
بسم الله الرحمن الرحيم

دانشگاه صنعتی اصفهان

مهندسی برق و کامپیوتر

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

پایان‌نامه کارشناسی مهندسی کامپیوتر

نامی نذیری

استاد راهنما

دکتر مازیار پالهنگ

۱۴۰۱

دانشگاه صنعتی اصفهان

مهندسی برق و کامپیوتر

پایان‌نامه کارشناسی رشته مهندسی کامپیوتر آقای نامی نذیری

تحت عنوان

بررسی سیستم گراف انیمیشن در موتور بازی سازی آنریل و و پیاده‌سازی 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 پایان‌نامه	دکتر مازیار پالهنگ
۲- استاد داور	دکتر داور اول
۳- استاد داور	دکتر داور دوم
سرپرست تحصیلات تکمیلی دانشکده	دکتر تحصیلات تکمیلی دانشکده

فهرست مطالب

عنـــوان **صفحه**

فهرست شکل‌ها

عنـــوان **صفحه**

فهرست جدول‌ها

عنـــوان **صفحه**

فهرست الگوریتم‌ها

چکیده

کلمات کلیدی

فصل اول
سیستم گراف پویانمایی در موتور بازی‌سازی آنریل

موتور آنریل مجموعه کاملی از ابزارهای تولید محتوا برای توسعه‌ی بازی، مصورسازی معماری و خودرو، ایجاد محتوا برای فیلم و تلوزیون، پخش رویدادهای زنده، آموزش، شبیه‌سازی و ساید برنامه‌های بلادرنگ است.

این موتور برای اولین بار برای توسعه‌ی بازی "غیرواقعی" در سال ۱۹۹۸ توسعه پیدا کرد. پس از آن نسخه‌های متعددی از این موتور منتشر شده است. [۱]

موتور آنریل مانند تمامی موتورهای بازی‌سازی دارای مولفه‌های فراوانی است که برای تولید بازی به کار می‌رود. مولفه‌های انیمیشن، هوش مصنوعی، رندر، رابط کاربری تنها تعداد اندکی از مولفه‌هایی است که می‌توان در آنریل استفاده کرد.

آنریل طیف گسترده‌ای از ابزارهای قدرتمند را برای مدیریت شخصیت‌ها، ایجاد محتوای سینمایی و انیمیشن‌سازی را ارائه می‌دهد. با استفاده از سیستم پویانمایی مش اسکلتی، کاربران می‌توانند شخصیت‌ها، اسکلت‌ها و کلیپ‌های پویانمایی وارد شده خود را مدیریت کنند. سپس این محتوا می‌تواند برای ایجاد گیم‌پلی تعاملی پویا شده با استفاده از ویژگی‌های مختلف مانند فضاهای ترکیبی [۱]، طرح‌های پویانمایی [۲] و ماشین‌های حالت [۳] استفاده شود. سکانس‌های

[1] Blend Spaces
[2] Animation Blueprint
[3] State Machines

سینمایی را می‌توان با استفاده از ابزار Sequencer ایجاد کرد. با استفاده از این ابزار می‌توان دوربین‌ها و شخصیت‌ها را متحرک ساخت. انیمیشن‌سازی شخصیت‌ها را می‌توان با استفاده از Control Rig که ابزار داخلی موتور آنریل است، انجام داد. با استفاده از این ابزار می‌توان ریگ‌های مناسبی ساخت تا درون Sequencer شخصیت را متحرک ساخت. [۲]

همانطور که مشخص است، سیستم پویانمایی موجود در موتور آنریل بسیار گسترده است. در این پروژه قسمت طرح پویانمایی آنریل که به گراف پویانمایی نیز شناخته می‌شود بررسی می‌شود.

برای اینکه بتوانیم در مورد سیستم گراف پویانمایی آنریل توضیح دهیم، ابتدا لازم است توضیحاتی را درباره‌ی نحوه‌ی معماری این انجین بیاوریم. بنابراین در این بخش ابتدا توضیح کوتاهی درباره‌ی معماری آنریل با محوریت نحوه‌ی رابطه‌ی اشیا با یکدیگر داده و پس از آن به بررسی ویژگی‌های سیستم گراف پویانمایی می‌پردازیم.

۱-۱ بازیگران، پیاده‌ها و شخصیت‌ها

اشیا در آنریل می‌توانند به سه کلاس کلی بازیگران [۱]، پیاده‌ها [۲] و شخصیت‌ها [۳] دسته‌بندی می‌شوند.

