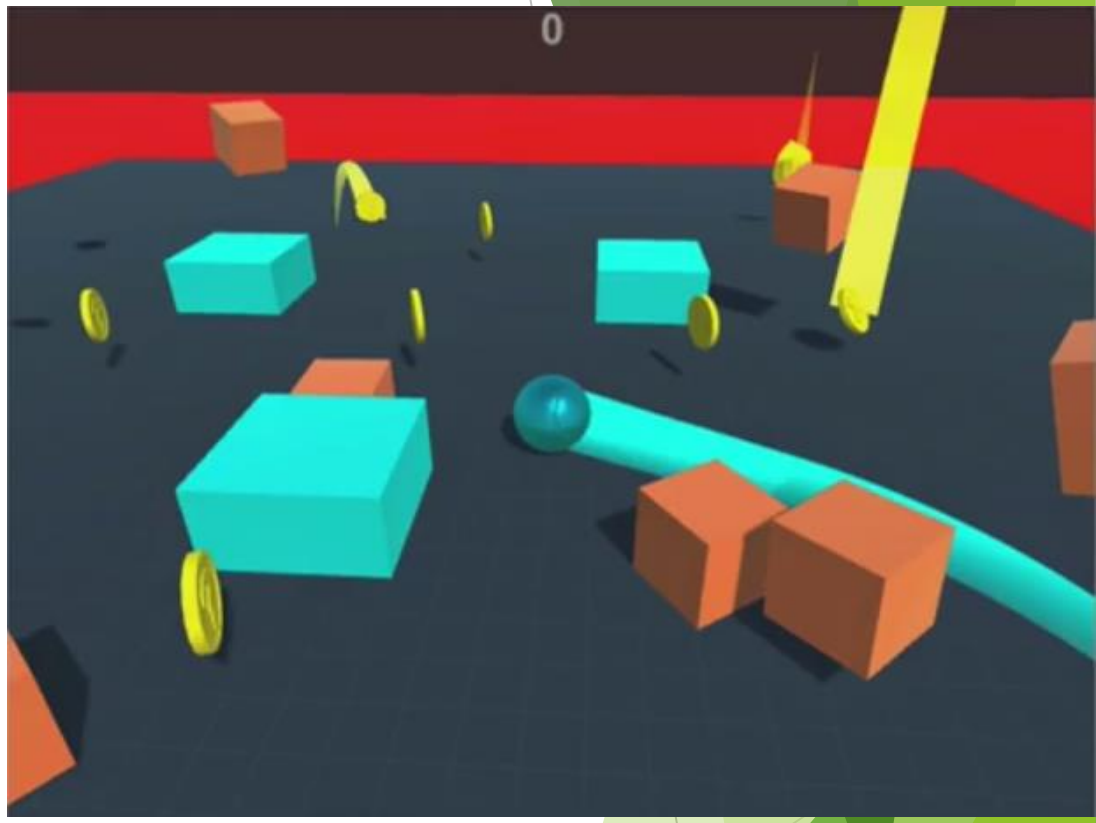
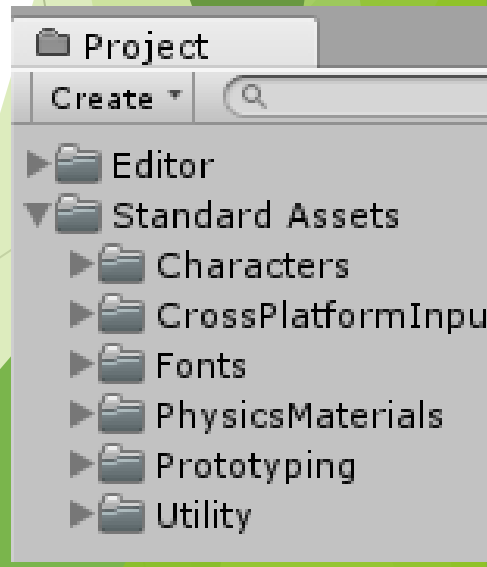
Roller Madness



- Neste novo jogo, trataremos os conceitos abaixo:
  - Importação de Pacotes de Assets
  - Uso dos Assets padrão
  - Boas práticas de organização do projeto
  - Física 3D
  - Fundamentos da Interface com o Usuário
  - Gerador de rastros
  - Sistemas de Partículas
  - Animação
  - Sistemas de Jogo : controle do player, pick-up, inimigos, placar de pontos
- Na pasta c:\temp, copie o arquivo RollerMadnessAssets.rar da pasta da nossa disciplina na rede e o descompacte. Observe a pasta Audios com arquivos wave e ogg. Há também o arquivo MSUGameScripts.unitypackage, que contém scripts C# que usaremos no jogo.

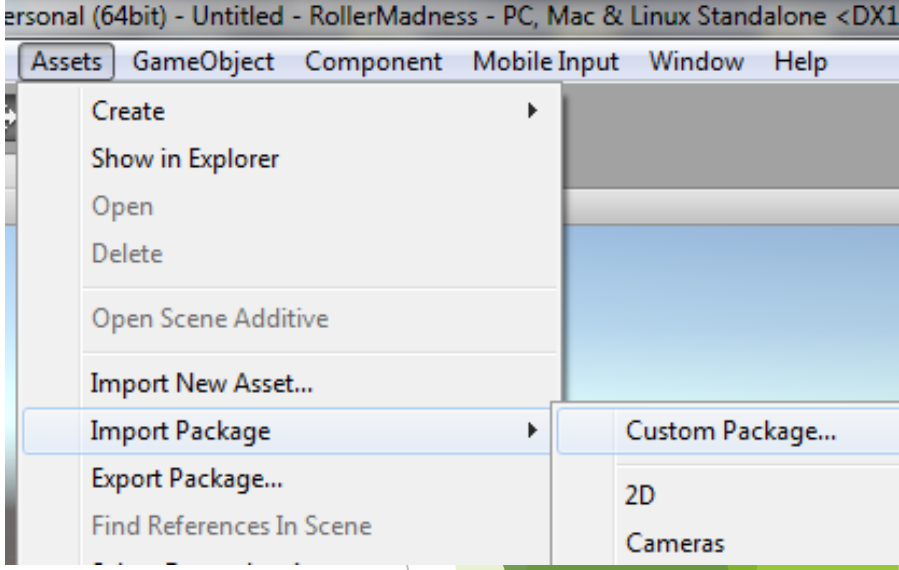
Roller Madness

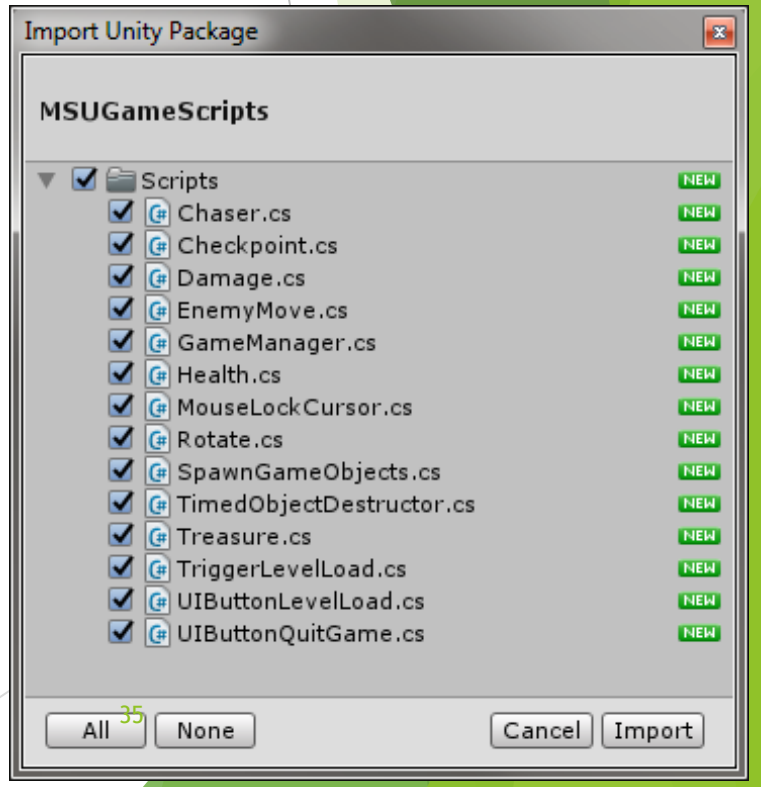
- Vamos criar um novo projeto, chamado RollerMadness e armazená-lo em uma pasta local do seu computador (por exemplo, Documents).
- Antes de clicar no botão que cria o projeto, clique no botão [Add Asset Packages]. Selecione os pacotes abaixo:
  - Characters
  - Cross Platform Input
  - Prototyping
  - Utility
- Clique em [Done]
- Clique no botão [Create Project].
- Como estamos importando assets, o Unity poderá demorar um pouco mais para mostrar o Editor.
- Você poderá importar posteriormente outros pacotes dos Assets Padrão como, por exemplo, Particle Systems.



Roller Madness –Pacote de Scripts prontos

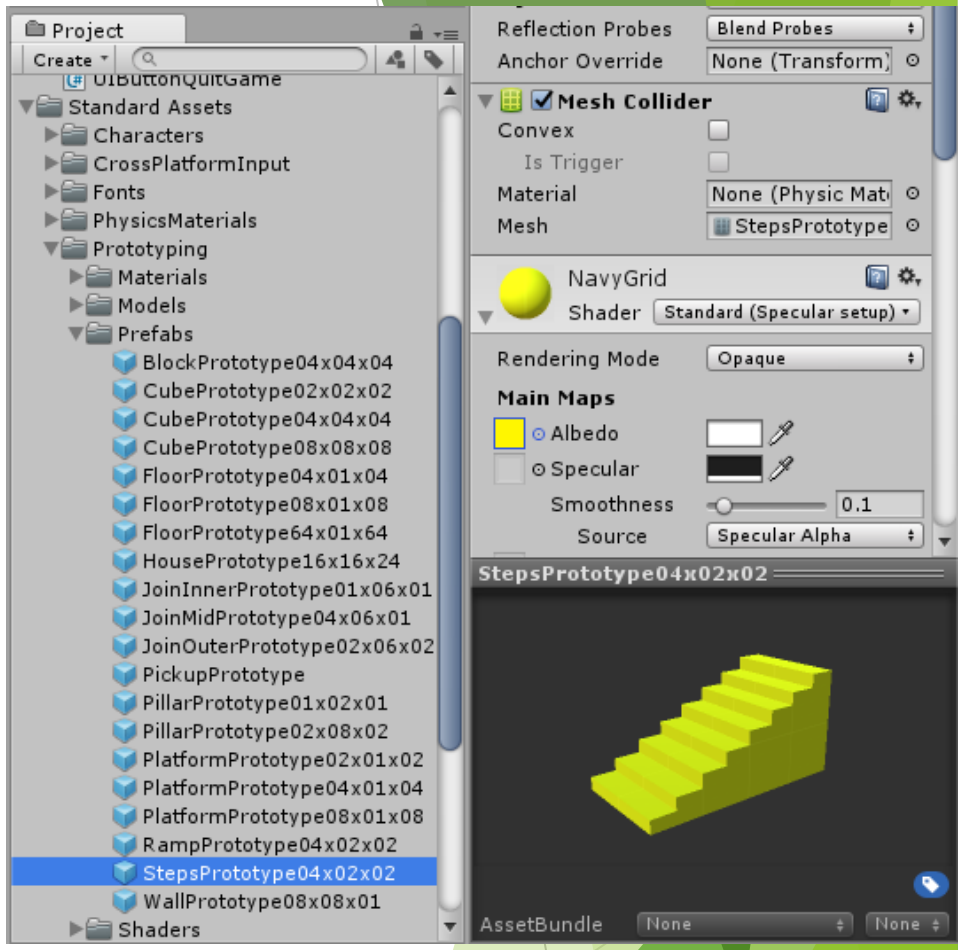
- Quando aparecer o Editor, acesse a opção Assets do Menu e selecione a opção Import Package e Custom Package.
- Selecione o arquivo c:\temp\MSUGameScripts.unitypackage.
- Clique [Import].
- Um package é um agrupamento de arquivos que pode ser usado para troca entre membros de um time de desenvolvedores de jogos ou para propósitos de arquivamento.
- No package que importamos temos scripts que controlarão os objetos de nosso novo jogo.

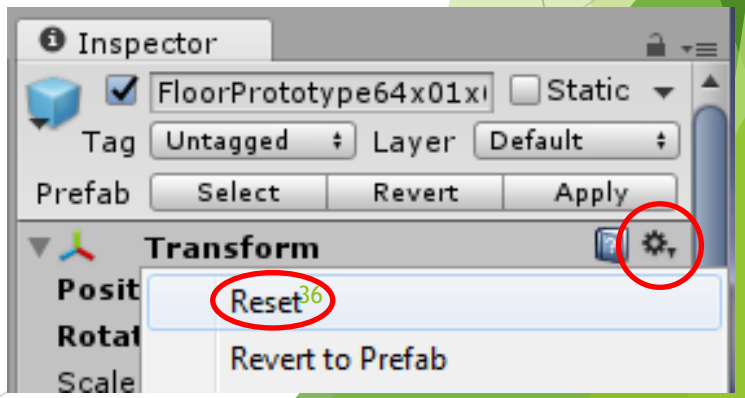




Roller Madness – Primeira fase

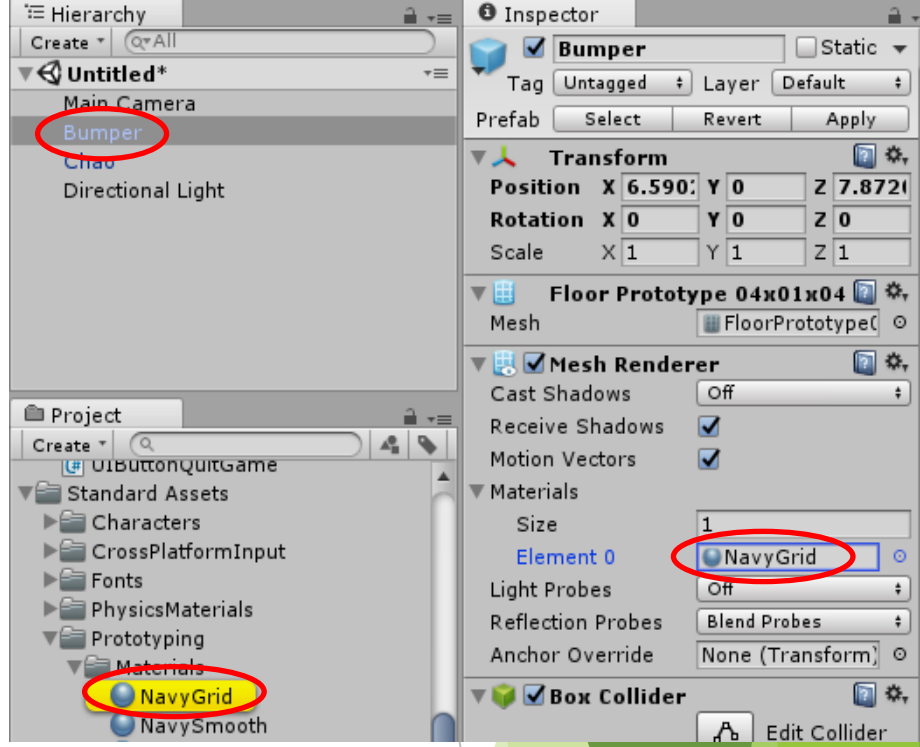
- Vamos criar a fase (level) base de nosso jogo, incluindo o chão e vários obstáculos.
- Vamos criar e usar objetos previamente fabricados, os prefabs.
- Vamos usar vários protótipos de assets vindos com o Unity, para representar objetos que posteriormente poderão ser colocados no jogo, já com sua arte final
- Na pasta Standard Assets existe uma pasta Prototyping | Prefabs com vários objetos prontos, como o que vemos na figura ao lado.
- Arraste para a cena o protótipo Floor, com 64 x 1 metros.
- Aparecerá um grande objeto na cena e, na Hierarquia, mude seu nome para Chao. Posicione-o na origem do mundo digitando (0,0,0) em Transform | Position ou selecione o botão da engrenagem no Inspector e clique em Reset.
- Modifique a escala do Chao para (0.5, 1, 0.5).

