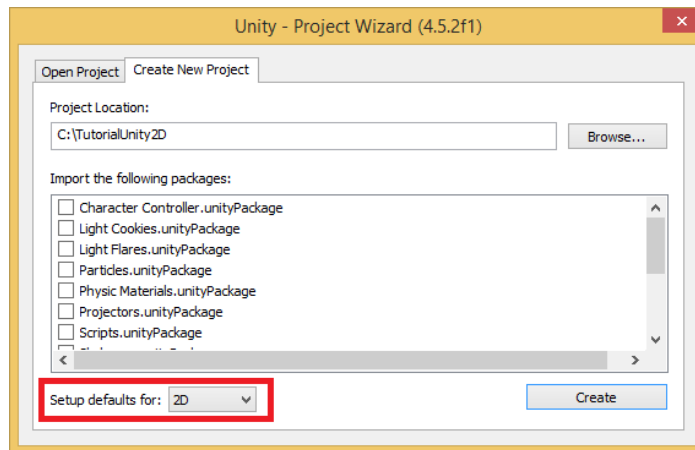


LABORATÓRIO: CRIANDO O JOGO **MEGAMAN 2D**

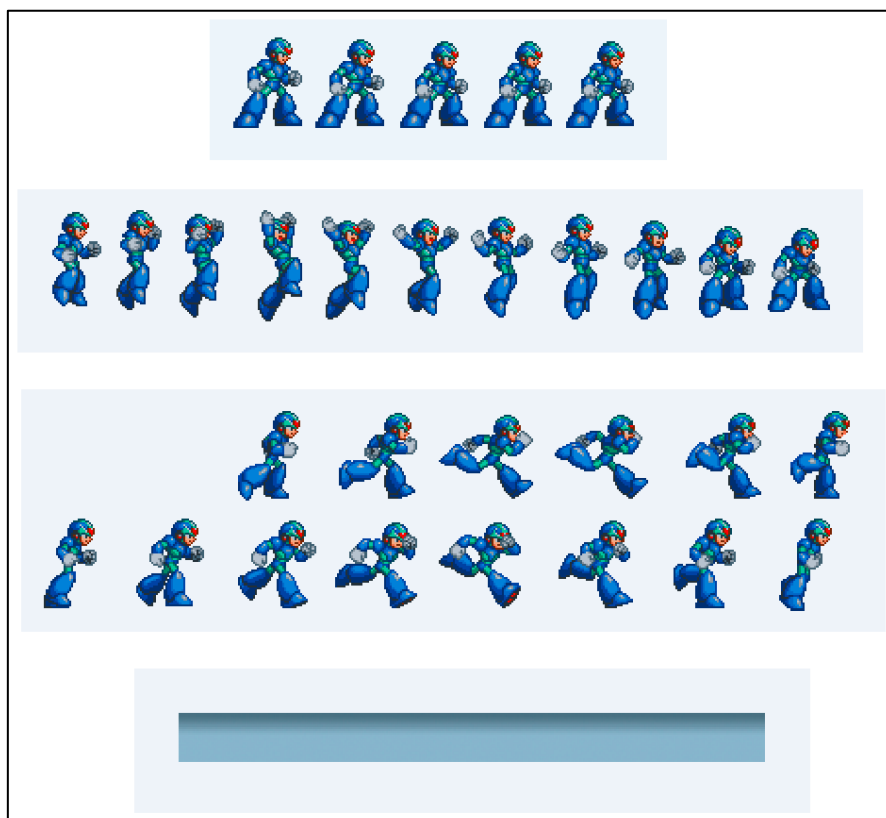
Parte-1

1. Crie um novo projeto **2D**.

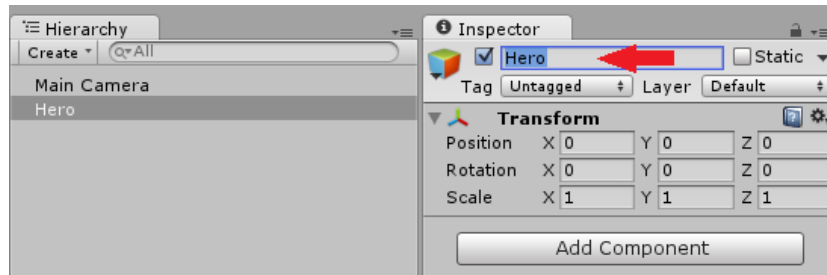


2. Criar as seguintes pastas na janela Project: `_scenes`, `_scripts`, `_sprites` e `_animations`.
3. Importe para o projeto, dentro da pasta `_sprites`, os arquivos de imagens disponibilizados no site do professor para esta aula.
4. Dentro de cada arquivo estão disponíveis uma ou mais imagens, as quais podem ser associadas aos `gameObjects`, podendo também serem utilizadas para criarmos as animações dos personagens do nosso jogo.

