

لَا إِلَهَ إِلَّا اللَّهُ



دانشگاه صنعتی اصفهان
مهندسی برق و کامپیوتر

بررسی سیستم گراف انیمیشن در موتور بازی سازی آنریل و و پیاده سازی یک سیستم انیمیشن با استفاده از OpenGL

پایان نامه کارشناسی مهندسی کامپیوتر

نامی نذیری

استاد راهنما
دکتر مازیار پالهنک



دانشگاه صنعتی اصفهان
مهندسی برق و کامپیوتر

پایان نامه کارشناسی رشته مهندسی کامپیوتر آقای نامی نذیری
تحت عنوان

بررسی سیستم گراف انیمیشن در موتور بازی سازی آنریل و
و پیاده سازی یک سیستم انیمیشن با استفاده از OpenGL

در تاریخ ۱۳۹۵/۱۰/۲۰ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت:

۱- استاد راهنمای پایان نامه دکتر مازیار پالهنک

۲- استاد داور دکتر داور اول

۳- استاد داور دکتر داور دوم

سرپرست تحصیلات تکمیلی دانشکده دکتر تحصیلات تکمیلی دانشکده

کلیه حقوق مالکیت مادی و معنوی مربوط به این پایان نامه متعلق به دانشگاه صنعتی اصفهان و پدیدآورندگان است. این حقوق توسط دانشگاه صنعتی اصفهان و بر اساس خط مشی مالکیت فکری این دانشگاه، ارزش گذاری و سهم بندی خواهد شد. هر گونه بهره برداری از محتوا، نتایج یا اقدام برای تجاری سازی دستاوردهای این پایان نامه تنها با مجوز کتبی دانشگاه صنعتی اصفهان امکان پذیر است.

فهرست مطالب

عنوان	صفحه
فهرست مطالب.....	شش
فهرست شکل ها.....	هشت
فهرست جدول ها.....	نه
فهرست الگوریتم ها.....	ده
چکیده.....	۱
فصل اول: مقدمه	
۲	
فصل دوم: مرور	
۵	
۱-۲ تاریخچه ی پویانمایی سستی.....	۶
۱-۱-۲ فریم های کلیدی و درمیان.....	۶
۲-۱-۲ چشم انداز چند منظوره.....	۷
۳-۱-۲ لایه های مختلف.....	۷
۲-۲ پویانمایی کامپیوتری.....	۸
۱-۲-۲ فریم های کلیدی و درمیان.....	۹
۲-۲-۲ رویه.....	۹
۳-۲-۲ مبتنی بر فیزیک.....	۹
۴-۲-۲ ضبط حرکت ^۱	۹
۳-۲ موتور بازی سازی.....	۱۰
۴-۲ موتور بازی سازی آنریل.....	۱۰
۵-۲ زبان برنامه نویسی در آنریل.....	۱۱
۶-۲ گرافیک کامپیوتری.....	۱۱
۱-۶-۲ OpenGL.....	۱۲
۲-۶-۲ سایه زنی فونگ.....	۱۲
۳-۶-۲ نور محیطی.....	۱۲
۴-۶-۲ نور پخش شده.....	۱۳
۵-۶-۲ نور آینه وار.....	۱۳
۷-۲ پویانمایی اسکلتی.....	۱۳
۱-۷-۲ شبکه ی ^۲ چند ضلعی.....	۱۴
۲-۷-۲ مدل.....	۱۵
۳-۷-۲ زیر مش ^۳	۱۶
۴-۷-۲ ماده ^۴	۱۷

^۱ Motion Capture

^۲ Mesh

^۳ Sub-Mesh

^۴ Material

۱۷.....	۵-۷-۲ بافت ^۵
۱۸.....	۶-۷-۲ اسکلت
۱۹.....	۷-۷-۲ Skinning
۱۹.....	۸-۷-۲ ژست شخصیت
۲۰.....	۹-۷-۲ کلیپ‌های پویانمایی
۲۰.....	۱۰-۷-۲ ترکیب کلیپ‌های پویانمایی

۲۲	پیوست‌ها
----	----------

۲۳	مراجع
----	-------

⁵Texture

فهرست شکل ها

<u>صفحه</u>	<u>عنوان</u>
۷.....	شکل ۲-۱: فریم های کلیدی و درمیان [۳].....
۷.....	شکل ۲-۲: چشم انداز چند منظوره [۴].....
۸.....	شکل ۲-۳: لایه های مختلف [۵].....
۱۳.....	شکل ۲-۴: مولفه های الگوریتم سایه زنی فونگ [۱۴].....
۱۴.....	شکل ۲-۵: نمایش یک دلفین به وسیله ی مش مثلثی [۱۵].....
۱۵.....	شکل ۲-۶: عناصر یک مش چند ضلعی [۱۵].....
۱۶.....	شکل ۲-۷: اجزاء مختلف ماشین [۱۶].....
۱۷.....	شکل ۲-۸: چند نوع ماده ی مختلف [۱۷].....
۱۸.....	شکل ۲-۹: چند نوع بافت مختلف [۱۸].....
۱۸.....	شکل ۲-۱۰: در سمت راست مش مدل و سمت چپ اسکلت مدل قابل مشاهده است. [۱۹].....
	شکل ۲-۱۱: عکس نشان دهنده ی میزان تعلق رئوس اطراف مفصل شانه است. هرچه راس به مفصل نزدیک تر باشد، تعلق بیشتری به آن دارد (رنگ قرمز تری می گیرند). [۲۰].....
۱۹.....	

فهرست جدول‌ها

صفحه

عنوان

فهرست الگوریتم‌ها

چکیده

کلمات کلیدی

فصل اول

مقدمه

پویانمایی هنر جان بخشیدن به اجسام بدون جان است. والت دیزنی درباره‌ی پویانمایی می‌گوید: ”پویانمایی می‌تواند هر آنچه را که ذهن انسان تصور می‌کند، توضیح دهد“

وقتی می‌گوییم جسمی را پویا کردیم، یعنی به آن جان بخشیدیم. زمانی که یک فیلم پویانمایی شده را در تلوزیون یا سینما می‌بینید، شخصیت‌های درون آن فیلم در حالت حرکت هستند. این حرکت معمولاً صاف و به هم پیوسته است. نوارهای حاوی فیلم متشکل از دنباله‌ای از تصاویر هستند که به عنوان ”فریم“ شناخته می‌شوند و درواقع با پخش شدن این فریم‌ها به صورت متوالی، توهم ایجاد حرکت به مخاطب ابراز می‌شود.

پویانمایی از گذشته تا امروز تغییرات فراوانی را دیده است. در پویانمایی سنتی، تصاویر به وسیله‌ی دست روی صفحات سلولوئیدی شفاف ترسیم یا نقاشی شده سپس از آن‌ها عکس گرفته و روی فیلم نمایش داده می‌شدند. امروزه اکثر پویانمایی‌ها با تصاویر کامپیوتری^۱ ساخته می‌شوند. [۱] علاوه بر این، دامنه‌ی استفاده از این پویانمایی نیز دستخوش بسیاری تغییرات شده است. در گذشته پویانمایی را می‌توانستیم در فیلم‌های پویانمایی شده یا کارتون‌ها مشاهده کنیم. اما اکنون با پیشرفت تکنولوژی، پویانمایی نقش بسیار اساسی‌ای در بازی‌های کامپیوتری پیدا کرده است. هدف بازی‌های کامپیوتری، به خصوص بازی‌های کامپیوتری داستان محور، غوطه‌ور کردن بازیکن در داستان است. همانطور که اشاره

^۱CGI

شد پویانمایی هنر جان بخشیدن به اجسام است و به وسیله‌ی آن است که می‌توانیم احساسات و اعمال شخصیت بازی را به بازیکن منتقل کنیم.

