

TPS Character System Guide

Guide for Product Version: 1.2.1

[Introduction](#)

[Guide Overview](#)

[Controls Mapping](#)

[Weapons System](#)

[Weapons System Overview](#)

[Customize Primary & Secondary Weapons](#)

[Customize Without Weapon State](#)

[Creating New Weapon](#)

[Creating New Weapon Blueprint Class](#)

[Weapon Blueprint Class & Skeletal Mesh Customize](#)

[Character Skeletal Mesh Customize](#)

[Weapons Customize](#)

[Basic Weapons Settings](#)

[Weapons Type](#)

[Recoil Settings](#)

[Recoil Recovery Settings](#)

[Spread Settings](#)

[Weapon Damage Settings](#)

[Take Damage Logic](#)

[Recoil & Spread Coefficient Settings](#)

[SFX & Sound Settings](#)

[Weapons Pickup](#)

[Player's Camera](#)

[Player's Camera Overview](#)

[System Structure](#)

[Customize Transition Camera Modes](#)

[Customize Camera Shakes](#)

[Cover System](#)

[Cover System Overview](#)

[Customize Covers Analyzer. Covers Settings](#)

[Customize Climb & Jump Over Movement](#)

[Main Menu](#)

[Setup Main Menu](#)

Introduction

Thank you very much for purchasing TPS Character System.

The product was developed as complex system, which includes main gameplay mechanics of game character.

The product will be constantly improved.

Please send all suggestions for improvement or new functionality, as well as bugs to the mail or fill out the form below.

Contact Email: gameparts.main@gmail.com

Bugs & Feedback Form: <https://goo.gl/forms/sqi47j7w89iVZUth2>

Guide Overview

This document describes how to customize each of the three main systems. The principle of operation system is not described in the guide. All blueprints complementing comments and the principle of operation will be clear when analyzing the blueprints themselves.

The document is divided into three parts: weapons system, player's camera and player's locomotion with covers system.

Controls Mapping

- Shooting: R2 (PS4 / Xbox), Left Mouse Button
- Mini Zoom: L2 (PS4 / Xbox), Right Mouse Button
- Reload Weapon: Square (PS4), X (Xbox), R keyboard button
- Crouch: Circle (PS4), B (Xbox), C keyboard button
- Climb / Jump Over: X (PS4), A (Xbox), Space keyboard button
- Character Movement / Camera Movement: Left / Right Stick (PS4 / Xbox), Mouse
- Swap Weapon: TAP Triangle (PS4), Y (Xbox), Q keyboard button
- Hide Weapon: HOLD Triangle (PS4), Y (Xbox), Q keyboard button
- Pick Up Weapon: HOLD Square (PS4), HOLD X (Xbox), E keyboard button

Weapons System

Weapons System Overview

Weapons System includes the following game mechanics:

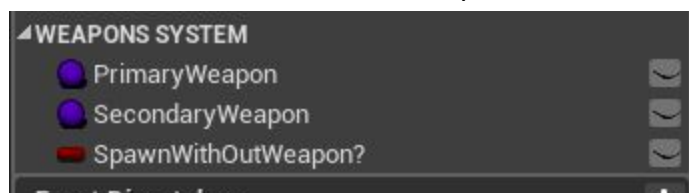
- **Various Weapons Types.** Supported auto, burst, single and autosingle weapons type.
- **Recoil.** Weapon can deviate vertically and horizontally, relative to a settings of a recoils curves.
- **Recovery Recoil.** Weapon have a functional of return to starting position after shooting stopped.
- **Spread.** Feature of bullets spread . Has a dependence on a rate of fire.
- **Recoil and Spread Coefficients.** System of coefficients depend on a state of game character (floor, crouch, mini zoom, walk end etc). Coefficients allow to increase or decrease recoil and spread of weapon.
- **Weapon Target System.** The system dynamically determines to aim a character and turns a weapon in this direction and sets character's off-set.
- **Front Checker System.** Systems solve the problem of shooting from jamb of wall.
- **Damage System.** Calculates a damage relative to a distance.
- **SFX and Sound System.** Creates a sound of a shot, a sound of a hit, effect at a point of hit and a muzzle effect.

Customize Primary & Secondary Weapons

The weapons system supports primary and secondary weapons. Weapons slots can be customized in BaseCharacter setting in Weapon setting group.

Customize Without Weapon State

Also can be set start state of the character: with weapon or without weapon.



