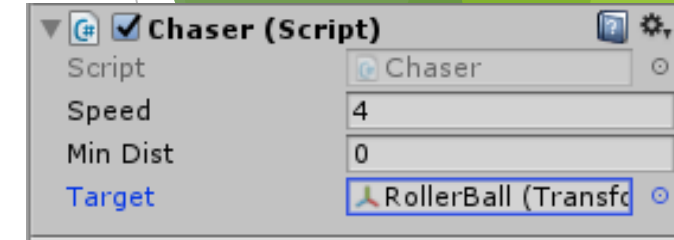
Roller Madness - Inimigos

- ▶ Nosso jogo seria muito mais desafiador e interessante se houvesse inimigos no mundo, que tivessem de ser evitados para sobrevivermos.
- ▶ Vamos aproveitar a criação de inimigos para rever alguns conceitos já estudados.
- ▶ Na hierarquia, crie um novo Cubo, usando [+] | 3D Object | Cube. Faça o reset desse cubo para que ele vá à origem do sistema de coordenadas.
- ▶ Suba-o na direção Y para que fique sobre o Chao e mude seu nome para Inimigo.
- ▶ Na pasta Materials, duplique o “Material ZonaDeMorte”. Mude o nome da cópia para “Material Inimigo”.
- ▶ Na propriedade Albedo do Material Inimigo, troque a cor por SwatchOrangeAlbedo. Troque a cor do quadrinho por branco puro.
- ▶ Arraste o novo Material para o cubo. Nosso inimigo será laranja.



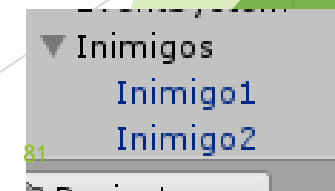
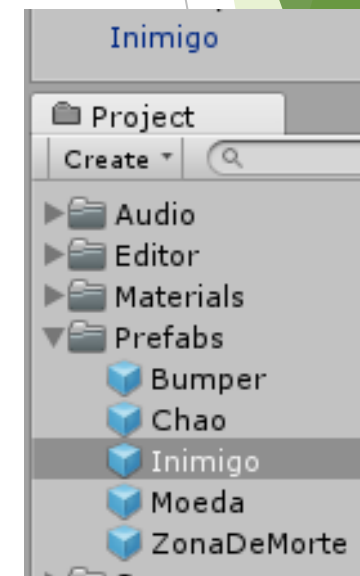
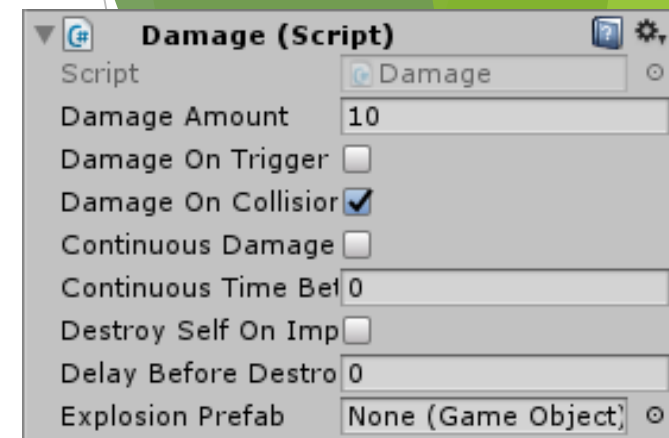
Roller Madness - Inimigos

- ▶ Ao objeto Inimigo, adicione o script Chaser, que o fará perseguir outro gameObject que, no nosso caso, será a Rollerball.
- ▶ O script Chaser possui 3 variáveis: Speed (mude para 4), Min Dist (mude para 0) e Target (arraste a Rollerball para cá).
- ▶ Esse script está escrito de tal forma que, se não houver um Target assinalado, ele procurará os objetos de jogo até encontrar um com tag igual a "Player", passando a seguir esse objeto. Estude seu código.
- ▶ Execute o jogo e perceba que o cubo laranja persegue nossa Rollerball.
- ▶ O inimigo atravessa os bumpers porque ainda não está pronto.
- ▶ Mova o inimigo para longe do player.
- ▶ Adicione um Rigidbody para que ele possa colidir com as coisas.
- ▶ Será afetado pela gravidade e outras forças.
- ▶ Mude o Physical Material do Box Collider para Bouncy e o Y para 2 metros.



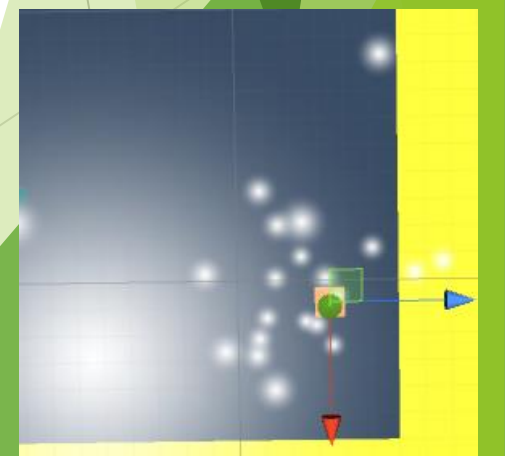
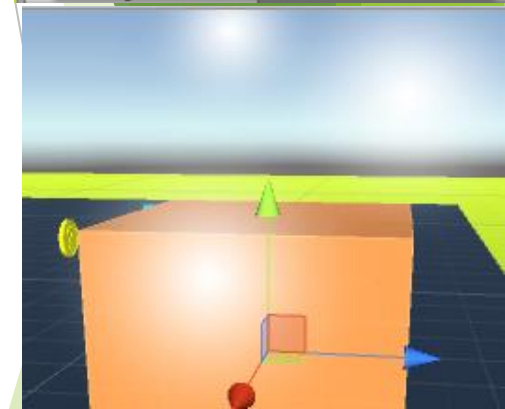
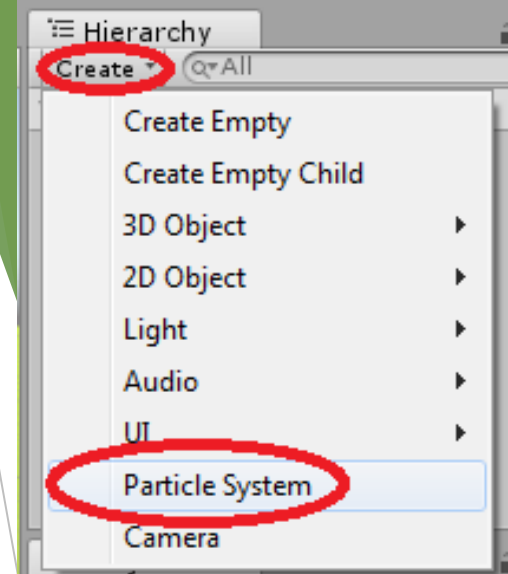
Roller Madness - Inimigos

- ▶ O Inimigo, neste momento, apenas segue nosso player, não infligindo nenhum dano a ele. Assim, vamos adicionar o script Damage ao Inimigo.
- ▶ Damage Amount pode ficar valendo 10, pois 10 é maior que 1, que é a saúde do player e queremos matar o player quando houver contato.
- ▶ Desmarque **Damage On Trigger** e marque **Damage On Collision**.
- ▶ Também queremos que o Inimigo tenha o script Health, para que sua saúde vá decaindo quando ele colidir com algum objeto, outro Inimigo ou cair para fora do Chao, como acontece com o Player.
- ▶ Adicione o Script Health ao Inimigo, deixe os valores default e execute o jogo. Se a velocidade do Inimigo estiver muito alta, você pode diminuí-la um pouco no script Chaser, para 3, por exemplo.
- ▶ Crie um Prefab a partir do objeto Inimigo.
- ▶ A partir do prefab Inimigo, crie uma nova instância do objeto na cena, em uma posição simetricamente oposta em relação ao primeiro.
- ▶ Da mesma forma que fizemos com as Moedas, seria interessante ter um objeto Vazio para armazenar e organizar os inimigos.
- ▶ Execute o jogo e observe que os inimigos se destroem caso colidam.