بازی‌های کامپیوتری به صورت معمول توسط موتورهای بازی‌سازی ساخته می‌شوند. اگر بخواهیم تعریفی برای موتور بازی‌سازی آوریم می‌توان گفت آن‌ها پلتفرم‌هایی هستند که ساخت بازی‌های رایانه‌ای را آسان‌تر می‌کنند. موتورهای بازی‌سازی متشکل از مولفه‌های مختلفی هستند که قابلیت‌های لازم برای ساخت بازی را فراهم می‌کنند. از رایج‌ترین مولفه‌های موتور بازی می‌توان به مولفه‌ی صدا، مولفه‌ی رندر، مولفه‌ی هوش مصنوعی و مولفه‌ی پویانمایی اشاره کرد. [۲]

هدف اصلی این پروژه آشنایی با روش‌های استفاده شده در محیط‌های گرافیکی مانند موتورهای بازی‌سازی با تاکید بیشتر بر روی سیستم‌های پویانمایی به کار رفته در این محیط‌ها است.

به همین جهت این پروژه به دو صورت این هدف را دنبال می‌کند. جهت آشناشدن با یک موتور بازی‌سازی و نحوه‌ی پیاده‌سازی سیستم پویانمایی آن، موتور بازی‌سازی آنریل انتخاب شده است. آنریل یکی از معروف‌ترین موتورهای بازی‌سازی در جهان است که اولین نسل آن توسط تیم سوینی، بنیانگذار اپیک گیمز^۱، توسعه یافت. آخرین نسخه‌ی این موتور به اسم موتور بازی‌سازی آنریل ۵ در سال ۲۰۲۰ معرفی و در سال ۲۰۲۲ انتشار یافت. سیستم پویانمایی این موتور بسیار وسیع است. به همین دلیل بخش کوچکی از این سیستم که گراف پویانمایی نام دارد، انتخاب شده و به بررسی ساختار و نحوه‌ی استفاده از این گراف می‌پردازیم.

پس از بدست آوردن تجربه‌ی اولیه از گراف پویانمایی برای آشنایی کامل تر با محیط گرافیکی و همچنین سیستم پویانمایی به پیاده‌سازی یک سیستم پویانمایی با استفاده از OpenGL پرداختیم. OpenGL یک واسط برنامه نویسی کاربردی^۲ است که با فراهم کردن توابع مختلف به توسعه‌دهندگان امکان دستکاری گرافیک و تصاویر را می‌دهد. با استفاده از این API می‌توان آشنایی خوبی در مورد گرافیک کامپیوتری و به صورت کلی محیط‌های گرافیکی بدست آورد. برای محیط سه‌بعدی پیاده‌سازی شده از روش Phong Shading برای نورپردازی محیط استفاده شده است. این روش یکی از معروف‌ترین روش‌های نورپردازی در محیط‌های سه‌بعدی بلادرنگ به‌خصوص بازی‌های کامپیوتری است. علاوه بر تولید صحنه‌ی سه‌بعدی، برای بدست آوردن آشنایی کامل با سیستم‌های پویانمایی که در بازی‌ها استفاده می‌شوند، به پیاده‌سازی یک نمونه از آن پرداختیم. در این پیاده‌سازی سیستم پویانمایی به چند بخش کلی تقسیم شده است که هر کدام هدف‌های مختلفی را دنبال می‌کند. برای اینکه یک سیستم پویانمایی داشته باشیم در ابتدا به یک شخصیتی نیاز داریم تا کلیپ‌های پویانمایی بر روی آن اجرا شود. شخصیت‌ها در این پیاده‌سازی توسط کتابخانه‌ی Assimp در ساختمان داده‌های مناسب ذخیره می‌شود. هر شخصیت در این پیاده‌سازی به دو قسمت کلی مش و اسکلت

^۱Epic Games

^۲API

تقسیم می‌شود. یکی از وظایف مهم این پیاده‌سازی، اتصال این دو قسمت به یکدیگر است. این اتصال به صورت کلی به اسم Skinning نام دارد. مرحله‌ی بعدی پیاده‌سازی به پخش کلیپ‌های پویانمایی بر روی این شخصیت می‌پردازد. در نهایت از ماشین حالت متناهی برای برای ترکیب کلیپ‌های پویانمایی متفاوت با یکدیگر استفاده شده است. خروجی این پروژه یک تحقیق در مورد سیستم گراف پویانمایی آنریل به همراه یک نرم‌افزار گرافیکی سیستم پویانمایی است.

در فصل‌های آتی به بررسی این موارد گفته‌شده پرداخته می‌شود. ابتدا در فصل دوم یک مروری بر تاریخچه‌ی پویانمایی می‌شود. سپس توضیحاتی درباره‌ی موتور بازی‌سازی و موتور بازی‌سازی آنریل داده می‌شود و در نهایت توضیحاتی کلی درباره‌ی روش‌های نورپردازی محیط و پویانمایی اسکلتونی که به وفور در موتورهای بازی‌سازی استفاده می‌شود داده می‌شود.

در فصل سوم به بررسی موتور بازی‌سازی آنریل با تاکید بر روی گراف پویانمایی می‌پردازیم و نحوه‌ی استفاده از آن را بررسی می‌کنیم.

در نهایت در فصل چهارم توضیحاتی درباره‌ی نحوه‌ی پیاده‌سازی سیستم پویانمایی به همراه توضیحات سیستم‌های موجود در این پیاده‌سازی می‌پردازیم.

در فصل "نتیجه‌گیری"، یک نتیجه‌گیری کلی از خروجی‌های این پروژه ارائه کرده و در فصل "کارهای آینده"، به بررسی مشکلاتی که می‌تواند در پیاده‌سازی برطرف شود به همراه پیشنهاداتی برای ادامه‌ی این پروژه پرداخته می‌شود.

فصل دوم

مرور

از گذشته تا امروز نحوه‌ی تولید و همچنین استفاده از کلیپ‌های پویانمایی تغییرات زیادی کرده است. در گذشته اکثر پویانمایی‌ها با استفاده از دست و ترسیم بر روی صفحات سلولوئیدی تولید می‌شدند. در صورتیکه که امروزه، اکثر کلیپ‌های پویانمایی توسط تصاویر گرافیکی کامپیوتری ساخته می‌شوند. اگرچه با پیشرفت تکنولوژی روش‌های جدیدی برای تولید پویانمایی به وجود آمده، همچنان نیز از اکثر روش‌هایی که در پویانمایی سنتی استفاده می‌شد در پویانمایی کامپیوتری نیز استفاده می‌شود.

امروزه نه تنها از پویانمایی برای تولید فیلم و کارتون، استفاده می‌شود، بلکه آن‌ها نقش تاثیرگذاری در بازی‌های کامپیوتری پیدا کرده‌اند. برای تولید بازی‌های کامپیوتری از موتورهای بازی‌سازی استفاده می‌شود. موتورهای بازی‌سازی دارای مولفه‌های متفاوتی هستند که هر کدام وظیفه‌ی مخصوص به خودشان را دارند. مولفه‌ای که وظیفه‌ی پخش کلیپ‌های پویانمایی را دارد، مولفه‌ی پویانمایی گویند. بدیهی است اگر بخواهیم از سیستم پویانمایی استفاده کنیم نیازمند یک محیط گرافیکی هستیم. محیط‌های گرافیکی به صورت معمول از کارت گرافیک استفاده می‌کنند. برای دسترسی به توابع کارت گرافیکی می‌توان از یک واسط برنامه‌نویسی کاربردی مانند DirectX یا OpenGL استفاده کرد. علاوه بر این برای مشاهده‌ی اشیاء موجود در محیط سه‌بعدی نیاز است که محیط را نورپردازی کنیم. روش‌های متنوعی برای نورپردازی محیط وجود دارند. یکی از این روش‌ها Phong Shading است که در این پروژه نیز استفاده

شده است.

در اکثر سیستم‌های پویانمایی بازی‌های کامپیوتری از پویانمایی اسکلتی برای پخش و ترکیب کلیپ‌های پویانمایی استفاده می‌شود.