Creating New Weapon

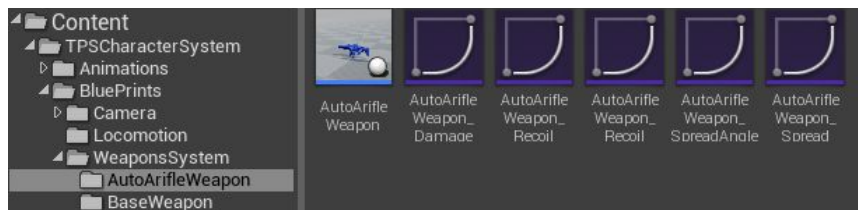
Creating New Weapon Blueprint Class

To creating a new weapon class, you need to go to the directory TPSCharacterSystem - BluePrints - WeaponsSystem.

In this directory the elements are: WeaponsSystem Component (component that controls the logic of weapons system, connects to the player's controller), WeaponSettings structure (structure describing a settings of weapon) and folders with weapons classes and curves for customization.



To creating a new types of weapon it is necessary to inherit a new class from the base class of weapons - BaseWeapon. Put the blueprint in a new folder and create new settings curves (for example, see how it was done for auto arifle).



Spawn weapons in a hands of game character there is in the blueprint Base Character in section SPAWN WEAPON & CREATE PLR'S HUD.

Weapon Blueprint Class & Skeletal Mesh Customize

SkeletalMesh in a weapons blueprint class must be properly configured.

- SkeletalMesh Root should be located in DefaultSceneRoot и and coincide with the place for which this weapon will be held by the character.
- Root of Main Arrow component should be located on one vertical with DefaultSceneRoot and SkeletalMesh Root.
- Main Arrow should be directed along the line of gun barrels, while indicating the direction of shots.

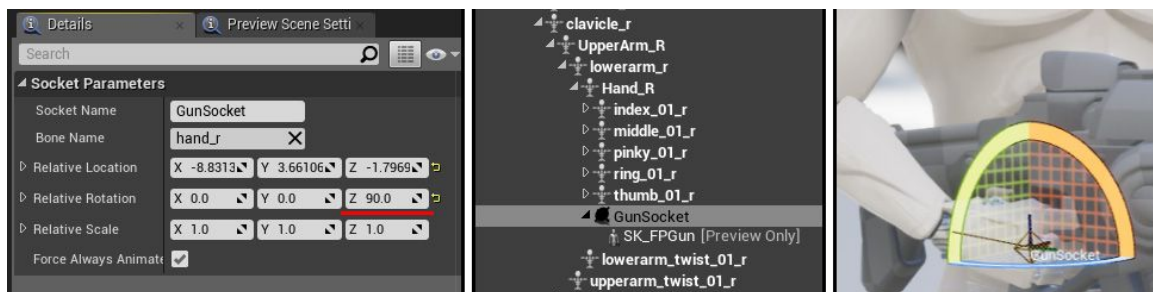


There are should be two sockets in weapons SkeletalMesh:

- **Muzzle Socket.** In place of the socket, where muzzle sfx is created.
- **L_HandSocket.** Required for animated blueprints, so that a character's left hand always holds a gun for this place.

Character Skeletal Mesh Customize

A Character SkeletalMesh must has a Gun Socket with certain settings (see screenshot below).



Weapons Customize

Weapons must be customize in from an inherited class. The settings there are in a structure of Weapon Settings.

Basic Weapons Settings

- **Fire Rate.** Rate of fire. Number of bullets per minute.
- **Clip Size.** Number of cartridges in clip.
- **Ammo Size.** Total number of cartridges for weapon.
- **Reload Time.** Reload animation corresponds this parameter. For correctly determine a duration of a reload animation, necessary set default time of an animation montage, which using for reload weapon. This parameter can be set in anim blueprint, in reload section.
- **Shooting Max distance.** How far the weapon can shoot. This parameter affects spread.