Roller Madness - Partículas e Efeitos Sonoros

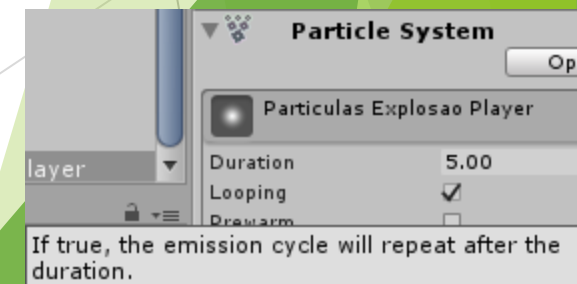
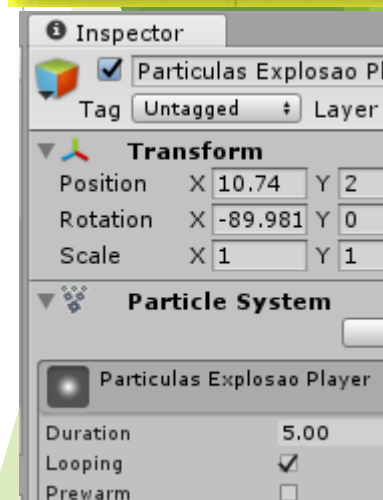
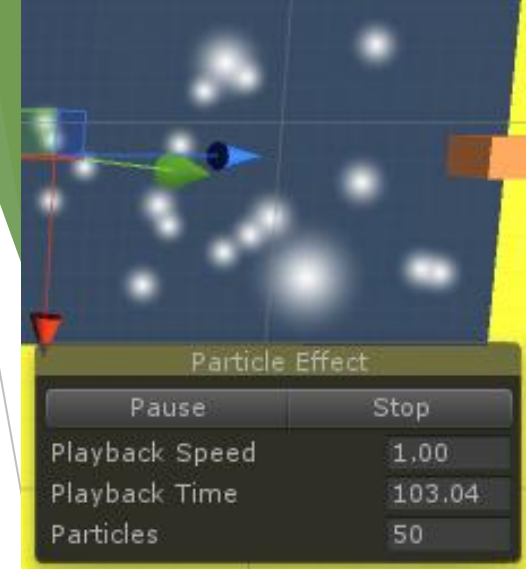
- ▶ Até este momento, temos apenas objetos sólidos em nosso jogo.
- ▶ Um Sistema de Partículas nos permite simular entidades semelhantes a fluidos, tal como fogo, fumaça e nuvens.
- ▶ Os melhores efeitos de partículas são obtidos pela combinação de imagens 2D simples ou malhas 3D movendo-se em conjunto.
- ▶ Vamos configurar, portanto, um sistema de partículas para a explosão de nosso player.
- ▶ Sob a janela Hierarchy, clique no botão [+] e selecione Effects | Particle System.
- ▶ Você verá que aparecerá uma emissão de partículas em nosso mundo de jogo, como um conjunto de bolas brilhantes flutuantes.
- ▶ Renomeie o objeto Particle System para “Particulas Explosao Player”.
- ▶ A melhor maneira de configurar um sistema de partículas é pelo seu uso, ajuste e visualização, até que atinja o padrão desejado.
- ▶ Clicando em outro objeto da cena, o sistema de partículas para de funcionar momentaneamente, até que você clique novamente na sua posição central.
- ▶ Assim, você pode testá-lo interativamente.



Roller Madness - Partículas e Efeitos Sonoros

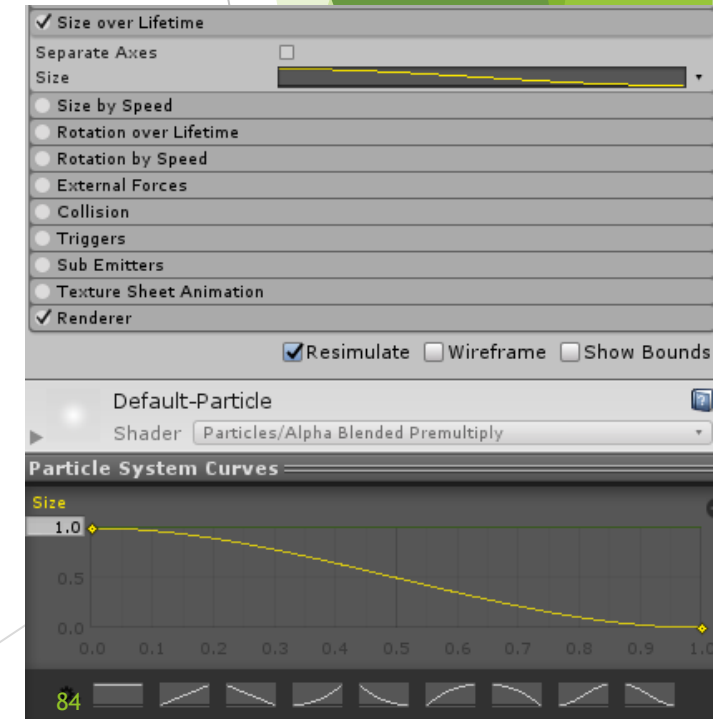
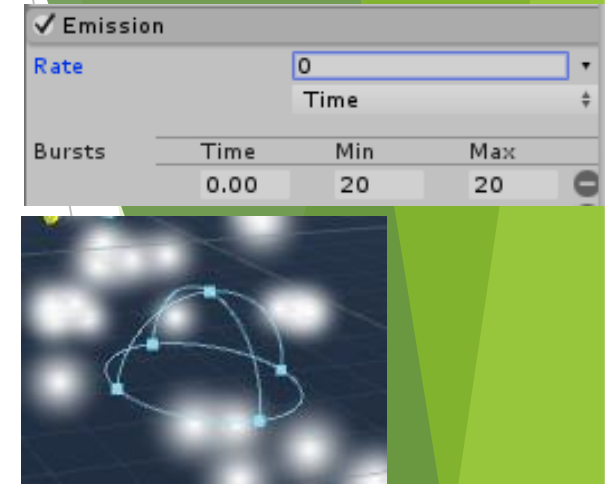
- ▶ Há, também, uma pequena janela de controle onde você pode pausar o sistema, pará-lo completamente, ajustar velocidade, tempo de reprodução e quantidade de partículas geradas.
- ▶ Observe no Inspector que o Sistema de Partículas é apenas um componente em um Game Object e que há várias abas que você pode abrir e fechar e, naturalmente, configurar suas propriedades.
- ▶ Em cada propriedade, há um menu suspenso que será exibido quando o mouse passar sobre o nome da propriedade.
- ▶ Mude a propriedade abaixo conforme definido:
 - ▶ Duration → 2
 - ▶ Looping → false
 - ▶ Start Lifetime → 1
 - ▶ Start Speed → 5
 - ▶ Start Size → 1
 - ▶ Gravity Modifier → 0.5um pouco

Você verá que as partículas sobem e caem ⁸³



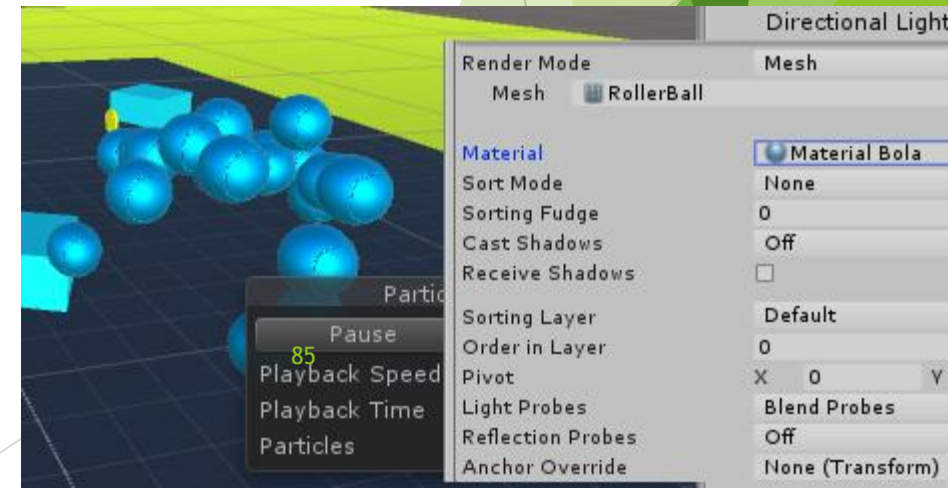
Roller Madness - Partículas e Efeitos Sonoros

- ▶ Em Emission, mude Rate para 0, clique em + e defina Burst com mínimo 20 e máximo 20. Assim, as partículas aparecerão no tempo = 0 todas de uma vez, como uma explosão.
- ▶ A próxima propriedade a alterar é Shape. Ela define o formato do conjunto de partículas. Mude de Cone para HemiSphere, para dar uma ideia mais próxima de uma explosão.
- ▶ Em seguida, selecione Size over Lifetime.
- ▶ Clique no item Size e aparecerá uma janela de configuração de curvas do sistema de partículas, com alguns itens pré-definidos na parte inferior.
- ▶ Você pode escolher a pré-definição mais à direita ou mover a curva até ela ficar semelhante à pré-definição. Ao fazer isso, notará que cada partícula individual começa maior e vai diminuindo até desaparecer.



