

Ultimate FPS Camera

© 2012 VisionPunk, Minea Softworks. All Rights Reserved.

Webplayer demo

<http://www.visionpunk.com/products/ufpsc/webplayer/demo.html>

Video preview

<http://www.visionpunk.com/products/ufpsc/video/preview.html>

Asset Store

<http://u3d.as/content/vision-punk/ultimate-fps-camera>

Forum thread

<http://forum.unity3d.com/threads/126886-Ultimate-FPS-Camera-RELEASED?goto=newpost>

Contact

info@visionpunk.com

<https://twitter.com/#!/VisionPunk>

v1.33 RELEASE NOTES

>> IMPORTANT: BACKUP ALL EXISTING PRESETS AND SCRIPTS BEFORE INSTALL <<

Installation

If you haven't previously installed v1.3, please read the v1.3 release notes below carefully before installing.

General info

This is a compatibility update to get Ultimate FPS Camera ready for Unity 4.

Special notes

- Unity 4 component hierarchy cannot be altered in "Awake" (this is good to start preparing for, since it'll be a breaking change for many projects). For UFPS it means that initialization code gets pushed to "Start". Subsequently, all code in "Start" that depends on finished initialization needs a new place to live (see changes to "vp_Component" below).
- In Unity 4, "GameObject.active" and "GameObject.SetActiveRecursively" have been made obsolete. In order to stay backwards compatible, Ultimate FPS Camera will keep using these for the time being. This will work fine in Unity 4, but will unfortunately generate warnings there until I drop support for Unity3.x.

Changelist

- vp_Component: Extended with "Init", a method complementing "Awake" and "Start" that will run once in the first call to "Update". "Init" can be overridden in any class derived from vp_Component to provide this option in a clean manner. Also, added descriptions outlining this class in a bit more detail.
- vp_FPSCamera: Added the "Init" override to allow the default "SetWeapon" call to happen after all "Start" calls but before everything else. Changed default setting from weapon 1 to weapon 0 (none) which often makes more sense. NOTE: vp_FPSPlayer, however, overrides this and in turn sets weapon 1 as its default, which is more beginner friendly.
- vp_FPSShooter: Muzzleflash initialization moved from "Start" to "Awake"
- vp_FPSWeapon: Weapon group, pivot and springs initialization moved from "Start" to "Awake". Removed a bug that would execute "base.Update" twice. Added a few more null checks for stability.
- vp_Component derived classes: All standard Unity methods made overridable. "m_Delta" is now a property named "Delta"

- Presets "WeaponMachinegun" and "WeaponMachinegunReload": Fixed an issue where the machinegun would accept too high input velocities - for example during very long falls - resulting in the arm sometimes flying around detached from your body. To fix this, changed "PositionMaxInputVelocity" from 1000 to 20 (this is a great example of the purpose of that parameter BTW. It's basically there to keep the springs in check).

v1.32 RELEASE NOTES

>> IMPORTANT: BACKUP ALL EXISTING PRESETS AND SCRIPTS BEFORE INSTALL <<

Installation

If you haven't previously installed v1.3, please read the v1.3 release notes below carefully before installing.

General info

This is a small bugfix update. Also, some simple code comments have been added for demonstrating how to play conventional animations on the weapon in vp_FPSPlayer.cs.

Bugfixes and misc. changes

- vp_FPSWeapon: fixed an issue with layer assignment on weapons not being recursive, resulting in weapon sub objects with the wrong FOV, clipping etc.
- vp_FPSWeapon: removed some obsolete code which modified the names of weapon models, breaking animation clip metadata
- vp_FPSPlayer: added commented-out code snippets showing how to play animations on the weapon object upon firing
- FPS editor classes: fixed an issue with inspector values being reverted upon pressing play
- CameraDefault.txt: updated a reference to an old parameter that was resulting in a preset error

v1.31 RELEASE NOTES

>> IMPORTANT: BACKUP ALL EXISTING PRESETS AND SCRIPTS BEFORE INSTALL <<

Installation

If you haven't previously installed v1.3, please carefully read the v1.3 release notes below before installing.

