Vol. 6, No. 18, Apr-Jun 2026 (114-134)
Journal of New Achievements in Electrical, Computer and Technology (JNAECT)
مجله دستاوردهای نوین در برق،کامپیوتر و فناوری
دوره ۶، شماره ۱۸، بهار ۱۴۰۵ (۱۳۴-۱۱۴)
journal homepage: https://jnaect.ir

بررسی همگام‌سازی عمودی (Vertical Synchronization یا به اختصار V-Sync) بر عملکرد و کیفیت بصری بازی‌های ساخته شده با انجین یونیتی (Unity)

امیرحسین اردلان*

استاد بازیسازی کالج بین المللی ابن سینا، تهران، ایران. تفلیس، گرجستان. (shsbbhorse@chmail.ir)

چکیده

همگام‌سازی عمودی (Vertical Synchronizationیا به اختصار V-Sync) یکی از تکنیک‌های اساسی در توسعه بازی‌های رایانه‌ای است. این تکنیک، با همگام کردن سرعت تازه‌سازی مانیتور با سرعت فریم خروجی بازی، پدیده شکستگی صفحه (Screen Tearing) را کاهش می‌دهد. همگام‌سازی عمودی با وجود مزایای بصری، می‌تواند تأثیرات متفاوتی بر عملکرد بازی داشته باشد؛ به‌ویژه زمانی که تعداد فریم بازی با سرعت تازه‌سازی مانیتور هم‌خوانی ندارد؛ ممکن است منجر به افزایش تأخیر در ورودی‌ها (Input Lag) یا کاهش تعداد فریم شود. در این پژوهش، ابتدا اصول فنی و مکانیزم عملکرد V-sync به صورت جامع شرح داده می‌شود. سپس به بررسی نحوه پیاده‌سازی این تکنیک در موتور بازی‌سازی یونیتی Unity با استفاده از زبان برنامه‌نویسی سی-شارپ پرداخته می‌شود. کدهای نمونه‌ای ارائه می‌گردند که نشان می‌دهند چگونه می‌توان همگام‌سازی را در پروژه‌های یونیتی فعال یا غیرفعال و ویژگی‌های آن‌را تنظیم نمود. علاوه بر این، با ارائه تصاویر عملی و تحلیل نتایج آزمایش‌های انجام شده؛ تأثیر این ابزار بر کیفیت بصری و عملکرد بازی به صورت ملموس نمایش داده می‌شود. در پایان، به نقد و تحلیل تصاویر اغراق‌آمیز موجود در منابع مختلف پرداخته می‌شود. هدف این پژوهش درک دقیق‌تری از مزایا و محدودیت‌های همگام‌سازی عمودی است.

واژه‌های کلیدی: بازیسازی، یونیتی، unity engine، مانیتور، v-sync، Vertical Synchronization، همگام‌سازی عمودی، سی شارپ، بازی‌های کامپیوتری، صفحه نمایشگر، monitor

۱. مقدمه

در دهه‌های اخیر، صنعت بازی‌های رایانه‌ای رشد چشمگیری داشته و به یکی از مهم‌ترین حوزه‌های سرگرمی و فناوری تبدیل شده است. با پیشرفت سخت‌افزارها و افزایش انتظارات کاربران، کیفیت بصری و عملکرد بازی‌ها به عوامل کلیدی در تجربه کاربر تبدیل شده‌اند[2]. یکی از چالش‌های اساسی در توسعه بازی‌ها، همگام‌سازی صحیح بین سرعت فریم تولید شده توسط بازی و سرعت تازه‌سازی مانیتور است. عدم همگام‌سازی صحیح این دو، می‌تواند منجر به پدیده‌ای به نام شکستگی صفحه (screen tearing) شود که در آن، تصویر نمایش داده شده به صورت ناهموار و شکسته دیده می‌شود و تجربه بصری کاربر را مختل می‌کند.

برای حل این مشکل، راه‌حل‌های مختلفی معرفی شده است که یکی از رایج‌ترین آن‌ها، همگام‌سازی عمودی (v-sync) است. این مکانیزم در کتاب Real-Time Rendering به تفصیل بررسی شده است[3]. **این** روش به متداول‌ترین روش تبدیل شده

است و تقریبا اکثر بازی‌ها از این ابزار استفاده می‌کنند. این روش با همگام کردن تعداد فریم بازی (خروجی کارت گرافیک) با سرعت تازه‌سازی مانیتور، از بروز شکستگی صفحه جلوگیری می‌کند. بدین صورت، تصاویر یکپارچه و روانی را ارائه می‌دهد. با این حال، استفاده از همگام‌سازی عمودی می‌تواند تأثیرات متفاوتی بر عملکرد بازی داشته باشد؛ به ویژه در مواردی که تعداد فریم بازی با سرعت تازه‌سازی مانیتور همخوانی نداشته باشد. در چنین شرایطی، ممکن است تأخیر ورودی (input lag) افزایش یابد یا سرعت بازی کاهش پیدا کند. هر دو می‌توانند تجربه کاربر را تحت تأثیر قرار دهند[4].

موتور بازی‌سازی یونیتی به عنوان یکی از محبوب‌ترین ابزارهای توسعه بازی، امکانات مختلفی برای پیاده‌سازی و تنظیم همگام‌سازی عمودی (v-sync) فراهم کرده است. این نرم افزار قدرتمند از سال ۲۰۰۵ عرضه شده و یکی از موتورهای اصلی بازیسازی در جهان به حساب می‌آید[۱]. با استفاده از زبان برنامه‌نویسی سی شارپ، بازیسازان می‌توانند به‌راحتی این تکنیک را در پروژه‌های خود فعال یا غیرفعال کنند و ویژگی‌های مربوطه را تنظیم نمایند. با این حال، بسیاری از توسعه‌دهندگان با چالش‌هایی در درک دقیق مکانیزم همگام‌سازی عمودی و تأثیرات آن بر عملکرد بازی مواجه هستند.

در این پژوهش، ابتدا اصول فنی و مکانیزم عملکرد همگام‌سازی عمودی به صورت جامع شرح داده می‌شود. سپس، به بررسی نحوه پیاده‌سازی این تکنیک در موتور یونیتی پرداخته می‌شود و کدهای نمونه‌ای ارائه می‌گردند که نشان می‌دهند چگونه می‌توان آن را در پروژه‌های یونیتی فعال یا غیرفعال کرد. علاوه بر این، تصاویری بازی‌ها ارائه شده است تا تأثیر این تکنیک بر کیفیت بصری بازی به صورت ملموس نمایش داده شود. در پایان، به نقد و تحلیل تصاویر اغراق‌آمیز موجود در منابع مختلف پرداخته می‌شود تا درک دقیق‌تری از مزایا و محدودیت‌های این روش ارائه گردد.

این پژوهش می‌تواند به عنوان راهنمایی عملی برای توسعه‌دهندگان بازی و اساتید آن، مورد استفاده قرار گیرد تا با آگاهی بیشتری از همگام‌سازی عمودی در پروژه‌های خود بهره ببرند و تعادل مناسبی بین کیفیت بصری و عملکرد بازی برقرار کنند.

۲. شرح مشکل شکستگی تصویر

یکی از چالش‌های اساسی در نمایش تصاویر متحرک، عدم همگام‌سازی بین تعداد تولید فریم توسط کارت گرافیک و نمایش آن فریم‌ها در مانیتور است. این عدم همگام‌سازی منجر به پدیده‌ای به نام شکستگی صفحه (Screen Tearing) می‌شود[3]. برای درک بهتر این مشکل، ابتدا باید با مفاهیم پایه‌ای مانند تعداد فریم بر ثانیه و سرعت بروزرسانی مانیتور آشنا شویم.