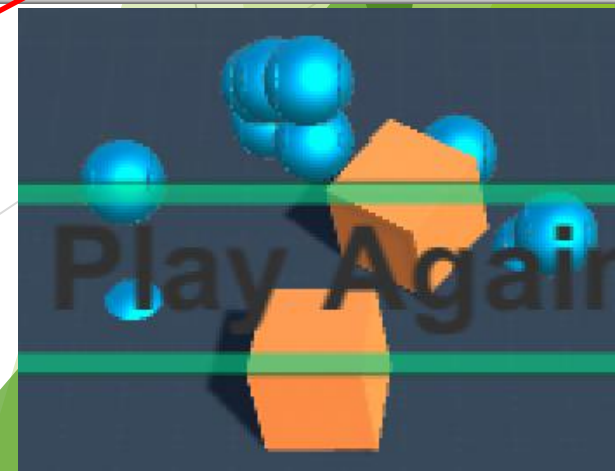
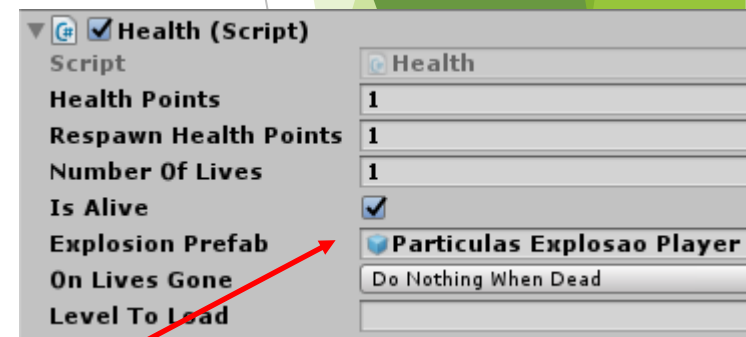
Roller Madness - Partículas e Efeitos Sonoros

- ▶ Renderer define o que será gerado para exibição. No momento, temos essas figuras brancas aparecendo.
- ▶ Atualmente, está com o valor Billboard, que gera esse grupo de imagens 2D com aparência de fumaça. Mas, para essas partículas especificamente, seria interessante gerá-las como se nossa rollerball estivesse se partindo em vários pedaços.
- ▶ Assim, vamos alterar de Billboard para Mesh (malha 3D).
- ▶ Clique no pequeno círculo ao lado da palavra Mesh e, na janela que aparecerá, selecione o item Rollerball.
- ▶ Precisamos definir o material da rollerball em explosão, pois o material está com o material default de partículas. Escolha o “Material Bola” e teremos o efeito de várias pequenas rollerball explodindo.
- ▶ Podemos continuar configurando mas está bom agora.
- ▶ Vamos adicionar um Audio Source em nossas partículas.
- ▶ Clique em [Add Component] | Audio | Audio Source.
- ▶ Na propriedade AudioClip, adicione o som PlayerExplosion.



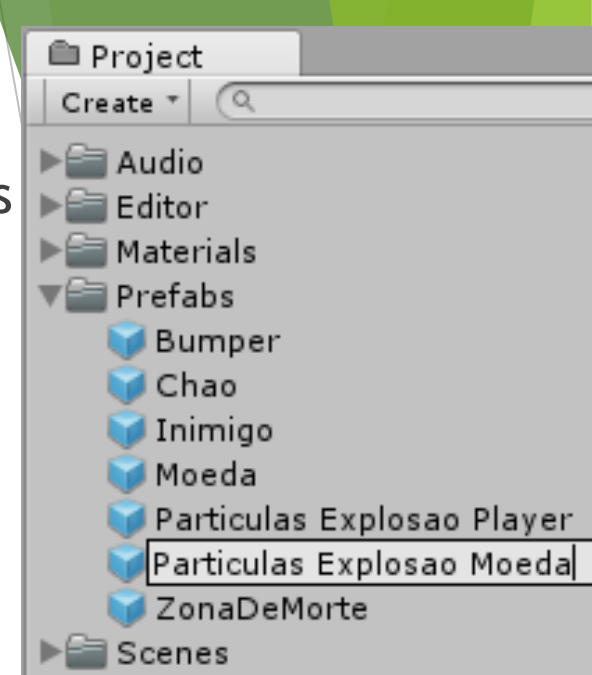
Roller Madness - Partículas e Efeitos Sonoros

- ▶ Agora, esse objeto de partículas será usado apenas quando o player morrer.
- ▶ Será importante que esse objeto seja desalocado da memória após seu uso e, para isso, vamos adicionar um script que o faça.
- ▶ Deixe selecionado o sistema de partículas e, no Inspector, clique em [Add Component] | Scripts. Selecione o script TimedObjectDestructor.
- ▶ Mude a variável Time Out para 3, de forma que o objeto destrua a si mesmo após 3 segundos.
- ▶ Agora que temos esse objeto pronto, vamos transformá-lo em um Prefab, arrastando-o da Hierarquia para a pasta prefabs da janela Project.
- ▶ Não queremos que essas partículas apareçam no jogo desde o início, apenas quando o player morrer, portanto, vamos apagá-la da Cena ou da Hierarquia, pois já temos seu modelo como um Prefab.
- ▶ Selecione a Rollerball e, no script Health, arraste o prefab “Particulas Explosao Player” para a variável “Explosion Prefab”. Quando o player morrer, o script Health disparará a geração da explosão que as partículas simulam.



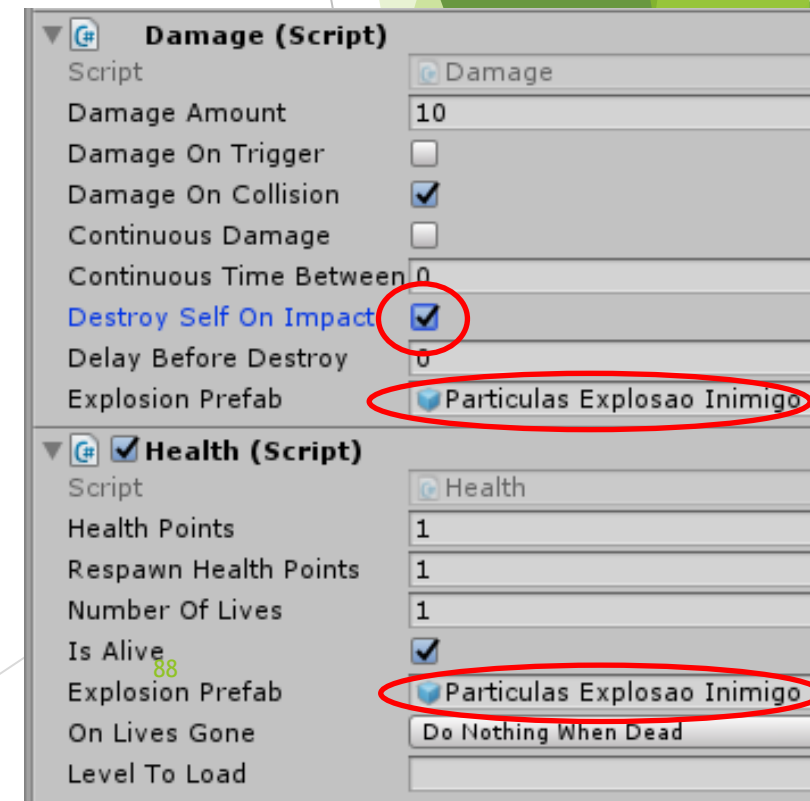
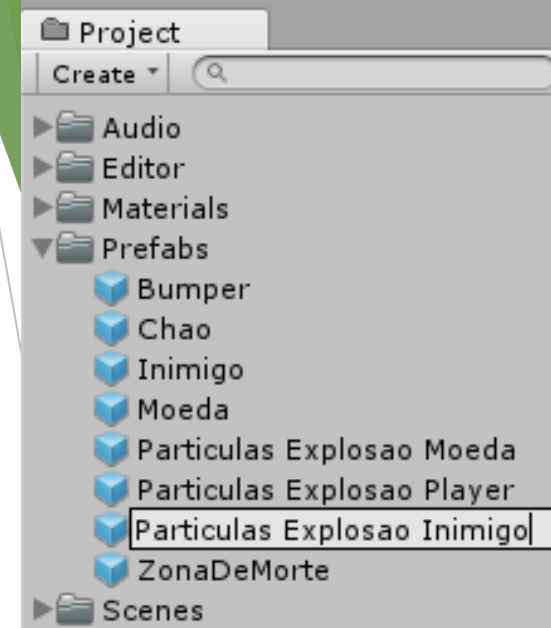
Roller Madness - Partículas e Efeitos Sonoros

- ▶ Agora que temos a explosão do player, podemos rapidamente criar sistemas de partículas para as moedas e para os inimigos.
- ▶ Assim, basicamente duplicaremos o prefab “Particulas Explosao Player” para criar outras partículas, específicas para cada situação.
- ▶ Duplique esse prefab e chame o resultado de “Particulas Explosao Moeda”.
- ▶ Você pode editar as propriedades diretamente no prefab selecionado, sem ter que fazer uma cópia dele na cena. Assim, no Inspector, altere as propriedades de Particle System abaixo:
 - ▶ Start Size → 0.2
 - ▶ Emission | Bursts → 10
 - ▶ Renderer | Material → Material Moeda
- ▶ Em Audio Source, troque AudioClip para CoinExplosion.
- ▶ Manteremos o script TimedObjectDestructor.
- ▶ Clique em uma das moedas na cena, vá ao script Treasure e, na variável Explosion Prefab, arraste o prefab que acabamos de criar (Particulas Explosao Moeda). Clique em [Apply], para que o prefab Moeda receba essas alterações e todas as demais moedas sejam atualizadas. Execute o jogo.



Roller Madness - Partículas e Efeitos Sonoros

- ▶ Agora já temos alguns efeitos visuais e sonoros interessantes em nosso jogo e vamos, agora, criar o sistema de partículas para a morte dos inimigos.
- ▶ Duplique o prefab “Particulas Explosao Player” e renomeie a cópia para “Particulas Explosao Inimigo”.
- ▶ No Inspector, mude a propriedade Renderer para Cube e Material para “Material Inimigo”.
- ▶ Modifique AudioSource | AudioClip para EnemyExplosion.
- ▶ No prefab Inimigo, vá ao script Damage e arraste o prefab “Particulas Explosao Inimigo” para a variável Explosion Prefab.
- ▶ Nesse script, faça Destroy Self on Impact valer true.
- ▶ No script Health, e arraste o prefab “Particulas Explosao Inimigo” para a variável Explosion Prefab.
- ▶ Execute o jogo.
- ▶ Agora, o jogo ficou um pouco mais balanceado pois, quando um inimigo colide com outro, eles se destroem.



