**DPJ\_02**

**Nível 1:**

No nivel 1 é instalamos o Input System através do Package Manager. Depois, criamos um Action Map dedicado ao movimento, definindo as teclas (w,a,s,d) direcionais que vamos utilizar para controlar o personagem (Fig.1).

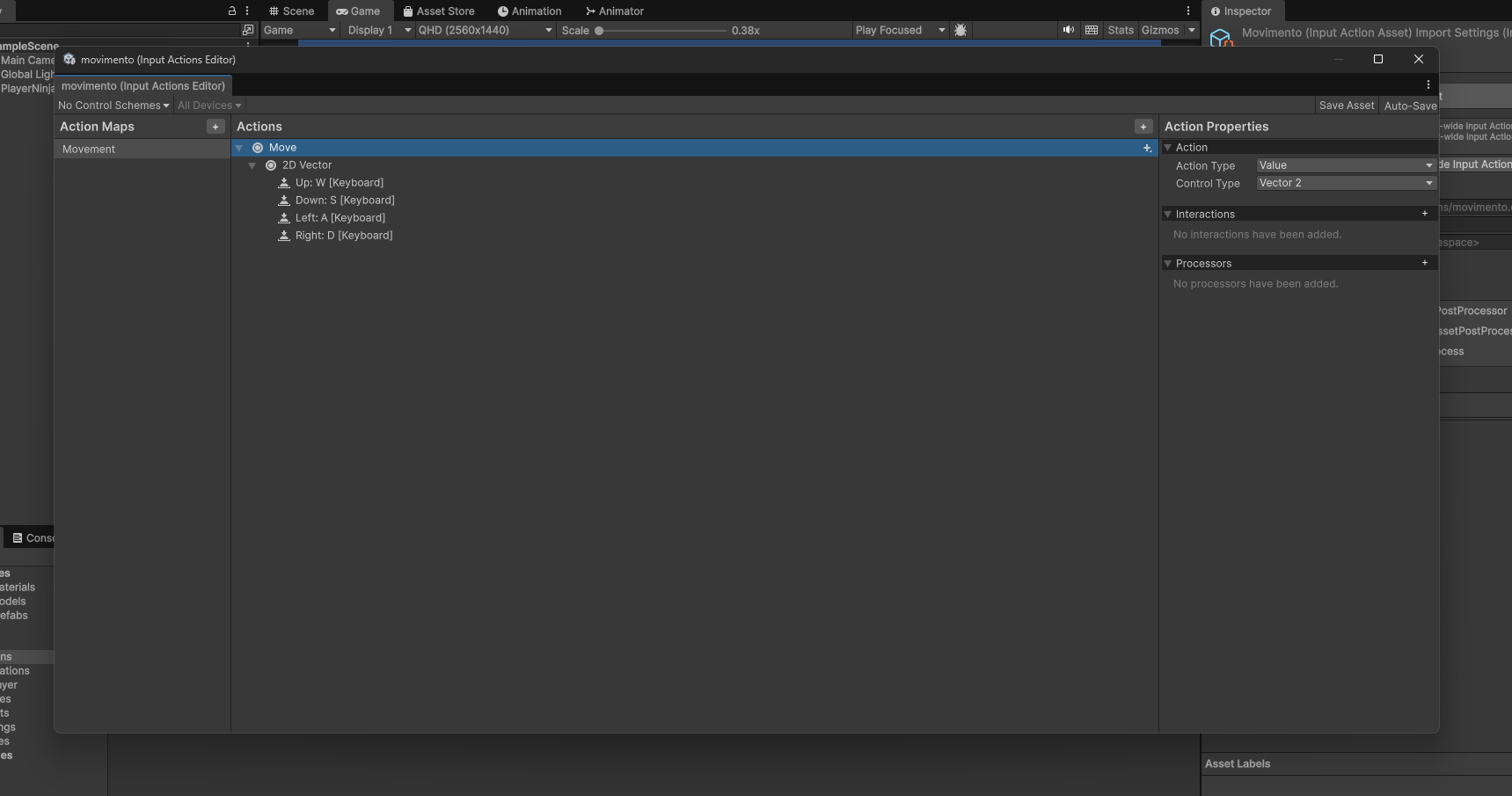


Fig.1

**Nível 2:**

No nivel 2 nos sprites do personagem Green Ninja ajustamos as propriedades como Pixels per Unit para que o personagem tenha a dimensão correta no cenário (Fig.2). Utilizamos o Sprite Editor para partir as spritesheets em frames individuais para criar as animações de movimento do personagem enquanto ele anda pelo cenário.



