

A Measurement Study of Vertical Synchronization Configurations in PC Video Games.

Jeremy Tang, jntang (at) go.wustl.edu (A paper written under the guidance of [Prof. Raj Jain](#))

[Download](#)



Abstract:

In computer graphics, vertical synchronization (VSync) refers to syncing the rate of drawn frames with a computer monitor's refresh rate, making the drawn frames have fluid motion while exhibiting no visual artifacts [[Wilson09a](#)]. In video games, both image fluidity and low input latency are important factors in providing a comfortable experience for a video game player. Traditionally, disabling VSync in video games will present the least amount of input latency at the cost of image fluidity and screen tearing, while enabling VSync will present a perfectly fluid and artifact-free image at the cost of significant input latency [[Wilson09b](#)].

The objective of this study is to determine a combination of settings which will minimize input latency while maintaining VSync in PC video games. The study used a 2^k*30 experimental design encompassing a variety of configurations for displaying a rendered frame onto a monitor, such as triple buffering and frame rate locking, and measuring the input latencies for each configuration. The video games used in this study are *Left 4 Dead 2*, *Super Meat Boy*, and *Sleeping Dogs*. We conclude that fullscreen VSync with frame rate locking is the most balanced configuration, providing minimal input latency while maintaining VSync.

Keywords: Video Games, PC Gaming, Frame Rate, Vertical Sync, VSync, Double Buffering, Triple Buffering, Input Lag, Input Latency, Performance Evaluation, Performance Analysis, Experimental Design.

Contents

- [1. Introduction](#)
- [2. Background](#)
 - [2.1 Input Latency](#)
 - [2.2 Frame rate and Vertical Synchronization](#)
 - [2.3 RadeonPro](#)
 - [2.4 Upcoming VSync techniques](#)
- [3. Performance Evaluation](#)
 - [3.1 System and Services](#)
 - [3.2 Goal and Metrics](#)
 - [3.3 Parameters and Factors](#)
 - [3.4 Workloads](#)
 - [3.5 Evaluation Technique](#)
- [4. Results and Analysis](#)
 - [4.1 Left 4 Dead 2](#)
 - [4.2 Super Meat Boy](#)
 - [4.3 Sleeping Dogs](#)
- [5. Conclusion](#)
- [References](#)
- [List of Acronyms](#)

1. Introduction

Video games refer to a class of games that are exhibited on a computer or console. In these games, players look at a display, such as a television or computer monitor, and use an input device, such as a controller or joystick, to perform actions which are then reflected on the display [[MerriamWebster](#)]. Modern video games can be played on either a dedicated video game console or on a consumer PC.

Two important factors in modern video games are image fluidity and input latency. The implementation of Vertical Sync (VSync) in a video game will enforce fluid image motion but will add input latency. Without VSync, input latency is minimal, but at the cost of screen tearing and jittery motion [[Wilson09a](#)].

In PC video games, there are many possible configurations for VSync. Configuration changes can either be performed within the game itself or through the use of an external tool. This measurement study aims to find a VSync configuration that will minimize input latency while maintaining VSync-like image fluidity.

The remainder of this paper is as follows. The Background section covers the important elements needed to understand the study, such as input latency and vertical synchronization techniques. The Design section covers the experimental design, including the hardware testbed, the video games under test, and the factors analyzed. The Results and Analysis section covers the outcomes of the study after measurements. The Conclusion section contains concluding remarks and gives a final verdict on the recommended configuration for balancing image fluidity and input latency.

2. Background

This section provides an overview of important terms and tools used in this study. It describes input latency, frame rate, VSync and its variants, the RadeonPro tool, and upcoming variable refresh rate technologies.

2.1 Input Latency

In video games, input latency, or "input lag," refers to the time between the player performing an input, usually a button press, and the outcome of that input appearing on the player's screen.

There are a variety of hardware factors that can affect input latency. While some input latency can exist within the PC or console itself, input latency can also come from the input devices and from the monitor. For example, input latency can be reduced by purchasing a higher quality controller that uses buttons that take less time to actuate, or by purchasing a higher quality monitor that has a faster processing time [[Wilson09b](#)].

There are also a number of software factors which have an impact on input latency. OS overhead and overhead from other running applications can potentially affect input latency. Another source of input latency is the use of VSync, in that disabling VSync also decreases input latency significantly [Wilson09b].

Input latency is a crucial factor in video games because it affects how much time the player has to react to events which occur on-screen. A player with low input latency will have an advantage over a player with high input latency. Too high input latency can also result in an uncomfortable gaming experience for the player [Wilson09b].

2.2 Frame rate and Vertical Synchronization

In modern video games, graphics are rendered by a graphics processing unit (GPU) found in the PC or console. The video game periodically calls the GPU to render an image, or frame, which will then be displayed by the monitor. The rate that the GPU draws frames is known as framerate and is measured in frames per second (FPS).