۱.۲. تعداد فریم بر ثانیه و سرعت تازه‌سازی مانیتور

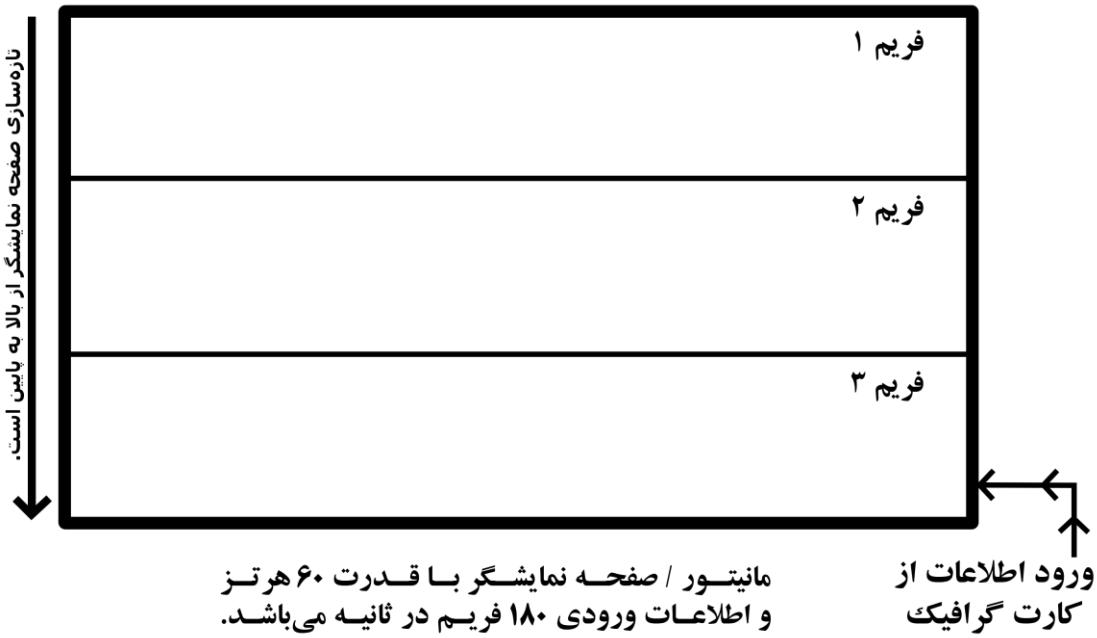
فریم بر ثانیه[۱]: به صورت فریم/ثانیه[۲] نیز نوشته می‌شود. به تعداد فریم‌هایی که کارت گرافیک در هر ثانیه تولید می‌کند؛ گفته می‌شود. برای مثال، اگر بازی با ۶۰ فریم در ثانیه اجرا شود یعنی در هر ثانیه ۶۰ تصویر با سرعت یکسان نمایش داده شده است. این حالت، حس توهم به حرکت نیز در انسان ایجاد می‌کند. در حقیقت تعدادی تصویر ثابت اما با سرعت یکسان نمایش داده می‌شوند و به همین دلیل حس حرکت و روان بودن را دارد. در یک فیلم یا پرده سینما، تعداد فریم‌ها بر ثانیه ثابت و از قبل تعریف شده‌اند اما در بازی‌ها چنین نیست. تعداد فریم‌ها در بازی متغیر است و کم و زیاد می‌شود. در صحنه‌های سبک‌تر، تعداد فریم‌ها بیشتر می‌شود زیرا کار برای کارت گرافیک سبک‌تر است. بدین ترتیب تعداد فریم‌ها زیادتر می‌شود و دیگر یک عدد ثابت نیست. ساخت فریم‌ها و ارسال آنها به مانیتور معمولا توسط کارت گرافیک (GPU) انجام می‌شود. البته ممکن است در برخی موارد توسط پردازنده (CPU) نیز انجام شود. به عنوان مثال، یک کامپیوتر می‌تواند بدون کارت گرافیک نیز روشن شود و کار کند. در این موارد، مسئولیت ساخت فریم‌ها بر عهده پردازنده است.

سرعت بروزرسانی[۳] مانیتور: تعداد دفعاتی که مانیتور در هر ثانیه می‌تواند بروز شود را سرعت بروزرسانی مانیتور می‌گویند. این مقدار معمولاً با واحد هرتز (HZ) نشان داده می‌شود. برای مثال، مانیتور ۶۰ هرتز، ۶۰ بار در ثانیه تصویر را تازه‌سازی می‌کند. به این دلیل از یکای هرتز استفاده می‌کنند که این‌کار به تناوب جریان برق درون مانیتور اشاره دارد و یکای آن هرتز است.

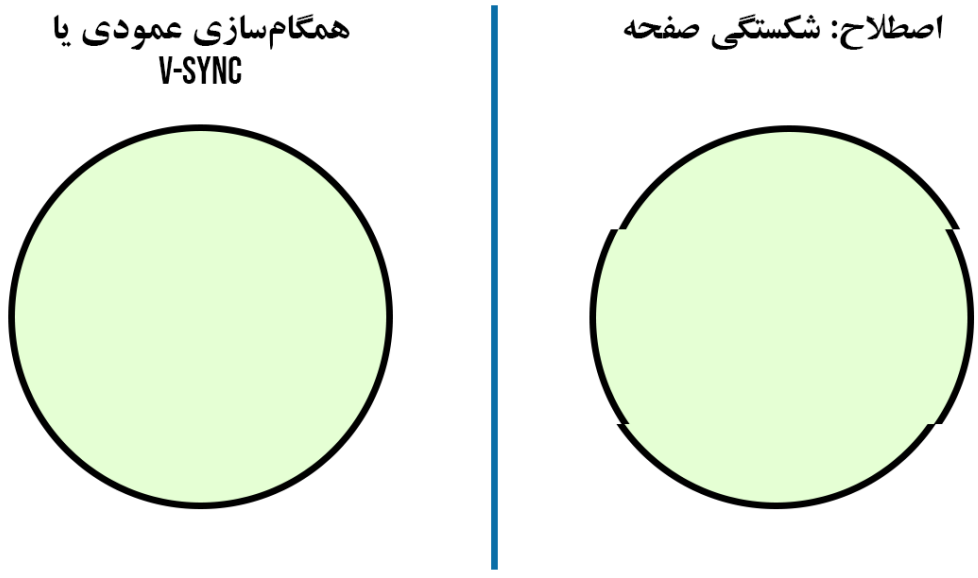
۲.۲. سازوکار بروز شکستگی صفحه

وقتی کارت گرافیک با سرعت بالایی فریم تولید کند و مانیتور قادر به پخش همه فریم‌ها نباشد؛ شکستگی صفحه ایجاد می‌شود. به عنوان مثال، کارت گرافیک ۱۸۰ فریم بر ثانیه تولید کند و از طرفی مانیتور ۶۰ هرتز باشد. یعنی فقط قادر به نمایش ۶۰ فریم در ثانیه می‌باشد. مانیتور نمی‌تواند همه فریم‌های تولید شده را نمایش دهد و در نتیجه در هر نمایش، بخشی از یک فریم اصلی و بخش‌هایی از فریم‌های بعدی را نمایش می‌دهد. این موضوع باعث می‌شود که تصویر به صورت ناهموار و شکسته دیده شود. انگار که صفحه از چندجا شکسته شده باشد و به همین دلیل به آن شکستگی صفحه می‌گویند. برای درک بهتر مثالی عرض می‌شود:

کارت گرافیک در یک ثانیه ۱۸۰ فریم تولید می‌کند. مانیتور فقط ۶۰ بار در ثانیه، تصویر را بروز می‌سازد. در هر نمایش، مانیتور فقط می‌تواند یک فریم را نمایش دهد اما از طرفی کارت گرافیک در همان زمان، سه فریم تولید کرده است. تصاویر مانیتور از بالا به پایین بروز می‌شود و فریم‌های جدید را نمایش می‌دهد. این امر براساس دستور تغییر رنگ پیکسل‌ها که از کارت گرافیک صادر شده؛ رخ می‌دهد. در نتیجه، مانیتور بخشی از فریم اول را در بالا و فریم دوم را در وسط و فریم سوم را در بخش پایینی صفحه نمایش می‌دهد. این عدم همگام‌سازی، باعث می‌شود که تصویر به صورت پاره‌پاره و شکسته دیده شود. به همین علت به آن شکستگی صفحه نیز می‌گویند. این مورد در شکل ۱ و ۲ به تصویر کشیده شده است.



شکل ۱ – مفهوم شکستگی تصویر در صفحه نمایشگر



شکل ۲ – نمایش مقایسه شکستگی صفحه و همگام‌سازی عمودی

۳.۲. تأثیر شکستگی صفحه بر تجربه کاربر

شکستگی صفحه، کیفیت بصری بازی را کاهش می‌دهد. همچنین می‌تواند باعث خستگی چشم و کاهش تمرکز کاربر شود زیرا چند فریم را در یک تصویر می‌بینند. این مشکل به ویژه در بازی‌های سریع و پویا که نیاز به واکنش سریع کاربر دارند؛ بیشتر به چشم می‌آید. البته یکی از شاخصه‌های بازی‌های قدیمی همین شکستگی تصاویر است. یک بازی جدید لاک‌پشت‌های نینجا اما با کیفیت قدیمی ,و پیکسلی به‌نام Teenage Mutant Ninja Turtles: Shredder’s revenge - 2022 عرضه شد که

برخی تصاویر به‌عمد شکستگی دارد تا یادآور دوران قدیم باشد. هم‌چنان در برخی موارد برای نمایش حالت قدیمی یا نمایش حالت‌های بسیار سریع، به عمد از این روش استفاده می‌شود.