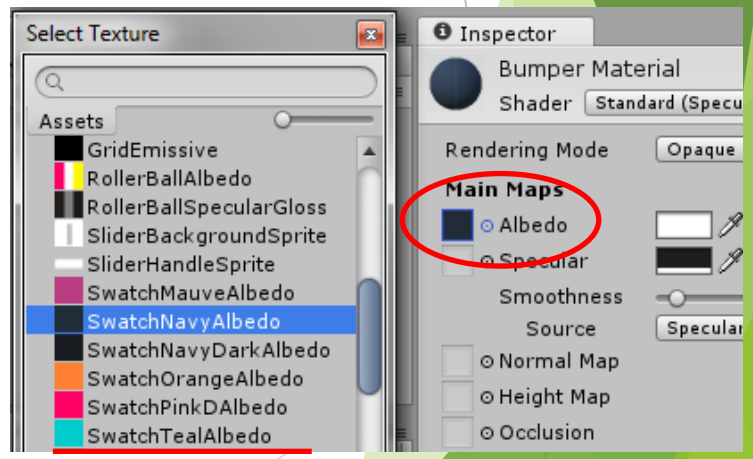




Roller Madness – Primeira fase

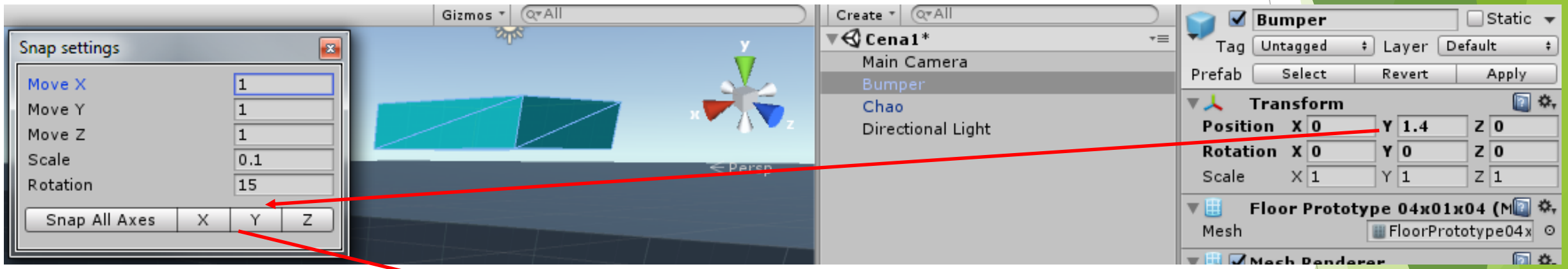
- Vamos colocar mais alguns assets como obstáculos na cena e modificá-los conforme nossas necessidades.
- Arraste para a cena o objeto FloorPrototype04x01x04 da pasta Prefabs. Ele ficará um pouco escondido, pois ainda tem o mesmo material cor e espessura que o Chao. Pressione a tecla F para dar zoom nesse objeto e mude seu nome para Bumper (ou para-choque).
- Na figura ao lado vemos o material no Mesh Renderer (NavyGrid). Selecione esse material na pasta Materials e o duplique (com Ctrl-D). Renomeie a cópia para Bumper Material e o selecione.
- Se você clicar no círculo ao lado da propriedade Albedo, aparecerão os materiais pré-feitos. Selecione o Swatch Teal Albedo ao invés do Swatch Navy Albedo. O Material mudará de cor.
- Arraste esse Material para o Bumper na Hierarquia.
- Faça reset no Bumper e o suba para (0,1,0).



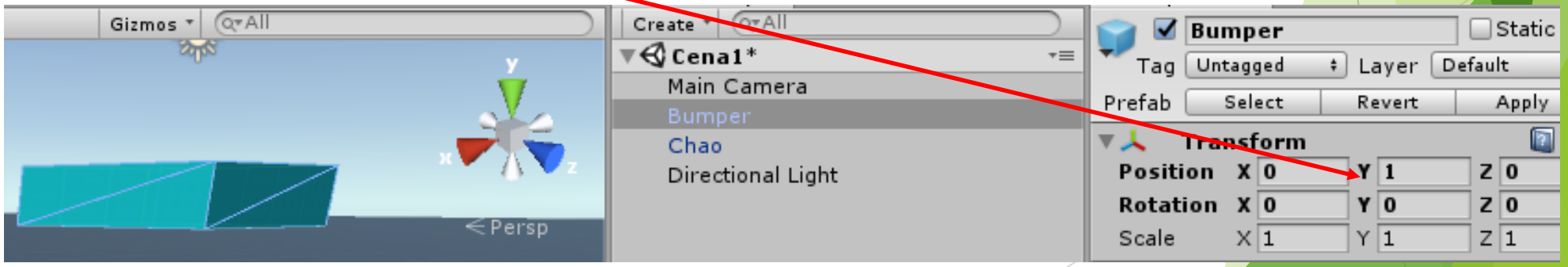


Roller Madness – Primeira fase – Snap Settings

- Para que os objetos sejam posicionados com maior controle, use as Snap Settings.
- Por exemplo, você pode mover o Bumper para cima usando a ferramenta de translação e selecionar Edit | Snap Settings. Se clicar em [Y], o Bumper será movido exatamente para a altura mais próxima de um múltiplo de 1 a partir de onde ele está, no eixo y.



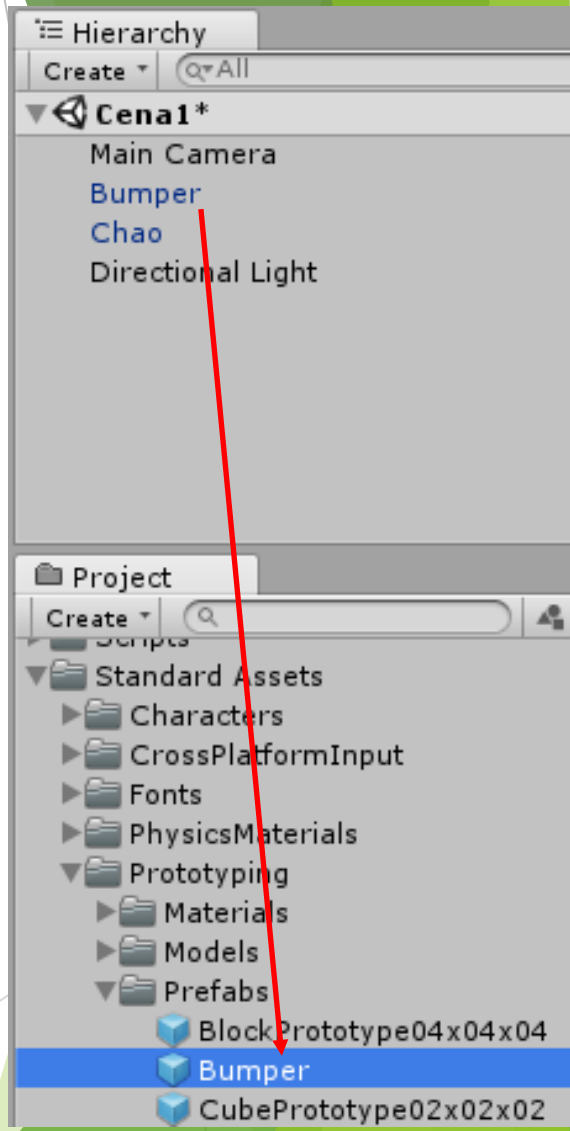
- Pressionando [Y] na janela Snap settings, o Bumper será posicionado exatamente em Y = 1, após ter sido colocado na altura y = 1.4 .



- Pressionando a tecla [**Ctrl**] e movendo o objeto, fará com ele mude de posição de acordo com os valores de Move X, Move Y e Move Z do Snap Settings.

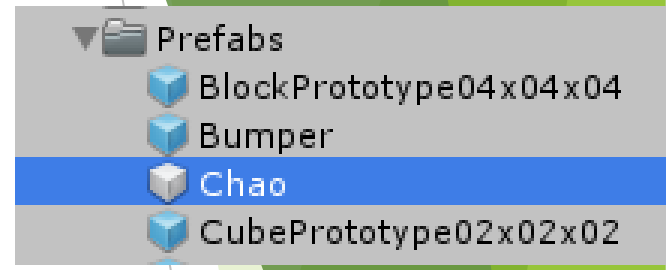
Roller Madness – Primeira fase – Criando Prefabs

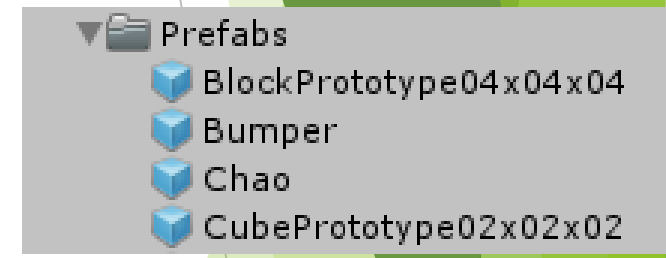
- Mova o Bumper do centro para alguma outra posição um pouco afastada, mantendo-o em contato com o Chao.
- Em seguida, fecha a janela Snap Settings.
- Queremos ter vários Bumpers em nosso mundo.
- Toda vez que planejarmos ter vários gameObjects que sejam muito idênticos nós devemos criar um Prefab a partir do objeto-modelo original.
- Para tornar nosso Bumper em um prefab, arraste-o da Hierarquia para a pasta Standard Assets | Prototyping | Prefabs.
- Observe que um Asset azulado apareceu nessa pasta, com o nome Bumper.

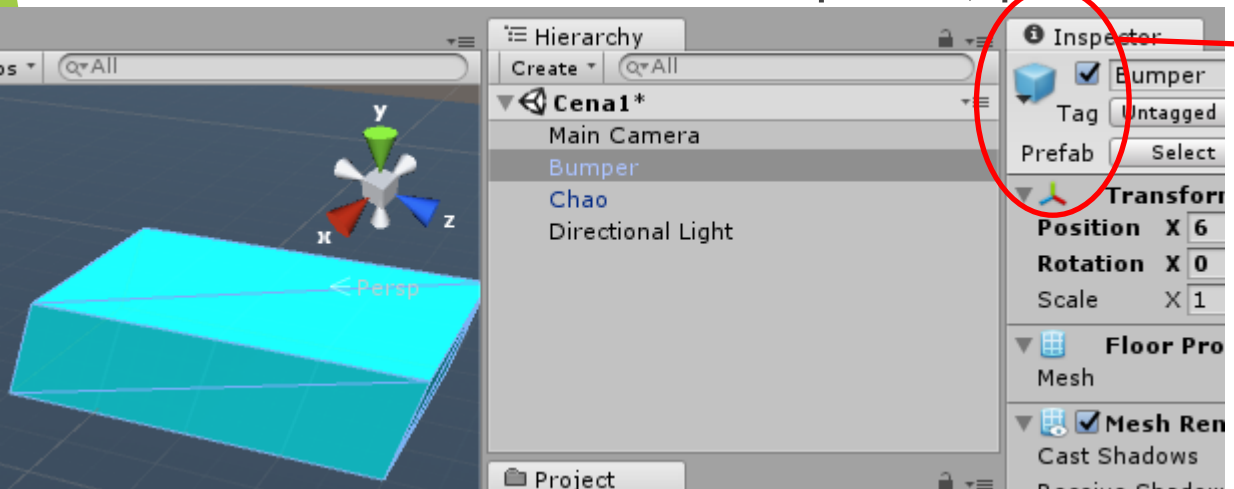


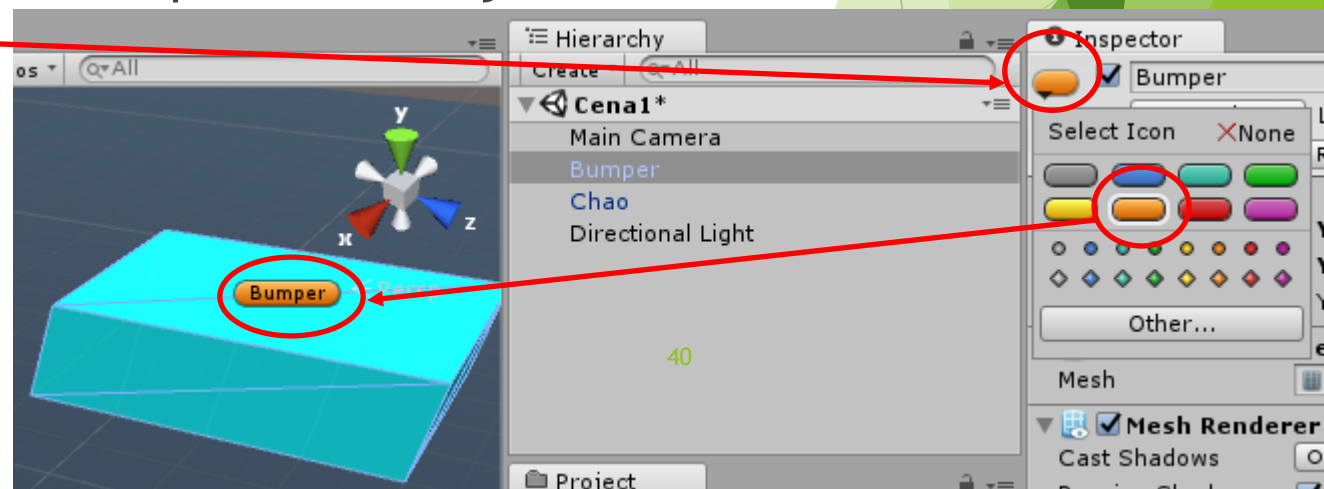
Roller Madness – Primeira fase - Prefabs

- Criemos, também, um Prefab para o Chao, mas agora de outra maneira:
- Clique no título da pasta Prefabs que usamos acima, e selecione o botão [+] abaixo da palavra Project e, em seguida, selecione [Prefab]. Aparecerá uma nova entrada na lista de prefabs. Chame-a de Chao.
- A cor cinza indica que nenhum objeto está associado ao prefab que, portanto, está vazio. A cor azul, como do prefab Bumper, indica que já há um objeto associado.
- Arraste o objeto Chao da Hierarquia para o prefab Chao, acinzentado, e ele passará a ficar azulado, indicando que um gameObject foi associado a ele e poderá ser usado como modelo para criação de outros “Chao”.
- Clicando na caixa azul no Inspector, poderá indicar o prefab do Objeto:



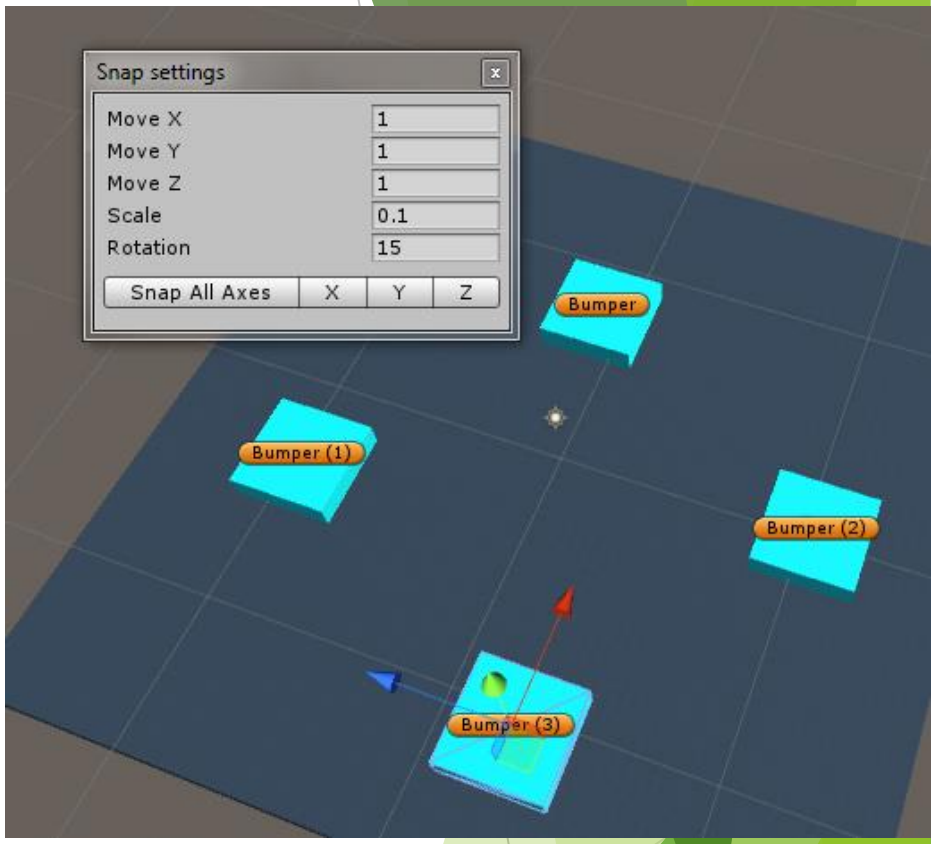


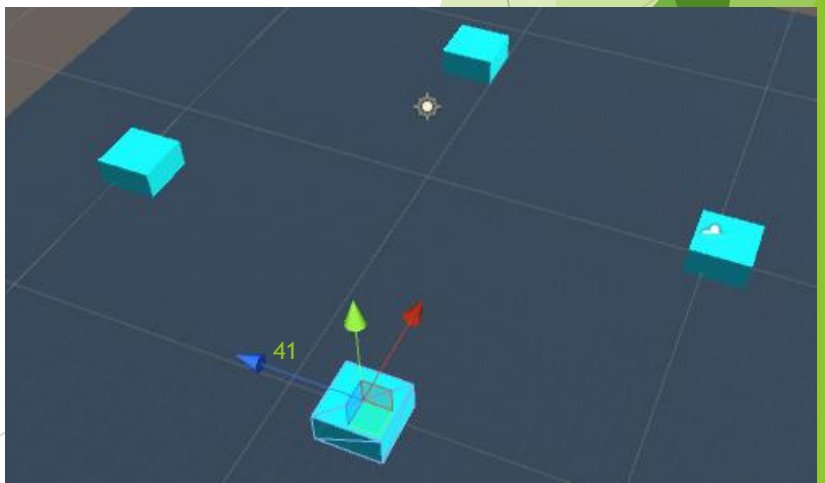




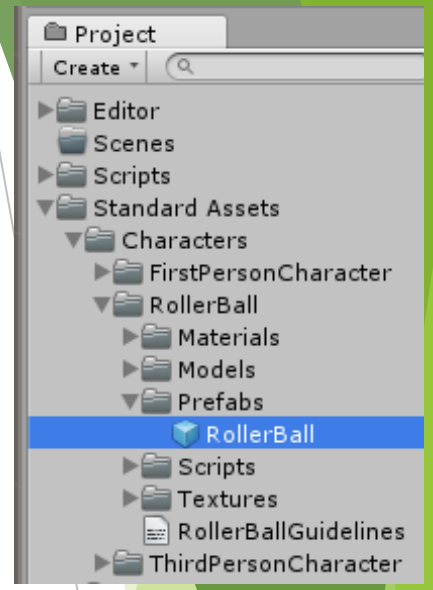
Roller Madness – Primeira fase - Prefabs

- Vamos arrastar mais alguns prefabs Bumper para a cena.
- Para posicioná-los com precisão, pode usar os Snap Settings como, por exemplo, Snap All Axes em múltiplos de 1, formando um padrão como o da figura:
- Em seguida, selecione o prefab Bumper na janela Projects para mudarmos sua escala, pois eles estão um pouco superdimensionados.
- Mude a escala em X e em Z para 0.5 metros, deixando a escala em Y valendo 1 metro.
- Esse objeto é um prefab. Caso você se arrependa e considere as alterações incorretas, basta pressionar as teclas Ctrl-Z.
- Caso considere que as alterações estão corretas e deseja que todas as instâncias do prefab passem a ter essa escala, saia da edição do Prefab na janela Hierarchy, como vimos no desenvolvimento 2D.
- As alterações serão propagadas aos Bumpers da cena.

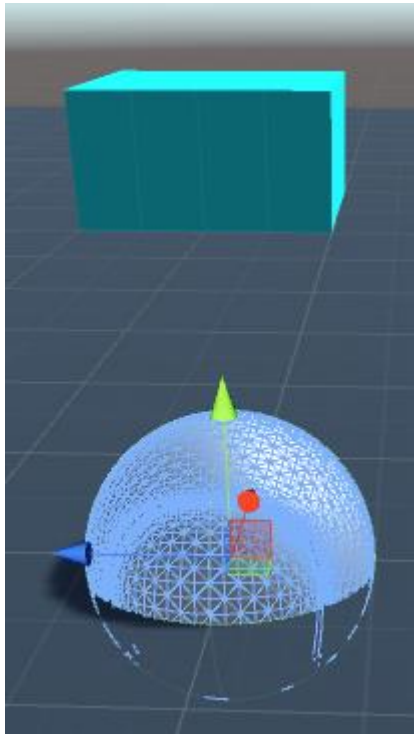




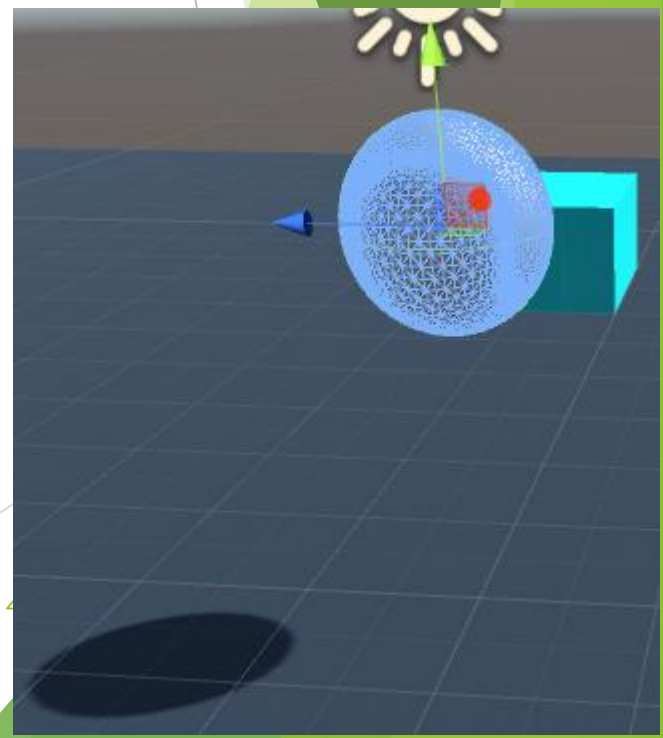
Roller Madness – Primeira fase – Player Character



- Para nosso Player Character (PC) usaremos um Asset Padrão.
- Na janela Project, abra as pastas Standard Assets, Characters, RollerBall e Prefabs.
- Arraste para a cena o prefab RollerBall.
- No Inspector, clique nos três pontinhos de Transform e, logo depois, em Reset para zerar o transform da esfera que aparecerá na cena, de forma que ela fique no centro do nosso mundo de jogo.

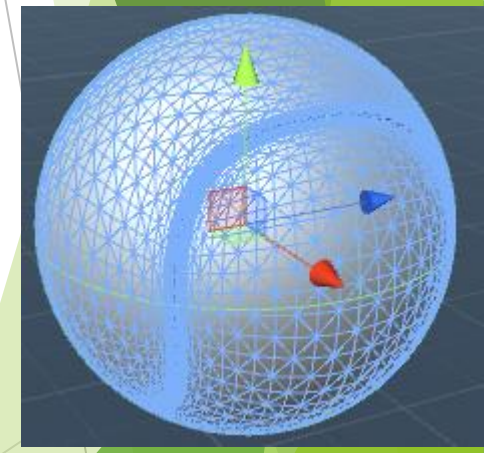


- Mude a posição Y da esfera para 2, de maneira que ela fique suspensa acima do chão.
- No inspector dessa esfera, observe os scripts já prontos : Ball e Ball User Control. Eles recebem os inputs do jogador e, com isso, movem e dão vida à bola.
- Salve e execute seu jogo, usando seu controlador de jogo ou as teclas de setas para mover a bola (espaço a faz pular).



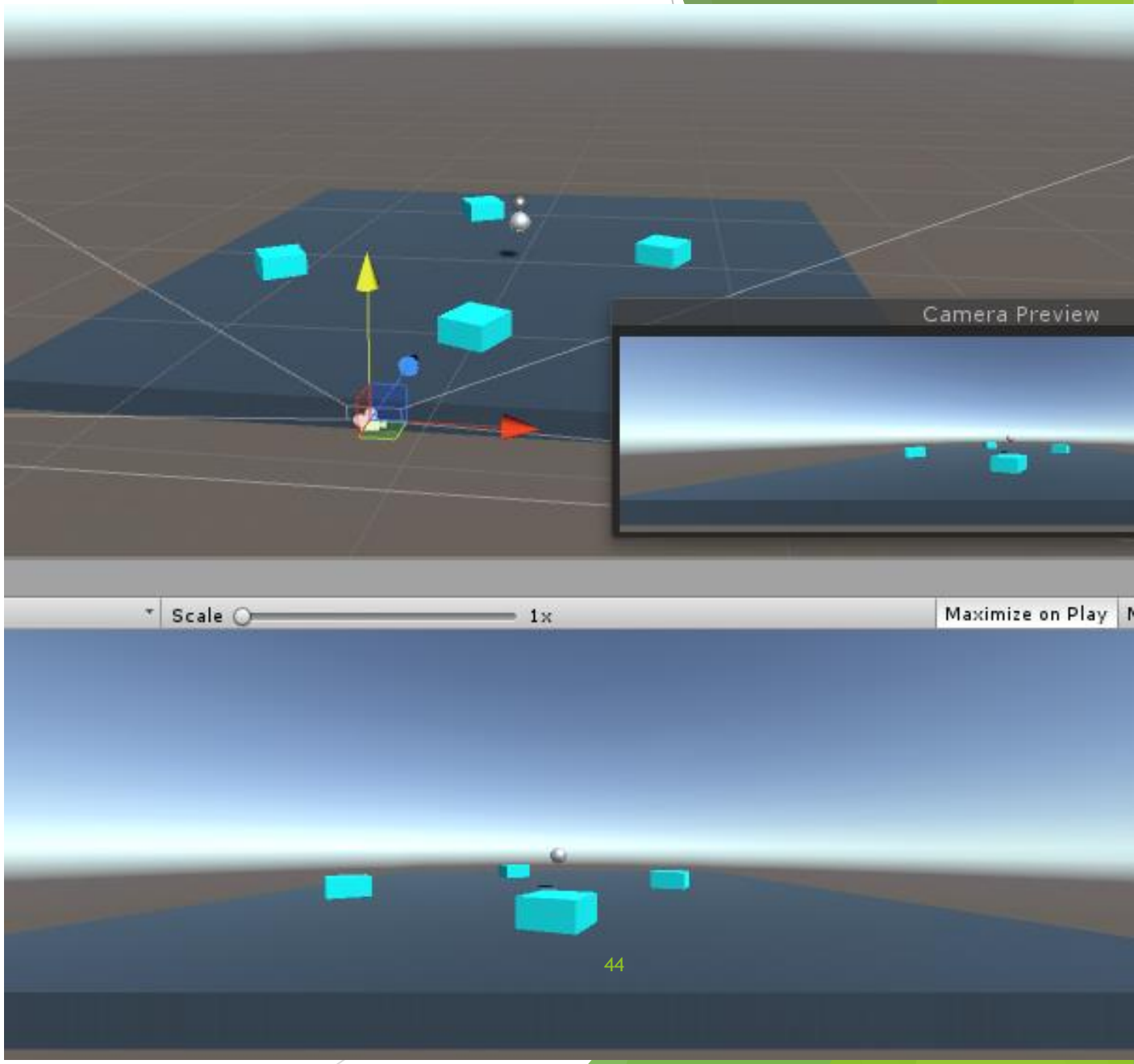
Roller Madness – Primeira fase – Câmera

- Nossos próximos objetivos são:
  - Posicionar a câmera
  - Fazer a câmera seguir nosso player character (a bola)
  - Modificar o fundo da tela visto pela câmera
  - Adicionar música à câmera
- Vamos posicionar a câmera. Clique na bola e pressione a tecla F para colocar esse objeto no foco.
- Observe o indicador de eixos da bola. O eixo z é o azul e sua seta aponta na direção positiva.
- O controlador funciona de maneira que, quando você pressiona o comando para ir para a frente, ele aplica uma força na direção positiva e a bola é empurrada na direção de z positivo.
- Assim, em essência desejamos que a câmera fique olhando na direção do eixo Z.
- Posicione a câmera um pouco atrás da bola, olhando para baixo, de forma que possamos ver nosso mundo.