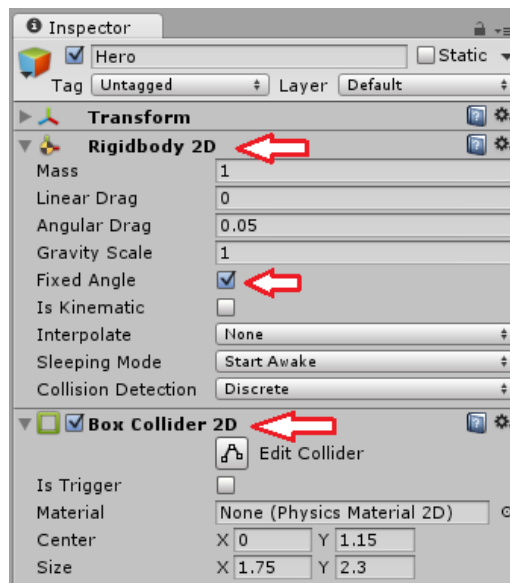
Na área de *Game Design*, tais arquivos de imagens são conhecidos como *Sprites Sheets*, ou somente *Sprites*. A seguir estão as imagens inseridas nos arquivos, sendo que as três primeiras possuem imagens que serão usadas na animação de cada “estado” do personagem (*idle* = parado, *jump* = pulando e *run* = correndo) e a última será usada como chão do jogo (*floor*).



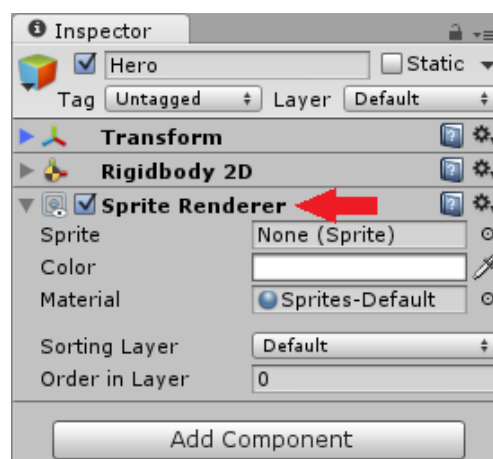
5. Para dar vida ao nosso personagem, vamos primeiro criar um *gameObject* Vazio (*Game Object* -> *Create Empty*) e renomeá-lo para *Hero*.



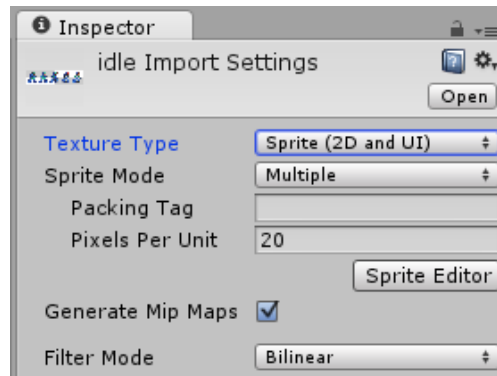
6. Agora vamos informar ao *Unity3D* que o objeto *Hero* vai ser afetado pela física, adicionando a ele um componente *Rigidbody* (botão *Add Component* -> *Physics 2D* -> *Rigidbody 2D*). Marque a propriedade *Fixed Angle* do *Rigidbody* para evitar que o nosso personagem gire na tela. Insira também um componente *BoxCollider* (*Add Component* -> *Physics 2D* -> *Box Collider 2D*).



7. Agora vamos adicionar em *Hero* um componente *SpriteRenderer*, que é necessário para renderizarmos os nossos *sprites* na tela do jogo (botão *Add Component* -> *Rendering* -> *Sprite Renderer*).

