



# **SprintR<sup>TM</sup> VR**

Game Conversion Guide

100 International Drive,  
23rd Floor Legg Mason  
Tower

p. 1-410-929-5054

[developers@sprintRVR.com](mailto:developers@sprintRVR.com)  
[www.sprintRVR.com/Developers](http://www.sprintRVR.com/Developers)

# Table of Contents

- I. **Project Overview**..... 2
  - Business Case
  - Process Overview
  - Objectives
  - The Movement Problem
- II. **“SprintR Enabled”** ..... 4
  - Best Practices
- III. **Nexus**..... 7
  - Submission Guide
  - Pricing
  - Advertising and Promotion
  - Business Terms Overview
- IV. **Facts & Figures** ..... 9
  - SprintR Facts and Beyond
  - SprintR Capabilities and Developer Enablers
- V. **Appendix**.....10
  - Links and Additional Information

# Project Overview

---

## Business Case

VR is taking off much like mobile was taking off several years ago, and developers that choose to wait to enter the market will miss the big VR opportunities. It was understandable to wait on mobile because converting a PC game to mobile was a challenge. With SprintR, the same is NOT true for VR. Our device makes it possible for developers to convert their existing FPS/Action games to VR using a very simple process; which is outlined below. There's no need to watch the train roll by for this upcoming market because we offer you the tools to capitalize on your existing content without making a "big bet". Re-monetize you existing content, and give players what they want!

## Process Overview

This video shows the need for SprintR: [https://youtu.be/ngdjDq\\_kPHM](https://youtu.be/ngdjDq_kPHM)

Here is a student using SprintR for the first time: <https://youtu.be/nlEvTScAMdM>

SprintR impressions by Puzzle Quest designer Steve Fawkner and Ted Lange (Exec at Reverb):

<https://www.youtube.com/watch?v=SMCnKkHp8DM>

<https://www.youtube.com/watch?v=rr9V63bwA7s>

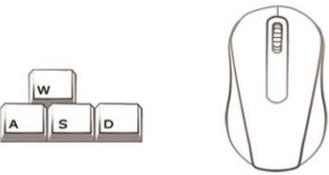
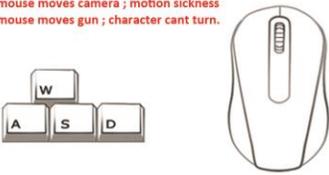
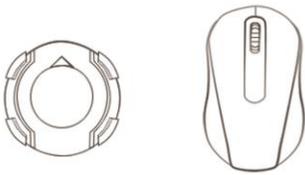
By offloading movement to the feet, SprintR enables developers to convert their games to VR and retain full functionality. The locomotion problem inherent in VR games is easily solved with SprintR, and this guide will help you get your game project VR ready and SprintR enabled. Using SprintR and good VR development practices, you can eliminate simulation sickness and provide your players with a rich experience. We will outline best practices for converting your game to VR using SprintR as the control solution, and show you how to keep the full control functionality of your game by doing it. This guide currently outlines the process for FPS titles, but will include special notes for other types of games that would work well in VR when matched with SprintR. The guide will continue to grow as we update it with best practices for all types of games, and we encourage developers to reach out with ideas and example of innovative control schemes you've tried with SprintR.

We'll also cover the basics our Nexus App store, which is bundled with the SprintR hardware. Nexus will allow anyone with SprintR hardware to buy your games and know that it's SprintR enabled. It's a marketplace for VR games, experiences, and innovative ideas. Your projects are fully hosted, and payment processing is handled by secure servers, and we provide analytics to track sales.