In computer graphics, vertical synchronization (Vertical Sync, or VSync) refers to synchronizing the framerate with the refresh rate of the computer monitor. A typical computer monitor refreshes at 60 Hz, so a video game with VSync enabled will render at 60 FPS. Without VSync being enabled, a video game will generally render as fast as possible, often up to hundreds of FPS.

Without VSync, the GPU and monitor share a memory buffer, referred to as a framebuffer. The GPU writes to the framebuffer at the current rendering framerate, and the monitor reads from the framebuffer at 60 Hz. This method of displaying images onto the monitor has extremely low latency. However, because the GPU is writing to the framebuffer at a different rate than the monitor reads from the framebuffer, the monitor will often read data from two separate GPU frames and display it onto the monitor as a single image, creating an undesirable screen artifact known as screen tearing (see Figure 1). Moreover, because the time between rendered frames is different from the time between monitor screen refreshes, the video game will look jittery or appear to stutter in motion [Bailey09] [Wilson09a].

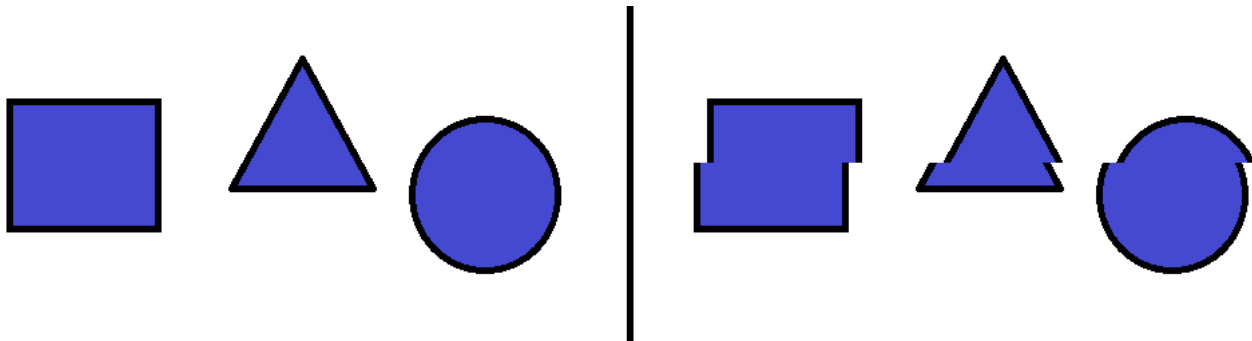


Figure 1: An illustration of screen tearing. Normal on the left, tearing on the right.

The most common form of VSync is double-buffering, in which a second framebuffer is used. The GPU writes to a dedicated "back-buffer" and the monitor reads from a dedicated "front-buffer." Frames from the back-buffer are only transferred to the front-buffer when the drawn frame is complete, and only when the monitor is ready to draw the next frame. This technique will guarantee fluid motion and even frame timing, but will limit the game's FPS to 60. More significantly, VSync increases the time needed to draw a frame onto a monitor, which in turn increases input latency [Wilson09a].

A modification to double-buffered VSync is adding an additional backbuffer for the GPU. This modification is often referred to as "triple-buffering." Normally, when a frame takes longer than one screen refresh to render, the monitor has no choice but to wait for that frame to be rendered before displaying it, which causes an uncomfortable stuttering. With triple-buffering, instead of waiting for a new frame from the GPU, the monitor simply loads the frame from the second backbuffer instead. Thus, this triple-buffering implementation improves image fluidity at lower framerates [Wilson09a].

Normally, PC video games have full control of the GPU when they are in "fullscreen." If a game is "windowed," then the game goes through the Windows Desktop Window Manager (DWM), which in turn controls the GPU. The DWM implements its own buffering scheme, meaning that windowed games will not exhibit screen tearing. However, the DWM will not enforce games to synchronize their framerate with the monitor's refresh rate [Schechter06].

For the remainder of this study, double-buffered VSync will be referred to as "VSync," and the implementation of the additional backbuffer will be referred to as "Triple-Buffering."

2.3 RadeonPro

RadeonPro is a PC tool developed by John Mautari and designed for PC gaming enthusiasts [RadeonPro]. It was originally designed to allow PC gamers to tweak their GPU's drivers and settings. It has since expanded to include many other tweaks, such as graphical filters and overlays. A notable feature of RadeonPro is its ability to act as an intermediary between the video game and the GPU driver. By doing so, it can override the VSync settings of video games, force triple-buffering for games that do not implement it, and enact a frame rate limit for games that would otherwise render frames as fast as the GPU would permit.

RadeonPro was used extensively in this study to override the VSync settings of the games under test. It was also used to enforce a 60 FPS frame rate cap.