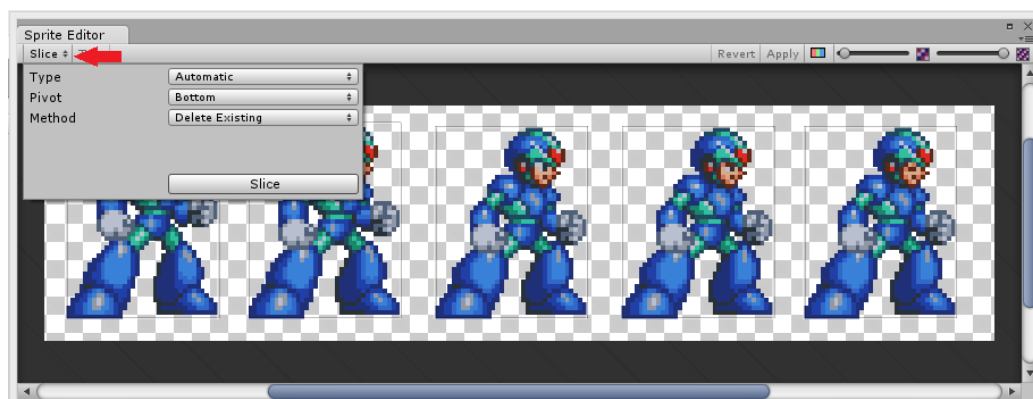


8. Selecione na janela *Project* o *sprite idle* e defina para ele as seguintes propriedades na janela *Inspector*:
 - a. *Texture Type*: certifique-se de selecionar a opção “*Sprite (2D and UI)*”;
 - b. *Sprite Mode*: mude para “*Multiple*”, pois o *sprite* possui várias imagens;
 - c. *Pixels per Unit*: mude para “*20*”. Aqui definimos quantos *pixels* do *sprite* correspondem a 1 unidade de medida do mundo do jogo.

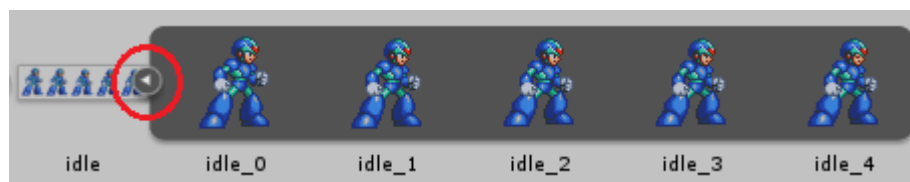


9. Agora clique no botão *Sprite Editor* da janela *Inspector*. Será exibida a janela de edição de *sprites* e será nela que iremos separar as imagens que formarão a animação do personagem.

Clique na opção *Slice* (canto superior esquerdo) e defina a propriedade *Type* para “*Automatic*” e *Pivot* para “*Bottom*”. Clique no botão *Apply* e perceba que as imagens serão recortadas automaticamente, ficando com o *pivot* (posição de referência entre os *sprites*) localizado na parte de baixo e centralizado em cada recorte. Feche a janela e salve a cena.

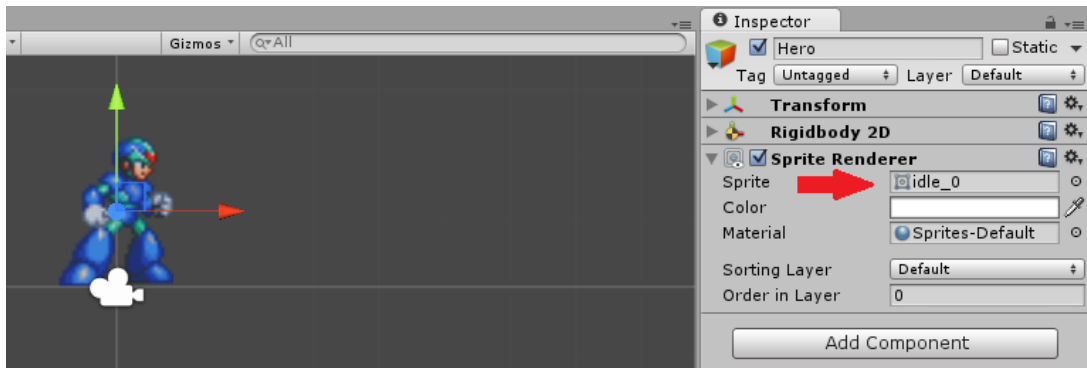


10. Note que o arquivo do *sprite* ficou com uma seta ao lado e clicando nela veremos todas as imagens separadamente.

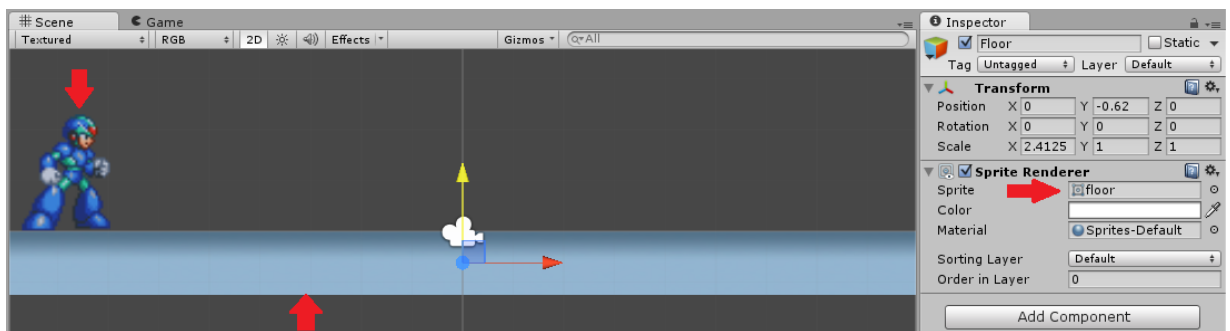


11. Repita os passos 8 e 9 para os outros arquivos de *sprite*, menos para o arquivo *floor*. Não se esqueça de mudar a propriedade *Sprite Mode* para “*Multiple*” e a *Pixels per Unit* para “*20*”.