پویانمایی اسکلتی تکنیکی در پویانمایی کامپیوتری است که به وسیله‌ی آن، شخصیت‌های درون بازی متحرک می‌شوند. این سیستم به دو بخش کلی تقسیم می‌شود. یک بخش، یک مش یا پوسته است که برای به نمایش کشاندن شخصیت در محیط سه‌بعدی استفاده می‌شود و بخش دوم یک اسکلت است. این اسکلت مجموعه سلسله مراتبی از قطعات به هم پیوسته است که به هر قطعه یک مفصل گویند. در این تکنیک، اسکلت شخصیت درون بازی متحرک شده و با روش‌های موجود، پوسته یا مش آن شخصیت، اسکلت را دنبال می‌کند.

در این فصل، ابتدا مروری بر پویانمایی سنتی و روش‌های تولید آن می‌کنیم. سپس نگاهی به پیشرفت صنعت تولید پویانمایی کامپیوتری می‌اندازیم و روش‌های فعلی تولید پویانمایی را خواهیم گفت. پس از آن توضیحی در رابطه با موتورهای بازی‌سازی و به ویژه موتور آنریل خواهیم داد.

در قسمت انتهایی توضیحی در مورد Phong Shading که یکی از روش‌های نورپردازی محیط سه‌بعدی است خواهیم داد و علاوه بر آن در انتها توضیحات جامعی را در مورد پویانمایی اسکلتی ارائه خواهیم کرد.

۲-۱ تاریخچه‌ی پویانمایی سنتی

پویانمایی سنتی که با اسم‌های مختلفی مانند "پویانمایی مرسوم"، "پویانمایی سل‌ای" و "پویانمایی بادست" شناخته می‌شود، روشی غالب برای تولید فیلم‌های پویانمایی شده در حدود قرن ۲۰ میلادی بود. در این روش، به صورت کلی پویانمایی به وسیله‌ی نقاشی با دست به وجود می‌آید. درواقع هر فریم از فیلم، یک عکس از نقاشی است و برای به وجود آوردن توهم حرکت، هر نقاشی اندکی با نقاشی قبلی خود تفاوت دارد.

برای تولید پویانمایی سنتی، از روش‌های مختلفی استفاده می‌شد. در اینجا به بررسی سه روش استفاده شده در پویانمایی سنتی می‌پردازیم.

۲-۱-۱ فریم‌های کلیدی و درمیان

از آنجایی که تولید پویایی با دست و کشیدن نقاشی کار بسیار طولانی‌ای بود، برای اینکه وقت پویانمایی‌های ارشد ذخیره شود، این پویانماها فریم‌های اصلی یک حرکت را بر روی کاغذ ترسیم می‌کردند و فریم‌های میانی را پویانماهای جوان پر می‌کردند.



شکل ۲-۱ - فریم های کلیدی و درمیان [۳]

۲-۱-۲ چشم انداز چندمنظوره

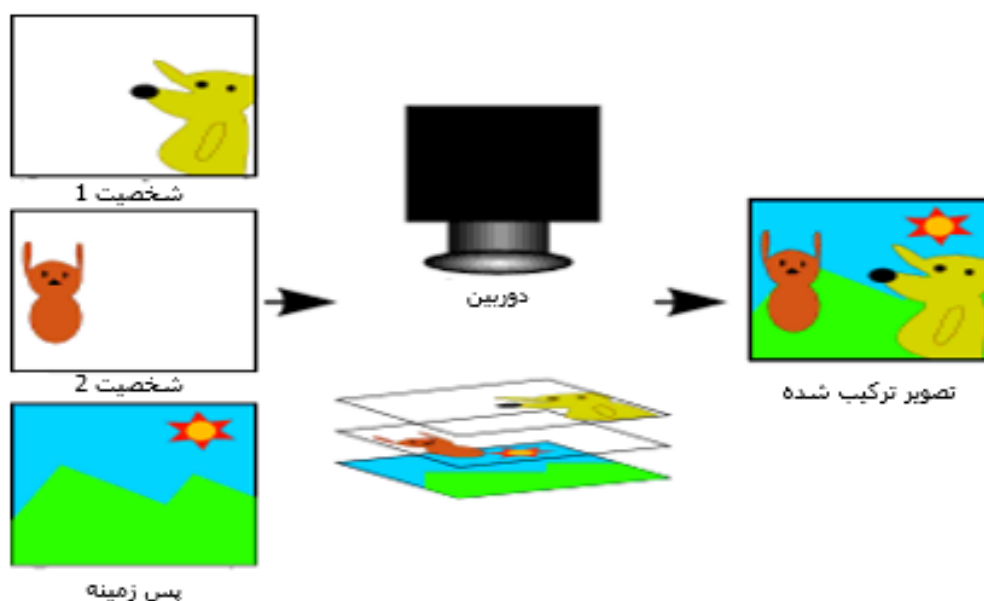
استفاده از چشم انداز چندمنظوره روش دیگری بود که در پویانمایی سنتی استفاده می شد. همطور که از تصویر زیر مشخص است، برای نمایش یک محیط از یک چشم انداز استفاده می شد. این چشم انداز می توانست نشان دهنده ی قرارگیری محیط در فواصل مختلف باشد. در این صورت، زمانی که دوربین در صحنه حرکت می کرد این توهّم را در مخاطب ایجاد می کرد که گویی در محیط در حال حرکت هستیم.



شکل ۲-۲ - چشم انداز چندمنظوره [۴]

۳-۱-۲ لایه های مختلف

با استفاده از این روش، پویانماها یک صحنه را به چند قسمت مختلف تقسیم می کردند. به عنوان مثال لایه های مختلفی برای هر شخصیت درون صحنه استفاده می شد. علاوه بر این یک لایه نیز برای تصویر پس زمینه استفاده می شد. از آنجایی که این لایه ها یک صفحه ی شفاف هستند بنابراین می توان لایه ها را بر روی هم انباشته کرد و با تصویر برداری از بالا تمام صحنه را تصویربرداری کرد. این روش در تصویر ۳-۲ آورده شده است.



شکل ۲-۳- لایه‌های مختلف [۵]

۲-۲ پویانمایی کامپیوتری

اگر بخواهیم نگاهی به تاریخچه‌ی انیمیشن‌های کامپیوتری بیاندازیم، مشاهده می‌کنیم که در حدود دهه‌ی ۱۹۸۰ میلادی شرکت دیزنی به عنوان یکی از اولین شرکت‌های جهان، شروع به دیجیتالی کردن خط لوله‌ی تولید پویانمایی سنتی خود کرد. در این دیجیتال‌سازی بسیاری از روش‌ها و ایده‌های استفاده شده در پویانمایی سنتی، به کار گرفته شد. اولین مقالات این حوزه توسط آقای جان لستر از کارمندان پیکسار به عنوان "اصول پویانمایی سنتی به کار رفته در پویانمایی کامپیوتری سه‌بعدی" ارائه شد. در این مقاله اصول اولیه پویانمایی سنتی دوبعدی ترسیم شده با دست و کاربرد آن‌ها در پویانمایی کامپیوتری سه‌بعدی شرح داده شده است. [۶]

پویانمایی کامپیوتری تنها محدود به دنیای سینما و فیلم‌های پویانمایی نمی‌شوند و به دنیای بازی‌های کامپیوتری نیز ورود پیدا کرده‌اند. بازی‌های کامپیوتری سعی می‌کنند دیوار میان تماشاگران و فیلم را بشکنند و با تعاملی بودن و دادن آزادی عمل به بازیکن، سعی می‌کنند داستان را به گونه‌ای تعریف کنند که گویی بازیکن یکی از شخصیت‌های اصلی داستان است. پویانمایی در بازی‌های کامپیوتری اهمیت بسیار بالایی دارد زیرا همانطور که گفته شد باعث جان بخشیدن به شخصیت‌ها می‌شود که اهمیت بسیار بالایی برای جلب توجه بازیکنان در هنگام داستان‌سرایی دارد.

با پیشرفت تکنولوژی، همراه با استفاده از روش‌های گذشته، روش‌های جدیدتری برای تولید پویانمایی توسعه یافته‌است که در ادامه به چند مورد از آن‌ها می‌پردازیم.

۱-۲-۲ فریم‌های کلیدی و درمیان

همانطور که اشاره شد در پویانمایی کامپیوتری از روش‌های موجود در پویانمایی سنتی استفاده شده است. در اینجا نیز فریم‌های کلیدی یک حرکت توسط پویانماها به وجود می‌آیند ولی فریم‌های میانی به جای اینکه توسط پویانماها به وجود آیند، توسط کامپیوتر با استفاده از روش‌های درونیابی به وجود می‌آیند.