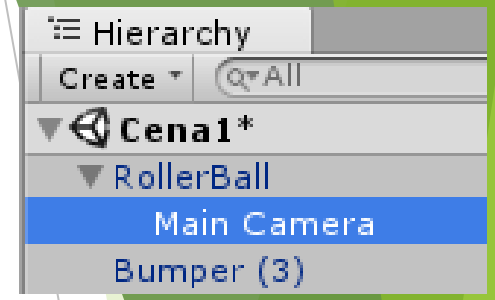
Roller Madness – Primeira fase – Câmera

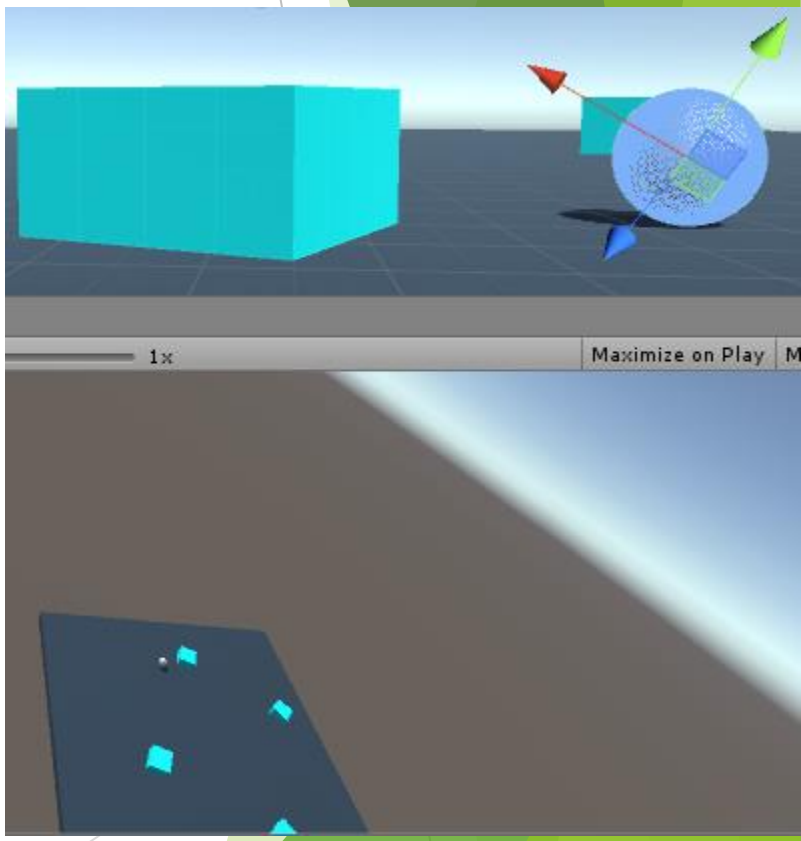
- Procure mover a câmera um pouco para trás e para o lado direito, de forma que a cena possa ser visualizada de forma semelhante à da figura ao lado:
- Agora que temos a visualização da cena de aspecto próximo ao que desejamos para o jogo, precisamos mudar a própria câmera.
- Selecione a câmera na hierarquia e ative a opção GameObjects | Align With View. Agora, a visualização [Game] ficará igual à da câmera na scene.
- Se executarmos o jogo, teremos essa visualização no modo de jogo, mas a câmera não segue a bola.
- Como fizemos no primeiro projeto, vamos subordinar a câmera à bola, arrastando-a na hierarquia para dentro do objeto Rollerball, criando um relacionamento pai-filho para os dois e fazendo a câmera seguir a bola conforme esta se move.



Roller Madness – Primeira fase – Câmera

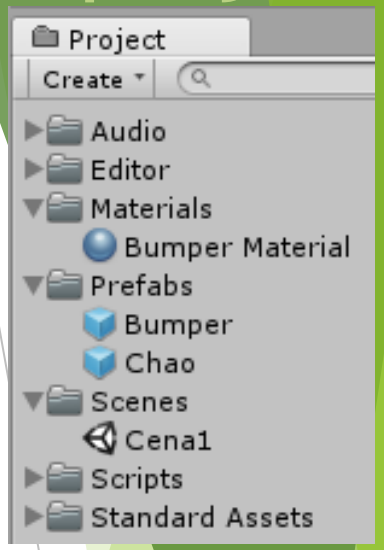
- Após fazer isso, salve e execute seu jogo.
- Observe que a câmera fica fora de controle. Isso ocorre porque ela está girando junto com a bola, que domina seus movimentos, pois a câmera está agora subordinada ao sistema de coordenadas da bola, o qual está girando continuamente conforme a bola gira.
- Pare a execução e recoloque a câmera fora da bola.
- Selecione a câmera e, em Standard Assets, abra a pasta Utility e inclua na câmera o script Smooth Follow. Configure a variável Target para apontar a RollerBall. Mude Distance para 6 e deixe Height valendo 5.
- Se você executar o jogo agora, verá que a câmera segue a RollerBall de maneira correta, pois está fora de seu sistema de coordenadas e segue a bola através do script Smooth Follow.
- Com a câmera selecionada, observe no Inspector a propriedade SkyBox. Mude Clear Flags para Solid Color e selecione uma cor vermelha escura (ou outra qualquer conforme seu gosto). Por exemplo, a cor (115, 0, 0, 0).
- A Câmera tem um componente Audio Listener e é um bom lugar para colocar sons, que não variarão com a distância à câmera. Clique em [Add Component], selecione Audio e Audio Source. Abra a janela Select AudioClip (círculo do lado direito da propriedade AudioClip e selecione a música Background. Mude **volume** para 0.25 e **loop** true. Execute o jogo.

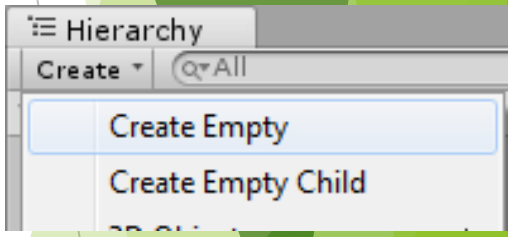


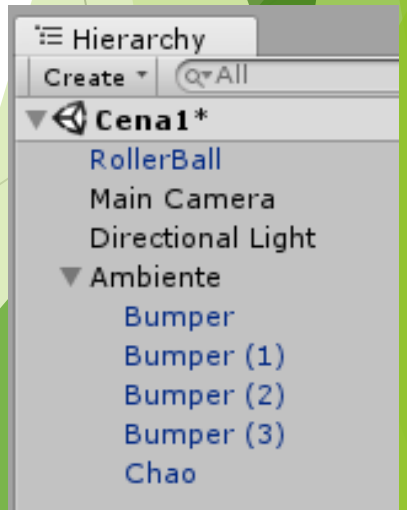


Roller Madness – Primeira fase – Organização do projeto

- Organizando Projects
  - Crie as seguintes pastas no Projeto : Scenes, Materials e Prefabs
  - Arraste para Materials o componente Bumper Material
  - Arraste para Prefabs os componentes Bumper e Chao
  - Arraste para Scenes o componente da nossa cena.
- Organizando Hierarchy
  - Do ponto de vista organizacional, uma relação pai/filho entre Game Objects é parecida a uma pasta
  - Podemos criar Objetos de jogo vazios (empty Game Objects) para atuar como pastas e, assim, organizar a hierarquia
  - Em Hierarchy, selecione [+] e [Create Empty], mude seu nome para Ambiente. Selecione a Engrenagem no Inspector e clique em Reset, para que o objeto fique no centro das coordenadas do mundo.
  - Arraste para dentro do Ambiente os 4 bumpers e o Chao.
  - Assim teremos nossos objetos de jogo organizados e agrupados.

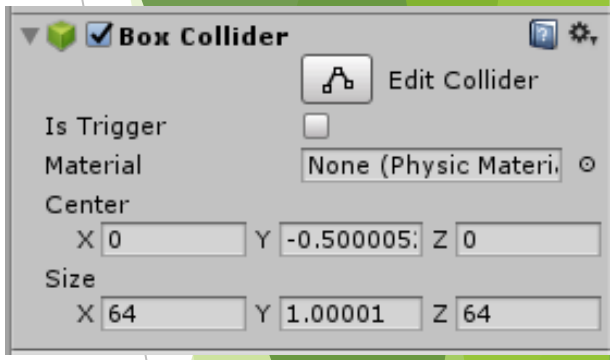


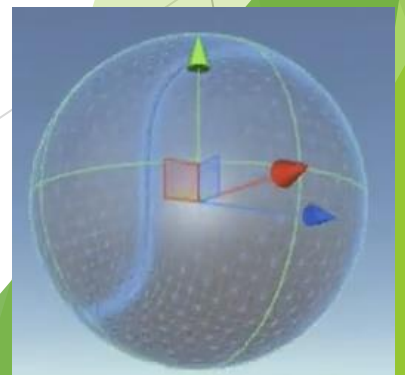




Roller Madness – Primeira fase – Física

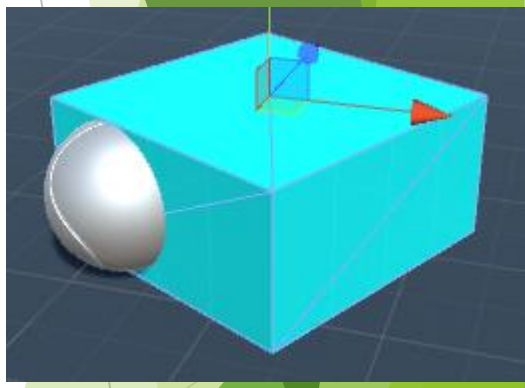
- Física é uma parte importante de muitos jogos. Unity tem duas engines de física:
  - Física 3D = PhysX Engine → é a que usaremos em RollerMadness
  - Física 2D = Box2D Physics Engine
- Nossos objetivos agora são:
  - Entender e modificar Colliders
  - Entender e aplicar RigidBodies
  - Fazer uso de Materiais Físicos
- Na cena, clique em Chao e observe que ele tem um Box Collider.
- Clique na RollerBall e observe que ela possui um Sphere Collider, que é parte do Prefab original que obtivemos dos Assets Padrão. Observe uma esfera com linhas verdes envolvendo a RollerBall. Essa é o Sphere Collider.
- Os bumpers também possuem Box Colliders que podem ser vistos quando clicamos um dos bumpers, como uma caixa de linhas verdes envolvendo o objeto.

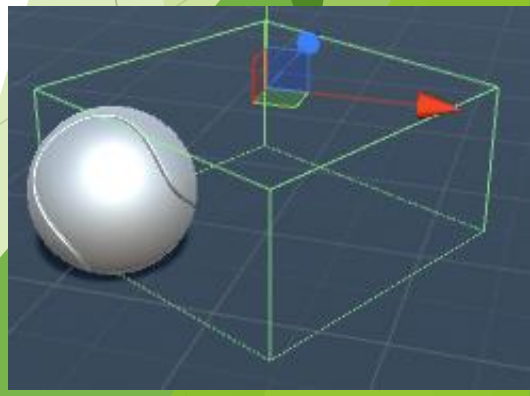




Roller Madness – Primeira fase – Física

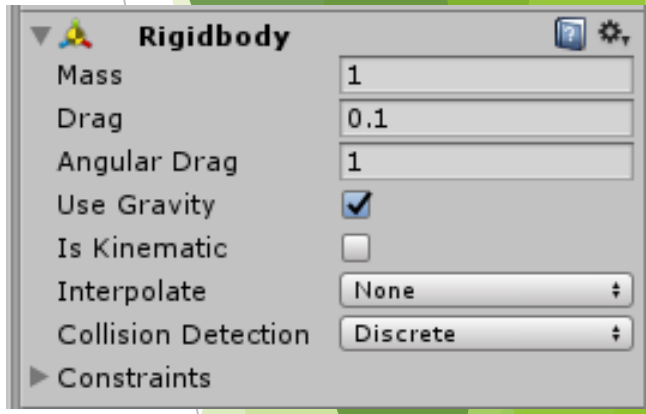
- O componente Collider é usado para detectar colisões entre game objects.
- Você pode modificar o tamanho do collider, pressionando o botão [Edit Collider] do Inspector e movendo os pequenos quadrados verdes que aparecerão ou, então, usando a propriedade Size do Collider no Inspector.
- Por default, Unity não permite que Colliders se sobreponham.
- Portanto, quando você faz a Rollerball bater em um bumper, ela não o atravessa.
- Se você ligar a opção Is Trigger de um Collider, ele deixará de usar a detecção de colisão. Fazendo isso num Bumper, ele permitirá que a RollerBall passe dentro dele, pois não mais impedirá a sobreposição, embora ainda detecte colisões. Inclusive podemos criar um evento OnTrigger em um script para tratar essa situação.
- Por exemplo, o bumper poderia ser um objeto invisível na cena mas que, quando tocado pelo PC, executaria outras ações, como abrir uma porta, ou mudar a iluminação da cena. Para tornar um objeto invisivel, desative seu Mesh Renderer no Inspector.





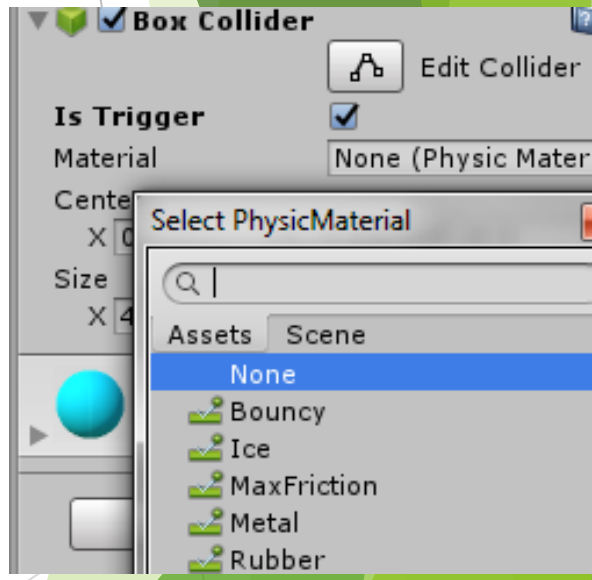
Roller Madness – Primeira fase – Física

- Se você colocar Is Trigger true para o Chao, ao executar o jogo verá que a bola passa por ele, caindo sob a ação da gravidade. No entanto, os bumpers não passaram pelo chão e cairam eternamente, ficaram onde estavam compondo a cena.
- Por que isso aconteceu? Observe a Rollerball e note que ela também possui um componente chamado RigidBody.
- Rigidbodies permitem que colliders sejam afetados por Física.
- Assim, como a Rollerball possui um RigidBody e uma e suas opções é Use Gravity, que está ligada, de forma que a Rollerball recebe uma força para baixo.
- Os bumpers não tem RigidBodies. É interessante colocar RigidBodies em componentes que possuam colliders e estejam se movendo de alguma maneira, como a Rollerball.
- Outra opção do Rigidbody é Is Kinematic. Quando ligada, essa opção faz com que o objeto não seja afetado pela Física, o que pode ser útil se você desejar mover o objeto pelo seu **transform**.
- Rollerball será afetada pela Física, portanto deixaremos Is Kinematic desligado. Desligue também Is Trigger do Box Collider do Chao, para que Rollerball não passe por ele.