Fig.2

**Nível 3:**

No nivel 3 renomeamos a sprite principal para PlayerNinja e criamos uma Sorting Layer para controlar a renderização dos personagens. Criamos as pastas dentro do nosso projeto e criamos o script PlayerMovement, associando-o ao PlayerNinja (Fig.3).



Fig.3

**Nível 4:**

No nivel 4 no script, definimos variáveis como a velocidade do personagem e o sistema de Input Actions (Fig.4), escrevendo métodos para ativar e desativar os comandos. Assim conseguimos fazer a leitura dos inputs do teclado e preparar o movimento do personagem conforme a tecla pressionada (Fig.5).

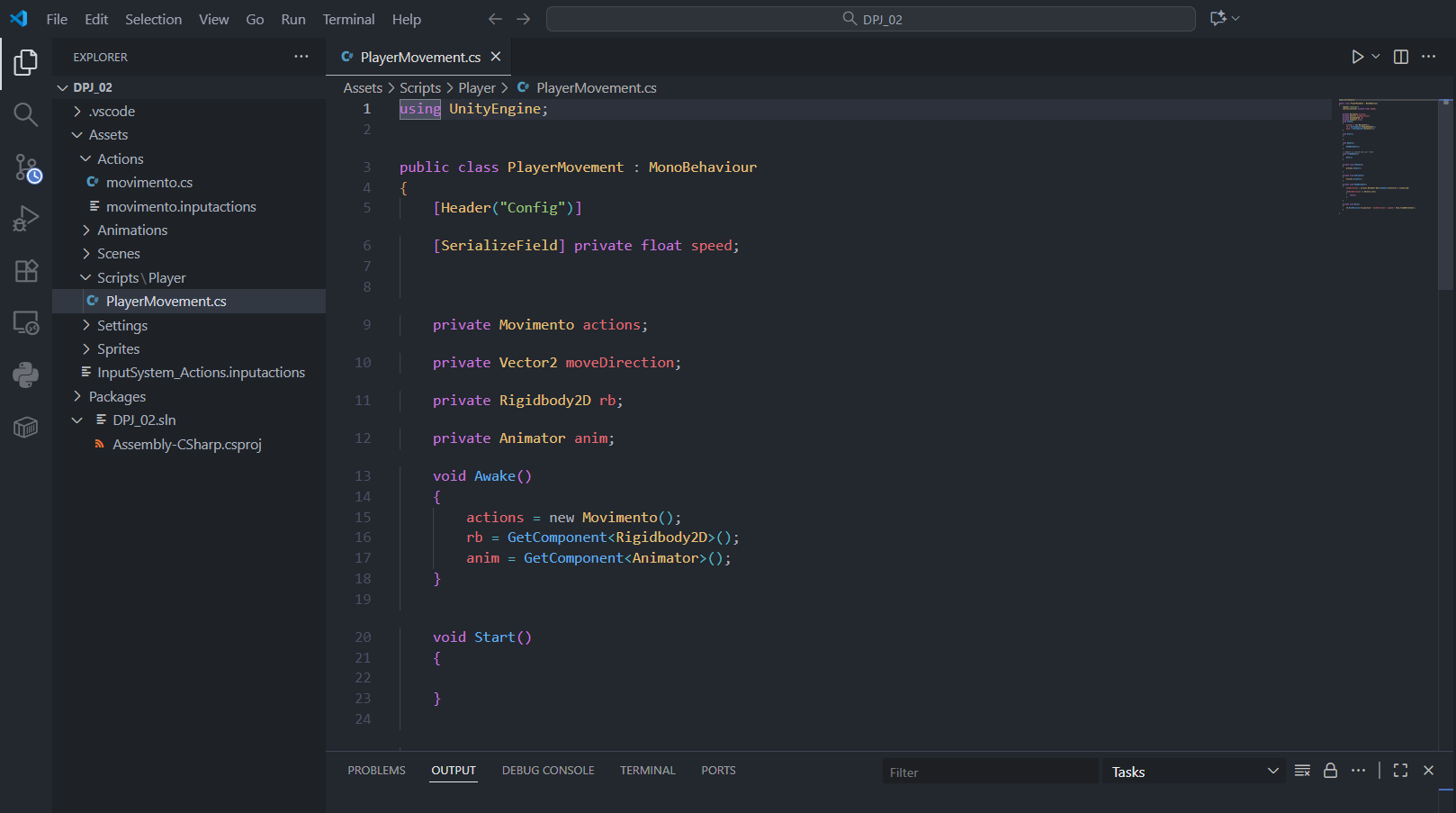


Fig.4

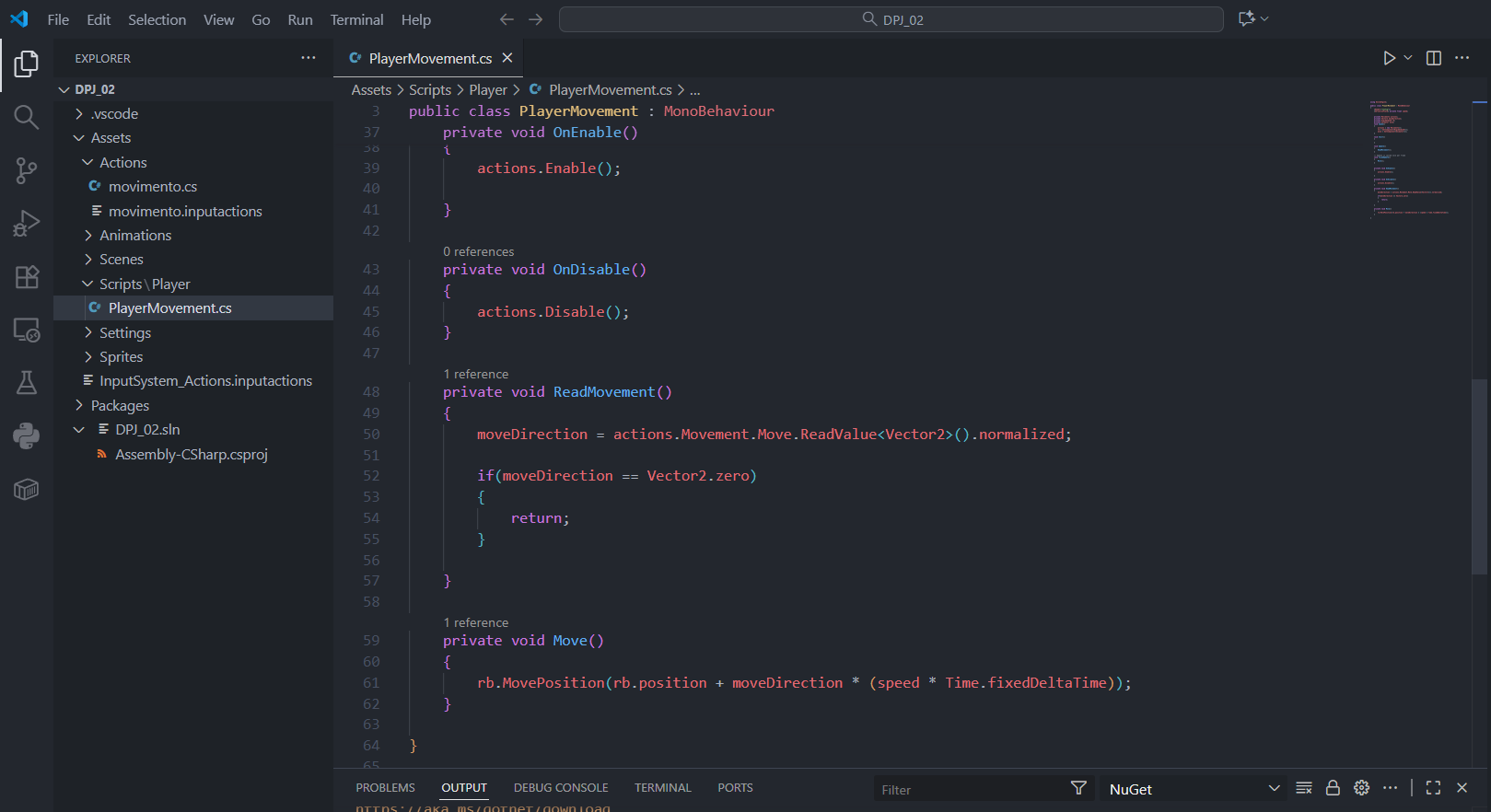


Fig.5

**Nível 5:**

No nivel 5 adicionamos o componente Rigidbody2D ao PlayerNinja para permitir a movimentação, bloqueando a rotação n o eixo Z para que o personagem permaneça sempre na posição correta (Fig.6). Em seguida, criamos animações para as diferentes direções (andar para cima, baixo, esquerda e direita), organizando-as na pasta Animations. Utilizamos o Animator e configuramos uma Blend Tree, que permite alternar automaticamente entre essas animações de acordo com os valores dos eixos Fig.7.