2.4 Upcoming VSync techniques

In the past few years, graphics and monitor companies have worked to produce technical standards for allowing a monitor's refresh rate to change dynamically, rather than be a fixed constant. These techniques include NVIDIA Corporation's proprietary GSync [Shimpi13], and the Video Electronics Standards Association (VESA)'s Adaptive-Sync [Hussain14]. By enabling monitors to have a variable refresh rate, monitors can synchronize with the rendering rate of the GPU, rather than the other way around, allowing the rendered frames to be completely displayed as they are ready and thus producing an ideal combination of image fluidity and input latency [Smith14]. As of early 2015, variable refresh rate monitors and the GPUs needed to support them are still targeted for the enthusiast market, and it will be a while before they reach the mainstream market.

3. Performance Evaluation

This section details the components of the measurement study. It consists of the definition of the system and services, the goals and metrics of the study, the parameters and factors relevant to the study, the evaluation technique used, and finally the experimental design.

3.1 System and Services

The system for this study is a computer which takes in input from a keyboard, renders frames with a GPU, and displays frames onto a computer monitor. The system offers the service of being able to run PC video games, and provides the ability to modify the settings for those games.

The specific implementation of the system used in this study is a Lenovo Y510P laptop. The laptop's built-in keyboard is used as an input device, and its built-in screen is used to display the video game. The laptop's NVIDIA GT755M GPU is used to render frames from the video game.

3.2 Goal and Metrics

The goal of this study is to evaluate input latency for video games under various VSync implementations and configurations, and ultimately determine the most ideal configuration to use. Naturally, the metric used in this study is input latency. Because the laptop monitor refreshes at 60 Hz, input latency will be measured in terms of the number of screen refreshes between a keyboard keypress and the screen changing in response. The input latency measurement itself is performed by using the Galaxy S4 smartphone's camera to record the keypresses and screen refreshes at 60 FPS. The number of frames between the keypress to the screen displaying a response will be equivalent to the number of screen refreshes needed for a keypress to be reflected on-screen.

3.3 Parameters and Factors

The following are the parameters that can potentially have an impact on input latency in video games.

Table 1: System Parameters

Parameters	Description
Operating System	Windows 8.1 x64
Processor	Intel i7-4700MQ
GPU	NVIDIA GT755M
Monitor	Built-in/external
Keyboard	Built-in/external

The system parameters consist primarily of the hardware specifications of the system, and are listed in Table 1. A typical PC gamer would not be able to change these parameters in the immediate short-term.

Table 2: Workload Parameters

Parameters	Description
The game being run	Configurable
GPU Driver	Configurable
Graphical settings	Configurable
VSync	Configurable
Triple-Buffering	Configurable
Locking the frame rate	Configurable
Playing in a window or fullscreen	Configurable
Other loads on the PC	Configurable

The workload parameters include the game being run itself, along with a variety of settings that can affect the performance of the game. These parameters can be easily modified by a typical PC gamer. The workload parameters are listed in Table 2.

Table 3: Factors under test

Parameters	Description
VSync	On/Off
Triple-Buffering	On/Off
Locking the frame rate	60 FPS/Unlocked
Playing in a window or fullscreen	Fullscreen/Window

The factors which were selected for this study are listed in Table 3. These parameters deal with how the video game interacts with the monitor, and thus are most likely to have a direct impact on input latency. VSync affects whether the video game's frame rate is synced with the monitor's refresh rate or not, and it inherently gets in the way of rendering a frame to the monitor. Triple-buffering affects the implementation of VSync but has an unknown impact on input latency. Locking the frame rate to 60 FPS using RadeonPro allows the game to match the monitor's refresh rate without actually syncing with the monitor, which may have an impact on input latency. Playing in a window allows the DWM to manage the rendered frames to the GPU, rather than letting the game manage itself.

All other parameters are fixed. The only two configurable system parameters are the choice of keyboard and the choice of monitor, and in both cases we are using the laptop's built in hardware. As for workload parameters, input latency is measured for one game at a time. The GPU driver is the latest driver from NVIDIA, version 347.88. All other loads on the laptop are disabled as much as possible.

3.4 Workloads

In this study, the workloads consist of three video games: *Left 4 Dead 2*, *Super Meat Boy*, and *Sleeping Dogs*. Each of these games uses a distinct rendering engine, and thus their own implementations of VSync. RadeonPro was used to override VSync settings and enforce a frame rate limit of 60 FPS as necessary for the study.

For each game, input latency was measured in the game's main menu. This is because menu selections change instantaneously and are easy to measure, while in-game actions are animated and much more difficult to measure.

3.5 Evaluation Technique

Table 4: Factor Symbol and Levels

Symbol	Parameters	Level -1	Level 1
A	VSync	Off	On
B	Triple-Buffering	Off	On
C	Locking the frame rate	Unlocked	60 FPS
D	Playing in a window or fullscreen	Fullscreen	Window

The evaluation technique used in this study is a $2^k \times r$ design; specifically, a $2^4 \times 30$ design. The four factors and their levels are described in Table 4.