۳. تاریخچه همگام‌سازی عمودی

برای درک بهتر این ابزار، لازم است نگاهی به تاریخچه و دلایل پدید آمدن آن بیندازیم.

۱.۳. ظهور شکستگی صفحه و نیاز به راه‌حل

در دهه‌های اولیه ظهور بازی‌های رایانه‌ای، سخت‌افزارهای گرافیکی و مانیتورها هنوز به بلوغ امروزی نرسیده بودند. با افزایش قدرت پردازشی کارت‌های گرافیک، تولید فریم‌ها با سرعت بیشتری انجام می‌شد اما مانیتورها قادر به نمایش همه آن‌ها نبودند. این عدم تطابق، منجر به بروز پدیده شکستگی صفحه شد. در آن زمان، توسعه‌دهندگان و تولیدکنندگان سخت‌افزار به دنبال راه‌حلی بودند تا تجربه بصری کاربر را بهبود بخشند. از این رو با تلاش دانشمندان و تولیدکنندگان و همکاری بین شرکت‌های مختلف، روشی ابداع شد که به آن همگام‌سازی عمودی می‌گفتند. یعنی به صورت عمودی تنها یک فریم نمایش داده شود. همچنین در شکل ۲و۳ شکستگی صفحه شرح و نمایش داده شده است.

شکل ۳ – مقایسه شکستگی تصویر در یک بازی قدیمی با و بدون V-Sync سمت راست همگام‌سازی دارد اما تصویر سمت چپ همگام‌سازی عمودی ندارد و به‌اصطلاح شکستگی صفحه اتفاق افتاده است. [5]

۲.۳. معرفی همگام‌سازی عمودی

در اوایل دهه ۱۹۹۰، با گسترش بازی‌های سه‌بعدی و افزایش سرعت کارت گرافیک‌ها، همگام‌سازی عمودی یک نیاز برای توسعه بازی‌ها و افزایش کیفیت تصاویر آنها بود. در آن دوره، شرکت‌های تولیدکننده کارت گرافیک مانند NVIDIA و AMD تکنیکی به نام همگام‌سازی عمودی یا همان V-Sync را معرفی کردند. این تکنیک با همگام کردن کارت گرافیک با سرعت مانیتور، مشکل شکستگی صفحه را حل نمود.

مکانیزم اولیه: در این روش، کارت گرافیک مجبور می‌شد تا تولید فریم جدید را با زمان‌بندی بروزرسانی مانیتور همگام کند. بدین ترتیب، مانیتور فقط فریم‌های کامل را نمایش می‌داد و دیگر بخشی از چند فریم مختلف را در یک قاب قرار نمی‌داد.

۳.۳. تکامل و بهبود V-Sync

با پیشرفت تکنولوژی، همگام‌سازی عمودی نیز بهبود یافت. در ابتدا، این تکنیک تنها در بازی‌های کامپیوتری استفاده می‌شد اما به مرور زمان در کنسول‌های بازی و گوشی‌هایی همراه نیز پیاده‌سازی شد. با این حال، استفاده از V-Sync همیشه بدون مشکل نیست. یکی از معایب اصلی، افزایش تأخیر ورودی است زیرا کارت گرافیک مجبور است تا منتظر مانیتور بماند و آماده نمایش فریم جدید شود. این موضوع باعث کاهش سرعت نمایش فریم‌ها می‌شود. بدین صورت که با زدن یک کلید، تأخیری هرچند کوتاه، به‌وجود می‌آید.

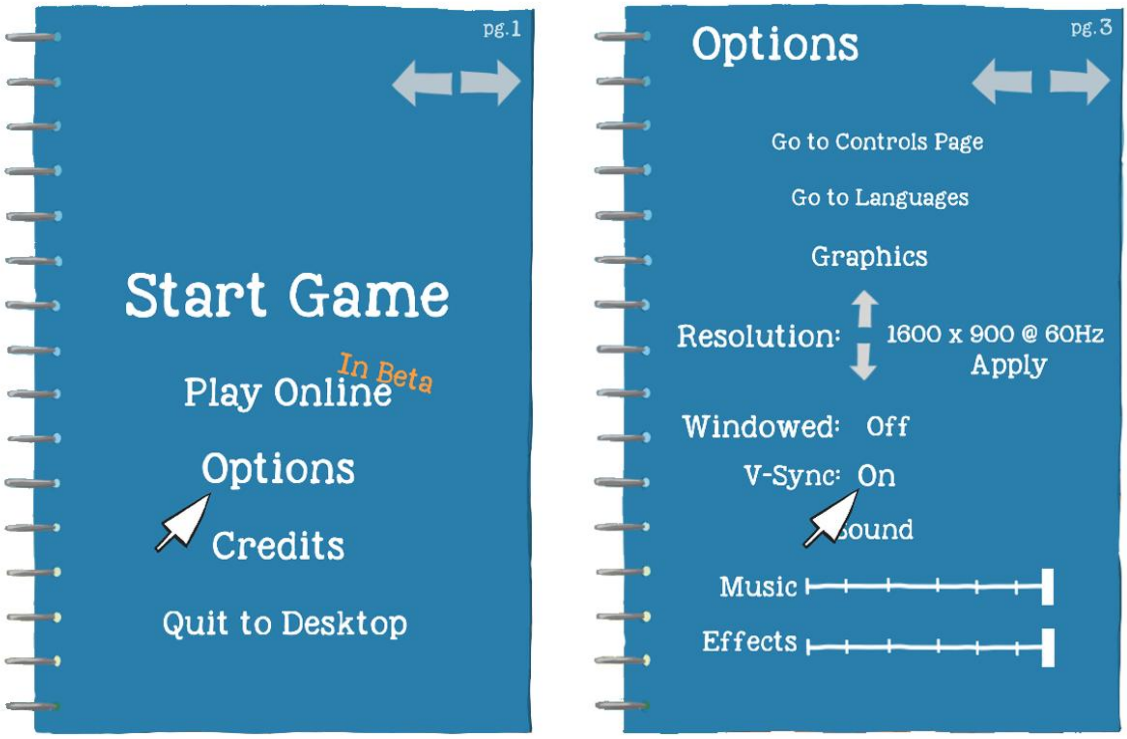
مثال: در بازی‌های سریع مانند تیراندازی اول شخص، تأخیر ورودی می‌تواند تجربه کاربر را تحت تأثیر قرار دهد. با زدن کلید شلیک، کاربر انتظار شلیک را از شخصیت درون بازی دارد. این انتظار باید بدون وقفه برآورده شود زیرا ممکن است نتیجه بازی را تغییر دهد. با زدن کلید مورد نظر، شخصیت کمی (در حدود ۳۰ میلی ثانیه) دیرتر تیر را شلیک می‌کند و همین باعث یک تجربه ناخوشایند می‌شود. به همین دلیل و در سال‌های بعد، تکنیک‌های جایگزین مانند Adaptive V-Sync و G-Sync توسط شرکت‌های مختلف معرفی شدند تا این مشکل را نیز کاهش دهند. البته امروزه معمولا از V-Sync استفاده می‌شود.

۴.۳. کاربردهای امروزی

امروزه، همگام‌سازی عمودی همچنان یکی از پرکاربردترین تکنیک‌ها در صنعت بازی‌سازی است. هرچند کمی تأخیر ورودی دارد اما امروزه با تقویت مانیتورها، این تأخیر بسیار کمتر و تقریبا ناچیز شده است. موتورهای بازی‌سازی مانند یونیتی، امکان فعال یا غیرفعال کردن V-Sync را به توسعه‌دهندگان می‌دهند. همچنین، با ظهور مانیتورهای با سرعت بروزرسانی بالا (مانند ۱۴۴ هرتز و ۲۴۰ هرتز)، شکستگی کمتر احساس می‌شود. همچنان در بسیاری از بازی‌ها، به عنوان یک گزینه استاندارد در نظر گرفته می‌شود.