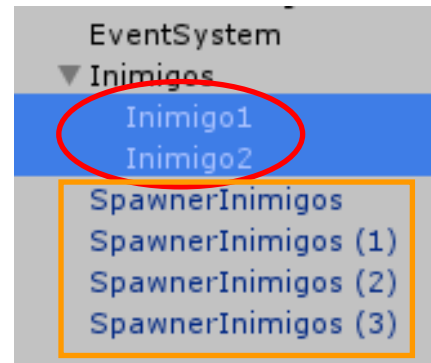
Roller Madness - Spawners

- ▶ A última coisa que faremos para tornar nosso jogo mais interessante e eficiente é adicionar spawners para as moedas e para os inimigos.
- ▶ Dessa forma, o jogo continuamente adicionará mais moedas para serem coletadas mas, também, mais inimigos serão adicionados para aumentar o grau de desafios do jogo.
- ▶ Primeiramente, criaremos um objeto vazio para representar o spawner de inimigos. Na Hierarquia, pressione [+] | Empty Object. Mude o nome para SpawnerInimigos e faça o reset de seu transform.
- ▶ A este objeto, adicione o script Spawn Game Objects, que já foi codificado anteriormente.
- ▶ Esse script espera um prefab para ser gerado (spawned) e, portanto, associamos a variável SpawnPrefab com o objeto Inimigo da pasta Prefabs da janela Project.
- ▶ Na variável Chase Target, associamos o player, ou seja, a Rollerball para que ela seja caçada pelos inimigos gerados.
- ▶ Copie o objeto SpawnerInimigos para a pasta Prefabs de Project, de forma que ele vire um prefab.



Roller Madness - Spawners

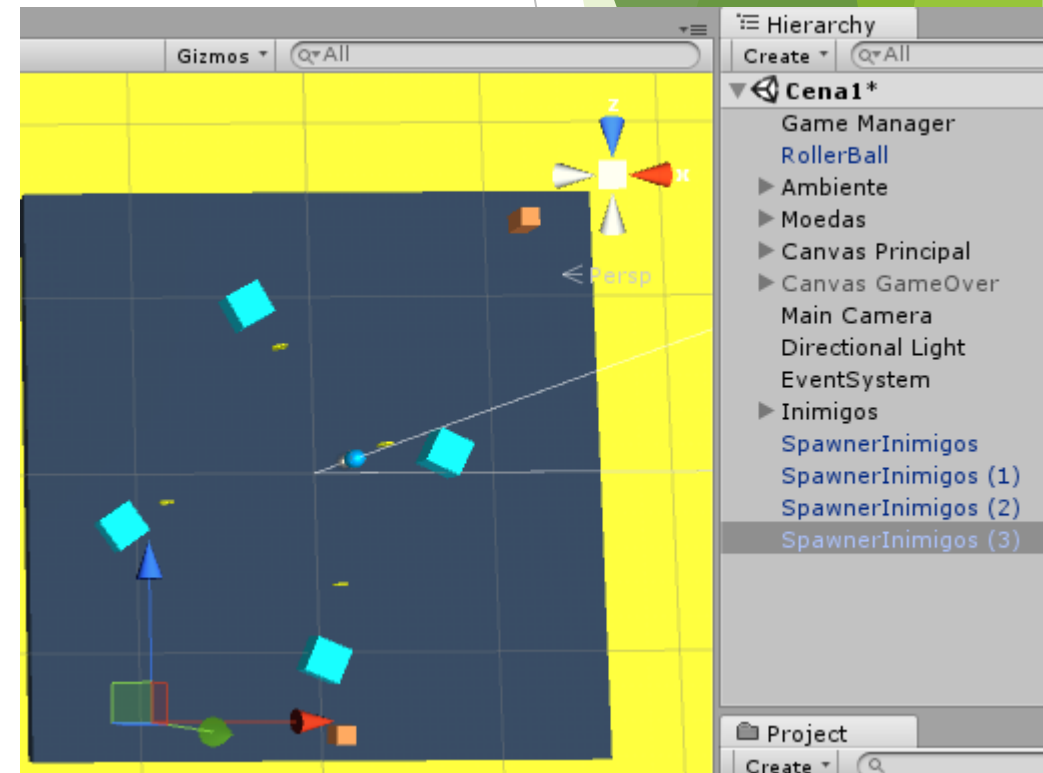
- ▶ Mude o Transform Y do objeto SpawnerInimigos da cena para 9, de forma que ele fique bem alto no jogo e, ao gerar novos inimigos, os faça cair de muito alto, o que resultará em inimigos que rebaterão bastante no jogo.
- ▶ Mude a visualização da cena para a visualização top down e mova esse objeto para o canto superior esquerdo da cena. Em seguida, faça duplicações do mesmo e as mova para os outros três cantos da cena.
- ▶ Apague os inimigos criados anteriormente e que compõe o objeto Inimigos da cena, pois apenas usaremos os que forem gerados pelos spawners.



- ▶ Copie os quatro spawners para dentro do objeto Inimigos. Mude o nome desse objeto para SpawnerInimigos.

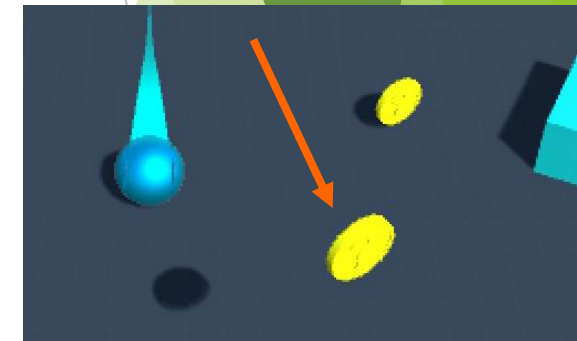


- ▶ Salve e teste o jogo.



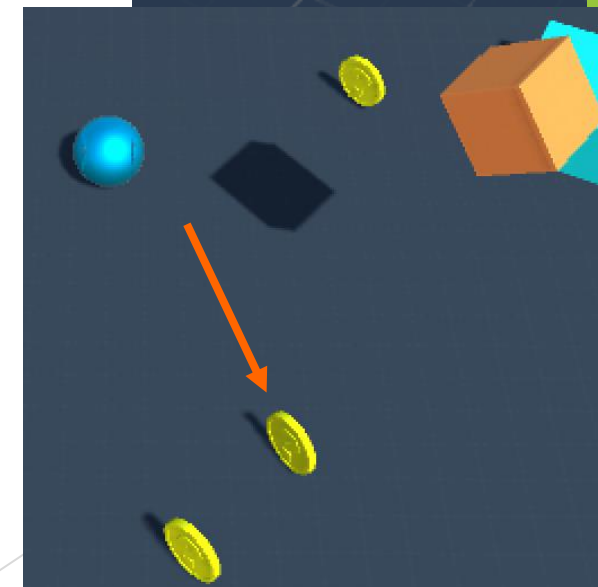
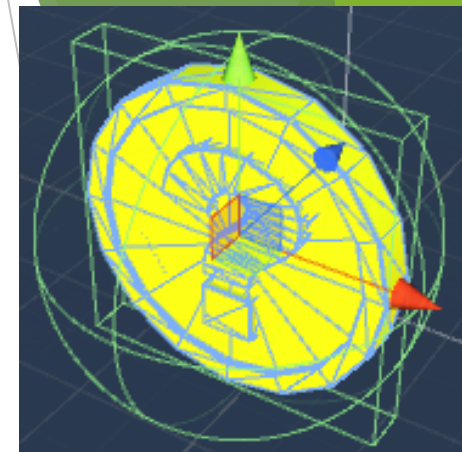
Roller Madness - Spawners

- ▶ Se você mantiver a visão top down da janela Scene enquanto joga (para isso, desative o botão [Maximize on Play] da janela Game), verá que os inimigos aparecem nos quatro cantos da área de jogo e perseguem a Rollerball.
- ▶ Vamos agora criar o spawner de moedas. Para torná-las mais interessantes, faremos com que elas interajam com Física, como os inimigos.
- ▶ Deixemos as moedas já existentes em suas posições.
- ▶ Vamos ao prefab das Moedas e o dupliquemos. Mude o nome dessa cópia para Moeda Saltitante.
- ▶ Precisamos de um Rigidbody para que as moedas saltitantes caiam.
- ▶ Portanto, observe o Rigidbody do prefab Moeda Saltitante. Mude sua propriedade Use Gravity para true.
- ▶ Arraste o prefab Moeda Saltitante para a cena e a deixe alta sobre o chão.
- ▶ Ao executar o jogo, verá que a moeda não cai. Isso é devido ao fato de que Is Kinematic ainda está ativado.
- ▶ Portanto, deixe Is Kinematic igual a false e execute o jogo. Ela cairá e passará pelo chão, desaparecendo.
- ▶ Isso ocorre porque ela tem um Sphere Collider onde Is Trigger está true.