The largest source of variability in this study is the uncertainty of when the keyboard keypress was actuated. Thus, a large number of replications is used to ensure that the measurements are as significant as possible.

For each combination of factors, a single video was recorded of the keyboard and of the screen. Each video consists of 30 keypresses, representing the 30 replications of the experiment design.

This design is performed three times, once for each game in the study. The games vary in their VSync implementations, meaning that a single study cannot encompass all games at once and that individual studies were required for each game.

Left 4 Dead 2 does not allow Triple Buffering to be enabled without VSync, and it ignores the VSync setting when playing in a window. Experiments that would have involved no VSync and Triple Buffering were treated as no VSync without triple-buffering, as triple-buffering by design is not expected to have an impact without VSync. Experiments that would have involved VSync and Windowed were treated as no VSync and Windowed, since they were considered identical for this particular game. 8 videos were recorded for this game.

Super Meat Boy defaults to VSync on and does not offer Triple Buffering, nor does it offer the option to disable VSync. For this particular game, VSync settings were managed by RadeonPro. 16 videos were recorded for this game.

Sleeping Dogs appears to enforce Triple Buffering at all times, regardless of VSync. Thus, for that game Triple Buffering was dropped as a factor, and the study was treated as a $2^3 \times 30$ design. 8 videos were recorded for this game.

4. Results and Analysis

This section describes the results of the experiment for each of the three games. For each experiment, the results are analyzed and interpreted, leading up to a few concluding remarks for the game. The full collection of data used in this analysis can be found at [[Tang15](#)].

4.1 Left 4 Dead 2

Table 5: Average latency data for *Left 4 Dead 2*

A	B	C	D	Latency
-1	-1	-1	-1	1.333333
-1	1	-1	-1	1.333333
-1	-1	1	-1	2.833333
-1	1	1	-1	2.833333
-1	-1	-1	1	3
-1	1	-1	1	3
1	-1	-1	1	3
1	1	-1	1	3
-1	-1	1	1	4
-1	1	1	1	4
1	-1	1	1	4
1	1	1	1	4
1	-1	1	-1	4.066667
1	1	1	-1	4.4
1	-1	-1	-1	7.3
1	1	-1	-1	7.5

For *Left 4 Dead 2*, we recorded 8 videos, with each video consisting of 30 keypresses. For each video, we counted the number of frames between the keypress and the screen changing to reflect the keypress, then averaged the count to obtain a final average input latency in terms of frames.

The average latencies across all 30 keypresses for each experiment are shown in Table 5. The bolded measurements represent actual measurements, and the italicized measurements are those which were not measured due to configuration limitations and were derived from another real measurement.

A quick visual analysis shows that enabling VSync in fullscreen without locking the frame rate has the most input latency, as much as 7.5 frames. Disabling VSync in fullscreen without locking the frame rate has the least input latency, with only 1.333 frames of latency. In general, turning off VSync results in lower input latency, and turning on VSync results in higher input latency.

Table 6: Effects and % variation explained for *Left 4 Dead 2*

	Effect	95% CI for effects		% Variation Explained
A	0.933333	0.89931	0.967357	28.8826
B	0.033333	-0.00069	0.067357	0.03684
C	0.041667	0.007643	0.07569	0.057563
D	-0.225	-0.25902	-0.19098	1.678525
AB	0.033333	-0.00069	0.067357	0.03684
AC	-0.583333	-0.61736	-0.54931	11.28226
AD	-0.933333	-0.96736	-0.89931	28.8826
BC	0.008333	-0.02569	0.042357	0.002303
BD	-0.033333	-0.06736	0.00069	0.03684
CD	0.458333	0.42431	0.492357	6.965071
ABC	0.008333	-0.02569	0.042357	0.002303
ABD	-0.033333	-0.06736	0.00069	0.03684
ACD	0.583333	0.54931	0.617357	11.28226
BCD	-0.008333	-0.04236	0.02569	0.002303
ABCD	-0.008333	-0.04236	0.02569	0.002303
Error				10.81255

The table of effects, percent variation explained by effects, and 95% confidence intervals for effects is shown in Table 6. What's notable is that all effects involving triple buffering are not 95% significantly different from zero; in other words, Triple Buffering has no significant effect on latency.

Of the effects that are significantly different from zero, not all of them are important, in that they do not explain much of the variation in input latency across measurements. The most important effects are VSync and the interaction between VSync and Windowed, both of which explained 28.9% of the variation. This makes sense given the data; enabling VSync caused the greatest increase in input latency in fullscreen, but VSync+Windowed was treated identically to No VSync+Windowed.

