Third Person Cover Shooter

Video tutorials

Character setup : https://youtu.be/8aCr4eIMjFc
Add Pistol : https://youtu.be/CIes-mAj8q8
Add Rifle: https://youtu.be/HwN4rB-jPvo

Covers

Walls usable for taking covers have to be marked by cover markers. A cover marker is any game object with Cover and Box Collider components attached. Markers can intersect, form a chain and act as if that chain is one big cover.

Cover orientation matters, the example scene contains markers with feet that mark facing directions.

There are two kinds of covers, low and tall. The kind is determined from the height of a BoxCollider attached to the marker. The height threshold is different for every character and is defined by a Character Motor. In a chain of covers of different height character will correctly change its stance when transitioning between tall and low covers.

A corner of a cover with no adjacent covers nearby is treated as a corner characters can peek from. However, there can be unmarked walls and therefore character can attempt to take a peek in impossible situations. Such cases are handled by Open Left and Open Right properties inside the Cover component, setting a value to false marks that corner as unusable for peeking.

Low covers can be climbed or vaulted over. Every cover has the type of climbing defined in a Cover component. Vaulting should be enabled for low walls players can jump over and climbing for covers alongside higher ground.

Character Motor

Characters must have a Character Motor component attached. It manages the character, it’s movement, appearance and use of weapons. It handles gravity and therefore gravity should be turned off in the RigidBody component to avoid conflicts.

There is an IK (inverse-kinematics) system that handles aiming and recoil. It can be configured manually but for ease of use there is a button to set it up automatically inside the CharacterMotor inspector. It is recommended to reduce the amount of bones used by IK on non-player characters for performance reasons.

IK is calculated by adjusting bones until certain objects reach defined targets. Target objects must be part of the skeleton in order for changes to modify their transform. Character’s sight is usually defined by a marker object that is part of the head, bones are transformed until marker’s forward vector points towards the target.

Each character has a set of weapons in its disposal. To add a new weapon to the character you must create an object with a 3D model and a Gun component and attach it to a hand. Additionally, you can create a version of the weapon that is put into its holster. The motor automatically enables and disables weapon and holster objects.

Characters can throw grenades from both hands. Left hand is used in some situations when the character is hiding behind a cover. Grenades are cloned when thrown.

A character can be setup to have many hitboxes for various body parts. Setup is done inside Character Health component.

Gun

Guns raycasts bullet, manage clip and recoil.

For player characters bullets originate at camera in order for player to be able to fire on targets they can see, even if there is a small obstacle in front of the gun. The fire origin is set by a camera. Since AI do not have Third Person Camera attached their bullets are fired starting from the Aim marker, which usually is at the end of the gun.

Each weapon has two marker objects. Aim defines point of origin for AI bullets and is also used when rotating character’s arms till the marker points towards the target. Left Hand object marks the position for character’s left hand. Naming of left and right hands is incidental and character’s handedness can be swapped.

The intended position of the left hand might differ in some animations, to handle that there are left hand marker overwrites you can use to set up IK for the left hand for some specific situations. Empty values are not used as overwrites.

Currently there are two kinds of weapons, pistols and rifles. The type defines character animations when using a weapon.

Grenade

Manage flight and explosion of a grenade. Flight and collisions are performed using raycasts. Grenades can bounce. Upon explosion a prefab is instantiated on grenades location. Grenade affects Character Health and Body Part Health objects in its explosion area. Damage is highest in the epicentre and decreases linearly to zero on the edge of the explosion area.

Third Person Controller

Takes player keyboard and mouse input and translates that to Character Motor commands.

Weapons are controlled by the number keys. Pressing 1 hides a weapon and keys starting with 2 make the character equip a weapon. The order of keys and weapons defined in the Character Motor is the same.

Enters grenade mode when pressing G. Displays a grenade preview using prefabs containing Path Preview and Explosion Preview components.

Mobile Controller

Takes touch screen input and translates that to Character Motor commands. Player input is generated using canvas objects containing components like Touch Movement or Touch Aiming. Those canvas components must be linked with the Mobile Controller.

When throwing a grenade, displays a grenade preview using prefabs containing Path Preview and Explosion Preview components.

AI Controller

Makes the character walk and fire towards an enemy. Stays idle until provoked. AI tries to maintain a certain distance from its threat. The AI can use cover.

It reacts to the appearance of player in its line of sight. The field of view and distance of it can be controlled. The AI will notice friends getting hurt using its line of sight. In addition to sight, AI reacts to various alerts generated in the world. Alerts can be generated by footsteps, gunfire, explosions, grunts, etc.