۲-۲-۲ رویه

در این روش، حرکت بر اساس یک الگوریتم بیان می‌شود. درواقع انیمیشن‌ها در این نوع پویانمایی، تابعی با تعداد کمی از متغیرها هستند. به عنوان مثال یک تابعی را در نظر بگیرید که به گرفتن ورودی ثانیه، دقیقه و ساعت، یک شیء ساعت را خروجی دهد که عقربه‌هایش در جای مناسب با توجه به ورودی‌ها قرار گرفته باشد. حال می‌توان با تغییر ورودی‌ها حرکت ساعت را شبیه‌سازی کنیم.

۳-۲-۲ مبتنی بر فیزیک

پویانمایی مبتنی بر فیزیک پلی میان دنیای پویانمایی با دنیای واقعی است. در این روش با نسبت دادن ویژگی‌های فیزیک به اشیاء سه‌بعدی و سپس حل کردن فرمول‌های فیزیک مانند فرمول حرکت یا فرمول‌های نیوتن، فیزیک را شبیه سازی می‌کند. پویانمایی‌های مبتنی بر فیزیک شخصیت را قادر می‌سازد تا حرکت‌های خود را به صورت پویا با محیط تنظیم کند.

۴-۲-۲ ضبط حرکت^۱

به فرآیند ثبت و دیجیتالی کردن حرکت یک شیء یا شخص، ضبط حرکت گویند. ضبط حرکت توسط دوربین‌های مادون قرمز که تعدادی زیادی از آن‌ها در صحنه‌ی ضبط قرار دارند، صورت می‌گیرد. این دوربین‌ها به صورت شبکه به یکدیگر متصل هستند و پس از کالیبره شدن، آماده‌ی استفاده هستند. این دوربین‌ها با استفاده از نشانگرهای سفیدی که بر روی لباس بازیگران ضبط حرکت قرار دارد، داده‌های مورد نیازشان را دریافت می‌کنند. قابل ذکر است این نشانگرها بازتابنده‌ی مادون قرمز هستند که توسط دوربین‌ها دریافت می‌شود. در نهایت پویانماها به پاکسازی و پردازش این داده‌ها پرداخته تا آن را برای استفاده‌ی شخصیت‌های سه بعدی آماده کنند.

^۱ Motion Capture

۳-۲ موتور بازی سازی

موتورهای بازی سازی پلتفرم هایی هستند که ساخت بازی های رایانه ای را آسان تر می کنند. آن ها به شما این امکان را می دهند تا عناصر بازی مانند انیمیشن، تعامل با کاربر یا تشخیص برخورد میان اشیاء را در یک واحد ادغام و ترکیب کنید. [۲] زمانی که از اصطلاح موتور بازی سازی استفاده می کنیم منظورمان نرم افزارهای قابل توسعه ای هستند که می توانند پایه و اساس بسیاری از بازی های مختلف باشند. [۷] موتورهای بازی سازی متشکل از اجزای مختلفی هستند که قابلیت های لازم برای ساخت بازی را فراهم می کنند. رایج ترین اجزای موتور بازی عبارتند از: [۲]

- مولفه ی صدا: نقش اصلی این مولفه تولید جلوه های صوتی در بازی است.
- موتور رندر: وظیفه اصلی این مولفه تبدیل داده های ورودی به پیکسل ها، برای به تصویر کشاندن بر روی صفحه است.
- مولفه هوش مصنوعی: این مولفه مسئولیت ارائه ی تکنیک هایی برای تعریف قوانین رفتار شخصیت هایی را دارد که توسط بازیکنان کنترل نمی شوند.
- مولفه انیمیشن: نقش اصلی این مولفه اجرای انیمیشن های مختلف مانند حرکت است.
- مولفه شبکه: وظیفه اصلی این مولفه قادر ساختن بازی همزمان بازیکنان با یکدیگر، از طریق استفاده از دستگاه های متصل به اینترنت است.
- مولفه منطق یا مکانیک بازی: این مولفه قوانین حاکم بر دنیای مجازی، ویژگی های شخصیت های بازیکنان، هوش مصنوعی و اشیاء موجود در دنیای مجازی و همچنین وظایف و اهداف بازیکنان را تعریف می کند.
- ابزارهای نرم افزاری: وظیفه اصلی این ابزارها افزایش راندمان و سرعت تولید بازی با موتور بازی سازی است. آن ها توانایی اضافه کردن بسیاری از عناصر مختلف را به بازی ها، از انیمیشن و جلوه های صوتی گرفته تا الگوریتم های هوش مصنوعی، را فراهم می کنند.
- یکی از مهم ترین مولفه های موجود در هر موتور بازی، مولفه ی انیمیشن آن است. در این پروژه به بررسی سیستم انیمیشن گراف که وظیفه ی پخش انیمیشن شخصیت های سه بعدی را در موتور بازی آنریل دارد می پردازیم.

۴-۲ موتور بازی سازی آنریل

اولین نسل موتور بازی سازی آنریل توسط تیم سوینی، بنیانگذار اپیک گیمز^۱، توسعه یافت. سوینی در سال ۱۹۹۵ شروع به نوشتن این موتور برای تولید بازی تیراندازی اول شخص به اسم غیر واقعی^۲ کرد. نسخه ی دوم موتور بازی سازی آنریل در سال ۲۰۰۲ منتشر شد.

^۱Epic Games

^۲Unreal

نسخه سوم نیز در سال ۲۰۰۴ پس از حدود ۱۸ ماه توسعه، منتشر شد. در این نسخه، معماری پایه‌ای موجود در نسخه‌ی اول مانند طراحی شی گرا، اسکریپت‌نویسی مبتنی بر داده و رویکرد نسبتاً مازولار نسبت به زیرسیستم‌ها وجود داشت. اما برخلاف نسخه دوم که از یک خط لوله با عملکرد ثابت^۱ استفاده می‌کرد، این نسخه به صورتی طراحی شده بود تا بتوان قسمت‌های سایه‌زنی سخت‌افزاری^۲ را برنامه‌نویسی کرد.

موتور بازی‌سازی آنریل ۴ در سال ۲۰۱۴ در کنفرانس توسعه‌دهندگان بازی^۳ منتشر شد. این نسخه با طرح کسب‌وکار اشتراکی برای توسعه‌دهندگان در دسترس قرار گرفت. این اشتراک به صورت ماهانه، با پرداخت ۱۹ دلار آمریکا به توسعه‌دهندگان این اجازه را می‌داد تا به نسخه‌ی کامل موتور، از جمله کد منبع C++ آن دسترسی پیدا کنند. البته در سال ۲۰۱۵ اپیک گیمز موتور بازی‌سازی آنریل را به صورت رایگان برای همگان منتشر ساخت.