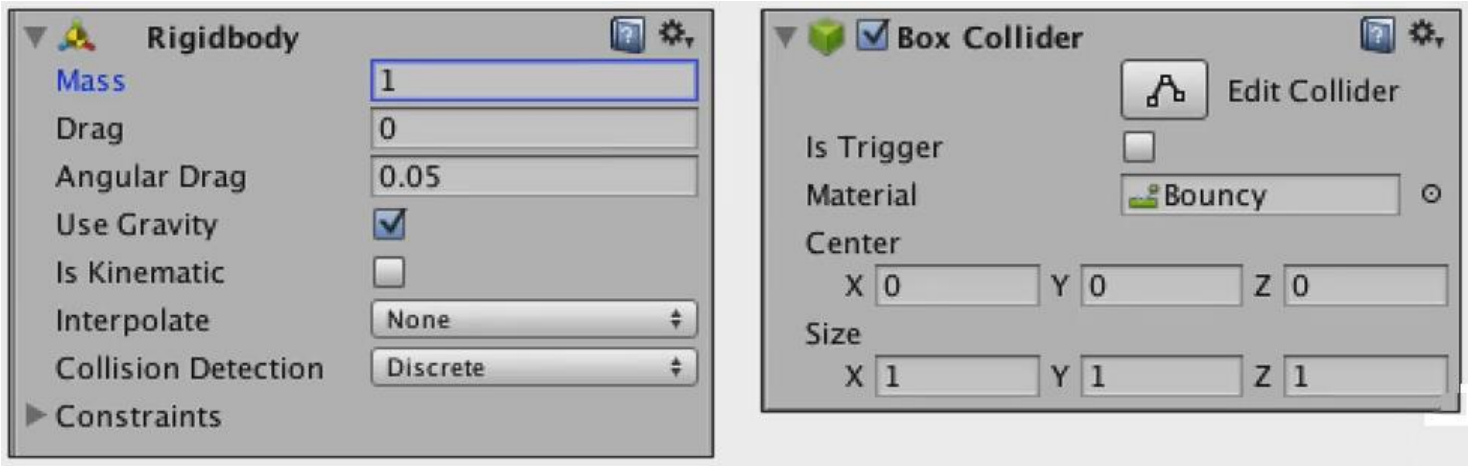
Roller Madness – Primeira fase – Física

- No Box Collider da Rollerball, observe que ela possui uma propriedade Material igual a Rubber. Quando essa bola bate em algum objeto, ela age como uma bola de borracha e volta para trás.
- Há vários tipos de Material, você pode testar o jogo com diversos e ver qual fica mais adequado à nossa bola.
- Selecione um dos bumpers e observe que ele não tem um material associado ao box collider. Clique no círculo à direita e selecione o material Bouncy. Execute o jogo e veja como a bola é rebatida quando bate no Bumper modificado.
- Pressione [Apply] no Inspector desse bumper para que todos os prefabs sejam atualizados com esse material.



Revisão de Física no Unity

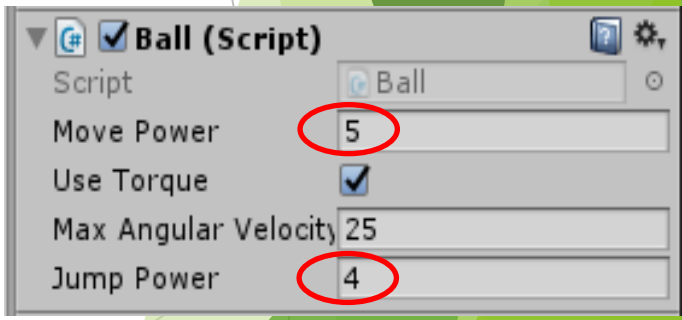
- Física é criada no Unity através de dois componentes aplicados aos game Objects:



- RigidBody habilita física num game object, incluindo gravidade.
- Você pode especificar propriedades físicas do objeto, como massa.
- Se um Collider é ligado ao game object, o RigidBody também poderá detectar colisões entre o objeto atual e outros.
- O Collider, mesmo sendo invisível para a câmera, determina o formato do game object e como ele interage com a física.
- Se Is Trigger for verdadeiro, os objetos podem passar através um do outro, mas o jogo pode detectar colisões (através de scripts) e atuar de acordo.
- Colliders podem ter um material físico ligado a eles para modificar suas características durante colisões, como rebatimento ou deslizamento.

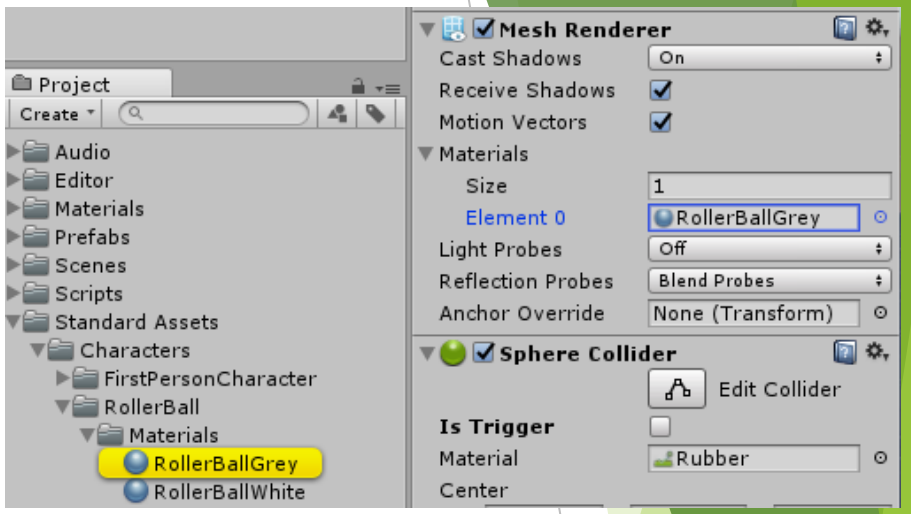
Roller Madness – Controle do Jogador

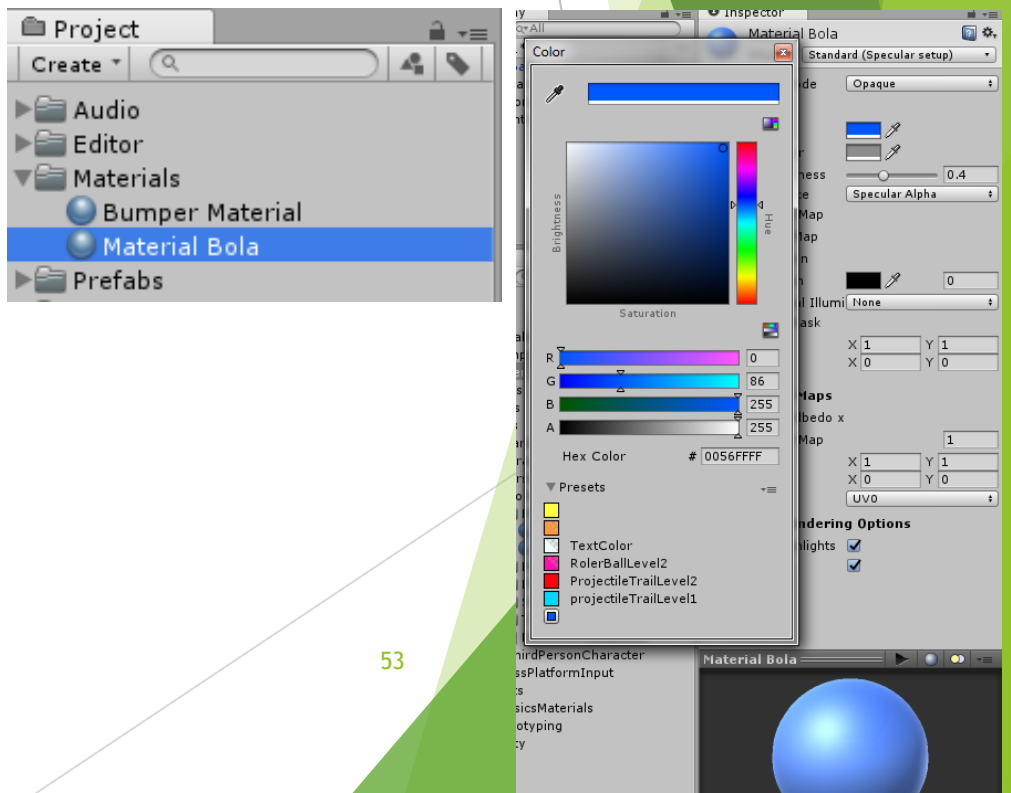
- Vamos aprimorar o movimento e a aparência de nosso jogador (a bola)
- Modificaremos a velocidade da bola, seu material e colocaremos um rastro que a segue
- O objeto RollerBall é um Asset Padrão, com características “físicas” e scripts prontos que facilitaram e tornaram mais veloz o desenvolvimento do jogo até o momento.
- Os scripts estão codificados para permitir a leitura dos valores de deslocamento pelo teclado (teclas de setas, WASD e controle de jogo), possibilitando várias experiências de jogabilidade.
- Observe no Inspector o script Ball. Ele possui uma variável chamada MovePower que define a quantidade de força aplicada ao movimento da bola.
- JumpPower indica a quantidade de força aplicada quando a bola pula. Se você não quiser que a bola do jogo pule, basta mudar esse valor para 0.
- Mude MovePower para 100 e JumpPower para 1, para que o movimento da bola fique mais controlável.
- UseTorque permite à bola derrapar enquanto a movemos. De certa maneira isso ajuda a controlá-la. Deixar UseTorque falso faz com que a bola role sem atrito e sua velocidade aumente muito, ficando fora de controle.



Roller Madness – Aparência do Jogador

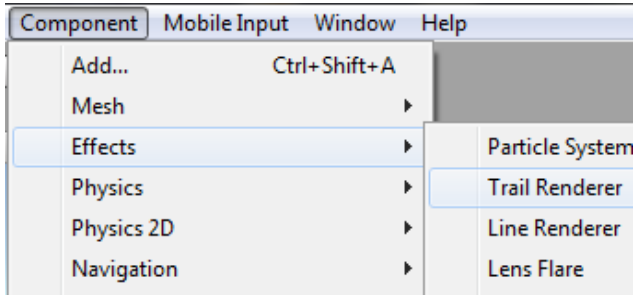
- Clique na RollerBall na Hierarquia. Observe no Inspector a propriedade Mesh Renderer e, dentro dela, Materials. Clique na indicação RollerBallGrey e observe que o material é cinzento. Observe também que esse material é exibido na janela Project, dentro dos materiais dos Assets Padrão (pasta Characters | Materials).
- Clique em RollerBallGrey na janela Project e tecle [Ctrl-D] para duplicar esse material. Mude seu nome de RollerBallGrey 1 para Material Bola.
- Arraste Material Bola para nossa pasta Materials, fora dos Standard Assets.
- No inspector desse material, clique no Color Pick do Albedo e digite, em Hex Color, 0056FFFF (ou outra cor que goste).
- Em seguida, clique no Color Pick de Specular, e digite 00C0FFFF no Hex Color, de forma que o material fique espelhado com cor Aqua Blue.
- Arraste esse material para o objeto RollerBall na Hierarquia, para que ele assuma a nova aparência.

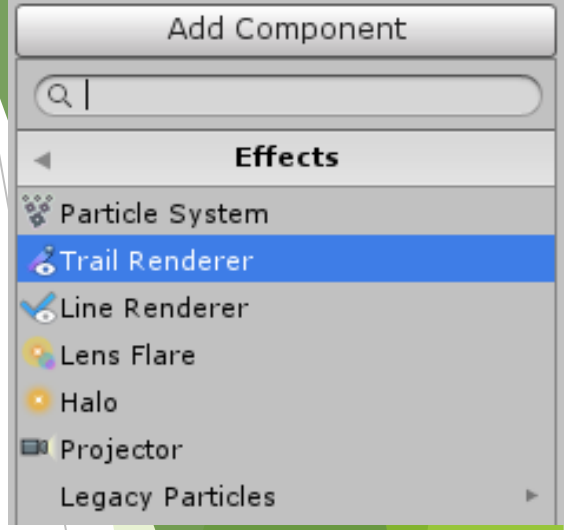




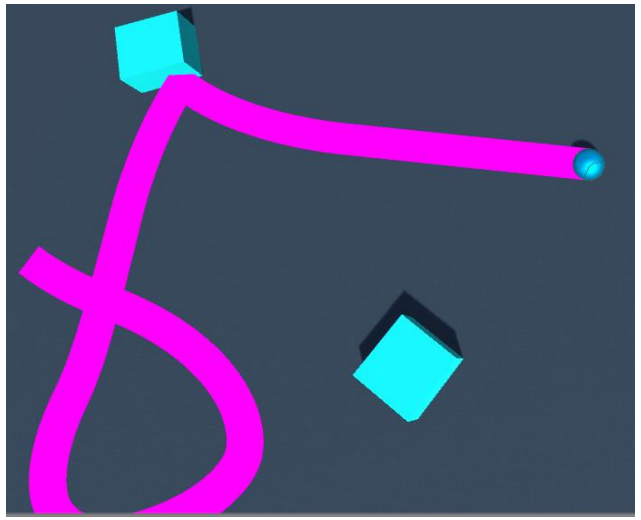
Roller Madness – Aparência do Jogador

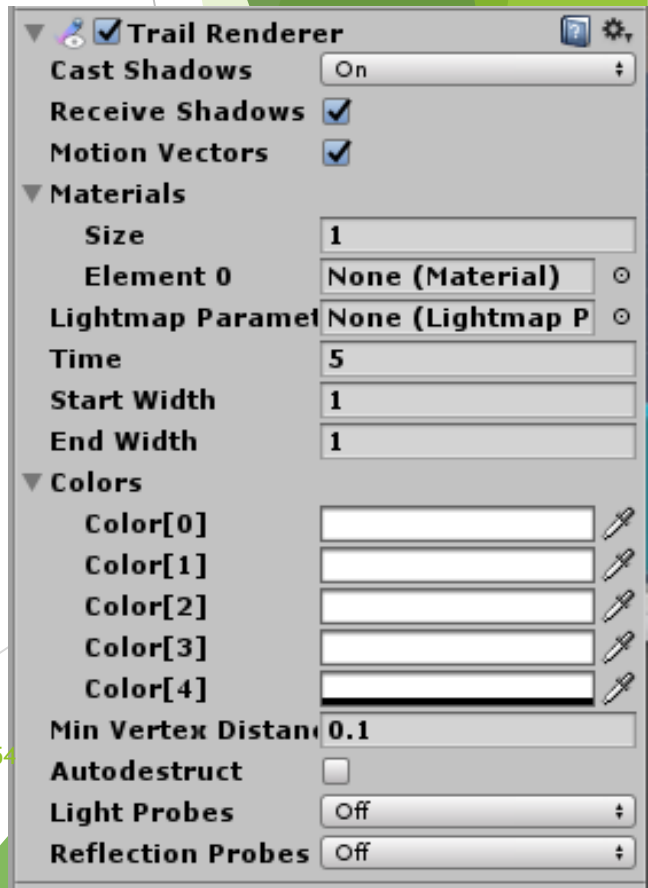
- Vamos agora adicionar um Trail Renderer (gerador de rastros) à bola.
- Com a bola selecionada na Hierarquia, clicamos em [Add Component] no Inspector, em seguida em Effects e escolhemos Trail Renderer. Ou fazemos isso pelo menu Component.





- Um Trail Renderer basicamente gera um rastro atrás de um game object conforme este se move na cena.