The next statistically significant effects are the interaction of VSync with locking the frame rate, and the interaction of VSync with locking the frame rate and windowed, both of which explained 11.28% of the variation. Again, this makes sense; not locking the frame rate with VSync caused 7.3 frames of input latency, while locking the frame rate dropped the latency to 4.067 frames. The latency differences are much smaller when playing the game windowed.

10% of the variation is explained by error in the measurements. Given the variability of attempting to measure keyboard presses with a camera, this is not surprising, if unfortunate.

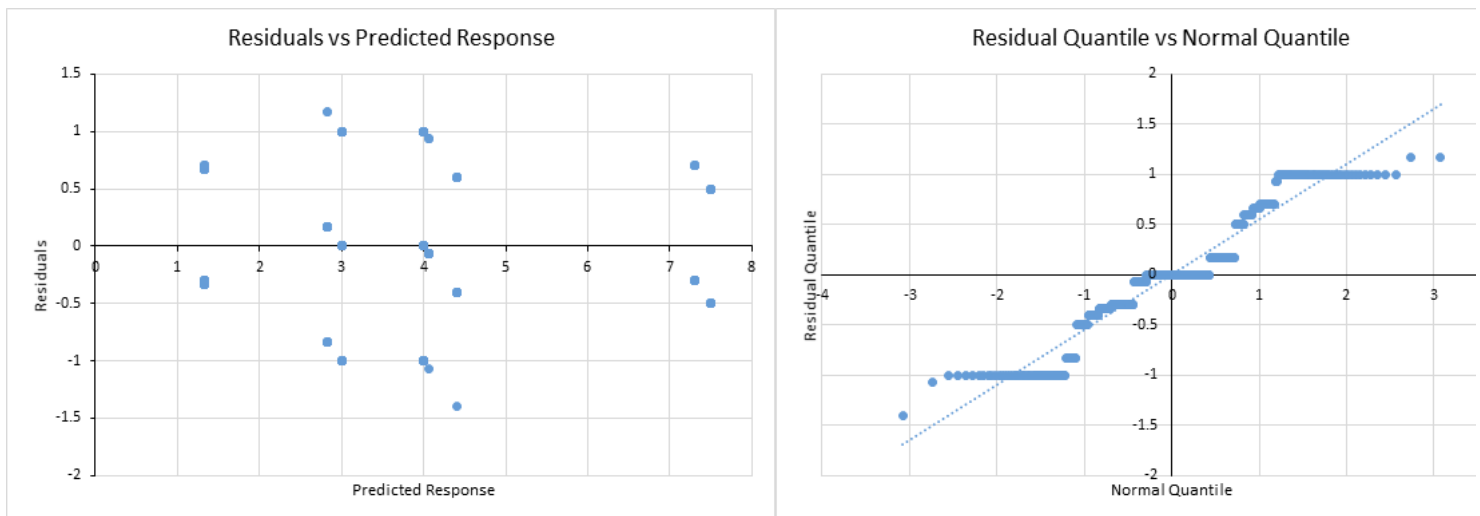


Figure 2: Plots for visual tests for *Left 4 Dead 2*

Fortunately, the errors appear to be normally distributed and independent, with a constant standard deviation, based on the plots in Figure 2. The errors appear to be normally distributed, and there does not appear to be a significant trend in errors based on the predicted response.

Of the possible configurations for *Left 4 Dead 2*, the configuration that gives the least input latency is no VSync + no frame rate lock + fullscreen, followed by no VSync + 60 FPS lock + fullscreen. Of the configurations that have VSync enabled, the configuration that gives the least input latency is VSync + no triple buffering + 60 FPS lock + fullscreen.

4.2 Super Meat Boy

Table 7: Average latency data for *Super Meat Boy*

A	B	C	D	Latency
-1	1	-1	-1	2.033333
-1	-1	-1	-1	2.1
-1	-1	1	-1	2.9
-1	1	1	-1	3.233333
-1	-1	-1	1	3.5
-1	1	1	1	3.533333
-1	-1	1	1	3.733333
1	1	1	-1	3.766667
1	-1	-1	1	4
1	-1	1	-1	4
1	-1	1	1	4.233333
-1	1	-1	1	4.4
1	1	1	1	4.466667
1	1	-1	1	4.566667
1	1	-1	-1	7.266667
1	-1	-1	-1	7.6

Super Meat Boy does not have a configurable VSync option, so all VSync options were enforced using RadeonPro. 16 videos were recorded, one for each configuration and 30 keypresses per video. The average latencies for the experiments are shown in Table 7. Once again, VSync off appears to decrease input latency in general, and VSync on appears to increase input latency in general.