Weapons Type

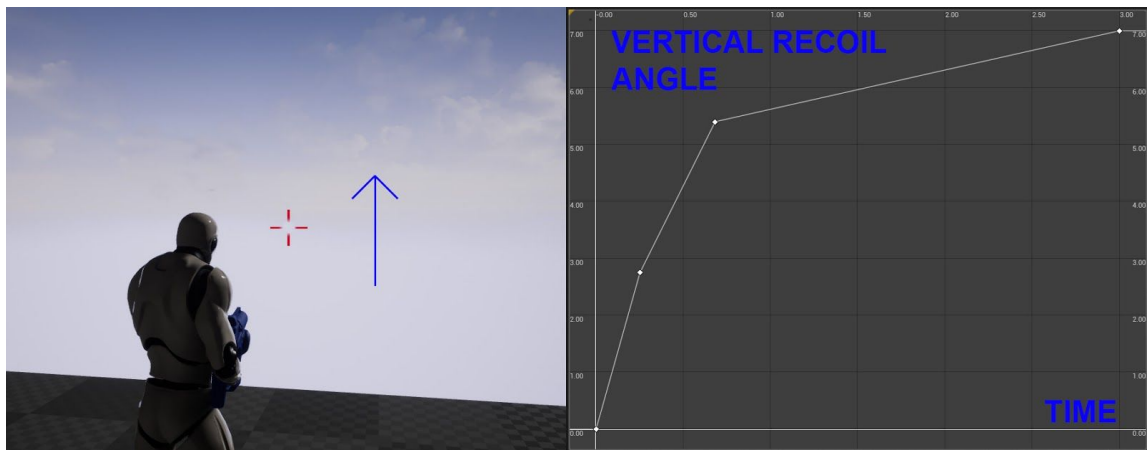
The Weapons System supported various weapons types. Weapons types is determined by the parameter **Shooting Type**.

- **Auto Fire**. Continuous shooting with a clamped trigger.
- **Single Fire**. A weapon that fires single shots. Play animation twitch shutter after each shot.
- **Auto Single Fire**. A weapon that fires single shots. A cartridge in a barrel is brought automatically after each shot
- **Burst Fire**. Several shots for one click of the trigger. In this case, with a rapid successive second push - a weapon remembers that it is necessary to produce a second shooting cycle. This weapons type is determined by the parameters: **BurstFirePause** (delay between turns) and **BurstFireShotsBeforePause** (shots in one turn)

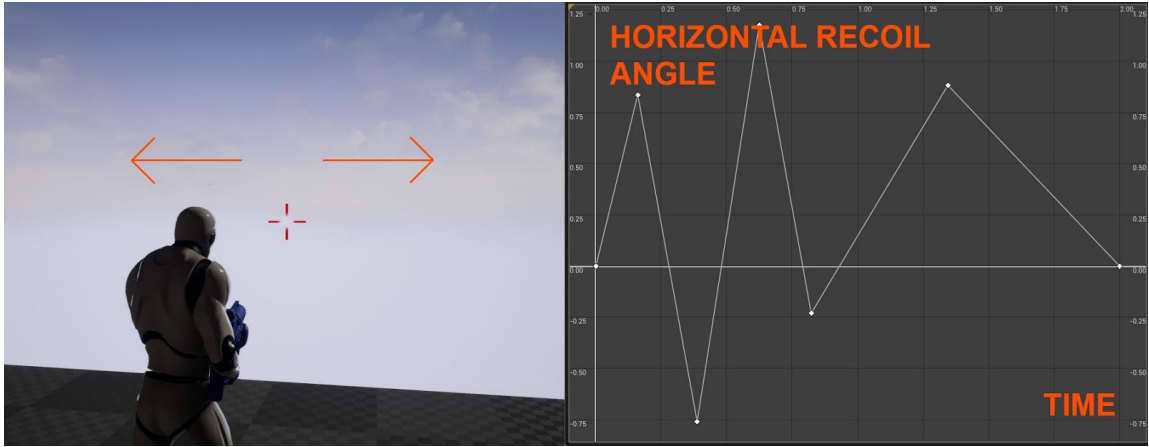
Recoil Settings

The setting of a weapon's recoil is set by 2 curves.

- **WeaponClassName_RecoilVertical**. The curve represents a dependence of a vertical deviation angle on a shooting time.



- **WeaponClassName_RecoilHorizontal**. The curve represents a dependence of a horizontal deflection angle on a shooting time.



Recoil Recovery Settings

A weapon return to starting position after shooting is configured by the following parameters.

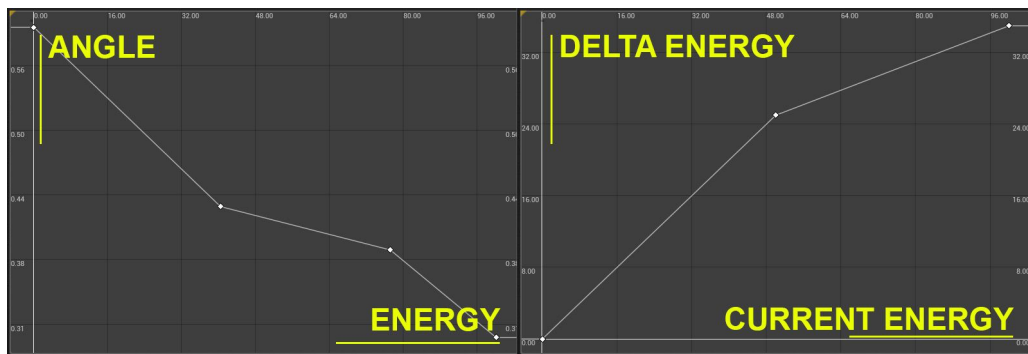
- **Recoil Recovery Vertical Speed.** Recovery speed vertically.
- **Recoil Recovery Horizontal Speed.** Recovery speed horizontal.
- **Recoil Recovery Delay.** After delay return of weapons will start.