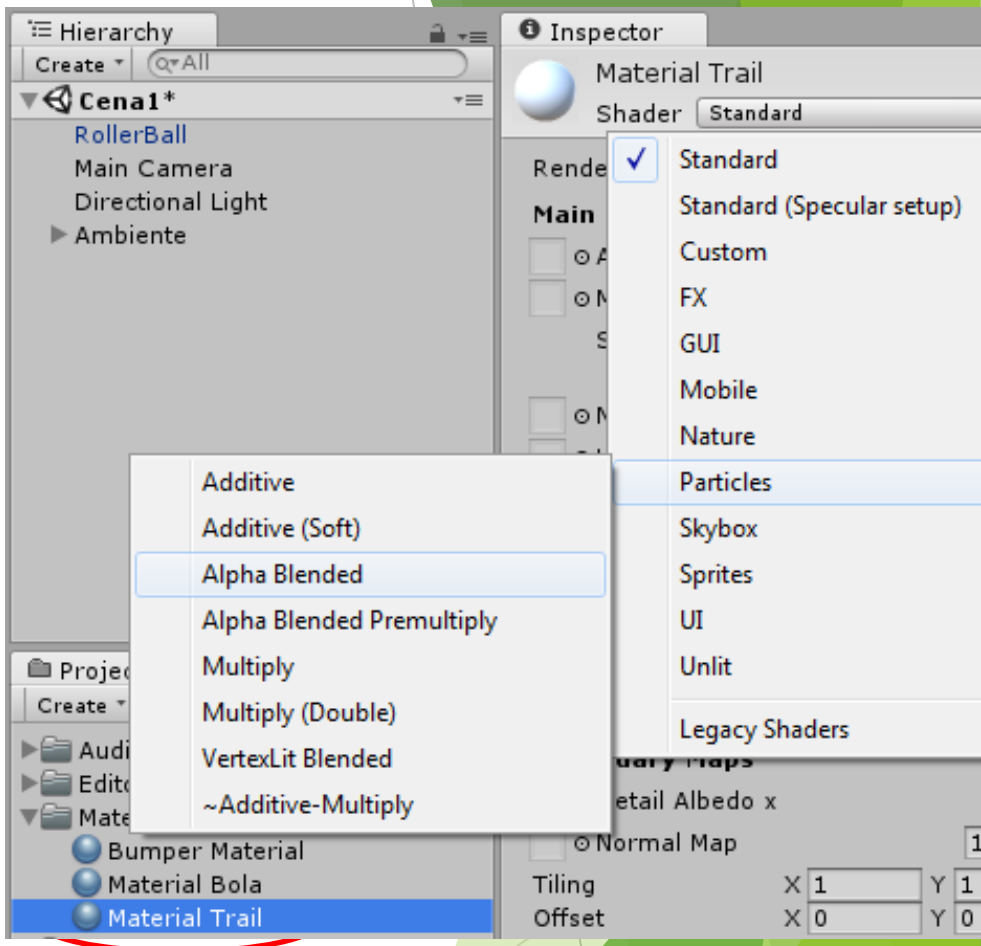


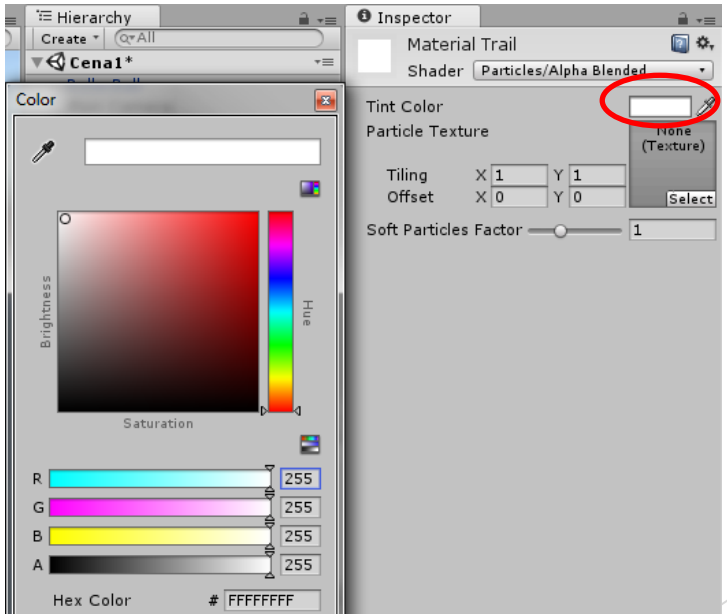


- Observe quantas propriedades esse componente possui. Ainda não o configuramos para ficar com uma boa aparência, mas vamos fazê-lo.

Roller Madness – Aparência do Jogador

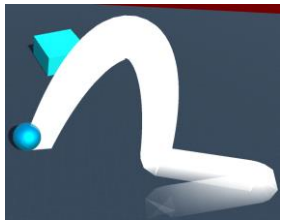
- Uma das propriedades mais importantes é o material usado para gerar o rastro.
- Vamos criar um novo material para o rastro. Clique no botão Create da janela Project, escolha a opção Material e chame o material criado de Material Trail.
- Este será um material bastante básico, pois mudaremos apenas seu Shader.
- No Inspector desse material, mude a opção Shader de Standard para Particles e selecione Alpha Blended.
- Mude Tint Color para branco (255, 255, 255, 255).



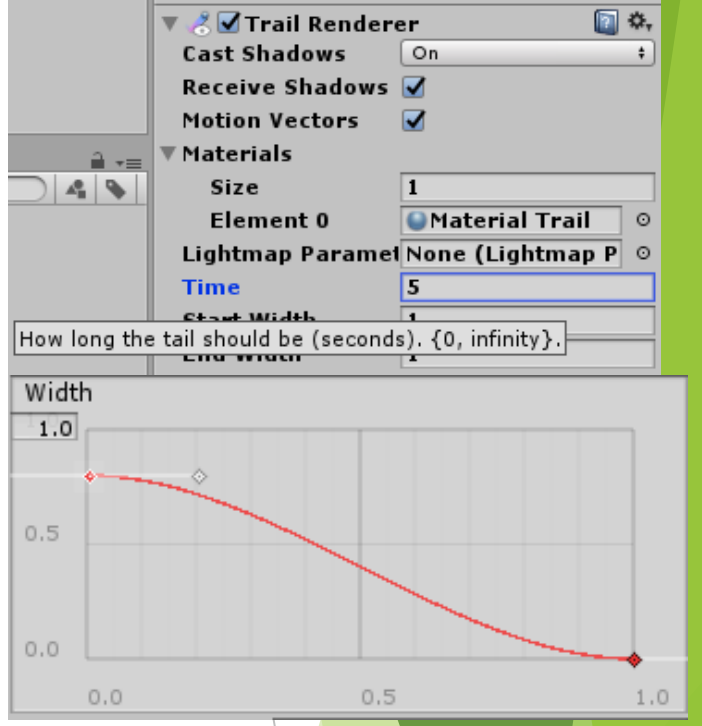


Roller Madness – Aparência do Jogador

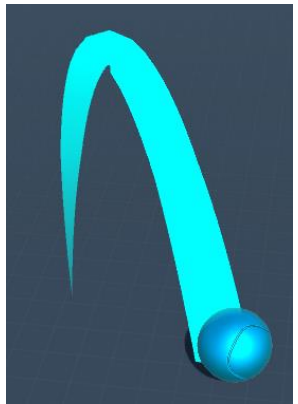
- Agora devemos arrastar o Material Trail para a propriedade Material do Trail Renderer.

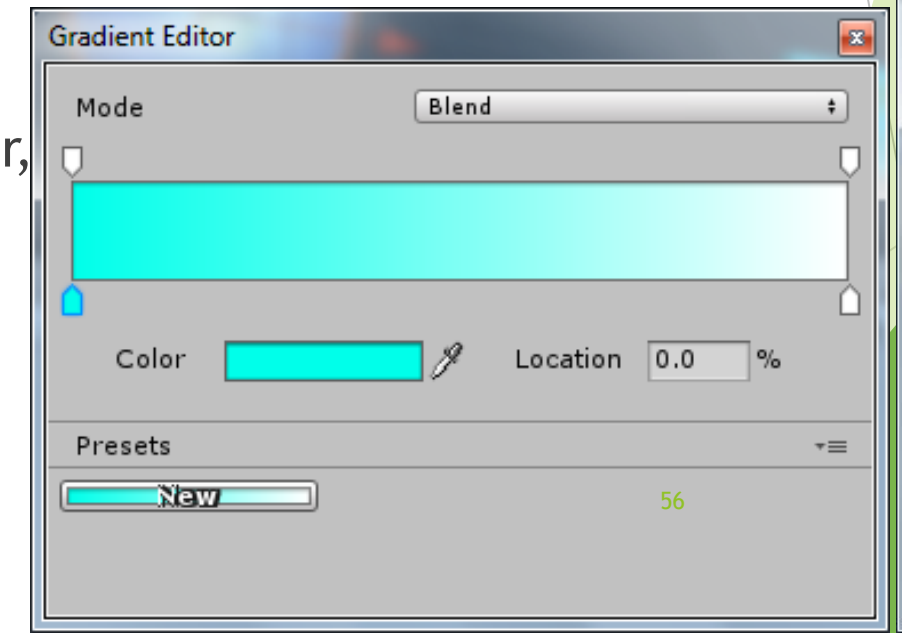


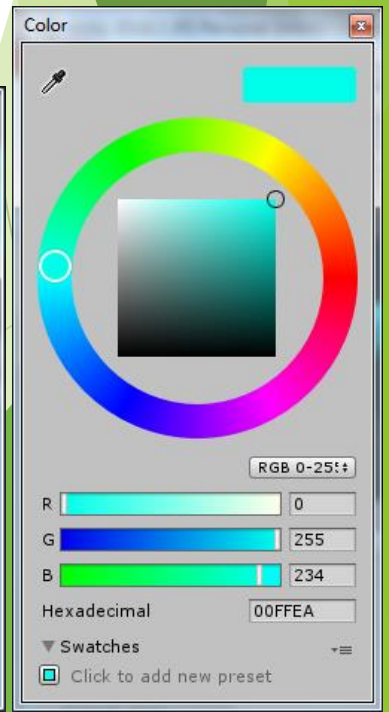
- O efeito do rastro mudou ao executar.
- Altere Time para 2 segundos.
- Na propriedade Width, clique na linha vermelha com o botão direito para selecionar a opção Add Key e crie pontos de interpolação da curva de diminuição da largura do rastro. Ao lado foi criado um ponto ao final da curva e puxado para baixo, resultando na curva apresentada.



- Assim, o rastro durará 2 segundos, começará com largura 0.8 e terminará com largura 0, de forma que aparecerá diminuindo até sumir.
- Por fim, mudaremos as cores do rastro durante sua existência.
- Clicando na propriedade Colors do Trail Renderer, aparecerá a janela Gradiente Editor. Clique no Color Pick e escolha a a cor 00FFEA na janela Color



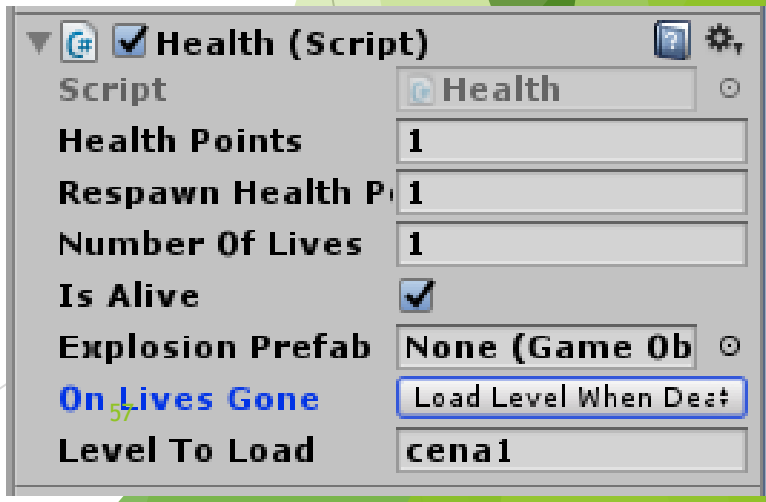




- A ferramente montará o gradiente.
- O efeito será o da figura ao lado.

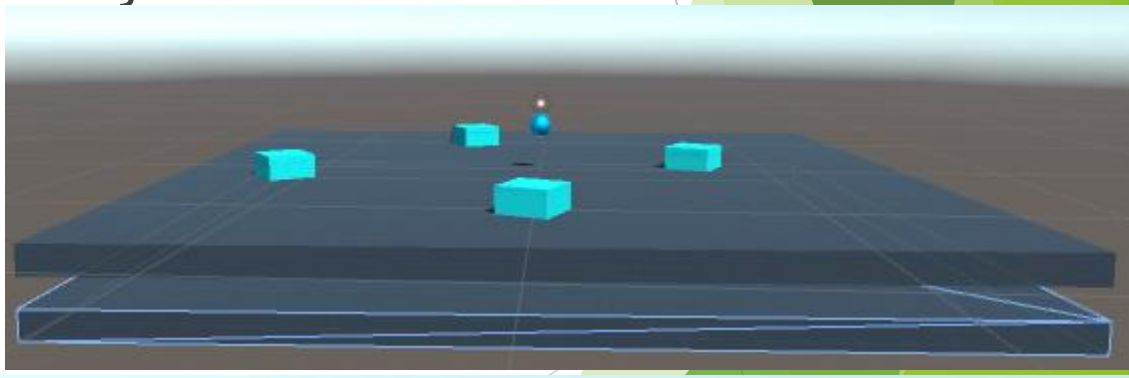
Roller Madness – Saúde e Danos

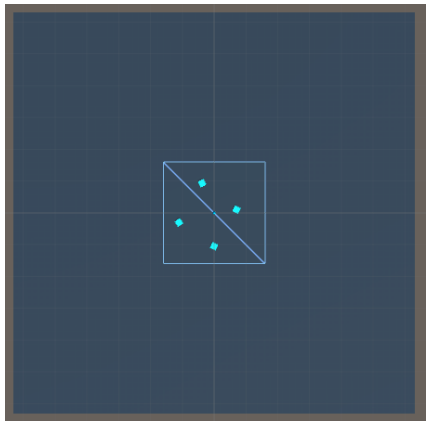
- Neste momento, se a bola cair para fora do chão, ela o fará para sempre. O ideal seria terminar ou reiniciar o jogo se isso acontecer.
- Vamos adicionar um Rastreador de Saúde e uma Zona de Danos que danifica o jogador se ele cai nela. A Zona de Danos será um game object que matará o jogador se ele colidir com ela.
- Para isso, temos dois scripts já prontos :
  - Health → rastreia a saúde de um personagem do jogo
  - Damage → danifica a saúde do personagem.
- Adicione o script Health à Rollerball. Esse script possui variáveis que informam HealthPoints do jogador, quantos HealthPoints ele terá se for gerado novamente (respawn), número de vidas. No momento, vamos manter esses valores com 1, pois o jogo será simples e se você morrer, você morreu.
- Mude a propriedade OnLiveGone para LoadLevelWhenDead, de forma que, quando o PC morrer, recarregaremos a cena.
- Em LevelToLoad, digite **cena1**, que é o nome da nossa cena atual, que será recarregada em caso de morte do PC.

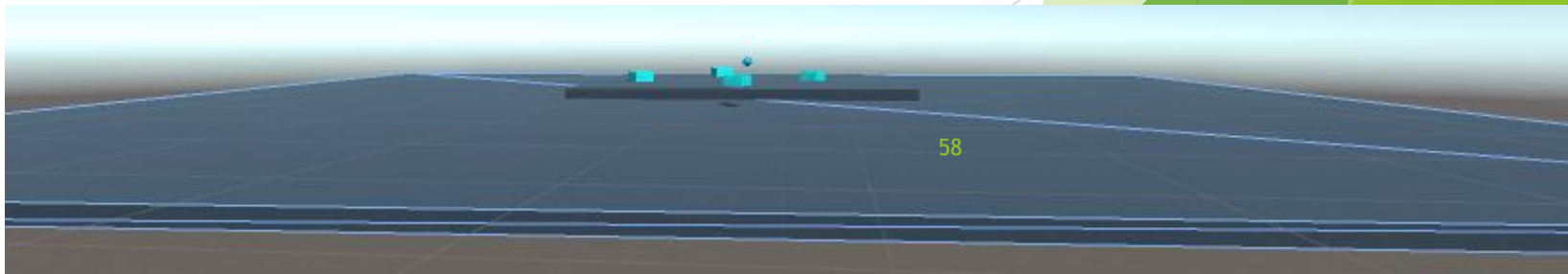


Roller Madness – Saúde e Danos

- Agora precisamos de algo que detecte quando o PC cai para fora do Chao e, então, realizar danos ao jogador.
- Há várias maneiras de conseguir isso, mas uma forma simples é criar um game object com o qual o jogador colida e, assim, dispare o evento que realiza danos.
- Clique no objeto Chao, pressione as teclas [Ctrl-D] para duplicá-lo. Mude o nome do objeto resultante para ZonaDeMorte.
- Esse objeto está, no momento, ligado ao prefab Chao. Vamos criar um prefab para ele, arrastando-o para a pasta Prefabs da janela Projects.
- Mude o transform Position do objeto ZonaDeMorte para (0, -2, 0), de forma que ele fique abaixo do Chao original. Mude sua escala para (2, 1, 2).

