

# Game Ready Futuristic Soldier

## Instructions on How to add the Soldier to the Scene:

- 1 - Drag the soldier and camera prefabs into your scene.
- 2 - Assign the camera to the Soldier's Camera attribute on the inspector.
- 3 - Assign the soldier to the Camera's Target attribute on the inspector.
- 4 - Hit play!

### Controls:

Movement - A, W, S, D or Arrow Keys.

Roll - Space.

Shot - Left Mouse Button.

## Scripts Documentation

**SoldierController.cs** - Controls the character movements and some actions like roll, death and reload.

### Inspector Attributes:

**Camera** - The 3rd person camera (It must have the SoldierCamera script attached to it).

**Speed** - Controls how fast the character moves, the higher the number the faster it is.

### Public Methods and Attributes:

**alive** - returns true when the player is alive.

**rolling** - returns true when the roll animation is playing.

**reloading** - returns true when the reload animation is playing.

**Die()** - Kills the character and triggers the death animation.

**Scale()** - Returns the localScale.y of the character, it's used as a multiplier for some other parameters that depends on the size.

---

**GunController.cs** - Responsible for firing the weapon.

### Inspector Attributes:

**Muzzle** - Particle System that represents the muzzle flash, it'll play when the weapon fires.

**Sfx** - gunshot sound effect.

**AudioSource** - Component that will play the gunshot sound effect.

**Character** - The character that holds the weapon (It must have the SoldierController script attached to it).

**FireRate** - Interval between shots (in seconds).

### Public Methods:

**Fire()** - Fires the weapon.

---

**SoldierCamera.cs** - Class that controls the 3rd person camera.

**Inspector Attributes:**

**Target** - The 3rd person character (It must have the SoldierController script attached to it).

**MaxZoom** - Max distance from the target when zooming out.

**MinZoom** - Min distance from the target when zooming in.

**MaxXRotation** - Max rotation on the X axis when orbiting the target.

**MinXRotation** - Min rotation on the X axis when orbiting the target.

**CollisionLayer** - Layers that the camera will be colliding with (Important: The character must not belong to any layer selected on this field, otherwise the camera will collide with it during some animations and cause some strange behavior).

---

**AnimationEventHandler.cs** - Class responsible for handling the animation events.

**Inspector Attributes:**

**StepWalk** - AudioClip to play on each footstep when the character walks.

**StepRun** - AudioClip to play on each footstep when the character runs.

**Reload** - AudioClip to play when the character reloads.

**Roll** - AudioClip to play when the character rolls.

**Death** - AudioClip to play when the character dies.