بازیگران کلاس پایه‌ی تمامی اشیا ای هستند که به صورت فیزیکی می‌توانند در محیط سه‌بعدی قرار گیرند. پیاده‌ها کلاسی مشتق شده از بازیگران هستند که بازیکنان می‌توانند کنترل آن‌ها را بدست گیرند و در محیط حرکت کنند. در نهایت شخصیت‌ها پیاده‌هایی هستند که دارای مش اسکلتونی، توانایی شناسایی برخورد و منطق حرکتی هستند. آنها مسئول تمام تعاملات فیزیکی بین بازیکن یا هوش مصنوعی، با جهان هستند و همچنین مدل های اولیه شبکه و دریافت ورودی را پیاده سازی می کنند. اگر بخواهیم شخصیت درون بازی از انیمیشن‌های اسکلتونی استفاده کند، باید از این کلاس بهره ببریم.

۱-۲ اجزاء

اجزاء [۴] مجموعه‌ای از توابع و ویژگی‌ها است که می‌تواند به یک بازیگر اضافه شود. بنابراین بازیگران می‌توانند حاوی مجموعه‌ای از ActorComponents باشند که این اجزاء می‌توانند برای موارد مختلفی از جمله کنترل نحوه‌ی حرکت بازیگران، نحوه‌ی رندر شدن و غیره استفاده شوند.

زمانی که یک مولفه به یک بازیگر اضافه می‌شود، آن بازیگر می‌تواند عملکردهای موجود در آن مولفه را استفاده کند. به عنوان مثل یک مولفه نور نقطه‌ای باعث می‌شود که بازیگر مانند یک نور نقطه‌ای، نور ساطع کند. یا یک مولفه

[1] Actors
[2] Pawns
[3] Characters
[4] Components

صوتی به بازیگر این توانایی پخش صدا را می‌دهد.

مولفه‌ها حتما باید به یک بازیگر متصل شوند و به خودی خود نمی‌توانند وجود داشته باشند. درواقع وقتی ما مولفه‌های مختلف را به بازیگر خود متصل می‌کنیم، در حال قرار دادن قطعه‌ها و تکه‌هایی هستیم که مجموع آن‌ها یک بازیگر را به عنوان یک موجودیت واحد که در محیط سه‌بعدی قرار می‌گیرد، تعریف می‌کنند. به عنوان مثال چرخ‌های یک ماشین، فرمان ماشین، چراغ‌ها و غیره همه به عنوان مولفه‌های ماشین درنظر گرفته می‌شوند در حالی که خود آن ماشین، بازیگر است.

۱-۳ شخصیت‌ها

هر شخصیت در آنریل از سه مولفه‌ی اصلی تشکیل شده است.

- Skeletal Mesh Component
- Character Movement Component
- Capsule Component

مولفه‌ی Skeletal mesh Component شامل طرح پویانمایی شخصیت است. طرح پویانمایی، سیستم پویانمایی شخصیت است که جلوتر آن را توضیح می‌دهیم.

مولفه‌ی Character Movement Component همانطور که از اسمش مشخص است برای منطق حرکت در حالت‌های مختلف از جمله راه‌رفتن افتادن و غیره استفاده می‌شود. این مولفه شامل تنظیمات و عملکردهای مربوطه برای کنترل حرکت است.

و در نهایت مولفه‌ی Capsule Component وظیفه‌ی تشخیص برخورد در هنگام حرکت را دارد.

۱-۴ مولفه‌ی مش اسکلتی

این مولفه، مولفه‌ای است که به شخصیت امکان پویا شدن را می‌دهد. این کلاس برای ساختن یک نمونه از کلاس SkeletalMesh است که بر روی آن کلیپ‌های پویانمایی اجرا می‌شوند. اینکه چه کلیپ پویانمایی بر روی آن اجرا شود از طریق کلاس AnimInstance که همان طرح پویانمایی [1] است، انتخاب می‌شود.

همانطور که در فصل گذشته اشاره شد، مش اسکلتی شامل یک هندسه‌ی چندضلعی است که به یک اسکلت که در واقع سلسله مراتبی از مفاصل است، متصل است و این اسکلت می‌تواند به منظور تغییر شکل آن هندسه‌ی چندضلعی یا مش، متحرک شود.

[1] Animation Blueprint

مش‌های اسکلتی از دو قسمت ساخته‌شده‌اند. مجموعه‌ای از چندضلعی‌ها که به منظور تشکیل سطح مش با یکدیگر ترکیب می‌شوند و یک اسکلت سلسله‌مراتبی که می‌تواند برای متحرک‌سازی چندضلعی‌ها استفاده شود. مدل‌های سه‌بعدی، اسکلت و کلیپ‌های پویانمایی در یک برنامه مدل‌سازی و ایجاد پویانمایی مانند Maya، 3DSMax و ابزارهای مدل‌سازی دیگر ایجاد می‌شوند.