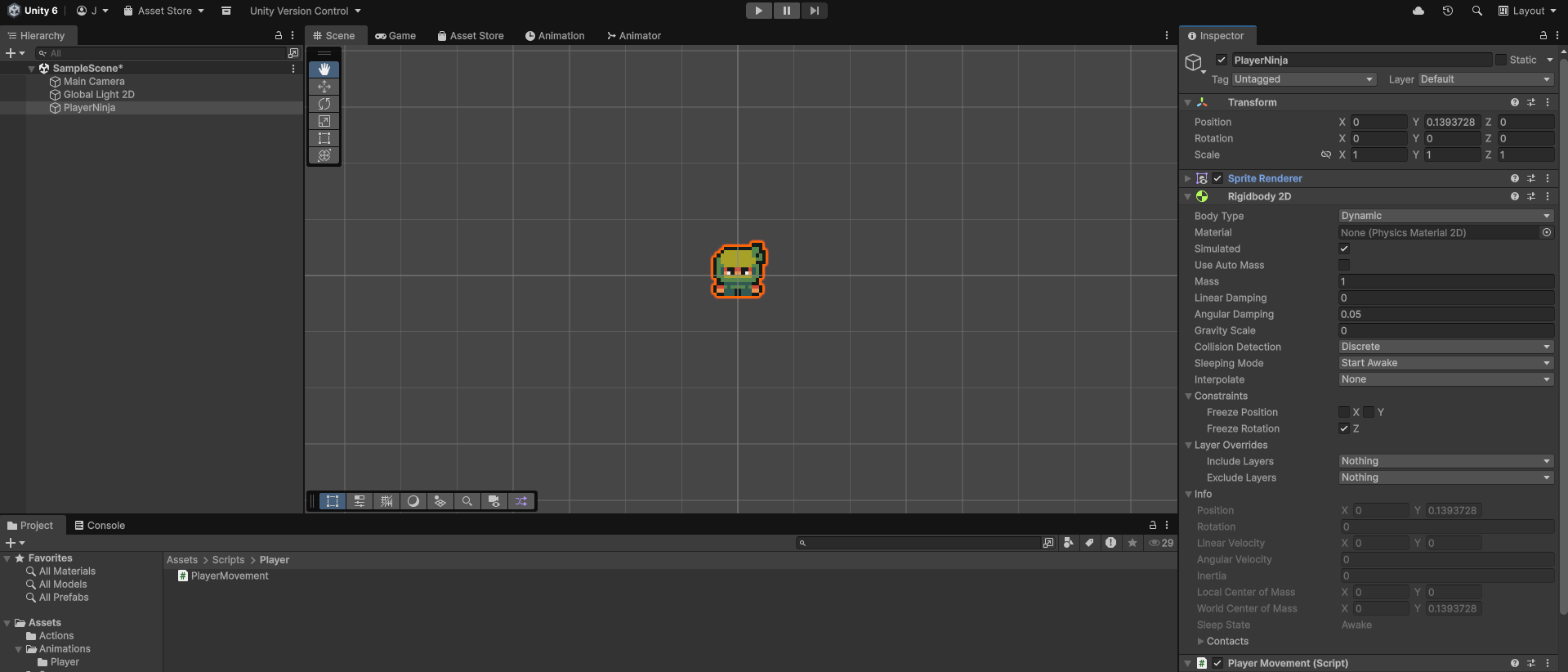


Fig.6