۴. پیاده‌سازی v-sync در یونیتی

به صورت پیش‌فرض این قابلیت در یونیتی فعال است. یعنی بازی‌های ساخته شده؛ همگام‌سازی عمودی دارند اما گاها لازم است تا آنرا خاموش نموده و یا تغییر آنرا به مخاطب بسپاریم. به عنوان مثال، یک بازی می‌تواند این اجازه را به کاربران بدهد تا خودشان تصمیم بگیرند که آیا همگام‌سازی روشن باشد یا خاموش. برخی از کاربران نیز گاه به عمد به یکی از این قابلیت‌ها نیاز دارند تا فریم‌های بازی را به دلخواه خود تغییر دهند. به عنوان مثال پایین آوردن تعداد فریم ملزوم به خاموش کردن کردن همگام‌ساز عمودی است. به این تنظیمات که در بازی قرار می‌گیرد تا کاربران خودشان به صورت دلخواه آنها را مدیریت کنند؛ سفارشی‌سازی تنظیمات می‌گویند. بازیسازها، باید این قابلیت را در تنظیمات قرار دهند تا کاربران بتوانند آنها را تغییر دهند. به عنوان مثال در شکل ۴و۵ تنظیمات دو بازی نمایش داده شده است که کاربران می‌توانند همگام‌سازی را تغییر دهند.



شکل ۴ – روشن و خاموش کردن حالت v-sync معمولا در بخش تنظیمات و قابلیت‌های بازی قرار دارد.

عکس از بازی Ultimate Chicken Horse که این بازی با انجین یونیتی ساخته شده است.

شکل ۵ – تغییر همگام‌ساز عمودی در بازی KNIGHTS OF BRAVELAND که با یونیتی ساخته شده است.

۱.۴. مدیریت همگام‌ساز

در موتور بازی‌سازی یونیتی، می‌توان v-sync را به دو روش اصلی تنظیم نمود. روش اول صرفا برنامه‌نویسی است. این روش در هر صورت اعمال خواهد شد. روش دوم، از بخش تنظیمات کیفیت بازی است که معمولا بخش بصری بازی را مدیریت می‌کند. برنامه‌نویسی بر تنظیمات ارجهیت دارد و ابتدا آن اجرا خواهد شد. تنظیمات بصری، به صورت کلی اعمال می‌شود و جزئیات را می‌توان با برنامه‌نویسی تغییر داد.

۱.۱.۴. برنامه‌نویسی

ابتدا باید یک برنامه ایجاد نمود. سپس می‌توان فعال/غیرفعال شدن همگام‌سازی را برنامه‌نویسی کرد. با استفاده از رابطه ۱ می‌توان عدد مورد نظر را نسبت داد. این اعداد تنظیمات مربوط به v-sync را تعیین می‌کنند.

```
1. QualitySettings.vSyncCount = عدد مورد نظر;
2. QualitySettings.vSyncCount = 0;
3. QualitySettings.vSyncCount = 1;
4. QualitySettings.vSyncCount = 2;                    (1)
```

در خط ۱ از رابطه ۱ ابتدا اصل کد نوشته شده است. این کد از متد تنظیمات کیفیت و زیرمجموعه همگام‌سازی استفاده می‌کند. عددی که در ادامه ذکر می‌شود می‌تواند صفر یا بیشتر باشد که در خط‌های ۲-۴ نمایش داده شده‌اند. اگر عدد برابر با صفر باشد یعنی همگام‌سازی خاموش می‌شود و در این حالت شکستگی صفحه داریم. عدد ۱ یعنی همگام‌سازی با سرعت مانیتور فعال است. عدد ۲ یعنی همگام‌سازی با نصف سرعت مانیتور فعال است. اگر عدد ۳ باشد یعنی با یک سوم سرعت مانیتور فعال است و همینطور این عدد می‌تواند بیشتر شود. هرچه بیشتر شود؛ سرعت پخش مانیتور را کمتر می‌کند. این رابطه با عرض یک مثال در رابطه ۲ آورده شده است.

$$\frac{refresh\ monitor\ rate}{VSyncCount} = MAX\ frame\ rate\ \ .\ VSyncCount > 0$$

$$\xrightarrow{example} \quad \frac{120}{2} = MAX\ 60\ fps \qquad (2)$$

رابطه ۲ حداکثر تعداد فریم بر ثانیه را نشان می‌دهد. در مثال زده شده؛ قدرت مانیتور ۱۲۰ هرتز بوده و عدد تنظیم شده در برنامه‌نویسی، برابر با ۲ بوده است. بدین صورت کارت گرافیک تا ۶۰ فریم بر ثانیه را تولید نماید.

عدد اصلی ممکن است با عدد محاسبه شده تفاوت داشته باشد. دلیل این تفاوت، میزان قدرت کارت گرافیک است. شاید یک مانیتور بسیار قوی و حتی ۵۰۰ هرتز باشد. اگر عدد همگام‌سازی را بر روی ۱ قرار دهیم؛ طبق فرمول باید ۵۰۰ فریم بر ثانیه داشته باشیم. این در صورتی درست است که کارت گرافیک نیز قدرت پردازش ۵۰۰ فریم از بازی را در یک ثانیه داشته باشد. به عنوان مثال اگر قدرت مانیتور را ۵۰۰ هرتز در نظر بگیریم و بازی توسط کارت گرافیک، در هر ثانیه حدودا ۱۰۰ فریم تولید

کند؛ هر فریم به اندازه ۵ دفعه تازه‌سازی مانیتور نمایش داده خواهد شد. پس عدد بدست آمده در رابطه ۲ حداکثر میزان ممکن است و عدد واقعی ممکن است کمتر باشد.

گاها برخی بازیکنان عدد همگام‌سازی را به عمد پایین می‌آورند تا فشار کمتری به کارت گرافیک وارد شود. این نیز به صورت کدهای تنظیمات سفارشی‌سازی در بازی می‌تواند وجود داشته باشد. بدین صورت، عمر کارت گرافیک نیز افزایش پیدا می‌کند و فریم‌های بیشتر تولید نمی‌شوند. به عنوان مثال یک مانیتور ۱۲۰ هرتز داریم. کارت گرافیک تلاش می‌کند تا با وجود v-sinc ۱۲۰ فریم را تولید نماید اما حدود ۸۰-۹۰ فریم تولید می‌کند. این باعث گرم شدن کارت گرافیک و پایین آمدن عمر مفید آن می‌شود. در اینجا پایین آوردن حداکثر تعداد فریم به عدد ۶۰ می‌تواند راهکار مناسبی باشد.

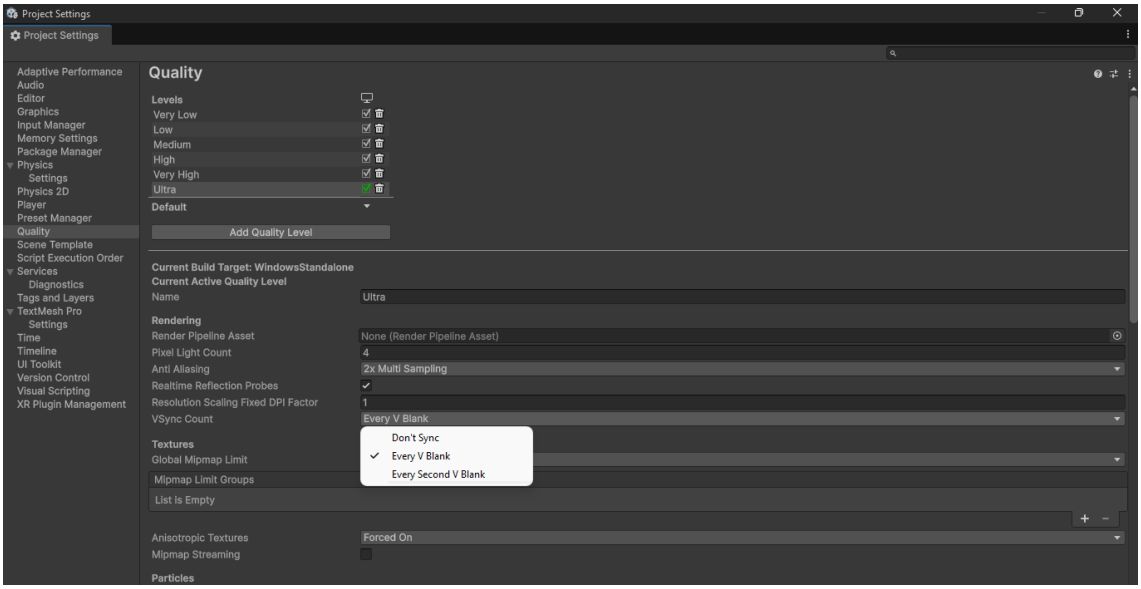
پس از خاموش نمودن همگام‌سازی، می‌توان به صورت دستی نیز حداکثر تعداد فریم بر ثانیه را تعیین نمود. در رابطه ۳ با استفاده از برنامه‌نویسی، تنظیم تعداد فریم در نظر گرفته شده است. در زمانی کار خواهد کرد که همگام‌سازی خاموش باشد. یعنی عدد آن برابر با صفر باشد.

```
1. Application.targetFrameRate = حداکثر تعداد فریم بر ثانیه;
2. Application.targetFrameRate = 60;                          (3)
```