## Objectives

- Explain the VR movement problem
- Outline best practices for VR
- Provide information on how to convert existing games to VR - "SprintR enabled"
- Outline Nexus App-store business model / developer commercialization plan

# The Movement Problem

Standard Games PC: Standard FPS Control Setup <span style="float: right;">1</span>	VR Games Virtual Reality PC: Standard FPS Control Setup <span style="float: right;">2</span>	Virtual Reality PC: SprintR / Mouse KB <span style="float: right;">3</span>
 <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Character and Camera</li> <li>✓ Turns Gun (Locked to camera)</li> </ul>	<p><b>Movement Problem in effect</b></p> <p>1) If mouse moves camera ; motion sickness 2) if mouse moves gun ; character cant turn.</p>  <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Character and Camera (Nausea)</li> <li>✗ Turns Gun</li> </ul>	<p><b>Head is Free to look around</b> ✓</p>  <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Gun/Hand (Free move)</li> <li>✓ Turns Character</li> <li>✓ Jumps / Leans / Crouch</li> </ul>
Console: Standard FPS Control Setup	Virtual Reality w/Gamepad: Standard FPS Control	Virtual Reality : SprintR / Gamepad
 <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Character and Camera</li> <li>✓ Turns Gun (Locked to camera)</li> </ul>	<p><b>Movement Problem in effect</b></p> <p>1) If Right Thumbstick moves camera/head ; motion sickness 2) if Right Thumbstick moves gun ; character cant turn.</p>  <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Character and Camera (Nausea)</li> <li>✗ Turns Gun</li> </ul>	<p><b>Head Free to look around</b> ✓</p>  <ul style="list-style-type: none"> <li>✓ Moves character</li> <li>✓ Turns Gun/Hand (Free move)</li> <li>✓ Turns Character</li> <li>✓ Jumps / Leans / Crouch</li> </ul>

While standard input may work fine for a screen based experiences, the same doesn't hold true for VR. We don't often analyze the typical WASD or gamepad movement scheme, but there's a few critical things to be aware of because of how they impact the player in VR. In a standard FPS, the WASD keys only allow for movement along 8 vectors (front back, sides, diagonals), but in order to properly walk around an environment, players must move the mouse left/right to "turn" the character. Doing this in VR amounts to using your hand to drag the head/body into a specific direction; and simulation sickness can strike rather quickly when players are "dragging the head" around with a mouse or thumbstick. With SprintR, this turning is offloaded to your foot. The player rotates their ankle left or right to indicate the direction they wish to turn the character. Now stand up! Turn 90 degrees to the right. If you pay attention to your foot as you begin to turn, you'll notice that your ankle does the same sort of rotation in real life that SprintR requires to turn your character in-game. This 1 to 1 movement is very natural for the body, and will be a natural way for your players to turn their characters in-game.

By offloading WASD movement and mouse based turning to the SprintR, you now have more options for the mouse (or thumbsticks); which we will review in more detail below. You may also consider hand based controllers, whether this is a VR glove, the Razer Hydra, or Sixsense Stem; SprintR works fine with all of these hand based controllers, so the possibilities are very intriguing. This guide will generally stick to the mouse/KB or gamepad for hand based controls, and outline how to best take advantage of these ubiquitous control solutions in conjunction with SprintR.

# “SprintR Enabled”

---

Making your game “SprintR enabled” is a very easy process. The controller loads as a standards HID joystick, so any game-level integration for baseline functionality is handled the same as any joystick; assign your movement axes and buttons as you generally would. [Here are the PSD](#) files need to implement SprintR in your controller/button assignment menu; should you wish to let the player alter the baseline controls you offer. If your game is already VR, you don’t have much to do; just map the controls and your game is SprintR enabled. If your game isn’t VR, and you’re creating a VR SKU of an existing title to take advantage of SprintR’s movement solution, there’s a few key considerations:

## Best Practices

1. **If you game is already VR:** Map the controls to SprintR and you’re done. ✓
2. **Making your game work in VR is your primary task:** If your game does not support stereoscopic rendering, or there isn’t an Oculus/Vive plugin for the engine you’re using, that’s your first, and probably biggest task. Look at the oculus developer center for their SDK; Located here: <https://developer.oculus.com/downloads/>. Engines like Unreal and Unity already have integration, so if you’re using one of those, make sure to check for the existing plugins and take advantage of them for your title if possible.
3. **VR best practices:** Here is a list of best practices the good folks over at Oculus have developed for VR. The more of these items you take into consideration, the better your VR port/game will be/feel:  
[https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp\\_intro/](https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_intro/)
4. **FPS Weapon/Prop set up:** In a traditional FPS, your gun is superglued to your character’s head. You move the mouse and this moves the gun as well as the character’s head; it also changes the character’s entire orientation. Notice that if you’re pressing “W” to walk, the mouse is what sets your character’s movement rotation (walking left or right) the “A” and “D” keys only enable strafing/side steps. VR FPS’s cause motion sickness because with this traditional movement scheme you’d have to move the mouse to rotate, and this in turn “drags” the players head (because it’s attached to the gun). SprintR detaches the hands / guns/ props