In order to make gameplay fairer the AI has a small delay which can be controlled by adjusting the Reaction Time. It’s amount of time in seconds it takes for AI to notice threats.

AI can perform patrols when it is not in an alert state. During patrols AI moves walks from one waypoint to the next. Waypoints are edited in the scene editor when the AI object is selected.

Actor

Actor depicts a character or an object that might be a subject of AI’s thoughts. All characters must have the Actor component attached to them.

Actors have a specified side number which depicts their team and is used by AI to find enemies and friends. AI doesn’t use covers when attacking an actor that is set to be non-attacking (for example, a security camera).

AI Group

AI groups are used to make AI controllers coordinate their attacks. AI group limits the number of characters performing attacks at the same time.

Third Person Camera

Manages the camera object by setting an appropriate orientation depending on the target object's state. For camera to work you have to link it to the target object that has a Character Motor attached.

The camera component also maintains and draws a crosshair. It hides the crosshair if the character is unarmed or unable to fire at a wall because is too close. The visibility of crosshair also can be turned off manually by setting Is Crosshair Enabled value to false when your game needs so.

Mobile Camera

Manages the camera object by setting an appropriate position depending on the target object's state. For camera to work you have to link it to the target object that has a Character Motor attached.

Camera shifts a bit depending on the orientation of the character. It also zooms out when enemies are close by in order to display a wider combat area.

Character Health

Manages health and sets Is Alive in Character Motor to false when it reaches 0. Registers damage done by bullets.

Multiple hitboxes can be setup inside the character. On setup Character Health will stop registering hits and instead will expect child Body Part Health components to pass taken damage to it.

Character Platform

Keeps character on top of a moving platform. It doesn’t require a Character Motor or any other component and therefore can be used on any object even if it’s not physical.

Additional character components

Body Part Health

Acts similarly to Character Health, but passed the taken damage to a first found Character Health component in the hierarchy.

Character Face

Reacts to character events and modifies blend shapes in a mesh. The set of faces can be expanded by modifying the script or making a new version of it.

Character Sounds

Spawns sound instances on various character events. Sounds are randomly picked from lists.

Character Effects

Spawns prefab instances on various character events.

Character Alerts

Generates alerts for the AI to pick up on various character events.

Character Name

Contains a character name used by the UI. If no Character Name is present the name of the game object is taken as the name.

Character Sleep

Turns off some components of character object when it is far away from the player’s character. Components are turned on when the player approaches the character.

Additional gun components

Gun Effects

Spawns effects prefabs on various gun events like reloads or gunfire.

Gun Sounds

Generates randomised sounds on various gun events.

Gun Alerts

Generates alerts for the AI to pick up on various gun events.

Hit Effect

Spawns an effect upon a bullet hit. Used mostly on static level geometry.

Additional grenade components

Path Preview

Builds a path mesh that depicts the approximated flight of a grenade. Manipulated by player controllers.

Explosion Preview

Displays a sphere depicting an area of grenade explosion. Used by both grenades and player controllers.

Extra

Damage Trigger

Deals damage to Character Health components attached to objects that enter its trigger area.

Enemy Sight

Generates and maintains a mesh depicting field of view for a character motor.

Player Arm Trigger

Causes player characters to pick a weapon when they enter the trigger area.

Player Disarm Trigger

Causes player characters to hide their weapon upon entering the area.

Random Audio

Picks a random audio sample from the supplied list and sets Audio Source to play it.

Delayed Destroy

Disables an object after a certain amount of time passes.

Delayed Disable

Destroys an object after a certain amount of time passes.

Exit to Escape

Exits the game when an escape key is pressed. Best used in the Unity Editor.

UI

Health Bar

Takes a relative health from a Character Health component and displays a bar that displays the value.

Enemy Display Manager

Creates and manages objects with Health Bar for all enemies visible on screen.

Enemy Health

Takes a target of a character motor and sets a sibling Health Bar component to take its values.

Ammo Bar

Displays information about specific weapons ammunition. When pressed makes a character pick that weapon.

Crouch Touch

When pressed toggles associated characters crouch state.

Grenade Bar

On touch makes the character pick up a grenade.

Touch Aiming

Takes directional input from the touch screen and passes it to Mobile Controller as aim direction.

Touch Movement

Takes directional movement input from the touch screen and passes it to Mobile Controller.

Support

Ask questions, send ideas, report bugs: eduardas.ninja@gmail.com

Thanks,

RedBee Team