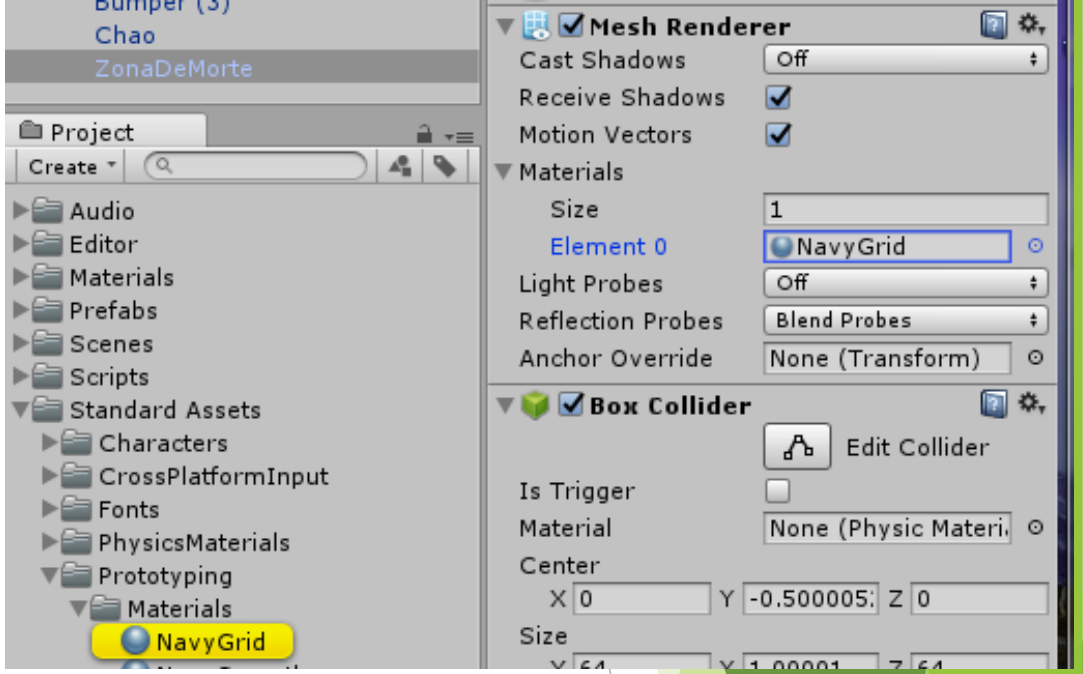


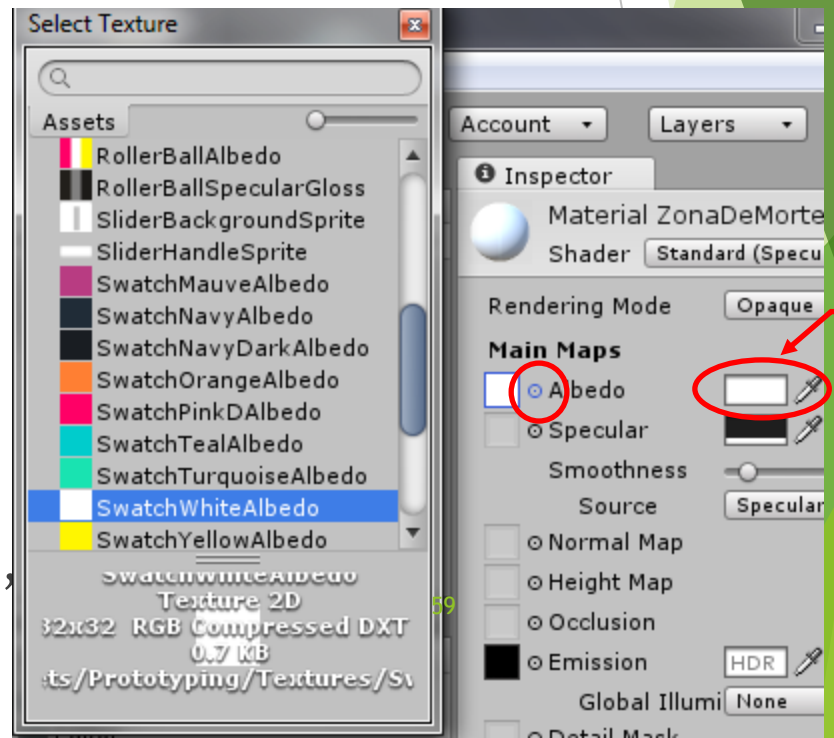




Roller Madness – Saúde e Danos

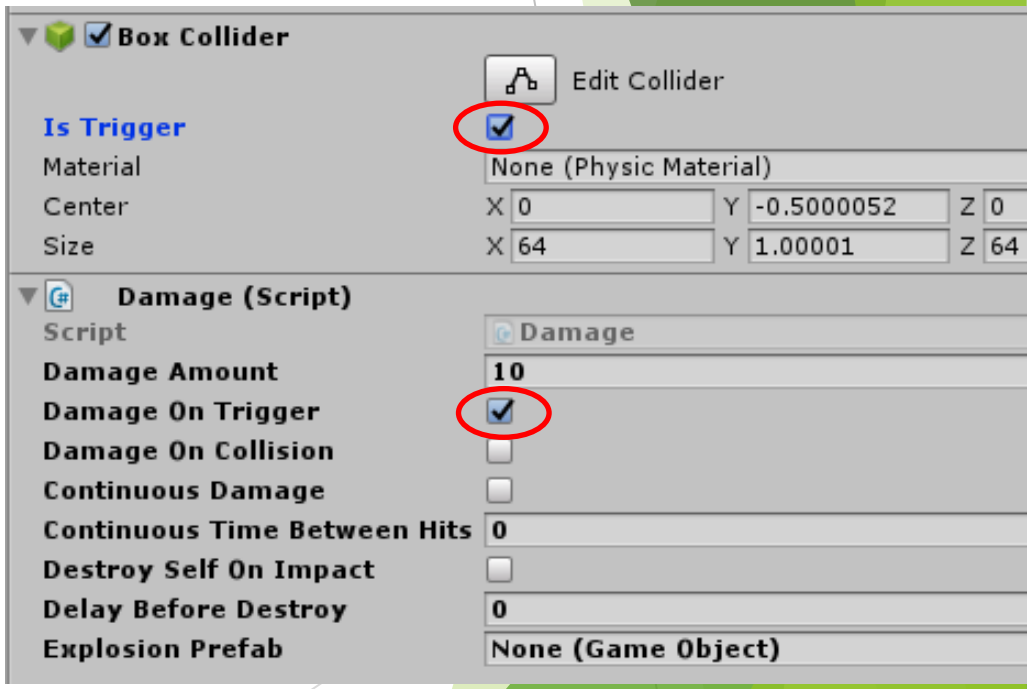
- Vamos criar um novo material para o objeto ZonaDeMorte, para diferenciá-lo do Chao.
- Clique na propriedade Mesh Renderer dele, acesse a propriedade Materials e veja que temos o Element 0 como NavyGrid. Clique em NavyGrid no Element 0 e aparecerá o Material na pasta Materials dos Standard Assets.
- Duplique esse material, chame-o de Material ZonaDeMorte e o arraste para a pasta Materials específica do nosso projeto, fora dos Standard Assets.
- Arraste esse novo Material para o objeto ZonaDeMorte, para que este passe a ter as características desse material (que modificaremos a seguir).
- Clique no Material ZonaDeMorte na pasta Materials, mude sua propriedade Albedo para SwatchWhiteAlbedo.
- Vá no Pick Color de Albedo e selecione uma cor amarela, por exemplo.





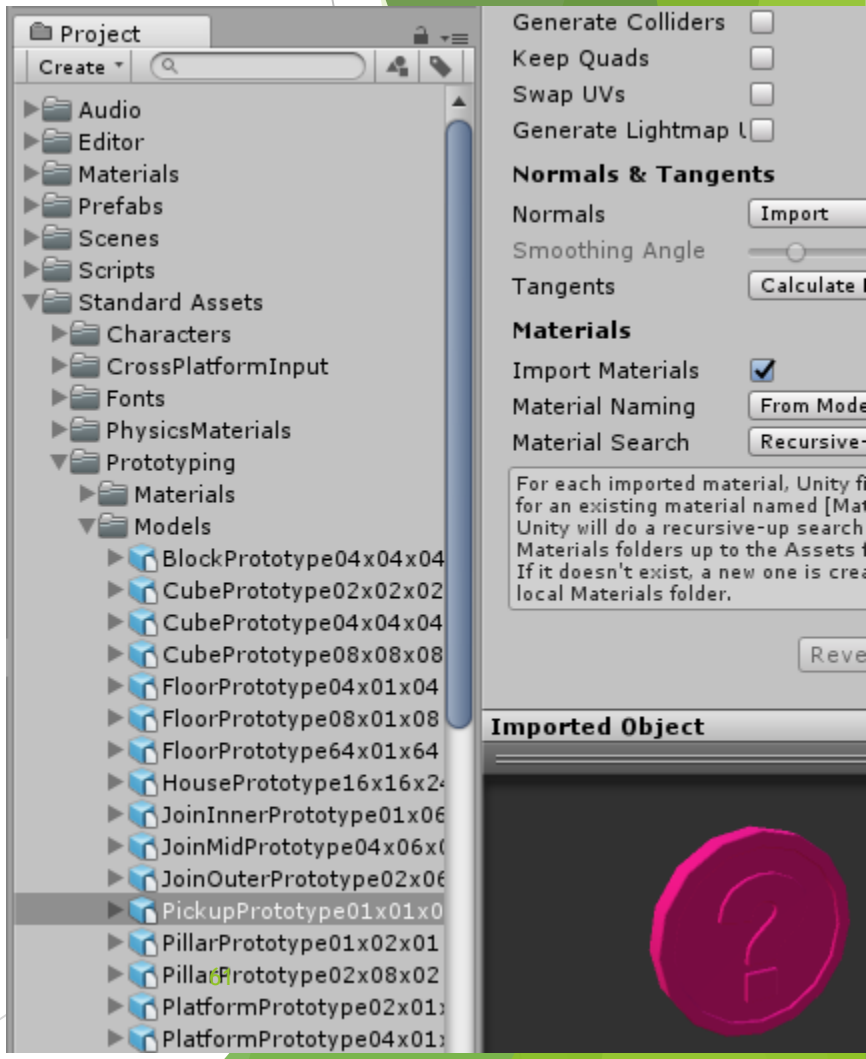
Roller Madness – Saúde e Danos

- Mude a propriedade Is Trigger do Box Collider de ZonaDeMorte para true, pois não queremos que a bola seja rebatida quando colidir com ZonaDeMorte, e sim queremos executar scripts quando ocorrerem colisões entre esse objeto e outros que o tocarem.
- Selecione o objeto ZonaDeMorte e adicione o script Damage a ele.
- Observe que a variável DamageAmount do script Damage vale 10, que é um valor maior que os HealthPoints do PC (a bola, cujo HealthPoints vale 1). Dessa maneira, o script subtrairá DamageAmount dos HealthPoints do objeto RollerBall e, como o resultado ficará negativo, matará a bola.
- A propriedade DamageOnTrigger deve valer true para que, no momento da colisão, o script danifique o objeto que colidiu com ZonaDeMorte.
- Ao executarmos o jogo, veremos que, quando a bola cai fora do Chao, ela se choca com a ZonaDeMorte e, nesse momento, a cena é recarregada, com a bola recomeçando no local original.



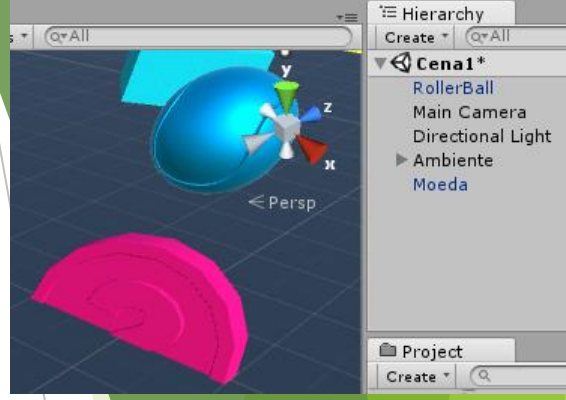
Roller Madness – Coleta de itens

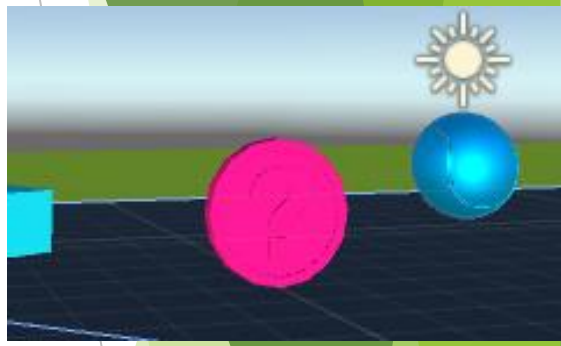
- O que é um jogo sem a capacidade de coletar coisas?
- Vamos criar objetos para que nosso jogador tenha algo para coletar e, portanto, algo para fazer no jogo.
- Esses objetos serão instâncias de um prefab de moedas.
- As moedas serão geradas e colocadas na cena.
- Reconheceremos a natureza de cada objeto do jogo através da propriedade Tag.
- Para criar nossas moedas, novamente usaremos os Assets Padrão de Prototipagem.
- Em Standard Assets | Prototyping | Prefabs, há um objeto PickupPrototype mas, por algum motivo, ele está vazio e não temos como usá-los.
- Por outro lado, em Standard Assets | Prototyping | Models, há um objeto PickupPrototype01x01x01 que pode ser usado para criarmos o prefab de moedas.
- Arraste esse objeto para a cena e chame-o de Moeda.