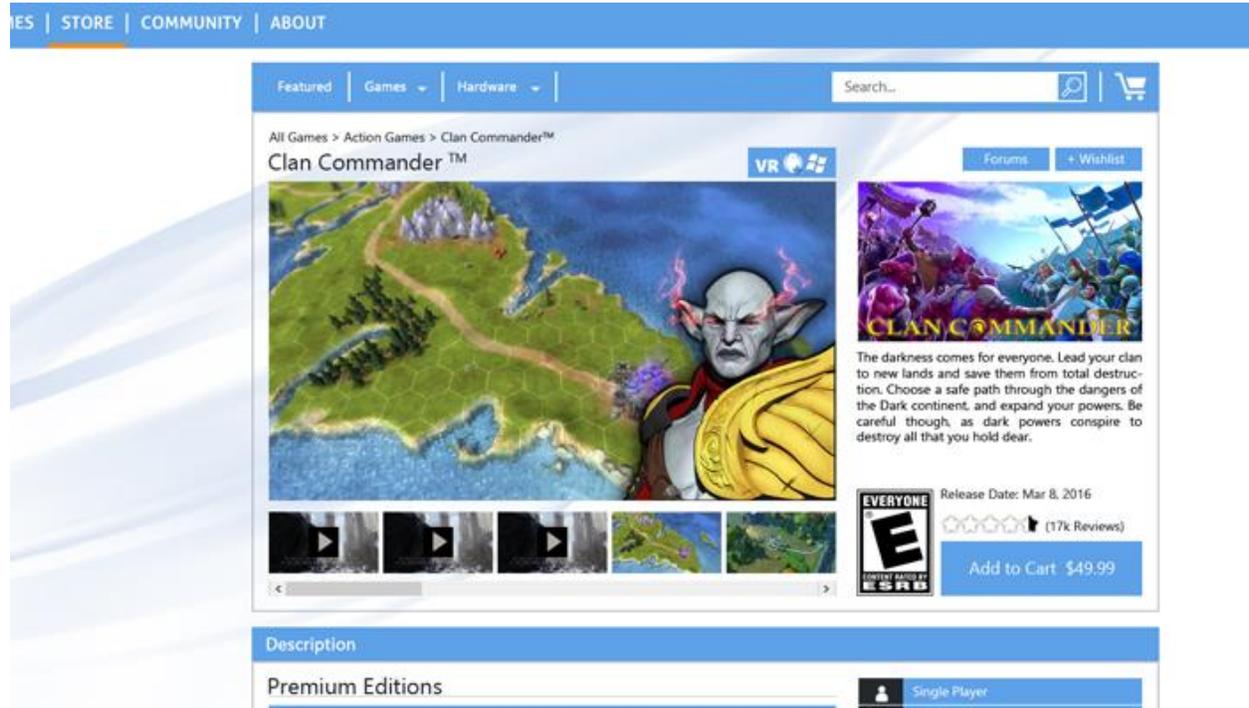


from the camera. This is critical in VR because the player's head is the character's head, and the rotation of the head is controlled by the player's real world head/HMD rotation.

- a. The hands/guns/prop should be controlled by the mouse (or right thumbstick) and move freely. SprintR enables players to turn their character left/right as they walk by rotating their foot. So you are free to use the mouse to point the gun at any location on the screen. This allows the player to have fine precision when shooting.
- b. Use a world-space crosshair some reasonable distance out from the player. Since the gun is no longer forced to shoot at the direct center of the screen, the player will need an indication of where they are firing. You're free to try other aiming concepts in your title, and perhaps develop innovative ideas around aiming and player gun mechanics, but current best practices indicate a world space crosshair/dot/laser beam.
- c. When the player moves their head (causing the headset/game to change the view frustum), pull the gun/prop along with them. This may seem counter-intuitive, but the reason is simple: If the gun goes off screen too often, the player will become frustrated because they have to keep finding it again. Without the ability to "feel" their virtual body, the player won't know where the gun is. Avoid this by implementing the solution outlined above.
- d. Watch for gimbal lock on your gun/prop solution. This can occur when the gun is moved along the vertical axis to a point where it flips. It will be odd for the player, and cause strange visual artefacts. Here is a quick tutorial on Euler rotation and gimbal lock issues: [YouTube Gimbal Overview](#)