در خط ۱ از رابطه ۳ یک نمونه برنامه شرح داده شده است. از کدهای اپلیکیشن است یعنی تنظیمات کلی برنامه را تنظیم می‌کنند. زیرمجموعه تعداد فریم هدف استفاده شده است. طرق خط دوم، باید عددی وارد نمود تا به عنوان حداکثر تعداد فریم بر ثانیه قلمداد شود. بدین صورت تعداد فریم‌های بازی از تعداد مشخص شده بیشتر نخواهند شد. این فریم‌ها با سرعت یکسان پخش خواهند شد و هرگز فریم‌ها یک‌باره پشت‌سرهم پخش نخواهند شد. به عنوان مثال، یک کارت گرافیک بسیار قوی وجود دارد و ۱۰۰ فریم را در نصف ثانیه نیز می‌تواند تولید کند. حداکثر تعداد فریم مشخص شده؛ ۶۰ فریم برای هر ثانیه است. در این صورت کارت گرافیک صرفا همان ۶۰ فریم را با فاصله زمانی مشخص (حدودا هر ۰.۰۱۶ ثانیه یکبار) به مانیتور می‌دهد و سرعت پخش زیاد یا کم نمی‌شود.

۲.۱.۴. تنظیمات کیفیت بازی

در نرم افزار یونیتی، از نوار بالا گزینه Edit و سپس زیرمجموعه Project Settings را انتخاب می‌کنیم. پنجره‌ای باز می‌شود که شکل آن در شکل ۶ آورده شده است. سربرگ Quality را انتخاب می‌کنیم. در اینجا می‌توان انواع سطوح بصری را مشخص نمود. در بخش‌های مختلف، تعیین کیفیت بصری را می‌توان تغییر داد. V-sync به عنوان یک ابزار در این بخش نشان داده شده است.



شکل ۶ - نمایش ابزارها و تنظیمات کیفیت، همگام‌سازی عمودی در تصویر نمایش داده شده است.

در بخش کیفیت، می‌توان سطوح مختلفی را برای بازی مشخص نمود. کیفیت پایین، متوسط، بالا، عالی و... را می‌توان مشخص نمود. در هر حالت کیفیت نیز می‌توان اطلاعاتی را در نظر گرفت. یکی از این اطلاعات، ابزار v-sync می‌باشد. این ابزار را در تنظیمات می‌توان به سه حالت مختلف قرار داد که در شکل ۶ نیز آورده شده است.

- Don’t Sync: همگام‌سازی انجام نشود.
- Every V Blank: همگام‌سازی همیشه فعال باشد.
- Every Second V Blank: یعنی همگام‌سازی فعال است اما حداکثر با نصف قدرت مانیتور کار می‌کند.

البته برنامه‌نویسی همیشه ارجهیت دارد. اگر برنامه دیگری در تنظیمات داده شده باشد و برنامه‌نویسی هم تنظیمات کیفیت را تحت تاثیر قرار دهد؛ تنظیمات در آن موضوع، نادیده گرفته می‌شود و برنامه اجرا می‌شود. به صورت پیش‌فرض، ابتدا تنظیمات کیفیت یونیتی اعمال می‌شود و سپس تنظیماتی که با برنامه‌نویسی تعیین شده باشند. به عنوان مثال، در تنظیمات، کیفیت همگام‌سازی خاموش باشد اما برنامه‌نویسی آنرا فعال کند؛ همگام‌سازی فعال می‌شود.

۲.۴. اجرای عملی کار و نکات مهم آن

برای مقایسه بازی با/بدون همگام‌سازی عمودی، ابتدا یک پروژه ایجاد نمودیم. در این مسیر نکاتی را مدنظر گرفتیم که در ادامه اشاره خواهند شد. یک صحنه ساده در یونیتی ایجاد شد که شامل یک دوربین، یک نورپردازی (خورشید) و یک مکعب است. مطلوب است که مکعب با سرعت مناسبی بچرخد تا در این چرخش‌ها، شکستگی صفحه را ثبت نماییم. طبق رابطه ۴ یک برنامه نوشته و آن‌را به مکعب نسبت می‌دهیم[6]. این برنامه برای چرخش با سرعت یکسان (۷۰ واحد) است.

```
void Update()
{
transform.Rotate(0, 70 * Time.deltaTime, 0);
}
```
(4)

حال به تنظیمات دوربین می‌رویم. بازی سه‌بعدی است پس پرسپکتیو بر روی خطوط راست نیز تأثیرگذار است. زیرا هرچه از دوربین دورتر شود؛ کوچک‌تر به‌نظر می‌رسد. این پدیده در نمایش خطوط صاف یک مکعب مناسب نیست. حالت پرسپکتیو را خاموش و دوربین را بر روی حالت OrthoGraphic قرار می‌دهیم.

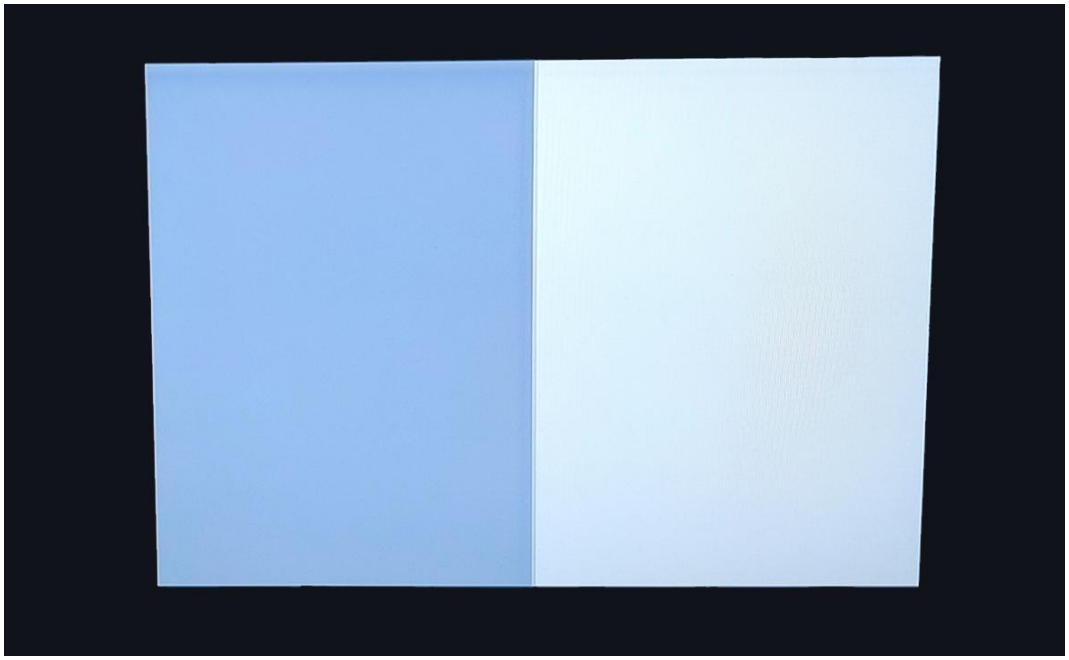
یک برنامه دیگر می‌سازیم تا v-sync خاموش شود. طبق رابطه ۵ برنامه را نوشته و به آبجکتی نسبت می‌دهیم. معمولا برنامه‌ای که تنظیمات نمایش را مدیریت می‌کند؛ به دوربین نسبت می‌دهند اما محدودیتی در انتخاب نیست.

```
void Awake()
{
QualitySettings.vSyncCount = 0;
Application.targetFrameRate = 300;
}
```
(5)