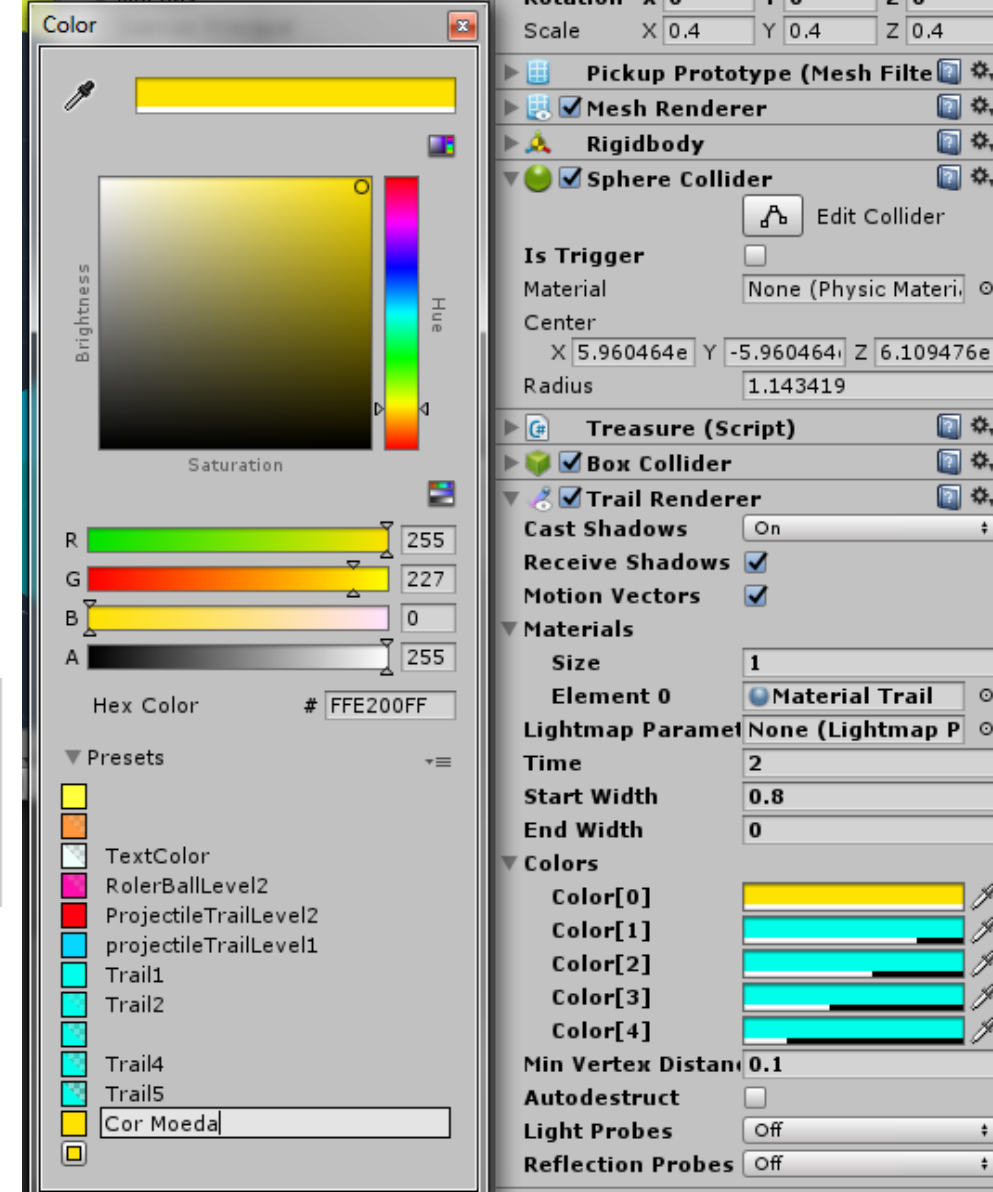
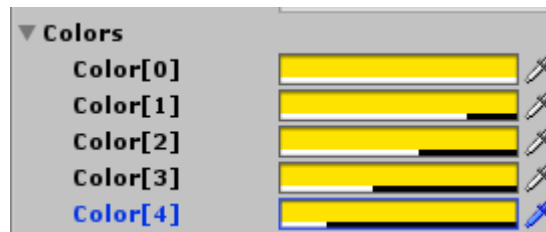
Roller Madness - Spawners

- ▶ Precisamos desse Sphere Collider true pois é com ele que ativamos o script Treasure, que conta as moedas que são coletadas.
- ▶ Portanto, para evitar que a moeda passe pelo chão, vamos adicionar um outro Collider a ela, este com Is Trigger igual a false.
- ▶ Clique em [Add Component] | Physics | Box Collider. Passaremos a ter dois colliders nesse objeto, como na figura ao lado.
- ▶ Deixe o Box Collider com Is Trigger valendo false, porque queremos que a moeda caia sobre o chão ou algum outro obstáculo da cena e salte, sem passar dentro do objeto com o qual colidiu. Dê a ela o Physic Material Bouncy para que ela fique saltitando quando tocar o chão.
- ▶ Remova o script Rotate da Moeda Saltitante e execute o jogo.
- ▶ Podemos também colocar um rastro nas moedas saltitantes que criarmos.
- ▶ Vá até o objeto Rollerball, clique no Trail Renderer de seu Inspector e copie esse componente.
- ▶ Clique na Moeda Saltitante e em alguma engrenagem do Inspector selecione Paste Component as New.



Roller Madness - Spawners

- ▶ Mude Start Width para 0.2, pois a moeda é menor que a Rollerball.
- ▶ Vamos ajustar as cores. Clique na primeira cor (ciano) e ajuste para amarelo, como vemos ao lado.
- ▶ Nomeie a nova cor como Cor Moeda.
- ▶ Em cada um das cores do Trail Renderer, selecione a cor Cor Moeda no Color Pickup e altere a parte alpha da cor (transparência).
- ▶ Terminamos o Trail Renderer.
- ▶ Como teremos várias moedas saltitantes sendo criadas, algumas poderão saltitar para fora do jogo e não mais as veremos, mas elas continuarão lá ocupando memória e tempo do processador. Para evitar isso, adicionaremos o script Timed Object Destructor à Moeda Saltitante.
- ▶ Mude TimeOut para 10 segundos e, após esse tempo, elas serão removidas da memória e deixarão de existir.



Roller Madness - Spawners

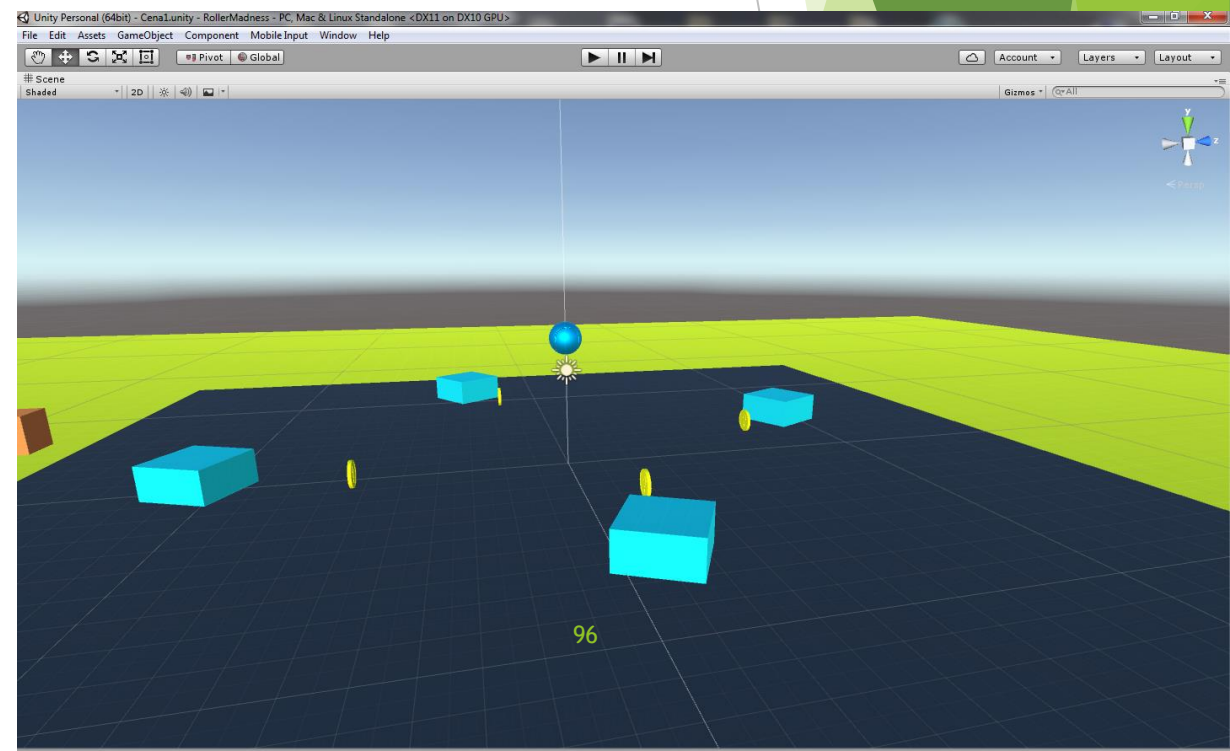
- ▶ Tendo selecionado a Moeda Saltitante, pressione o botão [Apply] no seu Inspector e, dessa forma, as alterações feitas serão aplicadas no Prefab associado a ela.
- ▶ Remova esse objeto da cena.
- ▶ Da mesma forma que fizemos com o Spawner de Inimigos, criaremos um Spawner de moedas.
- ▶ Crie um objeto vazio, chame-o de SpawnerMoedas. Faça o reset de seu transform.
- ▶ Adicione o Script Spawn Game Objects.
- ▶ Na variável Spawn Prefab, arraste o prefab Moeda Saltitante. Mude Min Seconds Between Spawning para 1 e Max Seconds Between Spawning para 5.
- ▶ A moeda não caçará nenhum objeto, portanto deixe a variável Chase Target vazia.
- ▶ Copie esse objeto SpawnerMoedas para a pasta Prefabs e faça várias cópias do novo prefab na cena, em diferentes posições, na altura 9 Y, por exemplo.
- ▶ Crie um objeto vazio chamado SpawnerMoedas para agrupar os spawners de moedas que você criou.