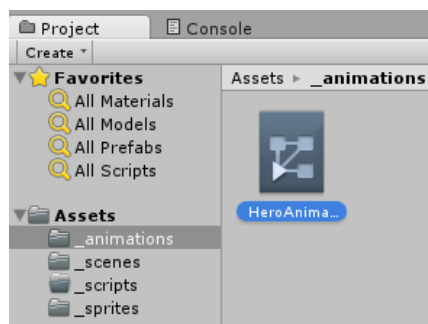
12. Selecione o objeto *Hero* na janela *Inspector* e defina a imagem "idle_0" como valor da propriedade *Sprite* do componente *Sprite Renderer*. Fizemos isso apenas para poder visualizarmos o objeto *Hero* dentro da cena do jogo, sendo que sua animação será definida nos passos mais adiante. Execute o jogo e veja que o nosso herói cairá no infinito.



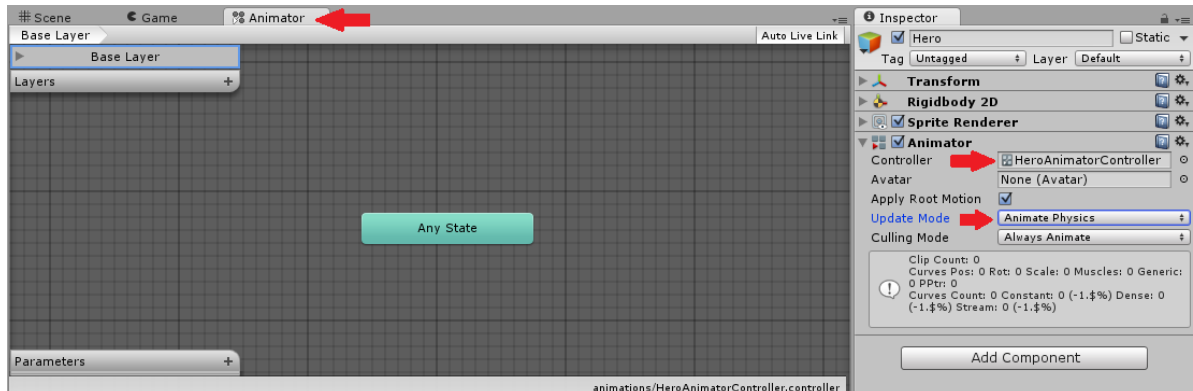
13. Agora selecione o sprite *floor* na janela *Project* e defina sua propriedade *Pixels per Unit* para "20". Crie um novo objeto vazio e chame-o de *Floor*. Insira nele um componente *Sprite Renderer* e defina sua propriedade *Sprite* para a imagem "floor". Insira também um componente *BoxCollider* (*Add Component -> Physics 2D -> Box Collider 2D*). Mova o objeto *Floor* na cena para que ele fique um pouco abaixo do objeto *Hero* e altere sua escala em X para que ele ultrapasse as fronteiras da cena. Posicione o objeto *Hero* na parte esquerda da cena.