آخرین نسخه آنریل به اسم موتور بازی‌سازی آنریل ۵ در سال ۲۰۲۰ معرفی شد. این نسخه از تمام سیستم‌های موجود از جمله کنسول‌های نسل بعدی پلی‌استیشن ۵^۴ و ایکس‌باکس سری X/S^۵ پشتیبانی می‌کند. کار بر روی این موتور حدود دو سال قبل از معرفی آن شروع شده بود. در سال ۲۰۲۱ نسخه‌ای از آن به صورت دسترسی اولیه منتشر شد. به طور رسمی در سال ۲۰۲۲ نسخه‌ی کامل این موتور برای توسعه‌دهندگان انتشار یافت. [۸]

۵-۲ زبان برنامه‌نویسی در آنریل

موتور بازی‌سازی آنریل از زبان برنامه‌نویسی C++ به همراه اسکریپت بصری به نام Blueprint استفاده می‌کند. Blueprint یک سیستم برنامه‌نویسی کامل گیمپلی مبتنی بر مفهوم استفاده از رابط‌های مبتنی بر گره برای ایجاد عناصر گیمپلی از درون ویرایشگر است. این سیستم بسیار منعطف و قدرتمند است زیرا این توانایی را در اختیار طراحان قرار می‌دهد تا از طیف گسترده‌ای از مفاهیم و ابزارها که عموماً فقط در دسترس برنامه‌نویسان هستند استفاده کنند. [۹]

۶-۲ گرافیک کامپیوتری

گرافیک کامپیوتری زیرشاخه‌ای از علوم کامپیوتر است که روش‌های ترکیب دیجیتالی و دستکاری محتوای بصری را مطالعه می‌کند. اغلب اوقات این اصطلاح به مطالعه‌ی کامپیوتر سه‌بعدی اشاره دارد. [۱۰] گرافیک کامپیوتری را می‌توان در موارد مختلفی مانند طراحی رابط کاربری، رندر اشیاء هندسی، انیمیشن و بسیاری موارد دیگر استفاده کرد. ابزارهای مختلفی برای پیاده‌سازی گرافیک کامپیوتری استفاده می‌شوند. یکی از این ابزارها OpenGL است. در این

^۱fixed-function pipeline

^۲shader hardware

^۳GDC

^۴PlayStation 5

^۵Xbox Series X/S

پروژه برای ایجاد محیط گرافیکی از OpenGL استفاده شده است. برای اینکه بتوان شخصیت‌ها و محیط سه‌بعدی را به وضوح مشاهده کرد از یک الگوریتم نورپردازی به نام سایه‌زنی فونگ استفاده شده است. در این بخش توضیح کوتاهی درباره‌ی OpenGL و این روش سایه‌زنی آورده شده است.

۱-۶-۲ OpenGL

OpenGL یک واسط برنامه نویسی کاربردی^۱ است که با فراهم کردن توابع مختلف به توسعه‌دهندگان امکان دستکاری گرافیک و تصاویر را می‌دهد. OpenGL یک کتابخانه‌ی رندرینگ است. یک "شیء" به خودی خود در OpenGL مفهومی ندارد و به صورت مجموعه‌ای از مثلث‌ها و حالات مختلف در نظر گرفته می‌شود. بنابراین وظیفه‌ی ما است که بدانیم چه شیء‌ای در کدام قسمت صفحه رندر شده است. این کتابخانه تنها وظیفه‌اش، کشیدن تصاویری که است که می‌خواهیم به تصویر کشیده شوند. در این صورت اگر می‌خواهیم تصویری را به‌روزرسانی کنیم و یا به عنوان مثال شیء‌ای را تحرک دهیم باید به OpenGL درخواست دهیم که صحنه را دوباره برای ما رندر کند. [۱۱]

به صورت کلی OpenGL را می‌توان یک ماشین حالت بزرگ در نظر گرفت. هر حالت شامل مجموعه‌ای از متغیرها است که نحوه‌ی عملکرد OpenGL را مشخص می‌کند. به مجموعه‌ی این حالت‌ها OpenGL context نیز می‌گویند. در واقع context را می‌توان یک شیء در نظر گرفت که کل OpenGL را دربر می‌گیرد. عموماً تمامی تغییرات، روی context فعلی اعمال می‌شود و سپس رندر می‌شود. [۱۱][۱۲]

۲-۶-۲ سایه‌زنی فونگ

نورپردازی در دنیای واقعی بسیار پیچیده است و به عوامل بسیار زیادی بستگی دارد. با توجه به قدرت محدود پردازش برای ما چنین امکانی وجود ندارد که رفتار آن را به صورت کامل تخمین بزنیم. بنابراین برای نورپردازی محیط‌های سه‌بعدی از تقریب واقعیت با استفاده از مدل‌های ساده‌شده‌ی فیزیکی استفاده می‌شود. یکی از این روش‌ها مدل نورپردازی فونگ نام دارد. این مدل بر اساس سه مولفه‌ی اصلی عمل می‌کند. این سه مولفه، نور محیطی^۲، نور پخش شده^۳ و نور آینه‌وار^۴ نام دارند. [۱۳]

۳-۶-۲ نور محیطی

نور معمولاً از یک منبع نور منفرد ساطع نمی‌شود، بلکه از منابع نوری زیادی که در اطراف ما پراکنده شده‌اند، حتی زمانی که به صورت مستقیم قابل مشاهده نیستند، نشأت می‌گیرد. حتی زمانی که هوا تاریک است، معمولاً هنوز مقداری

^۱ API

^۲ Ambient light

^۳ Diffuse light

^۴ Specular light

نور در جایی در جهان وجود دارد. مانند ماه در هنگام شب. بنابراین اجسام تقریباً هرگز به صورت کامل تاریک نیستند. اگر بخواهیم چنین مولفه‌ای را به صورت واقعی مدل‌سازی کنیم، الگوریتمی بسیار پر هزینه خواهد بود. به همین جهت در مدل فونگ برای اینکه بتوانیم نور محیطی را در اجسام سه بعدی مشاهده کنیم از یک رنگ ثابت کوچک نور استفاده می‌کنیم و آن را به رنگ نهایی هر شی اضافه می‌کنیم. در این صورت به نظر می‌رسد که همیشه مقداری نور پراکنده در محیط سه بعدی وجود دارد. [۱۳]

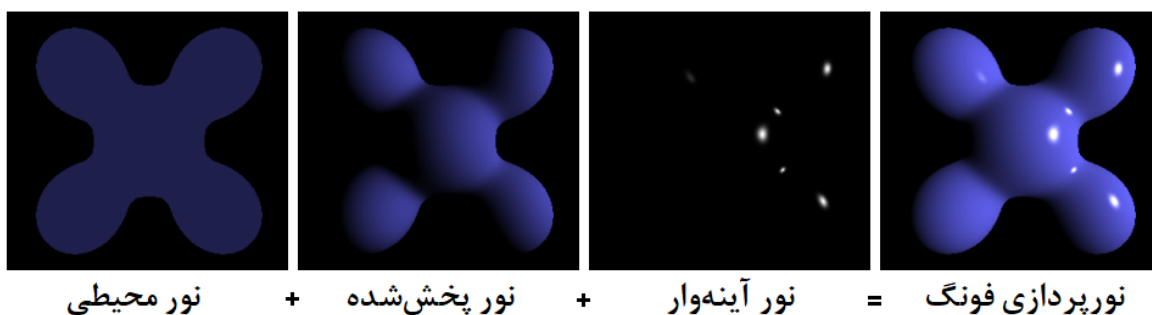
۴-۶-۲ نور پخش شده

می‌دانیم هرچه جسم به یک منبع نور نزدیک تر باشد و هرچه بخشی از یک جسم بیشتر به سمت منبع نور باشد، بیشتر روشن می‌شود. این مولفه، تاثیر جهت قرار گیری اجسام نسبت به منبع نور را شبیه‌سازی می‌کند. [۱۳]

۵-۶-۲ نور آینه‌وار

این مولفه وظیفه‌ی شبیه‌سازی نقطه‌ی روشن نوری که بر روی اجسام براق ظاهر می‌شوند، را دارد. جهت قرار گیری اجسام نسبت به جهت نور تاثیر زیادی بر نحوه‌ی شکل گیری و شمایل این نقطه‌ی روشن شده دارد. همچنین نحوه‌ی عملکرد نورپردازی آینه‌وار رابطه‌ی مستقیمی با جنس سطح و خواص بازتابی آن سطح دارد. هرچه سطح اجسام به جنس آینه‌ای نزدیک شود، نور بیشتری را بازتاب می‌دهد و بالعکس ممکن است جنس آن مانند گچی باشد که نور زیادی را جذب خود می‌کند. [۱۳]

در تصویر زیر می‌توانیم عملکرد تمامی این مولفه‌ها را در مدل فونگ مشاهده کنیم.