Table 8: Effects and % variation explained for *Super Meat Boy*

	Effect	95% CI for effects		% Variation Explained
A	0.904167	0.871944	0.936389	33.69276
B	0.075	0.042778	0.107222	0.231826
C	-0.35	-0.38222	-0.31778	5.048655
D	-0.02917	-0.06139	0.003056	0.03506
AB	-0.04583	-0.07806	-0.01361	0.086577
AC	-0.52083	-0.55306	-0.48861	11.17988
AD	-0.64167	-0.67389	-0.60944	16.96909
BC	-0.05833	-0.09056	-0.02611	0.14024
BD	0.1125	0.080278	0.144722	0.521608
CD	0.2875	0.255278	0.319722	3.406554
ABC	0.029167	-0.00306	0.061389	0.03506
ABD	0.058333	0.026111	0.090556	0.14024
ACD	0.616667	0.584444	0.648889	15.67258
BCD	-0.12083	-0.15306	-0.08861	0.601746
ABCD	0.066667	0.034444	0.098889	0.183171
Error				12.05495

The effects and percent variation explained by each effect are shown in Table 8. Nearly all effects were significantly different from zero, but only a few of them were important. In the table, the bolded effects represent those which were both significantly different from zero and important. Note that none of the important effects that are significantly different from zero include Triple Buffering.

Of the effects significantly different from zero, the most important of them was VSync, which explained 33.7% of the variation. Given the data and how the higher latencies lean towards VSync on, this is not surprising.

Other notable effects include the interaction between VSync and Windowed, and the interaction between VSync, Windowed, and the frame rate lock, with 16.97% and 15.67% of variation explained respectively. The interaction between VSync and the frame rate lock alone was also important, explaining 11.18% of the variation.

The variation explained by errors is even larger, i.e. 12.05%. Part of this had likely been caused by an outlier from one of the experiments. When VSync and Triple Buffering were on, and frame rate locking and windowed were off, the latency jumped to 10 frames instead of the average of 7.27 frames. This was likely caused by a GPU stall of some sort.

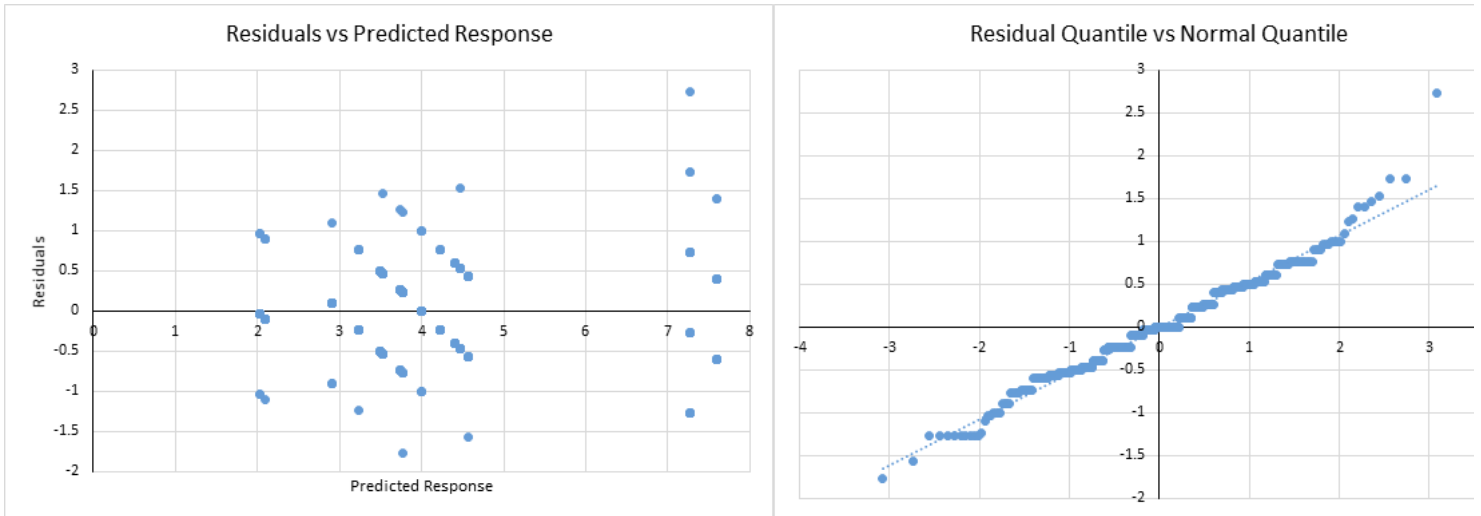


Figure 3: Plots for visual tests for *Super Meat Boy*

Looking at the plots in Figure 3 and disregarding the outlier, the errors appear to be normal. The errors also do not show a trend given the predicted response, indicating independence, and the spread of the errors seems even, indicating a constant standard deviation.

Of the possible configurations for *Super Meat Boy* and ignoring Triple Buffering due to its unimportance, the configuration that gives the least input latency is no VSync + no frame rate lock + fullscreen, followed by no VSync + 60 FPS lock + fullscreen. Of the configurations that have VSync enabled, the configuration that gives the least input latency is VSync + triple-buffering + 60 FPS lock + fullscreen.