General info

This is a very small hotfix update to address issues with trigger detection.

Bugfixes and misc. changes

- Broken trigger detection caused by crouching logic fixed
- An issue with the camera collision code jittering upon trigger collision fixed
- OnGUI method added to vp_FPSPlayer with snippets to display a simple health and current ammo HUD

v1.3 RELEASE NOTES

>> UPDATING? BACKUP AND DELETE YOUR OLD FOLDER FIRST! <<

Installation

Do NOT install this version over the previous version of the Ultimate FPS Camera folder! This is a very extensive overhaul of the system. It's more than likely to break your project. If you're close to a deadline you will want to wait. It might be a good idea to first install v1.3 into a clean project and have a look before integrating it into your project.

Recommended update steps:

1. Backup your existing project folder
2. Delete the "Ultimate FPS Camera" folder from your project
3. Go to the Asset Store and upgrade Ultimate FPS Camera to version 1.3
4. Read up on the new important concepts below
5. Integrate the new system with your existing game scripts

General info

Ultimate FPS Camera v.1.3 is a major update with many new features, optimizations, bugfixes and system overhauls. This release aims to make it easier to get started with Ultimate FPS Camera, to reduce scripting complexity and add many requested features, helper scripts and special effects for getting up and running quickly with a new FPS. The manual has also been expanded with 14 more pages, explaining new and old concepts in great detail.

Special notes

State Manager

This version introduces a new powerful State Manager to handle things like running, crouching and zooming. Every FPS component is now able to blend between multiple simultaneous Presets smoothly, depending on the current combination of States. This greatly reduces the need for scripting state changes.

"Resources" loading replaced by Text Assets

Loading from the "Resources" folder is being phased out in favor of Text Assets. This makes for a better editor workflow, reduces executable / webplayer file sizes and is better suited for asynchronous asset streaming. Furthermore, it removes tons of hardcoded paths from the scripts and is just more of the 'Unity way'. That said, it's still possible to load Presets from the Resources folder even though it's no longer used by the demo scripts

SimpleScript replaced by vp_FPSPlayer

v1.3 has a new well-featured player script that may be used as a starting point for a game. This script makes extensive use of the new State Manager. It is suited for heavy customization to meet your game's various needs. NOTE: this script completely replaces the old 'SimpleScript'. vp_FPSPlayer is now the recommended starting point for studying how this system works (the other demo scripts are highly specialized for the demo walkthrough and are not meant to be used as the starting point for a game, nor examples of best workflow practices).

Directory structure overhaul

The directory structure has been cleaned up and redesigned to separate the core systems from demo content and functionality. NOTE: the root "Presets" folder now contains "clean" presets for use with a new FPS game. The Demo folder has its own "Presets" folder with tons of less useful, demo specific presets.

Image Effects & weapon camera changes

The weapon camera functionality has been redesigned in order to allow for Unity Pro image effects - and to make the system more tweakable + less error prone in general. There is no longer one auto-generated weapon camera per weapon. Instead there is a global weapon camera that exists alongside the weapons in the hierarchy (also when the game is not running). Note that image effects should preferably be added to this new WeaponCamera, not the

FPSCamera (see the Image Effects chapter in the manual for detailed information about this).

New base class for FPS components

Base class `vp_Component` introduced to minimize code duplication. All FPS classes except `FPSPlayer` now inherit from it (`FPSPlayer` is a regular `monobehaviour` to allow for heavy customization).