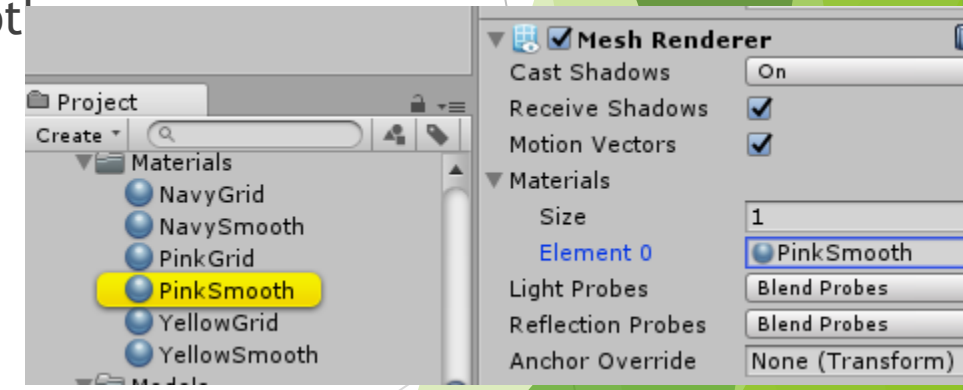


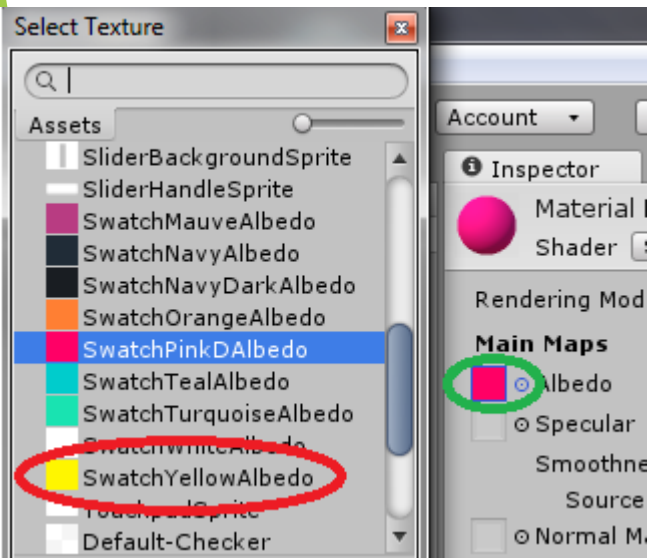
Roller Madness – Coleta de itens

- Faça o reset do transform da Moeda, de forma que ela fique na origem do sistema de coordenadas 3D.
- Pressionando a tecla [Ctrl], mude-a 0.5 unidade para cima e algumas para trás, de forma que fique sobre o Chao e afastada da bola.
- Ela ainda está grande em comparação com a bola, de forma que vamos mudar sua escala para (0.4, 0.4, 0.4) para que ela fique menor e mais difícil de ser tocada.
- Vamos mudar o material da moeda. Para isso, selecione a moeda e, no seu Inspector, clique em Mesh Renderer | Materials | Element 0. Note que aparece, na janela Project, o material do qual a moeda é "feita", PinkSmooth.
- Faremos como das outras vezes, vamos duplicar esse material.





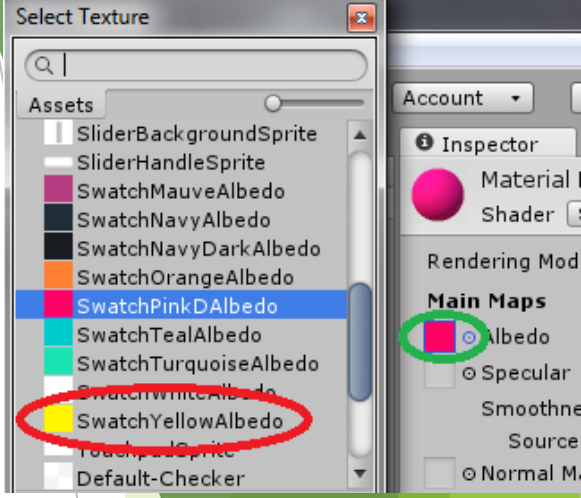


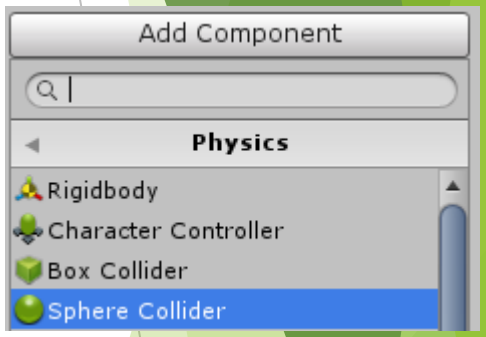


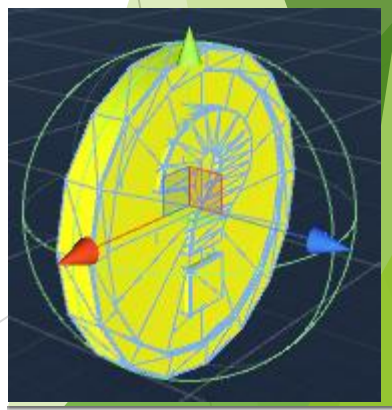
- O novo Material deve ser renomeado para Material Moeda.
- Arraste o Material Moeda para a pasta Materials do projeto.

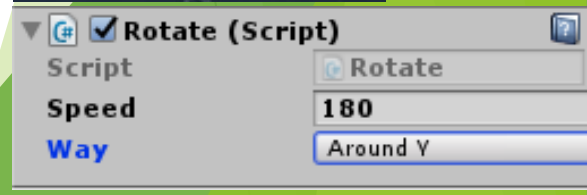
Roller Madness – Coleta de itens

- Selecione Material Moeda e, em seu Inspector, clique no círculo ao lado esquerdo da propriedade Albedo e o mude para SwatchYellowAlbedo.
- Arraste Material Moeda para o objeto Moeda na Hierarquia. A moeda ficará amarela.
- Gostaríamos que a moeda pudesse detectar colisões. Observe que no objeto Moeda não há um Collider, de forma que criaremos um.
- Selecione a Moeda na Hierarquia e, no seu Inspector, clique em [Add Component], selecione a opção Physics e, em seguida, Sphere Collider.
- A Moeda não é uma esfera, mas vamos detectar as colisões na área ao redor da moeda.
- Observe que, clicando em Sphere Collider no Inspector da Moeda, aparece uma esfera verde ao redor dela. Podemos mudar o tamanho do colisor para que ele fique um pouco mais que a moeda.
- Também será interessante que o player possa passar pela moeda ao coletá-la, de forma que vamos marcar **Is Trigger** do colisor como **True**.
- Vamos fazer a Moeda girar continuamente. Para isso, adicione o script Rotate após pressionar o botão [Add Component] |Scripts.
- Mude a variável **Way** para **AroundY** e **Speed** para **180** (graus por segundo).
- Execute o jogo para ver a moeda girando.



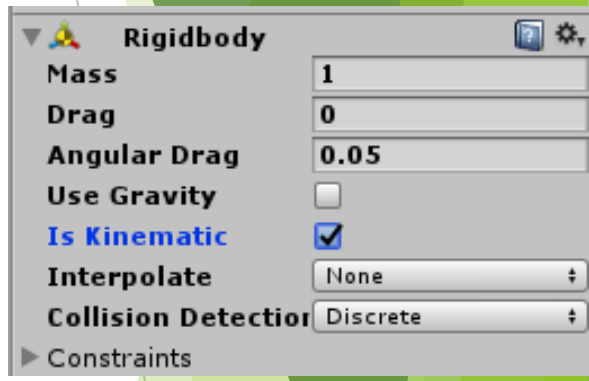


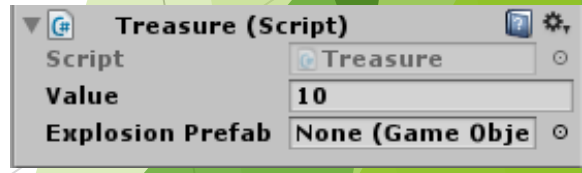


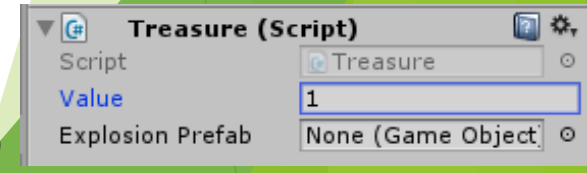


Roller Madness – Coleta de itens

- Vamos agora adicionar um RigidBody à Moeda ([Add Component] | Physics | RigidBody).
- A gravidade não irá atuar sobre a moeda mas, como ela será um objeto em movimento com um colisor, você sempre terá necessidade de um RigidBody para que o sistema de física saiba que esse é um objeto dinâmico no mundo do jogo.
- Deixe a propriedade **Use Gravity** como **false**.
- Nosso objeto Moeda está sendo dirigido apenas pelo script Rotate, que faz com que o transform desse objeto rotacione.
- Como não estamos aplicando forças ou torque à Moeda, apenas rotacionando-a por meio de um script, vamos deixar a propriedade **Is Kinematic** igual a **true**.
- A próxima atividade que faremos sobre nossa Moeda é adicionar um script que permita ao jogo saber que ela é um objeto a ser coletado (pick up), ou seja, que a moeda é um "tesouro".
- Assim, clique em [Add Component], selecione Scripts e adicione o script chamado **Treasure**. Mude o valor de Value para 1, ou seja, uma moeda equivale a 1 ponto.
- Execute o jogo e observe que, quando o player (a bola) colide com a moeda (o tesouro), a Moeda desaparece, ou seja, foi coletada.







Roller Madness – Coleta de itens

- O script Treasure basicamente verifica se o outro objeto que colidiu com o objeto que o possui (a Moeda) é o player. Se o outro objeto for o player, a Moeda é coletada e se destrói, saindo da cena.

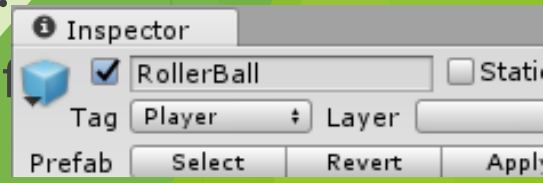
```
public class Treasure : MonoBehaviour {

    public int value = 10;
    public GameObject explosionPrefab;

    void OnTriggerEnter (Collider other)
    {
        if (other.gameObject.tag == "Player") {
            if (GameManager.gm!=null)
            {
                // tell the game manager to Collect
                GameManager.gm.Collect (value);
            }
            // destroy after collection
            Destroy (gameObject);
```

Evento que trata Colisões em Is Trigger = true

Este parâmetro referencia o outro objeto que colidiu com o dono do script.

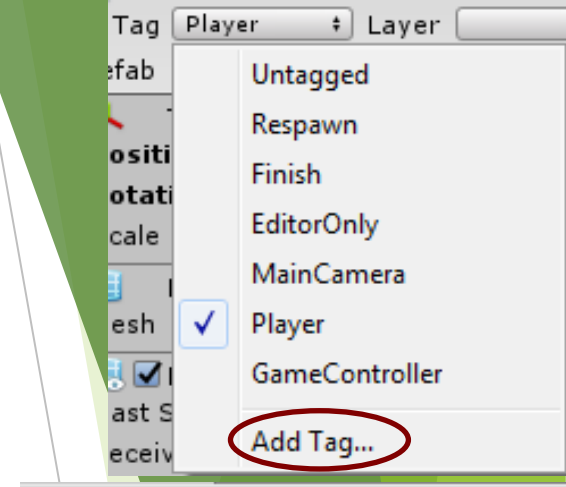
GameManager é um script que estudaremos depois e que gerenciará a mecânica do jogo

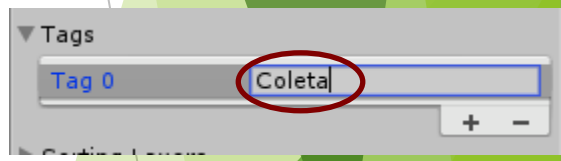
- Como o objeto Moeda sabe que o outro objeto que colidiu com ele é o Player?
- Observe que a RollerBall tem a propriedade Tag configurada como Player. O if acima compara se o objeto do parâmetro other tem tag igual a “Player”.

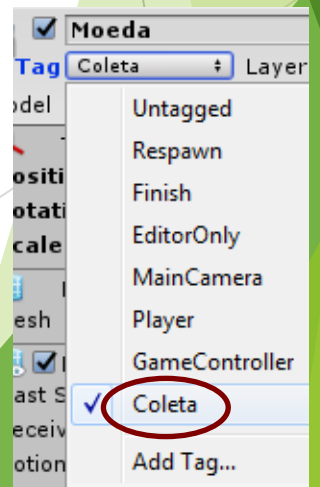


Roller Madness – Coleta de itens

- O uso da propriedade Tag é uma forma poderosa de trabalhar nos scripts, para determinar o tipo de atividade esperada de um objeto. Você pode criar suas próprias tags, mas já existem alguns pré-definidos, como vemos ao lado.
- Seu código poderá responder aos objetos do jogo de acordo com seus nomes, mas às vezes temos várias instâncias do mesmo objeto (por exemplo, várias moedas) no jogo e o uso das tags torna o processamento mais geral e unificado para cada tipo de objeto.
- Vamos colocar uma tag em nossa moeda. Para isso, clique na Moeda, abra a lista de tags no Inspector e clique em Add Tag… . Em seguida, entramos no editor de Tags e Layers. Pressione o símbolo + e digite o nome Coleta, que é um nome mais genérico que Moeda.
- Em seguida, clique na Moeda novamente e mude sua tag para Coleta, que passou a aparecer na lista suspensa de tags.

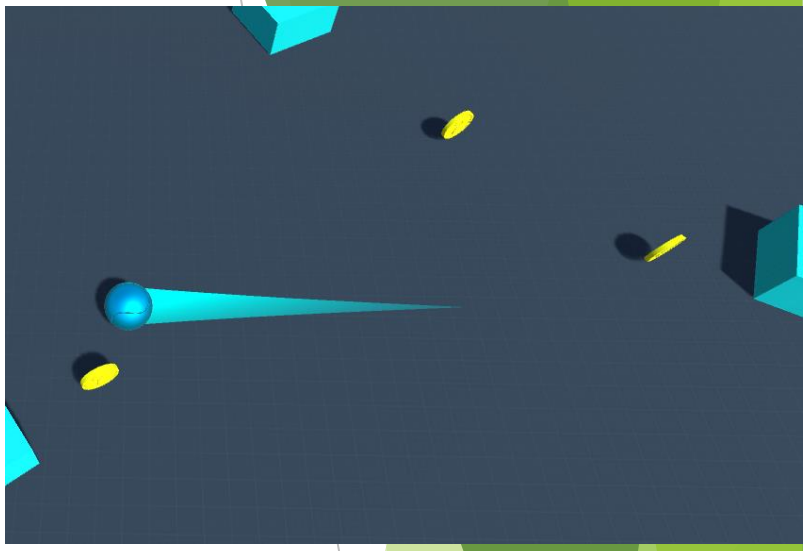






Roller Madness – Coleta de itens

- Temos nossa Moeda configurada. Como teremos várias moedas no jogo, é interessante torná-la em um Prefab.
- Arraste a Moeda para a pasta Prefabs da janela Project e ela virará um Prefab.
- Arraste o Prefab Moeda para a cena algumas vezes, para povoá-la. Ajuste suas posições para que fiquem ao redor da bola, cercando-a mas não muito perto.
- Organize a Hierarquia do seu jogo.
- Crie um objeto vazio, faça o reset de seu transform e chame-o de Moedas
- Arraste todas as moedas para dentro dele.
- Execute seu jogo e colete todas as moedas.
- Em seguida, criaremos um painel para exibir a quantidade de moedas coletadas, bem como adicionar uma tela de Game Over.



Na execução no Editor do Unity, a recarga da cena atual poderá voltar escurecida. Para ver como impedir isso, clique aqui.