Spread Settings

The mechanics of spread is based on the logic of spread energy.

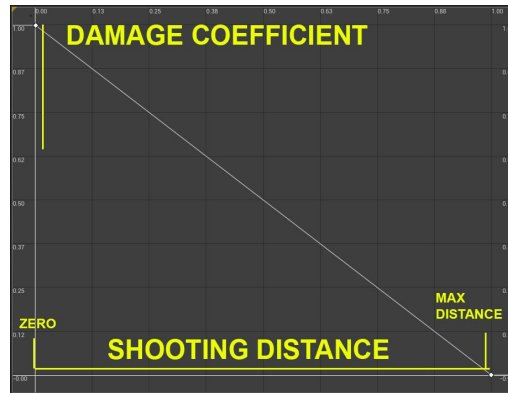
At the beginning of shooting the weapon have maximum value of spread energy, which is equal to 100. During shooting, spread energy drops. It depends on a energy curve **SpreadEnergyCurve**. The curve describes a energy reduction relative to a current energy.

The spread of bullets can be setup on spread angle. The angle is determined from a curve **SpreadAngleCurve** depending on a current energy level.



Weapon Damage Settings

- **Damage Amount.** Base damage per one bullet.
- **Damage Coefficient.** Dependence on curve of distance damage.



Take Damage Logic

Damage value is transferred to the event Take Damage from weapons system component, Create Trace & Damage System & SFX & Sounds Graph, section TAKE DAMAGE. After it processed in BaseCharacter blueprint. If you wish, a transferred damage variable can be connected to a different damage & health system.

Recoil & Spread Coefficient Settings

During shooting, certain logic calculates recoil and spread. After this, obtained recoil value and spread value are multiplied by the coefficients. The coefficients depend on a character's states.

- **Spread Floor Blind Coefficient / Recoil Floor Blind Coefficient.** Use when a character has standing position.
- **Spread Floor Minizoom Coefficient / Recoil Floor Minizoom Coefficient.** Use when a character has standing position and has state minizoom.
- **Spread Crouch Blind Coefficient / Recoil Crouch Blind Coefficient.** Use when a character has a crouch state.
- **Spread Crouch Minizoom Coefficient / Recoil Crouch Minizoom Coefficient.** Use when a character has crouch and has state minizoom.
- **Spread Move Coefficient / Recoil Move Coefficient.** The last multiplying coefficient. Use when a character is moving in any state.

SFX & Sound Settings

- **Hit Point SFX.** The effect that will create in a place collision of line trace, where a weapon gets bullet.
- **Muzzle SFX.** The effect muzzle that will create in near of weapon's end barrel.
- **WeaponShotSound.** The sound played in a weapon
- **HitPointShotSound.** The sound is play in a place where line trace hits.

Weapons Pickup

The product supports pickup of weapons. There is bp class which located in TPSCharacterSystem\Blueprints\PickUps folder. This class can be placed on a level. It has one setting - Weapon Item Class (class of a weapon which will be spawned on begin game).

Player's Camera

Player's Camera Overview

This system allows to change a states of a player's camera depending on any actions: transition to sprinting, running, crouching, etc. The system operates with four camera variables: Target Arm Length, Socket Offset, Target Offset, FOV.

System Structure

The camera system there is in the folder BluePrints - Camera. The logic of the work is in AdvancedTPSCamera actor component, which connects to BaseCharacter. Settings for camera mods are in CameraModeLib.

In section SET START CAMERA in BaseCharacter is logic of set start camera mode. Also in this section you specify the parameters of the maximum vertical deviation angles.

If you need to add a new mode or remake an old one, the transitions between modes are controlled by the logic in the event UpdateCameraState in CameraUpdater Graphs, BaseCharacter blueprint.

Customize Transition Camera Modes

In modes settings, you can set a curve for transition. There are 2 types of transitive curve: default and unique for transition from a certain mode. The logic of a camera mode selection depends on a correctness of a name modes in CameraModeLibrary and a

name of a curves. Consider example of CrouchIdle mode. This mode has 3 transitions curve:

- **CrouchIdle_Default**. Used to change camera settings in cases where the transition to CrouchIdle from any other mode, except CrouchMiniZoom or CrouchWalk.
- **CrouchIdle_CrouchMiniZoom**. Used when CrouchIdle mode camera goes from CrouchMiniZoom.
- **CrouchIdle_CrouchWalk**. Used when CrouchIdle mode camera goes from CrouchWalk.