Roller Madness - Projeto Final

- ▶ Baseando-se no que discutimos neste estudo do jogo RollerMadness, crie uma segunda fase, que deve ser lançada quando o player atingir um certo número de coletas sem ser atingido pelos inimigos.
- ▶ Na segunda fase, coloque mais obstáculos, de diferentes formas, mude os parâmetros dos inimigos para torná-los mais letais e, caso o Player adquira as moedas necessárias para mudar de fase, mostre um Canvas de Vitória e permita reiniciar o jogo ou terminá-lo (usando o script UIButtonQuitGame).
- ▶ Esse projeto deve ser desenvolvido em dupla.

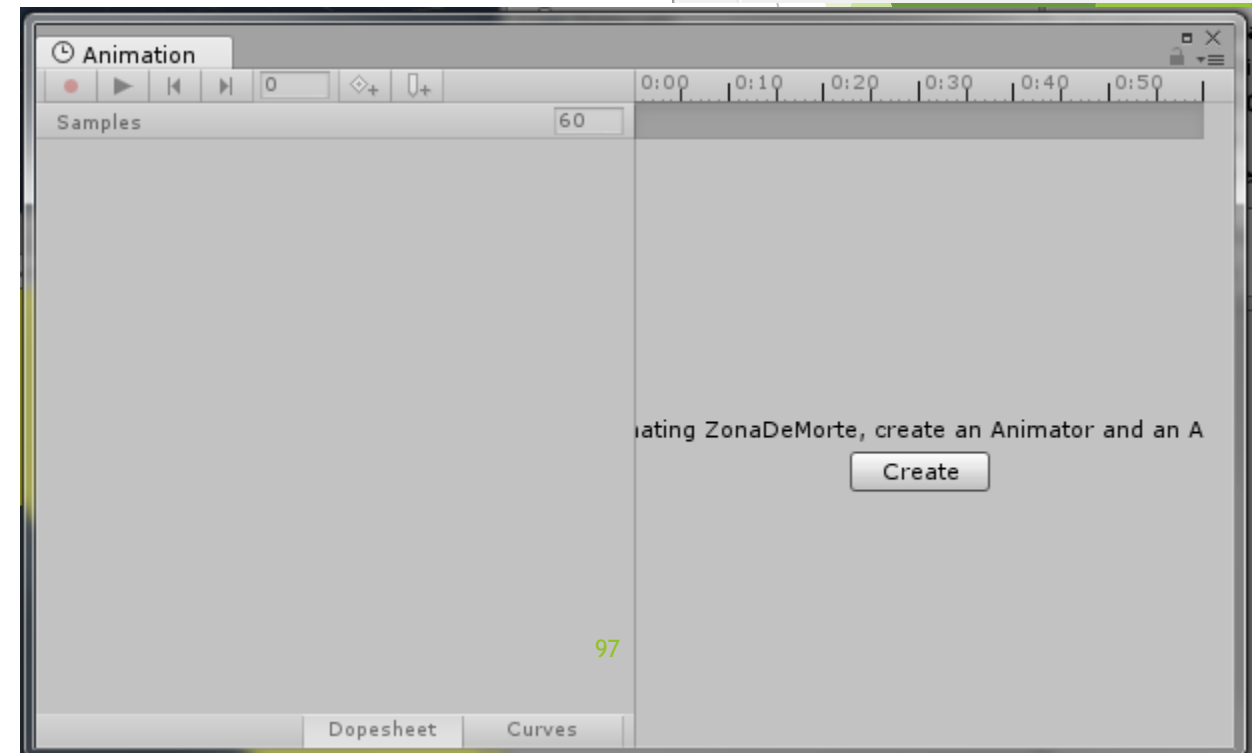
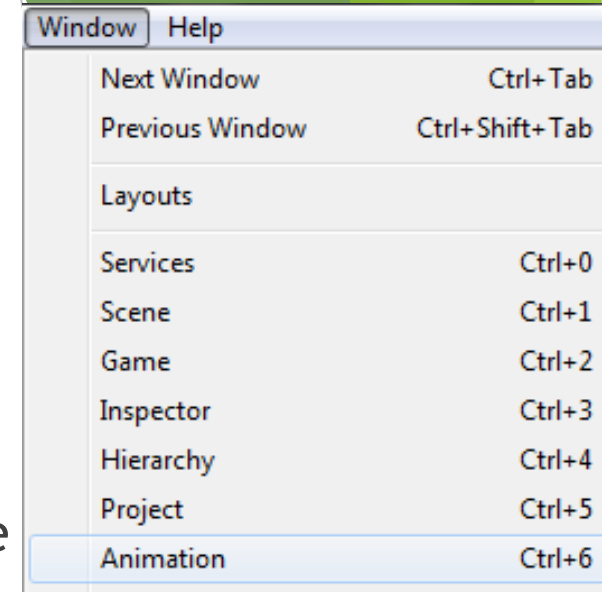
Roller Madness - Animação

- ▶ Animação é uma maneira de adicionar um polimento visual ao nosso jogo.
- ▶ Criaremos uma animação simples, explorando a Animation Timeline.
- ▶ Criaremos um clip de animação por meio de quadros chave (keyframe) no objeto ZonaDeMorte. Criaremos uma animação de luz difusa (shimmer) nesse objeto, para comunicar o fato de que é uma região perigosa para o jogador.
- ▶ Na Hierarquia, selecione o objeto “Zona de Morte” (para achá-lo, escreva a palavra “zona” no buscador ao lado do botão [+]).
- ▶ Clique em ZonaDeMorte para selecioná-lo e pressione o botão X do buscador, para mostrar toda a Hierarquia e manter selecionado ZonaDeMorte.
- ▶ Outra função de editor interessante que ainda não vimos é a fullscreen. Com o cursor sobre a janela Scene, pressione [Shift-Espaço].



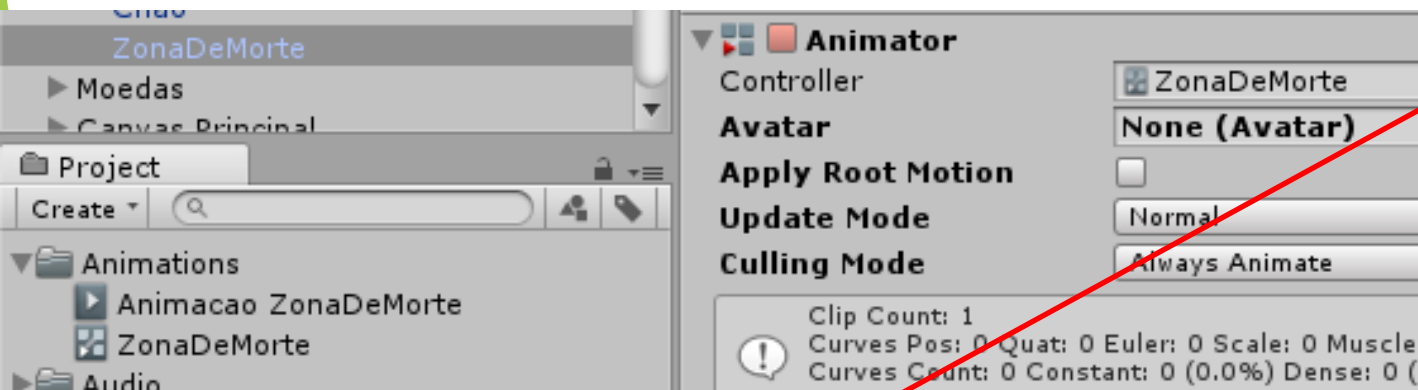
Roller Madness - Animação

- ▶ Outro [Shift-Space] volta a cena para a visualização anterior.
- ▶ Esse recurso funciona com qualquer janela do Unity.
- ▶ Se sua ZonaDeMorte estiver invisível, ative esse objeto ou seu mesh renderer, aquele que estiver desabilitado.
- ▶ Criamos animações através do Sistema de Animações.
- ▶ No menu, selecione Window | Animation. Aparecerá a janela do Editor de Animação, como vemos ao lado:
- ▶ Ela informa que, para criar uma animação, precisamos criar um Animator e um clipe de Animação.
- ▶ O sistema criará um componente Animator vinculado ao objeto ZonaDeMorte e esse Animator usa Animation Clips para criar os Quadros da animação.
- ▶ Clique no botão [Create] dessa janela.