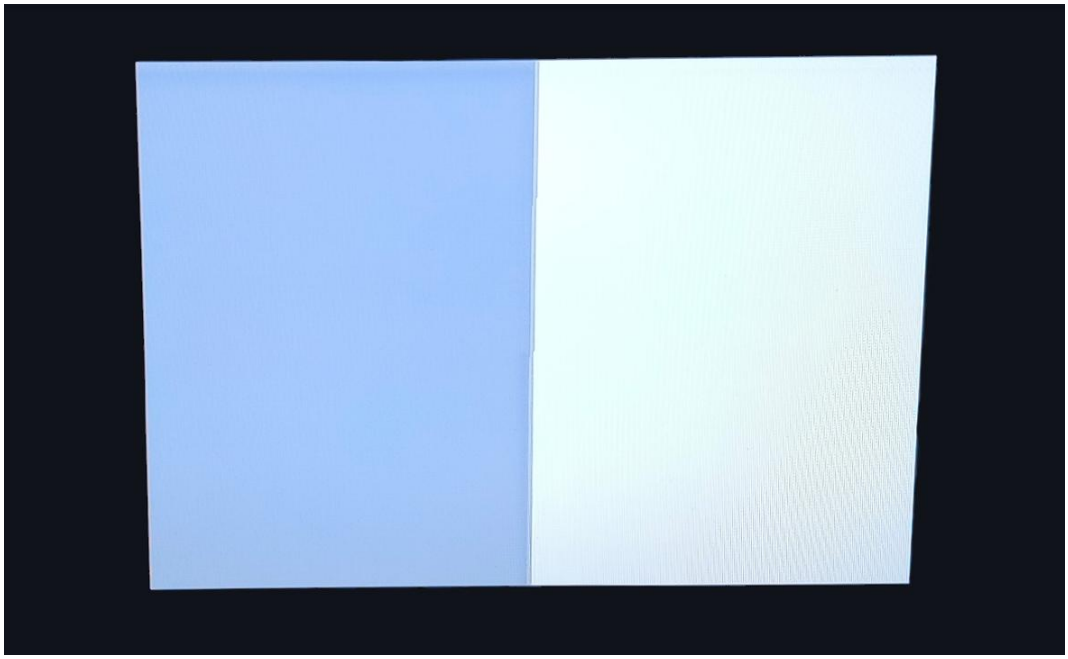
طبق رابطه ۵ همگام‌سازی عمودی غیرفعال می‌شود. حداکثر تعداد فریم بر ثانیه نیز ۳۰۰ فریم است.

حال کافی است دو دفعه بازی را خروجی گرفته و اجرا نماییم. بازی اول با همگام‌سازی و دفعه دوم بدون آن خروجی گرفته می‌شود. بدین صورت تغییرات را با چشم می‌توان مشاهده نمود. به دلیل محدودیت ضبط تصویر در buffer کامپیوتر، فقط می‌توان تصاویر سالم را در کامپیوتر ضبط نمود. برای ضبط تصویر با شکستگی، باید از دوربین استفاده کرده و از مانیتور تصویربرداری شود. بدین منظور، عکس‌هایی از مانیتور گرفته شده است. شکل ۷ توسط کامپیوتر ضبط شده است. شکل‌های ۸ تا ۱۱ با دوربین عکس‌برداری شده‌اند. شکل ۸ بازی را با همگام‌سازی نشان می‌دهد. شکل ۹ تا ۱۱ بازی را بدون همگام‌سازی ثبت و شکستگی تصویر در خط وسط شکل ثبت شده است.

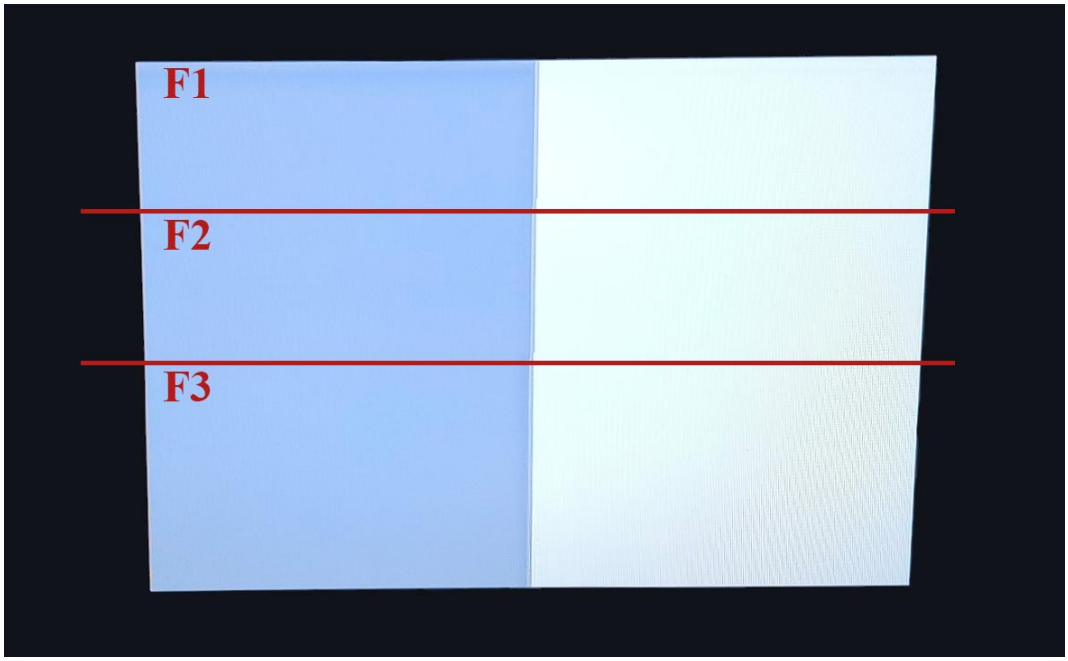
شکل ۷ – تصویر ساده مکعب، ضبط شده از داخل کامپیوتر



شکل ۸ – تصویر ضبط شده با دوربین، اجرای عملی با همگام‌سازی



شکل ۹ - تصویر ضبط شده با دوربین، شکستگی در خط وسط شکل مشخص است.



شکل ۱۰ – تقسیم بندی سه فریم در یک قاب: مکعب در فریم‌های بعدی به سمت چپ چرخش داشته است.

شکل ۱۱ – با زوم روی تصویر این شکستگی بهتر پیداست.

نکته: پخش رنگ و تغییر جزئی در تصاویر ۸ تا ۱۱ به علت عکس‌برداری از صفحه مانیتور با ابزار دوربین است.

۳.۴. اجرای برنامه به صورت پکیج

برای تغییر مقادیر v-sync می‌توان از برنامه‌های بهینه‌تری نیز استفاده نمود. پکیج‌ها راه بهینه‌ای هستند تا برنامه مورد نظر را ذخیره و بارها از آن، در پروژه‌های مختلف استفاده نمود. این پکیج برای فهم بهتر همگام‌سازی و استفاده‌های آن برای برنامه‌نویسان بسیار مناسب است. یک نمونه از آن در رابطه ۶ نشان داده شده است.

```
using UnityEngine;
public class Ardalan : MonoBehaviour
{
public uint myvsinccount = 1;
public uint mytargetframerate = 60;
void Awake()
{
QualitySettings.vSyncCount = (int) myvsinccount;
Application.targetFrameRate = (int) mytargetframerate;
}
}                                                                    (6)
```