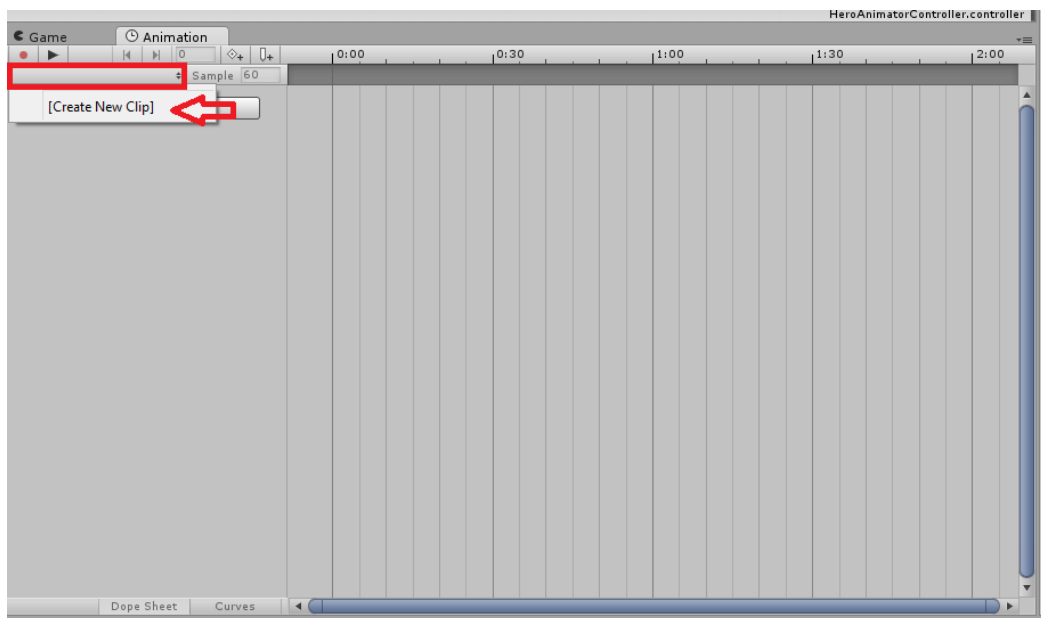
14. Vamos então criar as animações para dar vida ao nosso herói. Abra a pasta *_animations* dentro da janela *Project*. Crie dentro dela um objeto do tipo *Animator Controller*, através da opção *Create->Animator Controller* e chame-o de *HeroAnimatorController*.



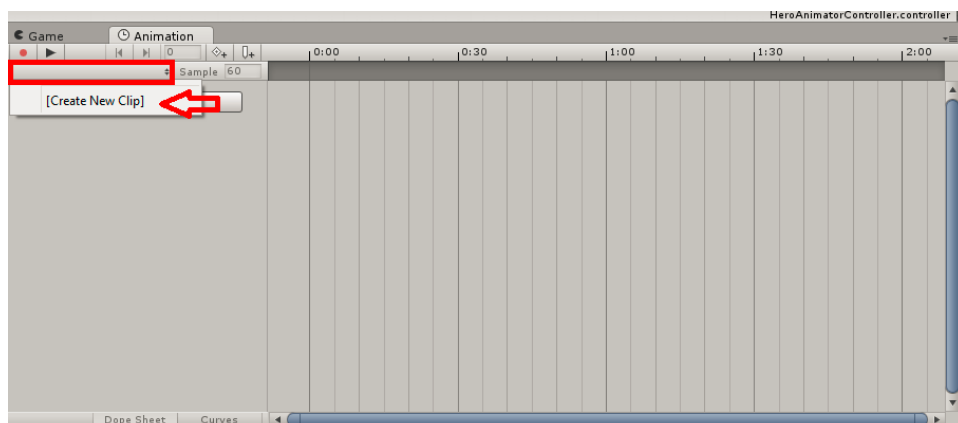
15. Abra o objeto criado (duplo-clique) para exibir a janela *Animator*. Essa janela irá controlar todas as animações do personagem. Para isso, precisamos ligar o personagem com o objeto *Animator Controller* que criamos. Selecione o objeto *Hero*, clique em *Add Component->Miscellaneous->Animator* e defina sua propriedade *Controller* para o arquivo *HeroAnimatorController*. Também mude sua propriedade *Update Mode* para “*Animate Physics*”.



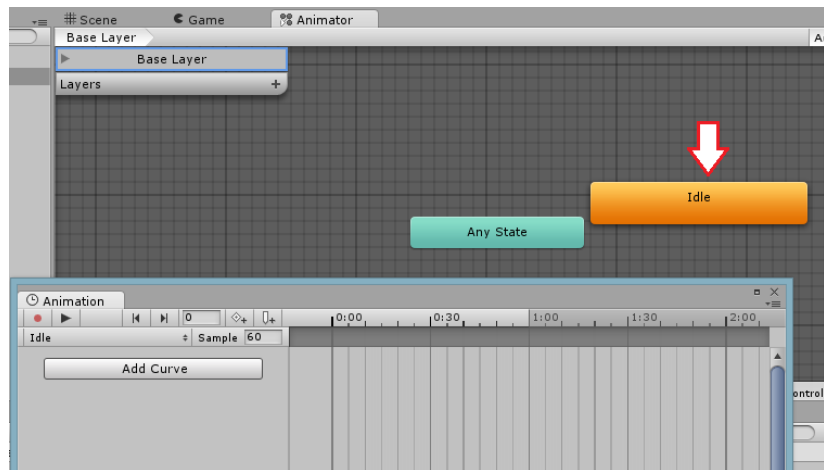
16. Agora acesse a opção do menu *Window->Animation* para abrir outra janela para definirmos as animações. Posicione-a onde você achar melhor na IDE.