New features

- New, more "game-ready" FPS player class designed as a starting point for a game or game prototype (replacing "SimpleScript").
- Shooter: New ammo & reload functionality
- New damage handler script to inflict damage on gameobjects and make them respawn with a delay. Bullets now execute a method on the hit object with a floating point damage parameter. Name of the damage method can be set per bullet type/prefab.
- Smooth weapon switching code integrated into the camera system
- Explosion class with area damage, a shockwave affecting camera shakes + player velocity along with example particle effects
- New advanced state system to blend between multiple presets in a controlled manner (reducing the need for scripting in many cases)
- Image effects: Big weapon camera overhaul to allow for Unity Pro image effects. Weapon cameras are no longer one per weapon & auto generated. There is one single weapon camera directly under the `FPSCamera`. System tested with all Unity Pro Standard Assets image effects.
- New feature to prevent the camera from going through walls. A ray is cast every frame from the camera spring's rest position to its current position. If it crosses a wall, the camera is moved back with a margin. A debug ray can be drawn to visualize the intersection.
- A new set of standard presets for the pistol, revolver and machinegun with crouch, run and reload presets
- Improved crouching logic, shrinking the character controller to allow for climbing through small spaces. NOTE: In order to pull this off without having the player fall through the ground, the center of the controller has been moved to its bottom. This may introduce strange bugs in existing projects but should be easy to fix.
- Presets can now be loaded into memory so you don't have to load them from the resources folder all the time (loading from resources as a methodology is being phased out in favor of loading from text assets, which is better practice)
- Parameters to limit springs and bob so they don't freak out under extreme velocities
- Feature to save 'tweaks only' to presets, greatly simplifying the process of creating additive presets
- A bunch of new sounds and particle effects
- New demo framework created to better separate the system from the walkthrough demo. Simply disable or delete the "Demo" gameobject in the example levels to remove the demo entirely and get a fresh FPS level.
- Demo walkthrough enhanced with some of the new features
- Earthquake and bomb functionality expanded to allow for much more customizable earthquakes and explosions
- `FPSCamera`: standard weapon switching functionality introduced. Weapons can be moved smoothly in and out of view with wield and unwield sounds.
- `FPSWeapon`: Added parameters to limit and scale spring and bob input velocity. This is useful if you change player velocity and spring/bob motion then freaks out - springs can now be limited and scaled back without having to adjust all values.

Bugfixes and misc. changes

- `SimpleScript` removed and replaced by `vp_FPSPlayer`
- `DemoScript` removed and replaced by the scripts `FPSDemo` and `FPSDemo2`, using the new demo framework

- The new preset & state system implementation is designed to be more forgiving to beginners (e.g. not resetting component changes upon play)
- Many (if not most) 'private' methods made 'protected' to allow for extending the system without altering the core classes
- Renamed "RefreshSettings" to "Refresh" in all classes. shorter and sweeter.
- vp_FPSCamera: Weapon sounds no longer cut off when switching weapons after recently firing a shot
- vp_Bullet: Bullets are now only added to the decal manager if they have a renderer, for mobile games that may not use decals.
- vp_ComponentPersister: Heavy modification of the component persister to make it work with the new state system and make it stable when loading another level.
- vp_ComponentPreset: Major rewrite of the preset system to work with textassets and the new state system.
- vp_FileDialog: Improvements to the file dialog class to make it work better with various warning message box states.
- vp_FPSCamera: Removed a questionable hard coded camera position offset. This change will probably mess with existing presets and projects but it's better long term.
- vp_FPSCamera: Angle property added to clamp pitch values better (replaces setangle)
- vp_FPSController: Support for making the charactercontroller (i.e. collision) smaller when crouching.
- vp_FPSController: Fixed a floating point issue with MotorThrottle. It didn't come to rest at 0.0f properly which messed with the anti-bump feature.
- vp_FPSController: Velocity property added.
- vp_FPSController: Added a check to prevent moveDirection from becoming NaN which would cause a crash.
- vp_FPSController: Jump force range increased in the controller editor
- vp_FPSShooter: Fixed a bug with next allowed fire time being set to a large value by default which sometimes prevented the weapon from being fired for a very long time
- vp_Layer: Added "Props" layer at position 28. Used in this case by the new DamageHandler for respawn logic
- vp_Timer: Changed the name of the 'At' method to 'In' which makes for more elegant readability, for example: to do something in 5 seconds: vp_Timer.In(5, DoSomething);
- vp_Timer: Small fix for more stable creation of timers

----- v1.21 RELEASE NOTES

>> IMPORTANT: BACKUP ALL EXISTING PRESETS AND SCRIPTS BEFORE INSTALL <<

General info

This is a bugfix release.