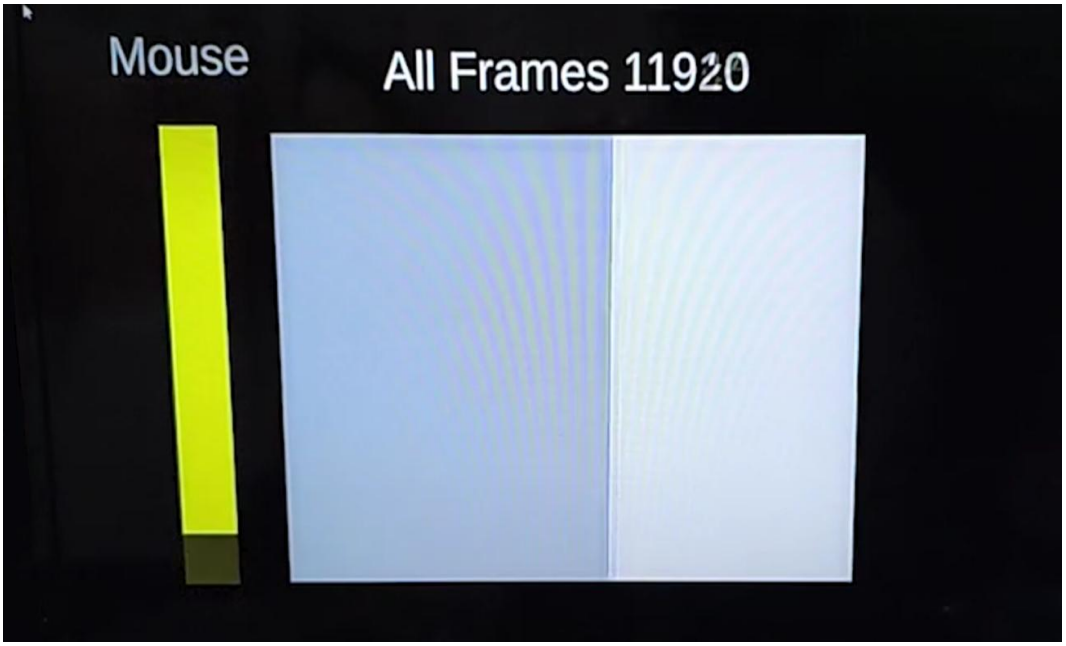
بهتر است تغییرات مربوط به تنظیمات که در رابطه ۶ اشاره شده؛ اصولی و در ابتدای اجرای بازی یعنی در تابع Awake یا Start نوشته شوند. همگام‌سازی فقط در زمانی تغییر کند که فراخوانی شود. مرسوم است که با یک دکمه در بخش تنظیمات فراخوانی شده تا تا خاموش/روشن شود.

همچنین در پکیج برای ورودی، از متغیر uint استفاده شده است. این متغیر عدد منفی نمی‌گیرد و بدین صورت، با تعیین اعداد مختلف، ایرادی در برنامه یافت نمی‌شود. اگر عدد تعداد فریم را منفی کنیم؛ یک کار بی‌معنی انجام داده‌ایم و چنین چیزی تعریف شده نیست.

۵. بررسی تاثیرات همگام‌سازی به صورت عملی

۱.۵. ایجاد پروژه و انجام آزمایش

در این بخش، آزمایش عملی تأثیر همگام‌سازی عمودی بر عملکرد و کیفیت بصری بازی‌های ساخته شده در یونیتی بررسی شده است. برای این منظور، دو نسخه از یک بازی ساده طراحی شد: یکی با فعال بودن V-Sync و دیگری با غیرفعال بودن آن. هدف اصلی، اندازه‌گیری تأخیر ورودی (Input Lag) است. شکل ۱۲ فضای بازی در آزمایش را نمایش می‌دهد. شکستگی تصویر در حالت غیرفعال بودن V-Sync، ناشی از عدم همگام‌سازی بین تعداد فریم بازی و سرعت تازه‌سازی مانیتور است.



شکل ۱۲ – بررسی عملی تاخیر ورودی با همگام‌سازی، عکسی از فیلم‌برداری‌ها

ادوات آزمایش:

- دوربین با ۲۴۰ فریم بر ثانیه: برای فیلم‌برداری از لحظه فشرده‌شدن کلید تا واکنش در صفحه نمایشگر، این واکنش را نمی‌توان با نرم‌افزاری ثبت کرد و استفاده از دوربین الزامی است.
- صفحه نمایشگر با سرعت تازه‌سازی ۶۰ فریم بر ثانیه (۶۰ هرتز)
- کیبورد و موس
- دو بازی ساخته شده توسط یونیتی:

بازی اول بدون همگام‌ساز است و حداکثر تا ۱۵۰۰ فریم بر ثانیه برنامه دارد.

بازی دوم همانند بازی اول است اما همگام‌ساز دارد. (حداکثر ۶۰ فریم و برابر با مانیتور است.)

شرح آزمایش

دوربین با سرعت ۲۴۰ فریم بر ثانیه، به گونه‌ای تنظیم شد که بتواند لحظه فشردن کلید و نمایش واکنش مربوطه در مانیتور را به صورت همزمان ثبت کند. برای هر ورودی (کیبورد و موس)، ۱۰ بار آزمایش تکرار شده است. سپس، فیلم‌های ضبط شده با استفاده از نرم‌افزار ادوبی پریمیر[۴] تحلیل و تعداد فریم‌های بین لحظه ورودی و نمایش واکنش در بازی، یادداشت شد. این اعداد در جدول ۱ و ۲ آورده شده است.

نتایج و تحلیل

جدول ۱ – آمار آزمایش بدون همگام‌سازی

دفعات آزمایش	کیبورد	موس
۱	۱۴	۱۷
۲	۱۶	۱۵
۳	۱۲	۱۶
۴	۱۶	۱۳
۵	۱۴	۱۳
۶	۱۶	۱۵
۷	۱۷	۱۷
۸	۱۵	۱۲
۹	۱۴	۱۲
۱۰	۱۴	۱۴
میانگین فریم	۱۴.۸	۱۴.۸
میانگین زمان	≈ ۰.۰۶۱ ثانیه	≈ ۰.۰۶۰ ثانیه

جدول ۲ – آمار آزمایش با همگام‌سازی

دفعات آزمایش	کیبورد	موس
۱	۲۱	۲۰
۲	۲۰	۲۱
۳	۲۳	۲۲
۴	۲۱	۲۰
۵	۲۱	۲۱
۶	۲۰	۲۲
۷	۲۳	۲۱
۸	۲۳	۲۲
۹	۲۱	۲۳
۱۰	۲۱	۱۹
میانگین فریم	۲۱.۴	۲۱.۱
میانگین زمان	≈ ۰.۰۸۹ ثانیه	≈ ۰.۰۸۸ ثانیه

نتایج آزمایش‌ها نشان داد که فعال بودن همگام‌ساز، به‌طور میانگین باعث افزایش تأخیر ورودی به میزان ۰.۰۲۵ - ۰.۰۳۵ ثانیه می‌شود. (بیست و پنج هزارم یا ۲۵ میلی ثانیه تا سی و پنج هزارم یا ۳۵ میلی ثانیه اختلاف دارند.) اگرچه این مقدار کم به نظر می‌رسد اما در بازی‌های سریع و رقابتی، می‌تواند تأثیراتی بر تجربه کاربر داشته باشد. همچنین، در سیستم‌های قدیمی‌تر با سخت‌افزار ضعیف، این تأخیر ممکن است بیشتر باشد.

نکته: تفاوت‌های مشاهده شده در نتایج تکرارهای مختلف، می‌تواند ناشی از عوامل متعددی مانند نوسانات پردازشی سیستم، تأخیر سخت‌افزاری ورودی‌ها (کیبورد و موس)، تأخیر نمایشگر، و خطای اندازه‌گیری ناشی از سرعت فریم دوربین باشد.

جمع بندی

نتایج آزمایش عملی نشان داد که فعال‌سازی همگام‌سازی عمودی، با حذف پدیده شکستگی صفحه، کیفیت بصری بازی را بهبود می‌بخشد. با این حال، این تکنیک باعث افزایش تأخیر ورودی می‌شود زیرا سیستم مجبور است منتظر سیگنال تازه‌سازی مانیتور بماند تا فریم جدید را نمایش دهد. این تأخیر، اگرچه کوتاه، می‌تواند در بازی‌های رقابتی یا آنلاین که نیاز به واکنش سریع دارند؛ تأثیر منفی بر تجربه کاربر بگذارد.

بنابراین، انتخاب بین فعال یا غیرفعال کردن V-Sync بستگی به اولویت‌های پروژه و انتظارات کاربران دارد. توسعه‌دهندگان بازی می‌توانند با ارائه گزینه‌های سفارشی‌سازی، امکان انتخاب بین کیفیت بصری و عملکرد فنی را به کاربران بدهند تا خودشان تصمیم بگیرند. همچنین، پژوهش‌های آتی می‌توانند به بررسی تأثیر این تأخیر در بازی‌های آنلاین، بازی‌های قدیمی با سخت‌افزار محدود، یا حتی بازی‌های واقعیت مجازی بپردازند.