Correct spelling of a name of the curve:

ModeToWhereWeWantToGo**From**ModeFromWhereWeAreGoing.

When adjusting curves, it is necessary to take into account that the transition time from one camera modes to another depends on the horizontal time scale.

Customize Camera Shakes

The camera system allows to create camera shakes for sprint and shooting. Sprint uses CamShake_Sprint. Shooting uses CamShake_Shooting.

Cover System

Cover System Overview

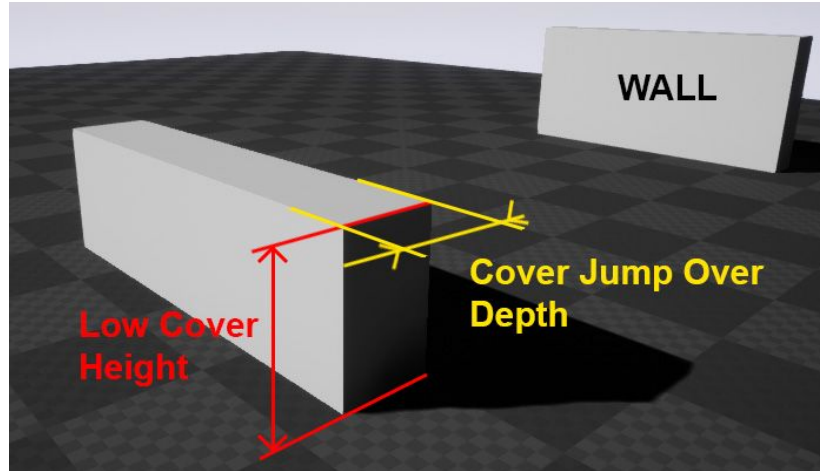
The logic of the covers system is in CoverAnalyzer component, which is connected to the player's controller. The system uses traces for analyze level geometry and gives an opportunity to use 2 possibilities: climbing and jump over.

In the first version of the product, mechanics are implemented only for low covers. In future, you will add new features for this system.

Customize Covers Analyzer. Covers Settings

The system settings there are in CoverAnalyzer in the group of variables Cover Settings.

- Low Cover Height (Default Value = 115). Height of low cover
- Wall Height (Default Value = 130) Height of high obstacle. The possibility of climb or jump over is disabled.
- Cover Jump Over Depth (Default Value = 90). Maximal thickness of a cover, for possibility of a jump over.
- Cover Check Angle (Default Value = 0.2). The maximum angle of the character's deviation from a cover



Customize Climb & Jump Over Movement

Cover Analyzer transfers variables for climb or jump over to BaseCharacter blueprint. The logic of moving a character when climbing and jump over is in CLIMB and JUMP OVER sections.

Moves of each type is divided into 2 parts, and uses set location node with time line nodes. Each time line node uses time curve. A time in timeline nodes determines a speed of movement in climb or jump over.

Main Menu

Setup Main Menu

In order for the main menu to function it is necessary to make 2 steps. Connect a game instance with the function of launching & join session. This setting is located in project settings - maps & Modes.



Need to add new strings in DefaultEngine.ini , which is located in TPSCharacterSystem\Config.

[OnlineSubsystem]
DefaultPlatformService=LAN

DefaultEngine — Блокнот

Файл Правка Формат Вид Справка

[URL]

GameName=TPSCharacterSystem

[/Script/HardwareTargeting.HardwareTargetingSettings]

TargetedHardwareClass=Desktop

AppliedTargetedHardwareClass=Desktop

DefaultGraphicsPerformance=Maximum

AppliedDefaultGraphicsPerformance=Maximum

[/Script/EngineSettings.GameMapsSettings]

GlobalDefaultGameMode=/Game/TPSCharacterSystem/Blueprints/DefaultGameMode.DefaultGameMode_C

EditorStartupMap=/Game/TPSCharacterSystem/Levels/playground.playground

GameDefaultMap=/Game/TPSCharacterSystem/Levels/MainMenu.MainMenu

GameInstanceClass=/Game/TPSCharacterSystem/Blueprints/MainMenu/MainGameInstance.MainGameInstance_C

[OnlineSubsystem]

DefaultPlatformService=LAN

[/Script/Engine.PhysicsSettings]

DefaultGravityZ=-980.000000

DefaultTerminalVelocity=4000.000000

DefaultFluidFriction=0.300000

.....