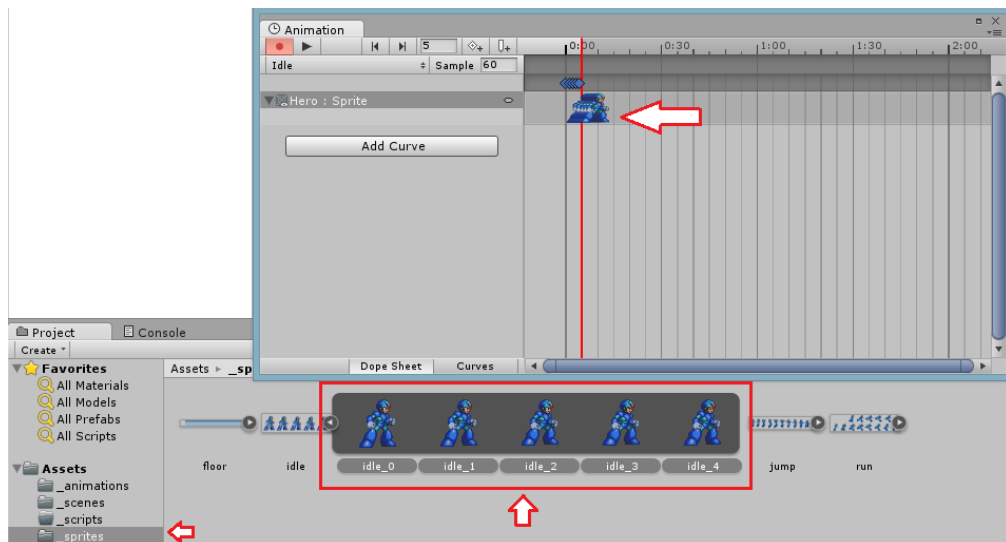
17. Na janela *Animation* iremos criar as animações para os estados do personagem, sendo eles: *Idle* (Parado), *Run* (Correndo) e *Jump* (Pulando). Agora selecione o objeto *Hero* e, na janela *Animation*, você verá um campo chamado *Sample* com um valor de 60. Do outro lado desse campo existe uma seta. Clique nela e selecione *Create a New Clip*. Salve o novo *clip* dentro da pasta *_animations* com o nome “*Idle.anim*”.



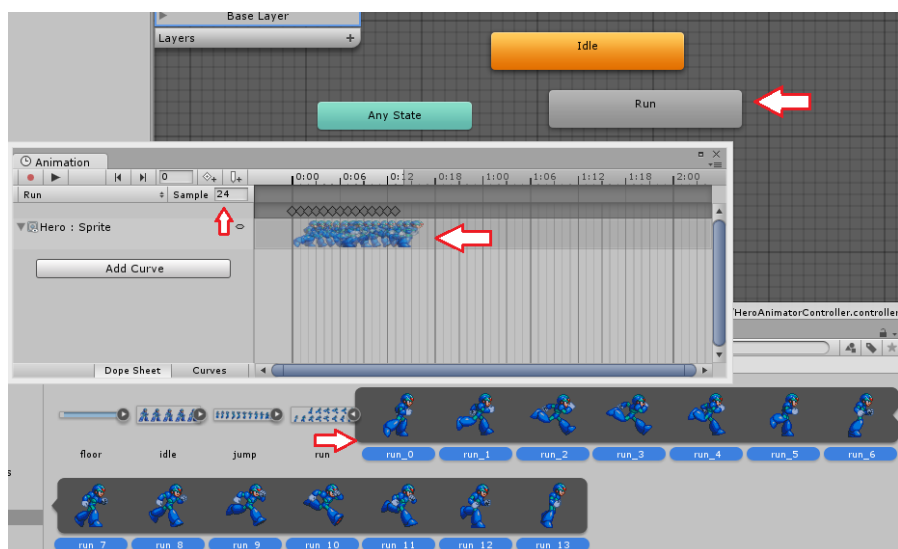
18. Se tudo deu certo, vai aparecer um outro retângulo dentro da janela *Animator*, chamado “*Idle*”. A cor laranja indica que essa animação é a padrão e será a primeira a ser executada.