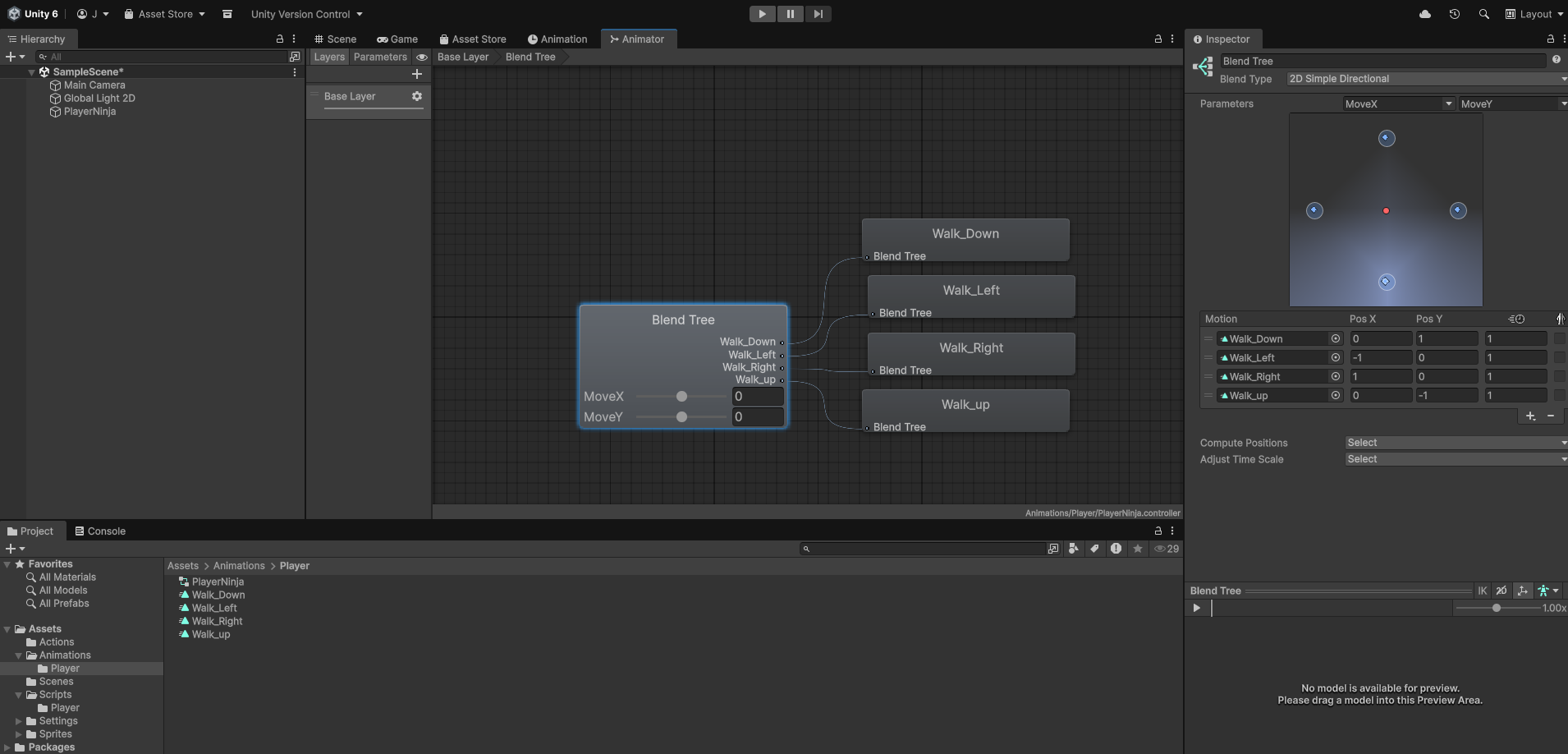


Fig.7