در آنریل کلاس SkeletalMesh وظیفه‌ی نگهداری این مش اسکلتی را دارد.

همانگونه که گفتیم، نیاز داریم تا یک سیستمی داشته باشیم که بتواند کلیپ‌های پویانمایی را بر روی این مش اسکلتی اجرا کند. به زبانی دیگر، این مش اسکلی را پویا و متحرک سازد. در آنریل کلاس AnimInstance وظیفه‌ی این عمل را دارد.

۱-۵ طرح پویانمایی

طرح پویانمایی یک طرح تخصصی است که پویانمایی یک مش اسکلتی را کنترل می‌کند. با ویرایش گراف‌های موجود در این طرح، میتوان کارهای مختلفی را روی پویانمایی شخصیت انجام داد. به عنوان مثال می‌توان کلیپ‌های مختلف را با یکدیگر ترکیب کرد، مستقیما مفاصل درون اسکلت را کنترل کرد و یا هر تنظیمات منطقی‌ای که باعث تعریف ژست نهایی شخصیت در فریم فعلی می‌شود را انجام داد.

دو جزء اصلی در طرح پویانمایی وجود دارد که با هم کار می‌کنند تا ژست نهایی شخصیت را برای هر فریم ایجاد کنند. این دو مولفه، گراف رویداد و گراف پویانمایی نام دارند.

به صورت کلی گراف رویداد مقادیری را که در گراف پویانمایی استفاده می‌شوند را به‌روزرسانی می‌کند تا در ماشین‌های حالت، فضاهای ترکیب و بقیه‌ی نودهایی که در گراف پویانمایی استفاده می‌شوند، به کار روند.

۱-۶ گراف رویداد

درون هر طرح انیمیشن، یک گراف رویداد وجود دارد. این گراف برای دریافت مقادیر منطقی از بخش گیمپلی و منطق بازی به‌کار می‌رود. به عنوان مثال اینکه شخصیت می‌خواهد به چه سمتی حرکت کند، یا اینکه چه سرعتی دارد را از طریق این گراف در متغیرهایی که تعریف می‌کنیم، ذخیره می‌کنیم. [۳]

۱-۷ گراف پویانمایی

گراف پویانمایی برای ارزیابی ژست نهایی مش اسکلتی در فریم فعلی استفاده می‌شود. به صورت کلی هر طرح پویانمایی دارای یک گراف پویانمایی است که این گراف شامل گره‌های مختلفی است که هر کدام از این گره‌ها

استفاده‌های متفاوتی دارند. به عنوان مثال می‌توان از این گره‌ها برای نمونه‌برداری [1] از دنباله‌های کلیپ‌های پویانمایی، انجام ترکیب‌های بین کلیپ‌ها یا کنترل تبدیل‌های مربوط به مفاصل استفاده کرد. سرانجام ژست نهایی بدست آمده روی مش اسکلتی در پایان هر فریم اعمال می‌شود.

۱-۸ گره‌های گراف پویانمایی

گراف پویانمایی با ارزیابی گره‌های موجود در گراف، عمل می‌کند. بعضی از گره‌های موجود در گراف، عملیات‌های خاصی را بر روی یک یا چند ژست ورودی انجام می‌دهند، در حالی که برخی دیگر برای دسترسی یا نمونه‌برداری از انوع دیگری از دارایی‌ها مانند فضاهای حالت [2]، مونتاژهای پویانمایی [3] و دنباله‌های پویانمایی [4] استفاده می‌شوند. ماشین‌های حالت نیز که حاوی شبکه‌ی نموداری خودشان هستند، می‌توانند به صورت تنهایی یا با ترکیب با یکدیگر در گراف پویانمایی استفاده شوند.

در این بخش ابتدا با نحوه‌ی جریان اجرا در گراف پویانمایی آشنا می‌شویم، سپس ساختار کلی گره‌های موجود را بررسی کرده و در نهایت به بررسی انواع گره‌های موجود در گراف پویانمایی می‌پردازیم.