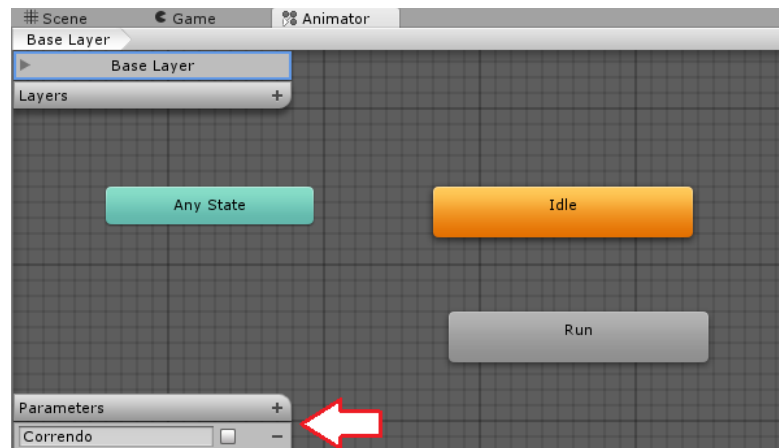
19. Agora vamos adicionar os *sprites* na animação *Idle*. Expanda o *sprite Idle* na janela *Project*, selecione todas as suas imagens e arraste-as para a janela *Animation*. Em seguida, altere o valor do campo *Sample* para 6. Com isso, o nosso herói já possuirá a animação de *Parado*.



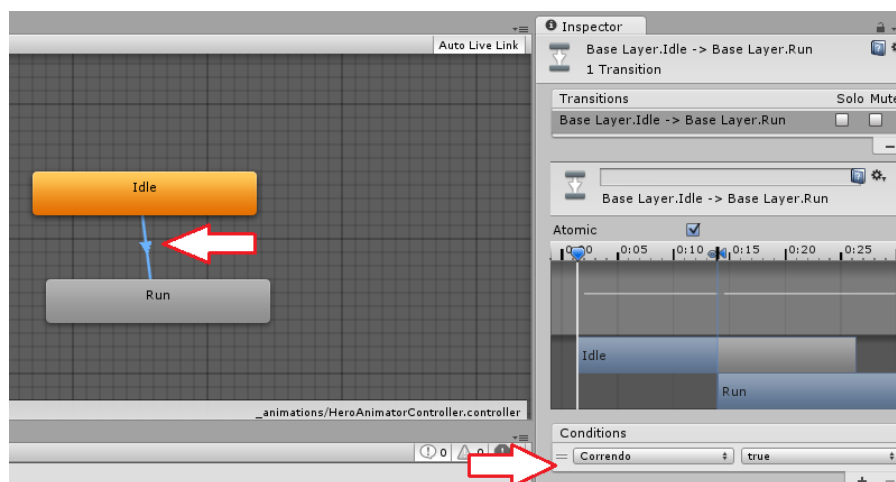
20. Vamos agora criar a animação para o estado *Correndo*. Selecione o objeto *Hero*, adicione um novo *clip* (*Create New Clip*) na janela *Animation* – da mesma forma que fizemos com a animação *Idle* – e salve-o como “*Run.anim*”. Selecione as imagens do *sprite run* e arraste-as para o *clip*. Mude o valor do campo *Sample* para 24.



21. Até agora já temos a animação do movimento do personagem. Precisamos fazer com que ele passe do estado *Parado* para o *Correndo* quando o jogador pressionar a tecla direcional para direita. Na janela *Animator*, temos uma área chamada *Parameters* que fica na parte inferior esquerda. Clique no símbolo “+” e em seguida em *Bool* para adicionarmos um parâmetro do tipo booleano. Chame o parâmetro de “*Correndo*”.



22. Agora clique com o botão direito em cima do estado *Idle* (cor laranja), selecione “*Make Transition*” e depois clique em cima do estado *Run*. Isso irá criar uma seta partindo de *Idle* para *Run*. Clique nessa seta e na janela *Inspector* você verá que existe uma área chamada “*Conditions*”. Defina a condição como “*Correndo*” e “*true*”.



23. Faça a mesma coisa, mas agora criando uma transição entre *Run* e *Idle* e definindo a condição como “*Correndo*” e “*false*”.

Até aqui, basicamente dissemos para o nosso *Animator Controller* que quando o seu parâmetro “*Correndo*” for “*true*”, deverá mudar para o estado *Run* e, quando for “*false*”, deverá mudar para o estado *Idle*.