نورپردازی فونگ = نور آینه‌وار + نور پخش شده + نور محیطی

شکل ۴-۲ - مولفه‌های الگوریتم سایه‌زنی فونگ [۱۴]

۷-۲ پویانمایی اسکلتی

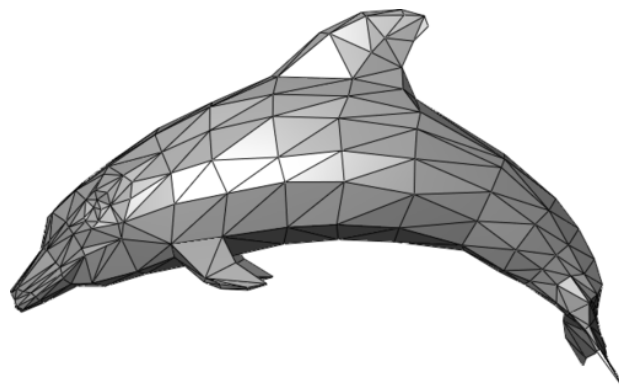
در پویانمایی اسکلتی از مدل‌های اسکلتی استفاده می‌شود. هر مدل اسکلتی از دو بخش مدل و اسکلت تشکیل شده است. برای اینکه بتوانیم پویانمایی اسکلتی را متوجه شویم، نیاز داریم دانش اولیه‌ای نسبت به بعضی از تعاریف گرافیک

کامپیوتری مانند "مش‌های چندضلعی"، "زیرمش"، "ماده" و "بافت" پیدا کنیم. در این بخش، ابتدا به این تعاریف می‌پردازیم. سپس به بررسی مدل اسکلتی پرداخته و پس از آن تعریفی از روش‌ها و تعاریف مربوط به پویانمایی اسکلتی را می‌آوریم.

۱-۷-۲ شبکه‌ی^۱ چندضلعی

در گرافیک کامپیوتری سه‌بعدی و مدل‌سازی جامد، شبکه چندضلعی مجموعه‌ای از رئوس، لبه‌ها و وجوه است که شکل یک جسم چندوجهی را مشخص می‌کند. وجوه معمولاً از مثلث‌ها (شبکه مثلثی)، چهارضلعی‌ها (چهار گوشه)، یا دیگر چندضلعی‌های محدب ساده (n ضلعی‌ها) تشکیل شده‌اند. دلیل استفاده از این نوع چندضلعی‌ها آسان‌تر بودن به نمایش کشیدن آن‌ها در محیط سه‌بعدی است. البته در حالت کلی اشیاء ممکن است از چندضلعی‌های مقعر و یا حتی چندضلعی‌های دارای سوراخ نیز تشکیل شده باشند.

اشیاء ایجاد شده توسط مش‌های چند ضلعی باید انواع مختلفی از عناصر، از جمله رئوس، لبه‌ها، وجوه، چندضلعی‌ها و سطوح را در خود ذخیره کنند. در بسیاری از نرم‌افزارهای سه‌بعدی، فقط رئوس، لبه‌ها و یکی از دو مورد وجوه یا چندضلعی‌ها ذخیره می‌شوند. در اکثر سیستم‌های رندر^۲ فقط از وجوه سه‌ضلعی (مثلث‌ها) استفاده می‌شود. بنابراین در این حالت چندضلعی‌های مدل، باید به شکل مثلث باشند. البته سیستم‌های رندر ای وجود دارند که از چهارضلعی‌ها یا چندضلعی‌های با تعداد اضلاع بالاتر نیز پشتیبانی می‌کنند و یا در لحظه این چندضلعی‌ها را به مجموعه‌ای از مثلث‌ها تبدیل می‌کنند که در این صورت باعث می‌شود نیازی به ذخیره‌ی مش به شکل مثلثی نباشد.



شکل ۲-۵- نمایش یک دلفین به وسیله‌ی مش مثلثی [۱۵]

بنابراین چهار قسمت اصلی یک مش چندضلعی، رئوس، لبه‌ها، وجوه و چندضلعی‌ها هستند. توضیح کوتاهی درباره‌ی هر کدام از این موارد را در زیر می‌توانیم مشاهده کنیم.

¹ Mesh

² renderer

راس

راس‌ها معمولاً یک موقعیت در فضای سه‌بعدی همراه با اطلاعات دیگر مانند رنگ، بردار نرمال و مختصات بافت را شامل می‌شوند. در راس‌های مربوط به مش‌های اسکلتی اطلاعاتی مانند تعداد مفاصلی که بر روی این راس تاثیر می‌گذارند همراه با وزن تاثیرگذاری آن‌ها، می‌تواند اضافه شود.

لبه

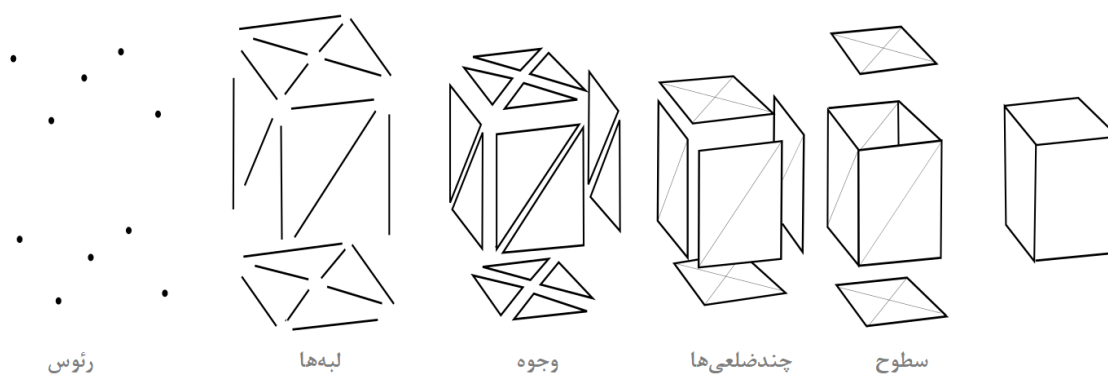
ارتباط بین دو راس را لبه گویند.

وجه

مجموعه‌ای بسته از لبه‌ها را وجه گویند. وجه‌ها می‌توانند از سه لبه (وجه مثلثی) یا از چهار لبه (وجه چهارگوش) تشکیل شده باشند.

چندضلعی

یک چندضلعی مجموعه‌ای همسطح از وجوه است. در سیستم‌هایی که از وجه‌های چند ضلعی پشتیبانی می‌کنند، وجوه و چندضلعی‌ها یکسان هستند ولی در صورتی که سیستم مورد نظر تنها از سه یا چهار ضلعی‌ها پشتیبانی کند، در این صورت به چند ضلعی‌ها، مجموعه‌ای از وجوه گفته می‌شود.



شکل ۲-۶ - عناصر یک مش چندضلعی [۱۵]

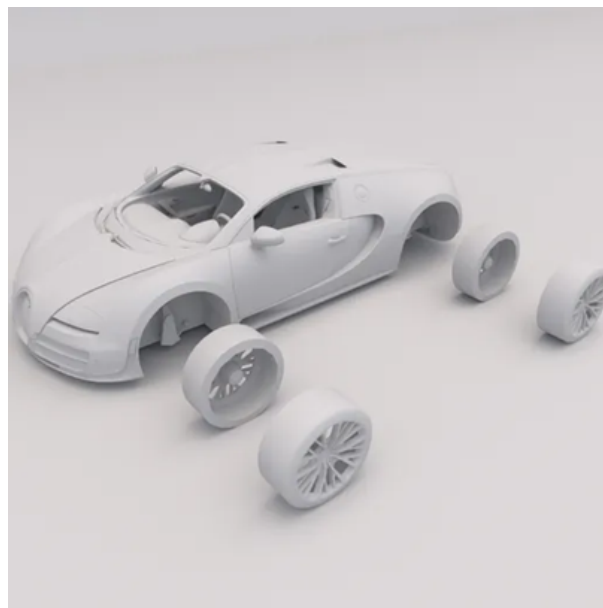
مدل ۲-۷-۲

مدل^۱ در واقع هر شی‌ای است که در محیط سه‌بعدی قرار می‌گیرد و به تصویر کشیده می‌شود. هر مدل می‌تواند از چند زیرمش تشکیل شود. به عنوان مثال یک ماشین را در نظر بگیریم. موجودیت ماشین می‌تواند یک مدل باشد که^۱ گاهی به جای استفاده از واژه‌ی مدل، از واژه‌ی مش هم استفاده می‌شود.