4.3 Sleeping Dogs

Table 9: Average latency data for *Sleeping Dogs*

A=vsync	B=lock60	C=window	Latency
-1	-1	-1	2.866667
1	1	-1	3.866667
-1	1	-1	4.4
-1	-1	1	4.833333
-1	1	1	5.566667
1	1	1	5.966667
1	-1	1	7
1	-1	-1	8.466667

Sleeping Dogs was unusual since it enforced Triple Buffering at all times. Thus, it seemed appropriate to perform a 2³*30 design and drop the Triple Buffering factor for this game. 8 videos were recorded, each containing 30 keypresses. The resulting average latency data is shown in Table 9. Once again, VSync off appears to be associated with low input latency, and VSync on appears to be associated with high input latency.

Table 10: Effects and % variation explained for *Sleeping Dogs*

	Effect	95% CI for effects		% Variation Explained
A	0.954167	0.914853	0.99348	30.18031
B	-0.42083	-0.46015	-0.38152	5.870775
C	0.470833	0.43152	0.510147	7.348684
AB	-0.9875	-1.02681	-0.94819	32.32581
AC	-0.3125	-0.35181	-0.27319	3.237242
BC	0.345833	0.30652	0.385147	3.964687
ABC	0.545833	0.50652	0.585147	9.876323
Error				7.196174

The effects and percent variation explained by each effect are shown in Table 10. All of the effects are significantly different from zero, and all of them are important to some extent. VSync once again is one of the most important effects, which is not surprising considering how low latency is associated with no VSync and high latency is associated with VSync. What is surprising is the interaction between VSync and the frame rate lock being even more important. This is likely a result of VSync + 60 FPS lock + fullscreen having the second least input latency of all the configurations and beating most of the non-VSync configurations. Additionally, the percent of variation explained by error is only 7.20% this time around.

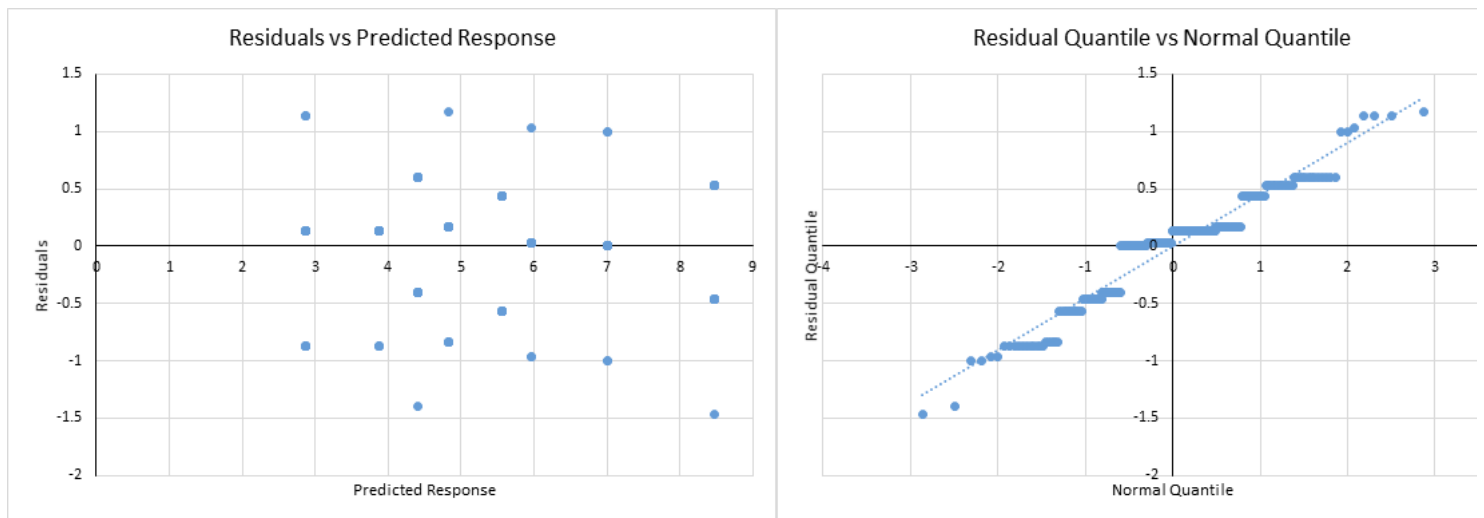


Figure 4: Plots for visual tests for *Sleeping Dogs*

The errors appear to have constant standard deviation and exhibit no trend, indicating independence, and the errors also appear to be normally distributed.

Of the possible configurations for *Sleeping Dogs*, the configuration that gives the least input latency is No VSync + No frame rate lock + Fullscreen. Of the configurations that use VSync, the configuration that gives the least input latency is VSync + 60 FPS lock + Fullscreen.