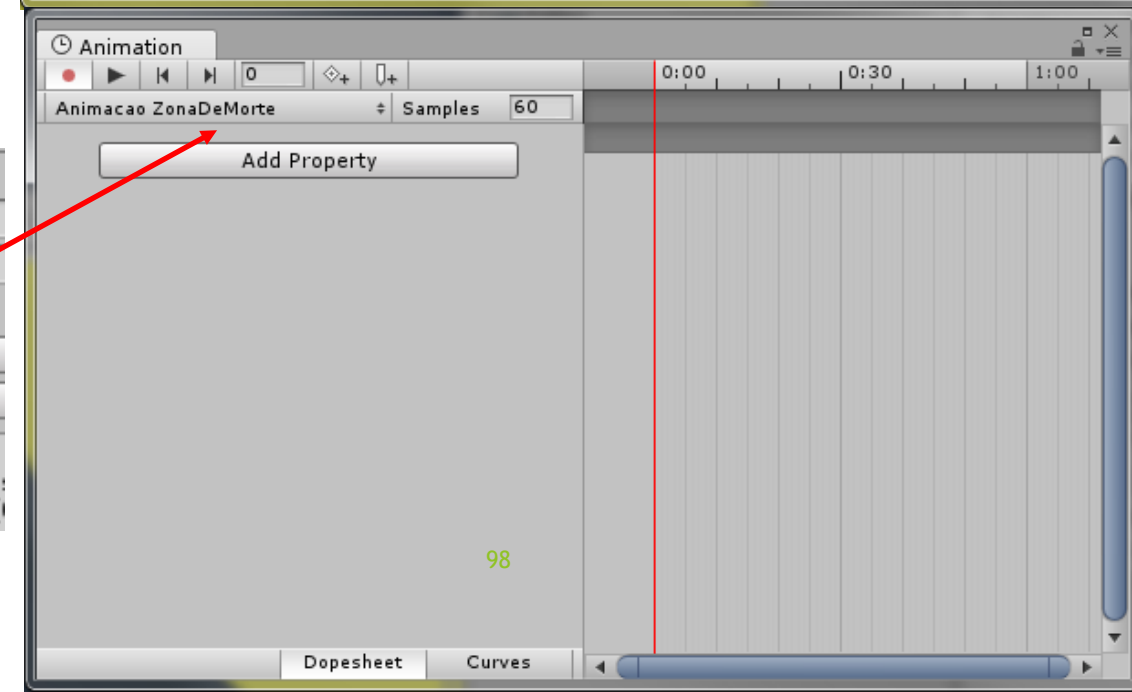
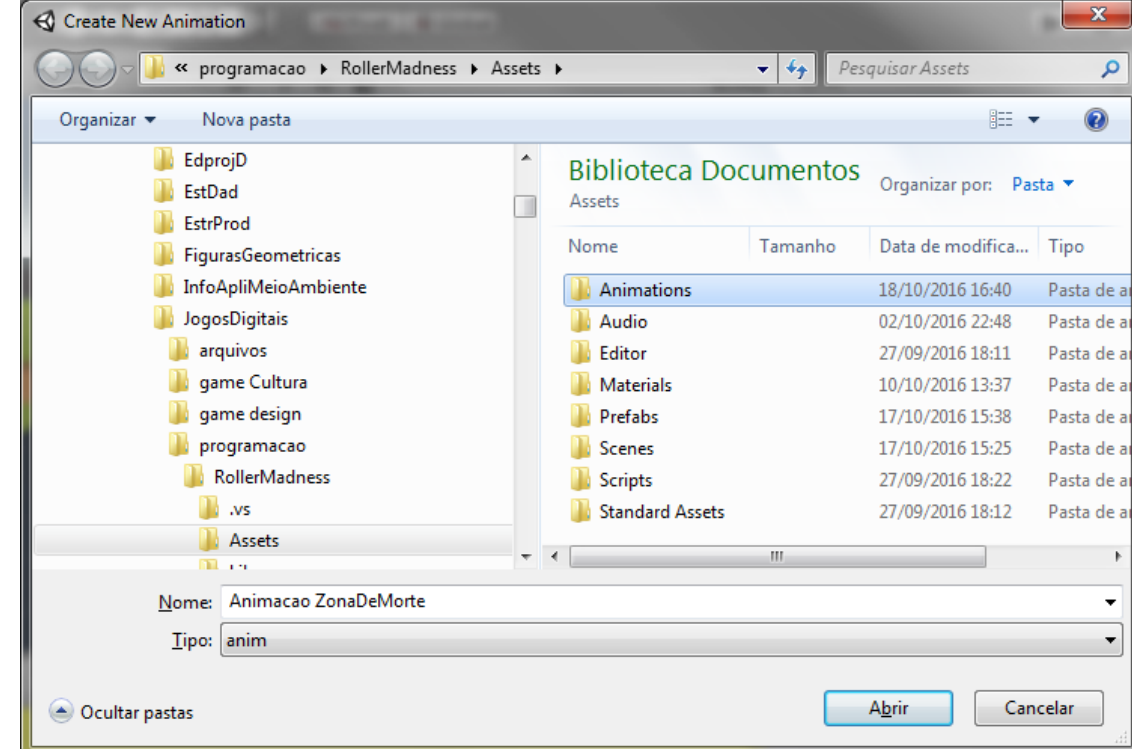


Roller Madness - Animação

- ▶ Aparecerá uma janela solicitando o nome do arquivo de animação.
- ▶ Crie uma pasta Animation nesse local, que representa o nosso projeto.
- ▶ Abra essa pasta e informe o nome do arquivo como “Animacao ZonaDeMorte”.
- ▶ Na pasta Animations, foi criado um objeto Animator e ele foi vinculado ao Inspector do game object ZonaDeMorte, que estava selecionado no momento da criação da animação.

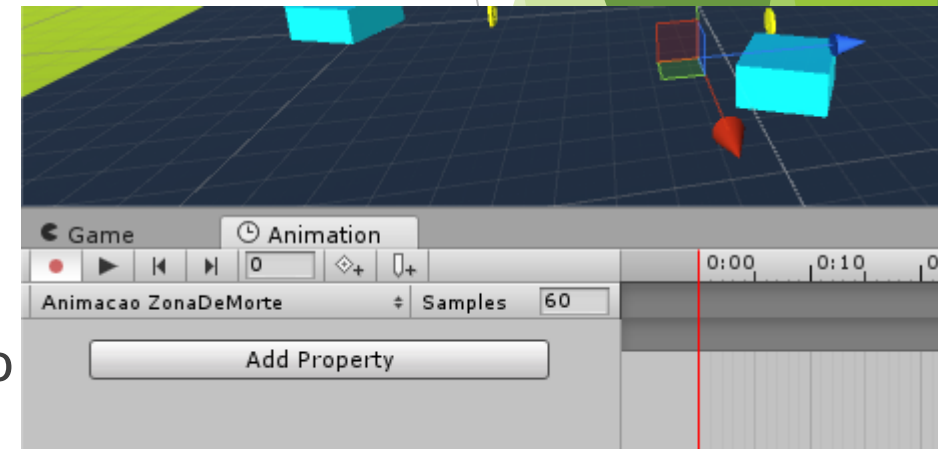
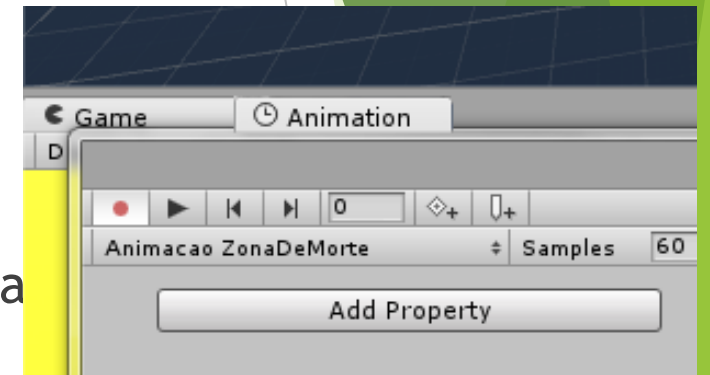
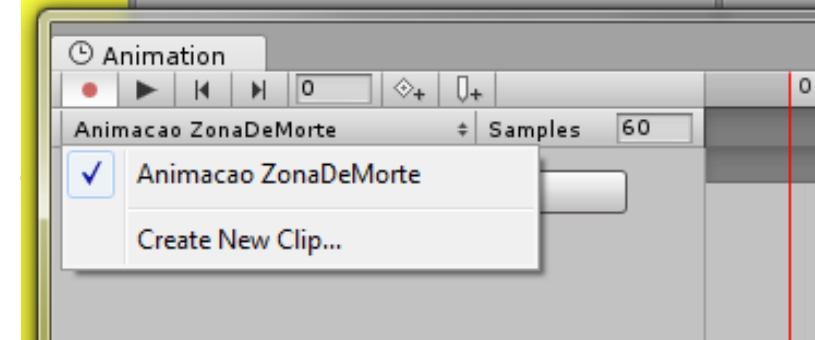


- ▶ Veja o objeto de animação no Editor de Animações.