5. **UI & Keyboard functions:** The player can't readily see the keyboard when they're in VR. This may limit the functions the player can access, and therefore limit your game. The solution is to use the 'spacebar' to enable a virtual function array. This means that when the player presses and holds the spacebar, their mouse is set free of the gun/prop, and becomes a cursor. You may use this to interact with items in the world (i.e click a button on a door), or you can bring up a 2d UI overlay that the player can access. This is very helpful if you have a lot of functions you wish to quickly present the player (spell selection, weapon switching, items & armor loadout). The reason we choose the spacebar to enable this is because it's the largest key, and there are no other keys beneath it. It's easier for the player to find unaided.
  - a. You may also wish to make your world more interactive by offloading items that were previously in your UI onto in-game surfaces. For example, instead of pressing "e" to get into a vehicle, you can have the player's target cursor change to a selection icon when they are close enough to a vehicle's door, then when they click on the door they can enter (highlight/outline the door to provide added visual indication of intractability).
  
6. **Cinematics:** This section will require some artistry. There are some times we DO want to move the player, and many times where we don't. If your cinematics involve constantly moving the camera, consider showing the cinematics via in-game screen. This means you have a "TV-like screen" which remains stationary, that the player can use to "watch" the cinematics in VR. Try NOT to drag the player around, you're almost assured to cause simulation sickness. This is mostly a VR best practice, but if you're porting your existing game to SprintR, it's a step we recommend in order to give your players the best experience. If you do perform an audit of your cinematics, be careful about movement, acceleration, and other strange distance based issues. If you absolutely need the player to look at specific thing/occurrence, make sure to draw attention there via sound, visual cues, and other real-life indicators. Grabbing the players head and forcing them to look is not a good solution.
  - a. Wait Cue: Since you're no longer the master of the camera, you may find that it takes the player a bit longer to look where you want, or see what you need them to. This will necessitate the creation of wait cycles for your NPC's or specific events while the player becomes aware of what they need to. This is an exciting challenge, because you must consider a more complete presentation. More often than not, the player will pick up on what they need to quickly.
  
7. **Beyond Standard Conversion:** You may consider implementing VR quality/specific spatial audio, and all of the extra niceties that will enhance the experience further.



Nexus is a free, stand-alone application / sales portal that comes with SprintR. The SprintR personalization settings are contained therein, and it's also a gateway for players to purchase SprintR enabled games. Publishers have access to a special admin area where they can upload files, create game store pages, and also manage analytics.

## Submission Guide

- 1) You can either sign up online or download the Nexus desktop application.
- 2) Create an account with an email associated with your studio.
- 3) Once you sign up, go to "settings -> apply for publisher status" and fill in the required fields.
- 4) Once approved you'll have access to an admin back-end.
- 5) Click "submit new product" and fill in the pertinent fields. The CMS will allow you to upload images, link videos, and create a game page link the screenshot above.
- 6) You can spend some time and setup your "publisher hub", which is very similar to a Facebook Page.

## Pricing

Pricing is up to the publisher, but we will help you determine the best sales price for your products. Nexus will feature periodic sales and feature placements for our partners; Participation in these events will be optional. Publishers can set pricing per title when they are creating a game's product page.

## Advertising and Promotion

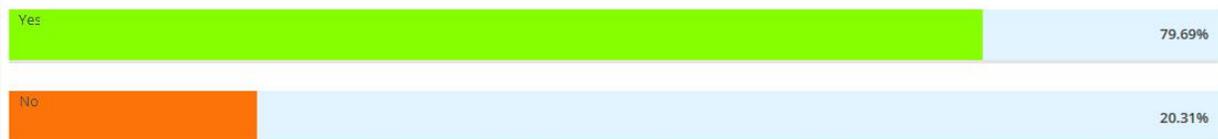
Our goal is to highlight quality titles. Front page listing will generally be based on titles that are proven quality, or have a stirring hook for the player. If a publisher feels they have a particularly exciting product available, we will work with you to determine the best placement.