24. Agora vamos inserir a codificação para controlar a animação do objeto *Hero*. Crie um novo script na pasta *_scripts* e chame-o de *HeroBehaviour*. Em seguida, adicione este script no objeto *Hero*. Abra-o e escreva o seguinte código nele:

```
//armazena o componente Animator do objeto Hero
private var animator: UnityEngine.Animator;

function Start () {
    //obtem o componente Animator do objeto Hero
    animator = gameObject.GetComponent.<UnityEngine.Animator>();
}

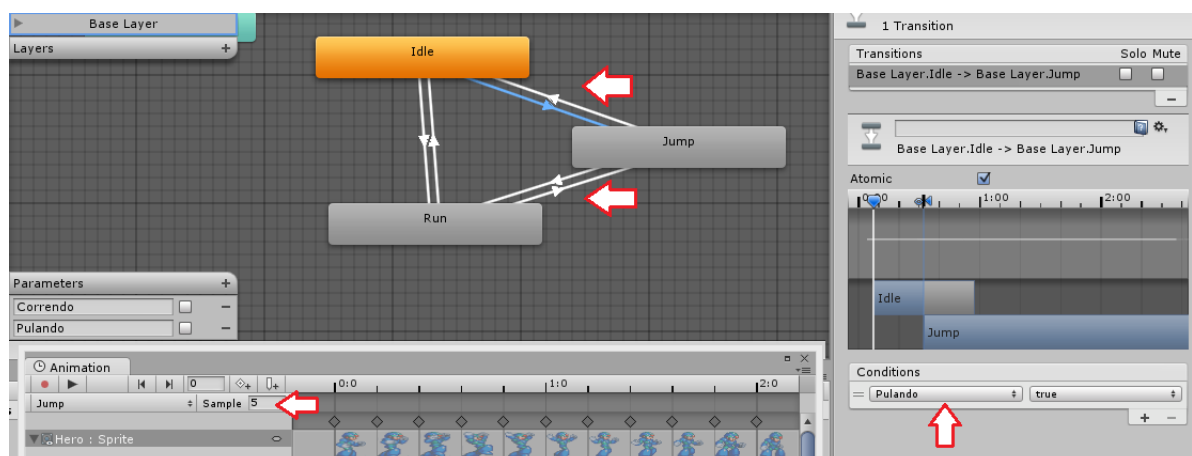
function Update() {
    //quando o jogador deixar pressionada a tecla "SETA PARA DIREITA"
    //atribui TRUE para o parametro "Correndo" do animator
    if (Input.GetKeyDown("right") ){
        animator.SetBool("Correndo", true); //animação "Run"
    }
    //quando o jogador liberar a tecla "SETA PARA DIREITA"
    //atribui FALSE para o parametro "Correndo" do animator
    else if (Input.GetKeyUp("right") ){
        animator.SetBool("Correndo", false); //animação "Idle"
    }
}
}
```

25. Execute e teste o jogo.

26. Agora vamos definir a animação para fazer o personagem pular. Selecione o objeto *Hero*, adicione um novo *clip* (*Create New Clip*) na janela *Animation* e salve-o como "*Jump.anim*". Selecione as imagens do sprite *jump* e arraste-as para o *clip*. Mude o valor do campo *Sample* para 5.

Ainda na janela *Animation*, clique no símbolo "+" e em seguida em *Bool* para adicionarmos um parâmetro do tipo booleano. Chame o parâmetro de "*Pulando*". Por fim, crie as seguintes transições:

- Entre *Idle* e *Jump* com *Conditions* "*Pulando*" e "*true*";
- Entre *Jump* e *Idle* com *Conditions* "*Pulando*" e "*false*";
- Entre *Run* e *Jump* com *Conditions* "*Pulando*" e "*true*";
- Entre *Jump* e *Run* com *Conditions* "*Pulando*" e "*false*".



27. Agora faça as seguintes alterações no script *HeroBehaviour*.

```
//armazena os componentes Animator e Rigidbody do objeto Hero
private var animator: UnityEngine.Animator;
private var rb: UnityEngine.Rigidbody2D;
//variaveis para controlar se o player estah pulando
private var noPiso: boolean = true;
private var pulando: boolean = false;

function Start () {
    //obtem o componente Animator do objeto Hero
    animator = gameObject.GetComponent.<UnityEngine.Animator>();
    rb = gameObject.GetComponent.<UnityEngine.Rigidbody2D>();
}

function Update() {
    //quando o jogador deixar pressionada a tecla "SETA PARA DIREITA"
    //atribui TRUE para o parametro "Correndo" do animator
    if (Input.GetKeyDown("right") ){
        if (noPiso){
            animator.SetBool("Correndo", true); //animação "Run"
        }
    }
    //quando o jogador liberar a tecla "SETA PARA DIREITA"
    //atribui FALSE para o parametro "Correndo" do animator
    else if (Input.GetKeyUp("right") ){
        animator.SetBool("Correndo", false); //animação "Idle"
    }
    //se apertou ESPAÇO, pula
    if (Input.GetKey("space") ){
        if (!pulando){
            rb.AddForce( new Vector2(0,300) );
            animator.SetBool("Pulando", true);
            noPiso = false;
            pulando = true;
        }
    }
    //se o objeto estah no piso, desativa animação de pulo
    if (noPiso){
        animator.SetBool("Pulando", false);
    }
}

//Esta função eh executada quando o player toca no piso
function OnCollisionEnter2D(coll: Collision2D) {
    noPiso = true;
    pulando = false;
}
```

28. Execute e teste o jogo