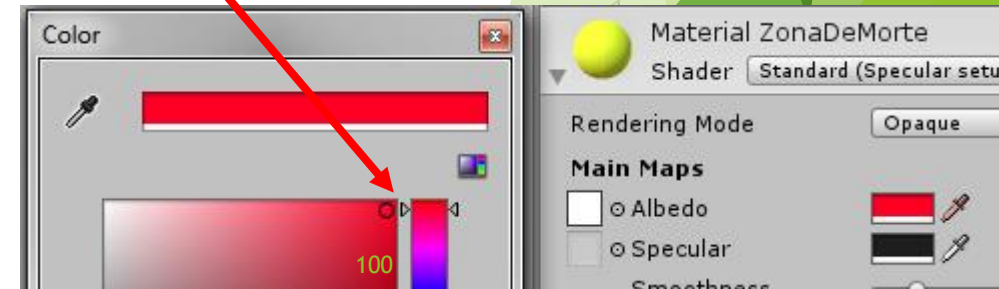
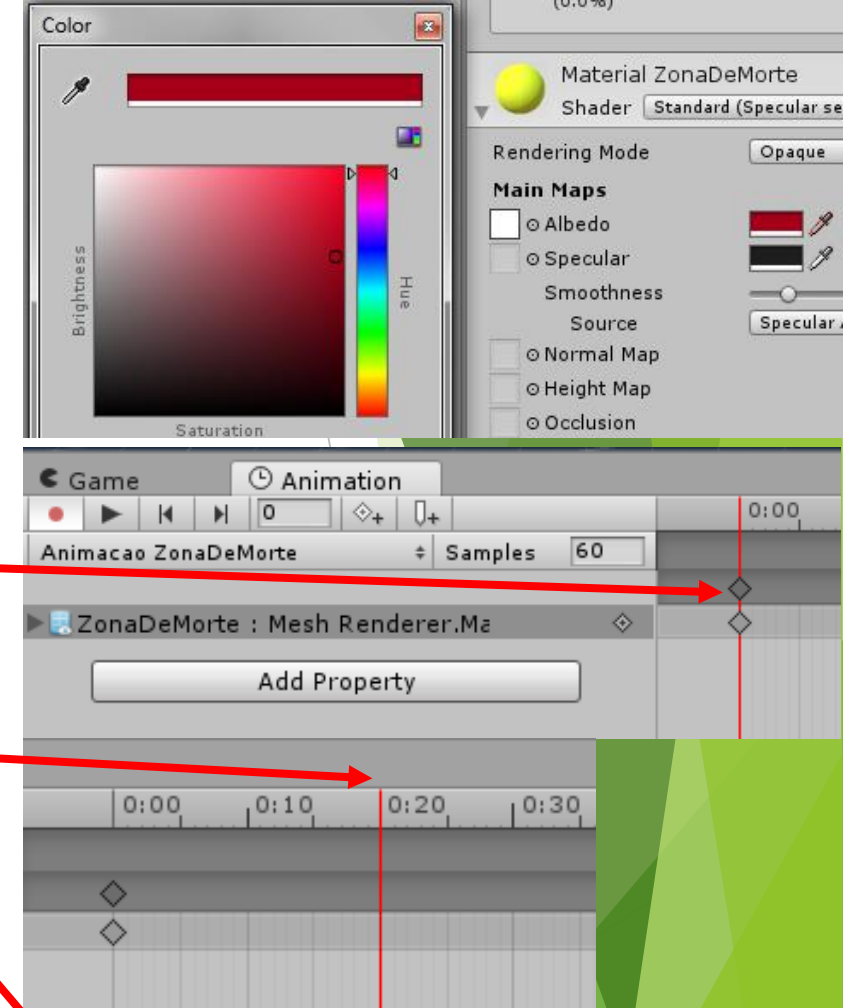
Roller Madness - Animação

- ▶ Você pode criar novos clipes de animação, na interface do Editor, vemos ao lado.
- ▶ Trabalhar com esse editor de animações seria melhor com várias monitores mas, como não os temos, vamos arrastar a janela do editor para baixo e encaixá-la ao lado da janela Game.
- ▶ O editor de animações é bastante direto.
- ▶ Temos uma timeline à direita, que começa em zero e o tempo de um em um segundo.
- ▶ Baseia-se em amostras (samples). O valor de 60 indica que serão 60 amostras por segundo.
- ▶ Note que o botão circular Vermelho (Recording) está ligado.
- ▶ Pode ser desligado mas, quando está ligado, tudo o que você fizer no editor do unity será gravado na timeline e reproduzido durante a animação.



Roller Madness - Animação

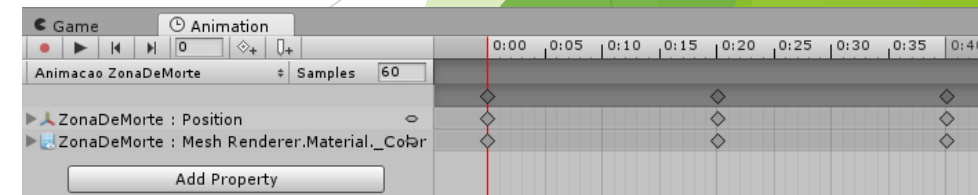
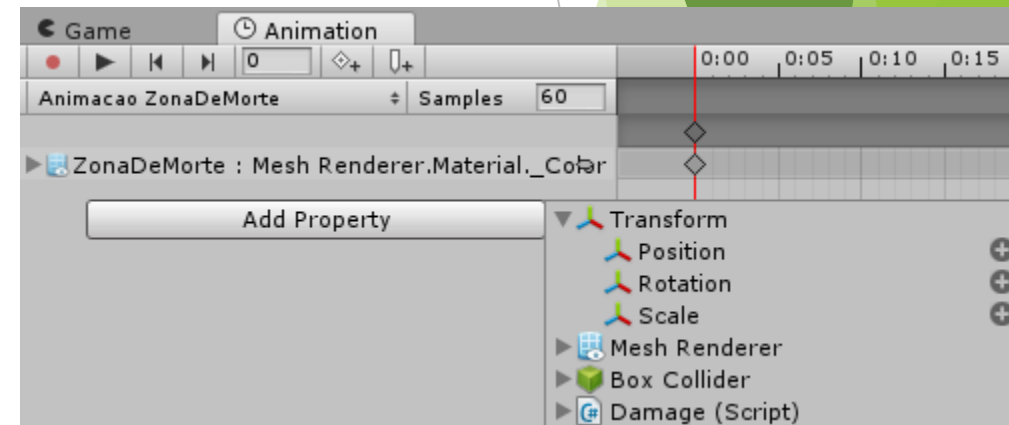
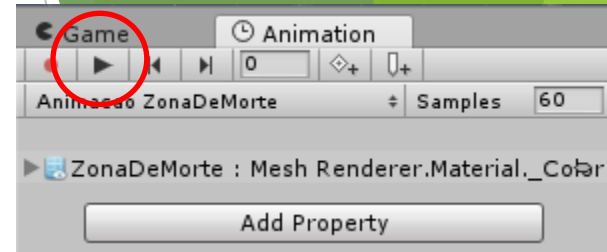
- ▶ Assim, com o objeto ZonaDeMorte selecionado, acessemos a propriedade Material. No quadro de cor do Albedo, mude a cor para uma versão mais escura de vermelho (A50017FF).
- ▶ Observem na timeline da animação que um keyframe foi marcado, pois tudo o que fizemos durante a gravação será registrado.
- ▶ Na timeline, cliquemos na marca 0.2 para movermos adiante no tempo.
- ▶ No Inspector do ZonaDeMorte, na propriedade Material, vamos voltar a cor para o vermelho vivo (FF0000FF) que havia antes.
- ▶ Essa ação também foi registrada na timeline.
- ▶ Avance agora o tempo para 0.4 na timeline.



- ▶ Marque novamente a cor do 1o keyframe no Materials do Inspector.

Roller Madness - Animação

- ▶ Se você clicar na seta ao lado do botão de gravação, poderá ver o resultado da animação, quando ZonaDeMorte ficará piscando entre tonalidades de vermelho.
- ▶ Em geral, em animações, o sistema de animação faz uma interpolação entre o valor estabelecido de uma propriedade num determinado momento do tempo e o valor estabelecido dessa propriedade no tempo seguinte. Em outras palavras, no caso da mudança de cor, ela não ocorre apenas no tempo 0.0 0.2, 0.4 e 0.0 mas, ao contrário o sistema calcula valores intermediários da propriedade nos intervalos de tempo entre as medidas (key frames), de forma que suaviza a transição de uma cor para outra.
- ▶ Trabalhamos com a animação de uma única propriedade mas podemos incluir outras propriedades e animá-las.
- ▶ Com o botão [Record] ainda selecionado na janela Animation, clique em [Add Property] e selecione Transform | Position.
- ▶ Clique no tempo 0:00, mude o valor de Y de -2 para -2.1.
- ▶ Clique no tempo 0.2 e mude o valor de Y para -2. Clique no tempo 0.4 e mude Y para -2.1.
- ▶ Desligue o botão [Record], pois temos nossa animação pronta.



Roller Madness - Animação

- ▶ Observe que o objeto ZonaDeMorte tem um box collider e estamos movendo-o com animation, provavelmente deveríamos adicionar um Rigidbody nele para propósitos de otimização de desempenho do jogo.
- ▶ Adicione um Rigidbody em ZonaDeMorte.
- ▶ Como o movemos por meio de Animation, certamente não desejamos que a gravidade atue nesse objeto e, portanto, desligamos o checkbox Use Gravity.
- ▶ Por estamos movendo-o com animação, ligamos Is Kinematic,
- ▶ Esses atos aumentarão o desempenho do jogo e diminuirão a carga na CPU e na GPU, tornando seu uso mais eficiente.