5. Conclusion

This study was intended to measure the input latency of PC video games in various configurations, and attempt to find a configuration that produced the best balance of low input latency and image fluidity. After this experiment, we feel there is adequate data to come to a conclusion.

This experiment revealed the limitations of using a 60 FPS camera for input latency measurement, which was demonstrated by the relatively high variability of the data points and the large percent of variability explained by error. In the future, we would like to re-attempt this experiment with better measurement equipment, such as a high-speed camera and a programmable input device. Such equipment would not only reduce the need for a large number of replications, it would also allow the precise measurement of minute input latency differences between configurations.

The input latency impact of Triple Buffering for *Super Meat Boy* was found to be unimportant, and in the case of *Left 4 Dead 2*, insignificant. Thus, we believe enabling Triple Buffering is optional, but suggested if the game frequently drops below 60 FPS.

For all three games, the least input latency came from having the game play without VSync, without triple buffering, without frame rate locking, and in fullscreen. This isn't surprising; the game is rendering as fast as it can, and the monitor is able to grab frames as fast as it can. Such a configuration is most suitable for competitive online video games, where having low latency is an advantage, but is not ideal for players who want their games artifact-free.

For two of the three games, the next lowest latency came from locking the frame rate to 60 FPS, while still playing with VSync off and in fullscreen. This configuration eliminates motion judder and partially eliminates tearing. It may be acceptable for some players, but not for all.

The configuration which we believe has the best balance of input latency and image fluidity is VSync on, fullscreen, with frame rate locked at 60 FPS. Such a configuration offers the full benefits of VSync, such as no judder and no screen tearing, while having a very low input latency.

References (in order of importance)

- [Tang15] Tang, Jeremy. "VSync Config Study Data." 11 April 2015, [vs_data.zip](#). All of the data that was collected for this experiment.
- [Wilson09a] Wilson, Derek. "Triple Buffering: Why We Love It." *AnandTech*, 26 June 2009, <http://www.anandtech.com/show/2794>. An article explaining Vertical Sync and Triple Buffering and its effects.
- [Wilson09b] Wilson, Derek. "Exploring Input Lag Inside and Out." *AnandTech*. 16 July 2009, <http://www.anandtech.com/show/2803>. An article which provides an overview of the sources of input latency.
- [Bailey09] Khan, S.; Bailey, D.; Sen Gupta, G., "Simulation of Triple Buffer Scheme (Comparison with Double Buffering Scheme)," *Computer and Electrical Engineering*, 2009. ICCEE '09. Second International Conference on , vol.2, no., pp.403,407, 28-30 Dec. 2009 doi: 10.1109/ICCEE.2009.226, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5380528>. A paper explaining the differences between double-buffering and triple-buffering in the context of image processing.
- [RadeonPro] "RadeonPro." John Mautari, <http://www.radeonpro.info/>. The RadeonPro tool website.
- [Schechter06] Schechter, Greg. "DWM's use of DirectX, GPUs, and hardware acceleration." 19 Mar 2006, http://blogs.msdn.com/b/greg_schechter/archive/2006/03/19/555087.aspx. A Microsoft blog post explaining how the Desktop Window Manager renders objects to the screen.
- [Shimpi13] Shimpi, Anand Lal. "NVIDIA's G-Sync: Attempting to Revolutionize Gaming via Smoothness." *AnandTech*. 18 October 2013, <http://www.anandtech.com/show/7436>. An article which introduces NVIDIA's GSync.
- [Smith14] Smith, Ryan. "VESA Adds Adaptive-Sync to DisplayPort 1.2a Standard; Variable Refresh Monitors Move Forward." *AnandTech*. 12 May 2014. <http://www.anandtech.com/show/8008>. An article which introduces Adaptive VSync in the context of computer monitors, and compares it to GSync.
- [Hussain14] Hussain, Syed Athar. "VESA DisplayPort Adaptive-Sync." VESA. Mar. 2014, <http://www.vesa.org/wp-content/uploads/2014/07/VESA-Adaptive-Sync-Whitepaper-140620.pdf>. A white paper explaining the details behind Adaptive VSync.
- [MerriamWebster] "Video-game." <http://www.merriam-webster.com/dictionary/video%20game>. The dictionary definition of "video game."

List of Acronyms (in alphabetical order)

- DWM: Desktop Window Manager
- FPS: Frames per second

- GPU: Graphics Processing Unit
 - PC - Personal Computer
 - VESA: Video Electronics Standards Association
 - VSync - Vertical Synchronization
-

Last Modified: May 1, 2015

This and other papers on performance analysis of computer systems are available online at <http://www.cse.wustl.edu/~jain/cse567-15/index.html>

[Back to Raj Jain's Home Page](#)