در محیط سه بعدی قرار می گیرد. مدل ماشین می تواند از چند زیرمش مانند چرخ ها، لاستیک ها و بدنه ی ماشین تشکیل شود. دلیل وجود داشتن یک موجودیت کلی به اسم ماشین این است که یک شخصی مانند طراح محیط و یا طراح مرحله نمی خواهد هر بار که ماشینی را در محیط قرار دهد، تک تک زیرمش ها را به صورت دستی در صحنه وارد کند و در سر جای خودش قرار بدهد.

۳-۷-۲ زیرمش^۱

چندضلعی های دارای یک نوع ماده^۲ را یک زیرمش گویند. همانطور که اشاره شد، هر مدل از چند زیرمش تشکیل می شود. دلیل این تقسیم این است که در هر عملیات به تصویر کشیدن^۳ تنها یک ماده می تواند به تصویر کشیده شود. مثلاً در مثال ماشین، قسمت های مختلف ماشین از ماده های مختلفی تشکیل می شود. به طور مثال چرخ ماشین می تواند از جنس آلومینیوم باشد، لاستیک چرخ از جنس پلاستیک باشد و یا حتی قسمت های داخلی ماشین مانند صندلی ماشین، از جنس چرم باشد. بنابراین باید این قسمت ها به صورت جدا قرار گیرند تا بتوان هر قسمت را با توجه به ماده ی موردنظر آن به تصویر کشاند.



شکل ۳-۷-۲- اجزاء مختلف ماشین [۱۶]

^۱Sub-Mesh

^۲Material

^۳Render

۲-۷-۴ ماده^۱

ماده‌ها شامل پارامترهای قابل تنظیمی هستند که با تنظیم آن‌ها، به گرافیک اعلام می‌شود که چگونه باید یک مثلث را به تصویر بکشد. این پارامترها می‌توانند شامل موارد زیر باشند ولی محدود به آن نمی‌شوند

۱. میزان کدورت و شفافیت شی

۲. میزان براقی شی

۳. رنگ (بافت) شی

۴. سایه‌زنی پیکسلی یا راسی^۲



شکل ۲-۸ - چند نوع ماده‌ی مختلف [۱۷]

۲-۷-۵ بافت^۳

بافت یک تصویر دوبعدی و یا سه‌بعدی است که می‌تواند در ماده استفاده شود. این تصاویر به عنوان ورودی در برنامه دریافت شده و پس از اینکه یک شناسه به آن‌ها تخصیص داده شد، در کارت گرافیک قرار می‌گیرند. ماده‌ها با استفاده از این شناسه می‌توانند در صورت لزوم به این بافت‌ها دستیابی پیدا کنند.

¹Material

²Vertex or Pixel shader

³Texture



شکل ۲-۹ - چند نوع بافت مختلف [۱۸]

۶-۷-۲ اسکلت

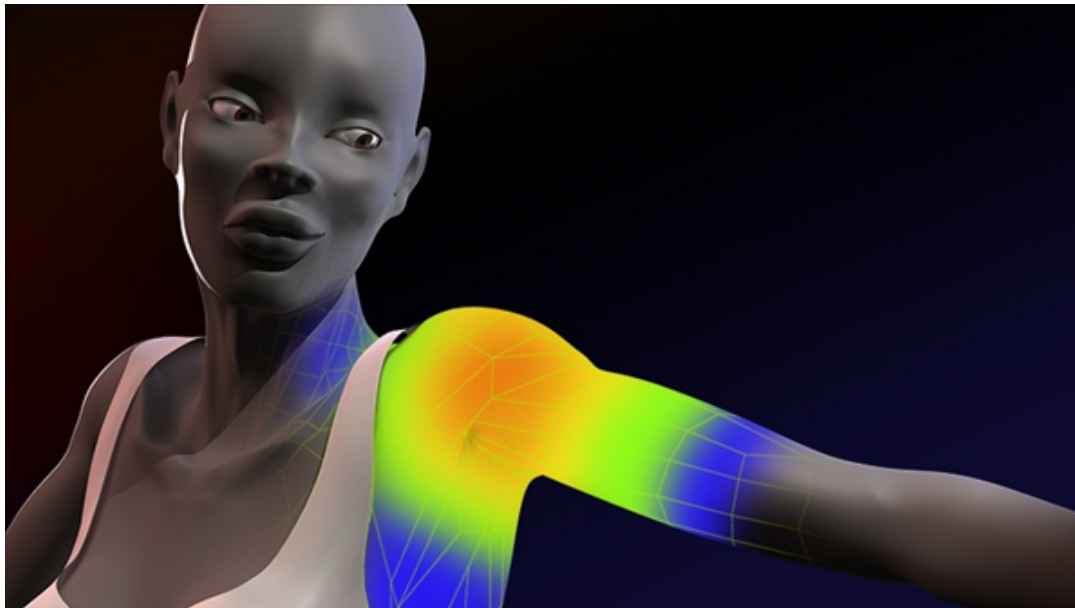
به مجموعه ای از مفصل که به صورت سلسله مراتبی به یکدیگر متصل می شوند، اسکلت گویند. پس از آنکه هنرمندان مدل شخصیت را طراحی می کنند در طی یک مرحله که به آن Rigging گویند، ساختار سلسله مراتبی اسکلت را به وجود می آورند. در پویانمایی اسکلتی درواقع این اسکلت است که حرکت می کند و با حرکتش باعث حرکت مدل شخصیت می شود.



شکل ۲-۱۰ - در سمت راست مدل و سمت چپ اسکلت مدل قابل مشاهده است. [۱۹]

۷-۷-۲ Skinning

تا اینجا با دو مفهوم مدل و اسکلت آشنایی پیدا کردیم ولی نگفتیم که این دو چگونه به هم مرتبط می‌شوند. به عملیاتی که طی آن مفاصل موجود در اسکلت به مدل متصل می‌شود skinning گویند. طی این مرحله هر راس موجود در پوسته‌ی مش به یک یا چند مفصل متصل می‌شود. برای اینکه چگونه رئوس مش، این مفاصل را دنبال کنند الگوریتم‌های مختلفی مطرح شده‌است که در فصل پیاده‌سازی به یکی آن‌ها اشاره خواهد شد.



شکل ۷-۱۱ - عکس نشان‌دهنده‌ی میزان تعلق رئوس اطراف مفصل شانه است. هر چه راس به مفصل نزدیک تر باشد، تعلق بیشتری به آن دارد (رنگ قرمز تری می‌گیرند). [۲۰]

۸-۷-۲ ژست شخصیت

ژست یک شخصیت نشان‌دهنده‌ی نحوه‌ی قرارگیری مفصل‌ها در اسکلت است. ژست‌های مختلف با دروان، حرکت یا تغییر اندازه‌ی مفاصل درون اسکلت به وجود می‌آیند. همانگونه که اشاره شد، اسکلت یک مدل در مرحله‌ی Rigging به وجود می‌آید و در همین مرحله با استفاده از Skinning به مدل متصل می‌شود. زمانی که این عمل صورت می‌گیرد مدل در یک ژست به خصوص قرار دارد که به آن ژست حالت اتصال^۱ یا ژست مرجع^۲ گویند. به صورت کلی شخصیت در این حالت به صورتی ایستاده‌است که پاهایش کمی از هم باز است و بازوهایش به شکل حرف T کشیده است. به همین جهت گاهی به ژست حالت اتصال، ژست T^۳ هم گفته می‌شود. دلیل انتخاب این ژست خاص دور نگه‌داشتن اندام‌ها از بدن است که باعث می‌شود فرایند اتصال رئوس به مفاصل آسان‌تر شود.