Bugfixes

- Fixed an issue with FPSShooter giving NullReferenceExceptions resulting in the FPSPlayer prefab working sporadically or not working at all in some circumstances.
- Fixed an issue where multiple weapons would be rendered inside each other.
- Fixed an issue where removing a weapon from the FPSPlayer would break the functionality of the other weapons.
- Improved zoom functionality in SimpleScript.cs.
- Fixed an issue in DemoScript.cs where the weapon layer would not be reset after leaving the 'Weapon Layer' demo screen with layer set to 'OFF'.

Special Notes

A crash can be caused by using the FPSPlayer prefab from 1.11 or earlier. This can happen since the FPSPlayer prefab used to be located in the root of the package, but the new FPSPlayer

is in the "Prefabs" folder. In other words, the old one is not overwritten by installing v1.2 or above. If you have an old FPSPlayer prefab in the asset root, please delete it and only use the one in the "Prefabs" folder.

v1.2 RELEASE NOTES

>> IMPORTANT: BACKUP ALL EXISTING PRESETS AND SCRIPTS BEFORE INSTALL <<

General info

With version 1.2, Ultimate FPS Camera has been expanded with a battery of new shooter features including advanced recoil settings, raycasting bullets, a decal manager for bulletholes, realistic shell case physics and muzzle flashes. Several example particle FX prefabs are included. The manual has been extended to discuss all new features in extensive detail.

Special Notes

Most features from v1.11 should be intact with one exception: The "Gameplay" foldout of the Shooter editor has been removed and the "FiringRate" parameter contained in it has been moved to the new "Projectile" foldout. Any instances of the parameter "GameplayFiringRate" in your existing preset scripts must be renamed to "ProjectileFiringRate". Other than that, it's all new goodies:

New Features

- Shooter: Big update. Added support for spawning projectiles with a muzzle flash and ejection of shell casings. Multiple projectiles can be fired simultaneously, with configurable conical spread (for scripting shotgun type weaponry or manipulating accuracy at runtime).
- Shell: A shell casing script with its rigidbody physics adapted for more realistic behavior. Has a simple optimization feature to control the amount of shells left on the ground. The shells are ejected with configurable velocity + random spin and bounce with random bounce sounds. Shell ejection can be delayed for e.g. shotguns or grenade launchers.
- Muzzle flash class: Renders an additive, randomly rotated, fading out muzzleflash.
- Decal manager: A class to handle fading out and removal of decals. The max amount of decals in the scene is fixed. A decal stays on the wall forever if no more than a certain amount of decals are added. After that, a number of the oldest decals will start fading out a tiny bit for every shot fired, until eventually being removed.
- Bullet class: A generic class for hitscan projectiles. Raycasts to the target and spawns a decal, a bunch of particle fx and a random impact sound.
- Layer manager: Provides an overview of allocated layer id's in one single place. Makes it easier to debug layer issues.
- Sounds: 4 bullet impact variants and 4 shell bounce variants for random variation.
- Bullet impact particle FX for firearms: Debris, Dust, Impact and Spark prefabs.
- New example models and textures: A shell casing, 2 muzzleflashes and a bullet hole.
- DemoScript, SimpleScript and FPSPlayer prefab adapted to the new Shooter systems.
- Shooters enabled on the Mech, Sniper Breath and Turret examples.
- Zoom examples now behave more like a modern fps (although this is just a simple example). Weapon goes to the middle of the screen and movement speed and accuracy is affected.

Bugfixes

- DemoScript: Firing while crouching is now handled better.
- FPSCamera: Camera cullingmask for weapon layer handled better for compatibility with other camera layer utilizing systems and shaders (e.g. scene reflective water or camera rendered gui systems).

v1.11 RELEASE NOTES

>> IMPORTANT: BACKUP ALL YOUR EXISTING PRESETS AND SCRIPTS BEFORE INSTALLING THIS VERSION. <<

General info

Version 1.11 is primarily a bugfix release with a couple of additional minor features. One substantial change is that most scripts have been renamed in order to avoid problems with other assets and Unity core classes. This will probably mess with your scenes. To get back up to speed quickly, follow these steps:

1. It is recommended to backup and delete the Ultimate FPS Camera folder structure before updating / reimporting in order to avoid problems with old file versions.
2. Re-connect all FPSController, FPSWeapon and FPSShooter scripts in the Inspector, where it says "Missing (Mono Script)".
3. If your old presets cause an error about the wrong component type, inside each preset text file add the prefix "vp_" to the component type name (reflecting the new class filenames).