این پژوهش می‌تواند به عنوان یک منبع علمی برای توسعه‌دهندگان و پژوهشگران حوزه کامپیوتر و بازی‌سازی مورد استناد قرار گیرد و راهنمایی عملی برای انتخاب بهینه تنظیمات گرافیکی در پروژه‌های مختلف باشد.

۲.۵. بررسی عکس‌های دیگر منابع

در این بخش به بررسی بیشتر v-sync خواهیم پرداخت. همانطور که مشاهده شد؛ فعال یا غیرفعال بودن همگام‌سازی تأثیر مستقیم بر کیفیت بصری دارد. البته این تأثیر آنچنان زیاد نیست که در برخی مطالب اینترنتی می‌توان یافت. به عنوان مثال شکل شماره ۱۳ یک تصویر اغراق‌آمیز از شکستگی عمودی است.

شکل ۱۳ – نمایش اغراق‌آمیز شکستگی صفحه در غیاب همگام‌سازی عمودی

این تصویر بسیار پربازدید شده است. این تصویر اغراق‌آمیز است و با مانیتورهای جدید و رزولوشن‌های بالا، شکستگی‌ها انقدر عمیق نخواهند بود. معمولا برای مبالغه و فهم شکستگی از این دست تصاویر استفاده می‌شود اما از لحاظ علمی، کاملا صحیح نیستند. در شکل ۱۴ یک نمونه شکستگی تصویر واقعی آورده شده است. در وسط تصویر، یک شکستگی وجود دارد. در اینجا فقط یک شکستگی داریم یعنی ۲ فریم همزمان نمایش داده شده است.

شکل ۱۴ – یک تصویر شکستگی عمودی واقعی و بدون اغراق

شکل ۱۴ از بازی Kamikaze Veggies است که با نرم‌افزار یونیتی ساخته شده و در تنظیمات نیز اجازه تعیین و تغییر قابلیت همگام‌ساز عمودی را می‌دهد. محل شکستگی صفحه یک خط افقی است که با دایره‌های آبی نشان داده شده است.

حتی اگر ظرفیت مانیتور بسیار پایین و قدرت کارت گرافیک بسیار بالا باشد؛ قوانین داخل بازی اجازه نمی‌دهند که سرعت تغییرات به قدری زیاد باشد که شکستگی‌های بزرگ پدید بیایند.

۶. نتیجه‌گیری

ابزار همگام‌سازی عمودی (V-Sync) به عنوان یکی از تکنیک‌های اساسی در توسعه بازی‌های رایانه‌ای، نقش مهمی در بهبود کیفیت بصری و تجربه کاربری ایفا می‌کند. نتایج این پژوهش نشان داد که فعال‌سازی V-Sync با همگام کردن تعداد فریم بازی و سرعت تازه‌سازی مانیتور، پدیده شکستگی صفحه (Screen Tearing) را به‌طور مؤثر کاهش می‌دهد و تصاویر یکپارچه و روانی را ارائه می‌کند. با این حال، استفاده از این تکنیک می‌تواند تأخیر ورودی (Input Lag) را افزایش دهد زیرا سیستم مجبور است منتظر سیگنال تازه‌سازی مانیتور بماند تا فریم جدید را نمایش دهد. این تأخیر، اگرچه کوتاه است؛ در بازی‌های رقابتی و سریع که نیاز به واکنش فوری دارند؛ می‌تواند تأثیر منفی بر تجربه کاربر بگذارد.

در این پژوهش، با پیاده‌سازی عملی V-Sync در موتور بازی‌سازی یونیتی و انجام آزمایش‌های دقیق، تأثیرات این تکنیک بر عملکرد و کیفیت بصری بازی‌ها بررسی شد. نتایج نشان داد که تفاوت تأخیر ورودی در حالت‌های با و بدون همگام‌سازی، به‌طور میانگین حدود ۰.۰۳ ثانیه است. این تفاوت، هرچند کم، در بازی‌های حساس به زمان می‌تواند اهمیت داشته باشد. همچنین، تحلیل تصاویر عملی نشان داد که شدت شکستگی صفحه در مانیتورهای امروزی، کمتر از تصاویر اغراق‌آمیز موجود در برخی منابع است.

با توجه به یافته‌های این پژوهش، انتخاب بین فعال یا غیرفعال کردن همگام‌ساز، بستگی به نوع بازی و انتظارات کاربران دارد. توسعه‌دهندگان بازی می‌توانند با ارائه گزینه‌های سفارشی‌سازی در تنظیمات بازی، امکان انتخاب بین کیفیت بصری و عملکرد فنی را به کاربران بدهند. این پژوهش می‌تواند به عنوان یک منبع علمی و عملی برای اساتید، توسعه‌دهندگان بازی، پژوهشگران و علاقه‌مندان به حوزه بازی‌سازی مورد استفاده قرار گیرد و راهنمایی مفید برای انتخاب بهینه تنظیمات گرافیکی در پروژه‌های مختلف باشد.

در نهایت، پیشنهاد می‌شود پژوهش‌های آتی به بررسی تأثیر V-Sync در بازی‌های آنلاین، بازی‌های واقعیت مجازی و بازی‌های قدیمی با سخت‌افزار محدود، یا ابزارهای دیگر مانند G-sync و free-sync بپردازند تا درک جامع‌تری از مزایا و محدودیت‌های این تکنیک در شرایط مختلف حاصل شود.

۷. قدردانی

با تشکر ویژه و فراوان از آقا امام زمان (عج) برای تمام الطافی که داشتند و دارند.

با تشکر و قدردانی ویژه و فراوان از آقا امام خامنه‌ای شهید و بزرگوار که در تمام زندگی الگوی من و یاور جهانیان بوده‌اند.

۸. منابع و مراجع

[۱] اردلان، امیرحسین، تابستان ۱۴۰۴. « *فرآیند بازیسازی رایانه‌ای: از ایده تا انتشار؛ راهنمای جامع برای علاقه‌مندان به هنر و صنعت بازی سازی*»، معماری سیستم‌های کامپیوتری، دوره ۵، شماره ۲، صص. ۶۲-۷۲.

[2] Liu, S., Kuwahara, A., Scovell, J. J., & Claypool, M. (2023). The effects of frame rate variation on game player quality of experience. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1–10).

[3] Akenine-Möller, T., Haines, E., & Hoffman, N. (2018). *Real-Time Rendering*. CRC Press.

[4] Halbhuber, D., Schauhuber, P., Schwind, V., & Henze, N. (2023). The effects of latency and in-game perspective on player performance and game experience. *Proceedings of the ACM on Human-Computer Interaction*, 7(CHI PLAY), 1308–1329. https://doi.org/10.1145/3611070

[5] A. J. Brook, "Fix screen tearing and boost game performance with vertical sync," ReneeLab, 2023. [Online]. Available: https://www.reneelab.com/vertical-synch-and-methods.html

[6] J. Hocking, Unity in Action: Multiplatform Game Development in C# with Unity 5. Shelter Island, NY, USA: Manning Publications, 2015.

Investigating the Impact of Vertical Synchronization (V-Sync) on Performance and Visual Quality in Unity Engine Games

Amirhosein Ardalan*
Game Design Instructor, Avicenna International College, Tehran, Iran; Tbilisi, Georgia.
(shsbbhorse@chmail.ir)

Abstract—Vertical Synchronization (commonly referred to as V-Sync) is a fundamental technique in computer game development. By synchronizing the monitor's refresh rate with the game's output frame rate, this technique mitigates the phenomenon of screen tearing. Despite its visual benefits, V-Sync can have varying effects on game performance; particularly when the game's frame count is not aligned with the monitor's refresh rate, it may lead to increased input lag or a reduction in frame rates .In this research, the technical principles and operational mechanisms of V-Sync are comprehensively explained. Subsequently, the implementation of this technique within the Unity game engine using the C# programming language is examined. Sample code snippets are provided to demonstrate how to enable or disable synchronization in Unity projects and configure its properties. Furthermore, through practical images and an analysis of experimental results, the tangible impact of this tool on both visual fidelity and game performance is illustrated. Finally, the study critiques and analyzes exaggerated imagery found in various sources. The primary objective of this research is to achieve a deeper understanding of the advantages and limitations of Vertical Synchronization

Keywords: Game Development, Unity, Unity Engine, Monitor, V-Sync, Vertical Synchronization, C#, Computer Games, Display Screen

[1] FPS: Frame Per second

[2] Frame\seconds
[3] Refresh Rate
[4] Adobe Premiere