^۱ Bind Pose

^۲ Reference Pose

^۳ T Pose

همانگونه که اشاره شد مفاصل به صورت سلسله مراتبی به یکدیگر متصل هستند. یعنی نحوه‌ی قرارگیری آن‌ها متناسب با نحوه‌ی قرارگیری والدشان است. این کار باعث می‌شود که مفاصل به صورت طبیعی حرکت کنند. یعنی در صورتی که والد حرکت کند، به واسطه‌ی آن فرزند نیز حرکت می‌کند. زمانی که ژست شخصیت در این حالت والد، فرزند را قرار دارد به آن ژست محلی^۱ گفته می‌شود. حالت دیگری نیز وجود دارد که موقعیت هر مفصل نسبت به فضای مختصاتی مدل در نظر گرفته می‌شود. به ژست شخصیت در این حالت ژست جهانی^۲ گفته می‌شود.

۹-۷-۲ کلیپ‌های پویانمایی

در یک فیلم پویانمایی شده، تمام بخش‌های یک صحنه قبل از ساخت هر کلیپ پویانمایی به دقت برنامه‌ریزی می‌شود. این شامل حرکات هر شخصیت، لوازم موجود در صحنه و حتی حرکات دوربین نیز می‌شود. این بدان معنی است که کل صحنه را می‌توان به عنوان یک دنباله طولانی و پیوسته از فریم‌ها، متحرک ساخت. در این حالت در صورتی که شخصیت‌ها خارج از دوربین باشند لازم نیست که متحرک شوند.

کلیپ‌های پویانمایی متفاوت از این هستند. یک بازی، یک تجربه‌ی تعاملی است بنابراین نمی‌توان از قبل چگونه حرکت کردن شخصیت‌ها و رفتار آن‌ها را پیش‌بینی کرد. حتی تصمیمات شخصیت‌های غیربازیکن کامپیوتری نیز می‌توانند تابعی از اقدامات غیر قابل پیش‌بینی بازیکن انسانی باشد. به این ترتیب، کلیپ‌های پویانمایی مربوط به بازی تقریباً هیچ‌گاه از مجموعه‌ای از فریم‌های طولانی و به هم پیوسته تشکیل نمی‌شوند. در عوض، حرکت شخصیت بازی باید به تعداد زیادی حرکات ریز تقسیم شود. منظور از کلیپ‌های پویانمایی این حرکات کوتاه و یکتا است.

بنابراین هر کلیپ به صورتی طراحی شده است که یک عمل کاملاً مشخص را انجام دهد. برخی از این کلیپ‌ها به گونه‌ای طراحی شده‌اند که بتوان آن را به صورت حلقه شونده تکرار کرد. به عنوان مثال چرخه‌ی راه رفتن یا دویدن می‌توانند از این نوع کلیپ‌ها باشند و حرکاتی مانند پریدن یا دست تکان دادن از نوعی هستند که تنها یک‌بار پخش می‌شوند.

بنابراین به طور کلی حرکات هر شخصیت بازی معمولاً به هزاران کلیپ تقسیم می‌شود. [۷]

۱۰-۷-۲ ترکیب کلیپ‌های پویانمایی

اصطلاح ترکیب کلیپ‌ها به هر تکنیکی اطلاق می‌شود که در آن بیش از یک کلیپ پویانمایی در ژست نهایی کاراکتر سهیم می‌شود. به صورت دقیق‌تر در این عمل دو یا چند ژست برای ایجاد یک ژست خروجی برای اسکلت شخصیت، با یکدیگر ترکیب می‌شوند. همانطور که در بخش قبل گفته شد، کلیپ‌های پویانمایی، کلیپ‌های کوتاه

^۱Local Pose

^۲Global Pose

و یکتایی هستند. با استفاده از روش ترکیب می‌توان مجموعه‌ای از کلیپ‌های پویانمایی را با یکدیگر ترکیب کرد تا مجموعه‌ی جدیدی از کلیپ‌های پویانمایی را بدون نیاز به ایجاد دستی و از پایه‌ی آن‌ها تولید کنیم.

به عنوان مثال، با ترکیب یک کلیپ راه رفتن آسیب دیده با راه رفتن بدون آسیب دیدگی، می‌توانیم سطوح مختلفی از آسیب دیدگی در هنگام راه رفتن را به وجود آوریم. از ترکیب می‌توان برای درونیابی بین حالات مختلف چهره، حالت‌های مختلف بدن و حالت‌های مختلف حرکتی استفاده کرد. علاوه بر این می‌توان از آن برای یافتن یا حالت میانی بین دو حالت شناخته شده در زمان‌های مختلف نیز استفاده کرد. این کار زمانی استفاده می‌شود که بخواهیم ژست یک شخصیت را در نقطه‌ای از زمان پیدا کنیم که دقیقاً با یکی از فریم‌های نمونه موجود در داده‌های کلیپ پویانمایی مطابقت ندارد. همچنین می‌توانیم از ترکیب موقتی پویانمایی برای انتقال هموار از یک کلیپ به کلیپ دیگر، با ترکیب تدریجی کلیپ پویانمایی مبدا به مقصد در مدت زمان کوتاهی استفاده کنیم.

پیوست‌ها

مراجع

- [1] "Animation wikipedia", <https://en.wikipedia.org/wiki/Animation>.
- [2] Barczak, A. M. and Woźniak, H., "Comparative study on game engines", *Studia Informatica. Systems and Information Technology. Systemy i Technologie Informacyjne*, No. 1-2, 2019.
- [3] "Keyframe animation berkeley", <https://cs184.eecs.berkeley.edu/sp19/lecture/17-28/intro-to-animation-kinematics-mo>.
- [4] Wood DN, Finkelstein A, H. J. T. C. and DH, S., "Multiperspective panoramas for cel animation", *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.
- [5] "Animation cells wikipedia", https://en.wikipedia.org/wiki/File:Animation_cells.png.
- [6] Lasseter, J., "Principles of traditional animation applied to 3d computer animation", *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987.
- [7] Gregory, J., *Game Engine Architecture*, A K Peters/CRC Press, 3rd ed. , 2018.
- [8] "Unreal engine wikipedia", https://en.wikipedia.org/wiki/Unreal_Engine.
- [9] "Unreal engine blueprint", <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/>.
- [10] "Computer graphics wikipedia", [https://en.wikipedia.org/wiki/Computer_graphics_\(computer_science\)](https://en.wikipedia.org/wiki/Computer_graphics_(computer_science)).
- [11] "Using opengl", https://www.khronos.org/opengl/wiki/Getting_Started.
- [12] "Opengl state machine", <https://learnopengl.com/Getting-started/OpenGL>.
- [13] "Phone shading learnopengl", <https://learnopengl.com/Lighting/Basic-Lighting>.
- [14] "Phong shading wikipedia", https://en.wikipedia.org/wiki/Phong_reflection_model.
- [15] "Polygon mesh wikipedia", https://en.wikipedia.org/wiki/Polygon_mesh.
- [16] "Different parts of a car", <https://cults3d.com/en/3d-model/game/bugatti-veyron-printable-car-3d-digital-stl-file>.
- [17] "Different materials", <https://cleverlottery.weebly.com/vray-materials-download.html>.
- [18] "Different textures", <https://www.pinterest.com/pin/56787645294557990/>.
- [19] "Skeletal model", <https://www.fudgeanimation.com/experiments/top-5-rigging-tips/>.
- [20] "Skinning", <https://www.pluralsight.com/blog/film-games/understanding-skinning-vital-step-rigging-project>.

Analysis of the animation graph in Unreal Engine and implementation of an animation system using OpenGL

Nami Naziri

nami.naziri@yahoo.com

May 22, 2022

Department of Electrical and Computer Engineering

Isfahan University of Technology, Isfahan 84156-83111, Iran

Degree: Bachelor of Science

Language: Farsi

Supervisor: Maziar Palhang, Assoc. Prof., palhang@cc.iut.ac.ir.

Abstract

Keywords



Isfahan University of Technology

Department of Electrical and Computer Engineering

Analysis of the animation graph in Unreal Engine and implementation of an animation system using OpenGL

A Thesis

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science

By

Nami Naziri

Evaluated and Approved by the Thesis Committee, on May 22, 2022

- 1- Maziar Palhang, Assoc. Prof. (Supervisor)
- 2- First Examiner, Assoc. Prof. (Examiner)
- 3- First Examiner, Assist. Prof. (Examiner)

Department Graduate Coordinator: Reza Tikani, Assist. Prof.