This is hopefully the last version for a while to require this level of re-installation.

Bugfixes

- Better class naming convention, in order to avoid collision with other assets and Unity core classes. All VisionPunk system classes are now prefixed with "vp_".
- Initial camera & weapon rotation: Fixed an issue where weapon model would be offset if the player had been rotated in the editor. Controller initial rotation is now handled much more robustly.
- Anti-bump offset: Pushes the controller into the ground to prevent the character from "bumpety-bumping" when walking down slopes or stairs. This makes a huge difference in terrain.

New Features

- Slope speed multiplier: Allows slowing down the CharacterController moving uphill, and giving it a speed boost running downhill.
- Fall sway factor if grounded: You can now make weapon vertical sway less pronounced when moving in slopes than when falling.
- Random fire sound pitch: Optionally pitches the sound of each shot slightly different to get a more organic firing sound.

v1.1 RELEASE NOTES

>> IMPORTANT: BACKUP ALL YOUR EXISTING PRESETS AND SCRIPTS BEFORE INSTALLING THIS VERSION. <<

General info

Version 1.1 is an interim release with a significant overhaul of the component structure. The "Weapon" part of the camera class has been moved to its own class, FPSWeapon. This will pave the way for more advanced and flexible weapon features in upcoming releases. It also results in a better object oriented structure. If you're an existing user, this means you will have to manually update your camera presets from version 1.0. See below for a walkthrough on how to do this.

New Features

- New component: FPSShooter - with an advanced recoil functionality. Lots of parameters to tweak for making your weapon kick like a mule or twist in any direction when fired.
- Timer class, a small but very powerful and easy to use timer system for scheduling all sorts of client side game events.
- 6 high quality sound effects: Earthquake, Explosion, Machinegun, Pistol, Revolver and Stomp.
- Better transitions between weapons in the demo.

Updating your existing camera presets

Old camera presets need to be split into two; one for the camera and one for the weapon settings.

NOTE: This only affects camera preset scripts that you may have created yourself. All demo example presets from v1.0 have been updated in v1.1.

1. Open an existing camera preset saved from v1.0.
2. Cut and paste all lines that begin with the word "Weapon" into a new text file (and preferably save the file as a name with "Weapon" in it).
3. In the original camera preset file, remove the prefix "Camera" from all lines beginning with "Camera" (Note: be careful not to remove the name of the ComponentType, "FPSCamera"). Save the camera script.
4. In the newly created weapon preset file, remove the prefix "Weapon" from all lines beginning with "Weapon".
5. At the top of the weapon preset file, add the following line and save the script:
`ComponentType FPSWeapon`
6. See the manual for information on how to set up FPSWeapon components on your weapons.
7. Remember to load your newly saved presets onto your components using Preset->Load in the Inspector.

----- v1.0 RELEASE NOTES

General info

Easily enhance your game with the ultra smooth motion seen in today's top first person shooters. Ultimate FPS Camera animates your camera and weapons in response to player input, resulting in super lifelike behaviour that animations alone simply cannot match! Ultimate FPS Camera feeds player movements (mouse input, walking, jumping) into the camera and weapon transforms using realtime spring physics, sinus bob and procedural noise shaking. The system uses over 50 parameters to manipulate the camera and weapon model, allowing for a vast range of complex, realtime-generated behaviors.

Features

- Spring physics: Breathe life into everything from melee weapons to guns and cockpits
- Configurable Mouse Smoothing and Mouse Acceleration: say goodbye to jerky input!
- Procedural Weapon and Camera Shakes
- External forces like Earthquakes, Shockwaves, Stomping, Impacts
- A powerful Preset System
- Persist run-time component changes
- 3-axis versatile Bob
- 4 weapon models: Pistol, Machinegun, Mace and Revolver
- Full, well-commented C# Source Code