Promoting your VR ports is a practical point to consider when releasing on Nexus. Users who have a SprintR device will all have access to your titles, but it's recommended that you reach out to your player base and let them know about your forthcoming or recently released VR projects.

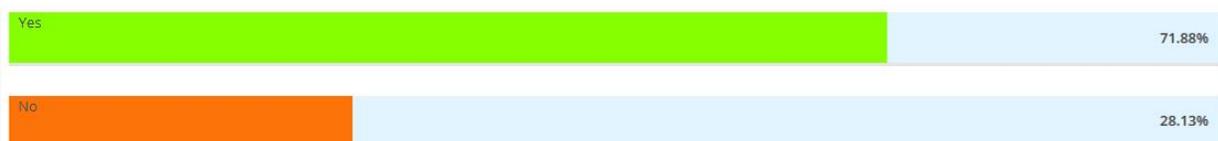
Games that are SprintR enabled will require clear marking indicating this hardware requirement. Make sure that your customers understand that in order to play the games on Nexus, they will require a SprintR controller. This is important because players who buy a game on Nexus, and don't have the required hardware will not be able to properly interact with your title, and in this instance we will process a return for them.

We've conducted user surveys which show that players want their favorite franchises in VR. And what's more, they're excited to play games that have recently been released, but in VR form. 74% indicated that they'd buy a game again if it had a VR port. Here are some of the responses from our survey:

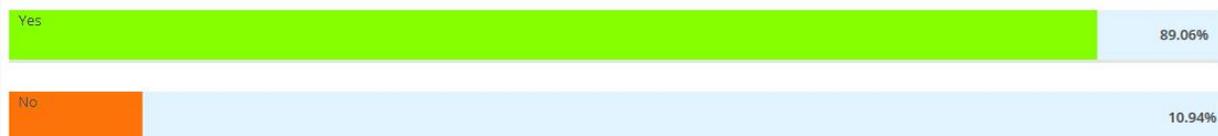
Do you believe VR will enhance a game experience even if the game remains otherwise the same?



Are you interested in playing some of your favorite games from the last year in VR?



If you could play your favorite FPS/Action game in VR by using a specific input device, would you buy that device?



This is a great opportunity for publishers to enter the market without taking a big risk, and a great opportunity to give players what they want: their favorite games in VR.

## Business Terms Overview

- Publisher favoring revenue share
- Simple CMS interface for game submission and management
- Store management tools
- Detailed sales analytics
- A detailed [content distribution agreement](#) is available for review. The terms are simple, and hassle free.

## Facts & Figures

---

### SprintR Facts and Beyond

- SprintR will launch in 2017
- Developer version will be available before launch
- Nexus App-store bundled with device to sell “SprintR enabled” games
- Headset Agnostic
- PC & mobile enabled
- Hand controller agnostic
- 90% of surveyed players said that VR generally lacks content
- 74% surveyed said they’d buy an existing game if it’s VR enabled

### SprintR Capabilities and Developer Enablers

- Allows full character movement: WASD + Turn + Jump + Crouch
- Use the mouse to aim a gun or cursor (free from the head movement)
- Use spacebar (the biggest key) to enable a 3d cursor / interact with UI
- Easily drive vehicles with SprintR & operate a turret with the mouse
- Has analog movement (move a little to sneak, move a lot to run)
- Software functions like quick turn - based on player foot gestures
- Enhance player experience with haptics.

# Appendix

---

## Links and Additional Information

<http://www.SprintRVR.com>

<http://www.SprintRVR.com/Developers>

<http://www.sprintrvr.com/distribution/>

<https://www.htcvive.com/us/>

[https://www.htcvive.com/us/develop\\_portal/](https://www.htcvive.com/us/develop_portal/)

<https://developer.oculus.com/downloads/>

[https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp\\_intro/](https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_intro/)