۱-۹ ساختار گره‌های گراف پویانمایی

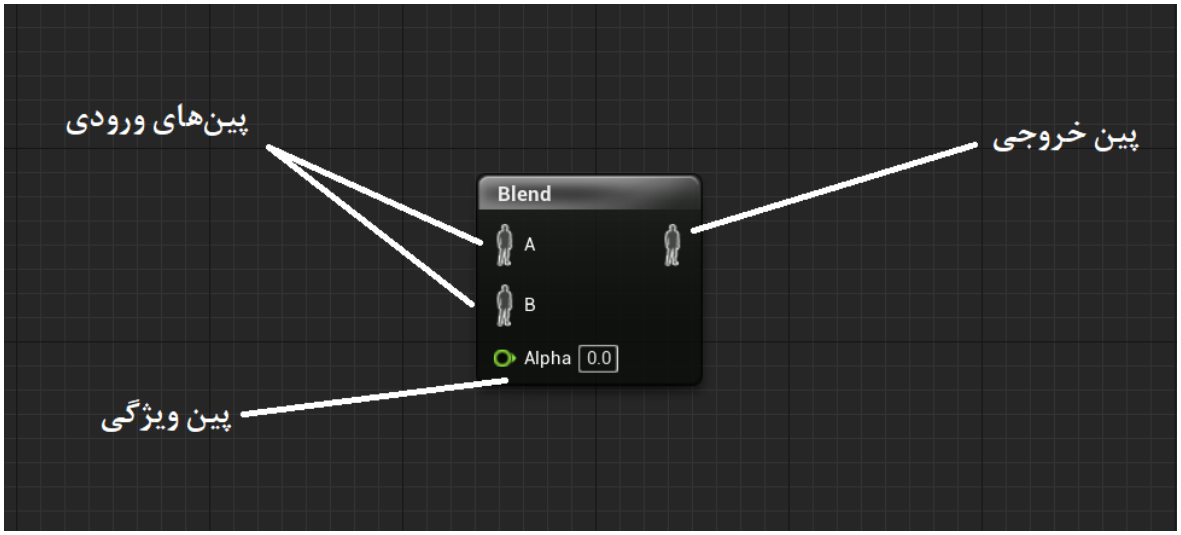
گره‌ها می‌توانند شامل چند پین ورودی که درواقع ژست‌های ورود هستند، باشند. به صورت کلی پین‌ها شامل یک خروجی هستند که این خروجی نشان‌دهنده‌ی ژست شخصیت پس از انجام عملیات‌های مربوط به آن پین است. همچنین می‌توانند شامل پین‌های ویژگی باشند. مقادیر این پین‌های ویژگی از متغیرهایی که در گراف رویداد تعریف و مقداردهی شده‌اند، می‌توانند بدست آیند.

قابل ذکر است با انتخاب هر نود می‌توان به پنل جزئیات آن نود هم دسترسی پیدا کرد که به وسیله‌ی آن می‌توان تنظیمات لازم را بر روی آن نود انجام داد.

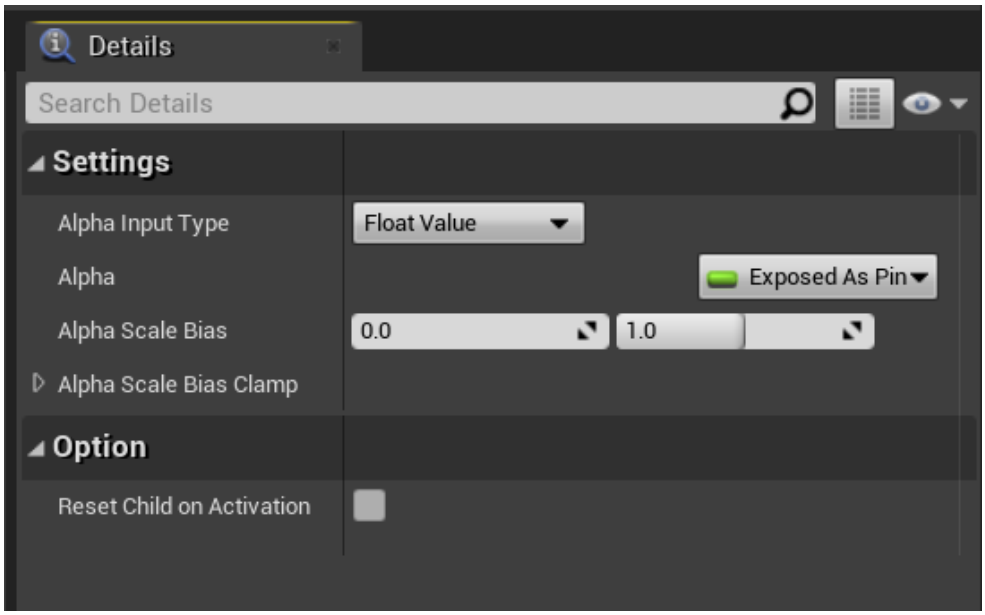
۱-۱۰ جریان اجرا در گراف پویانمایی

تمامی گراف‌ها دارای یک جریان اجرا هستند که به صورت پیوندهای ضربانی میان ورودی و خروجی پین‌ها قابل مشاهده هستند. این جریان‌ها درواقع نحوه‌ی حرکت داده را در گراف ترسیم می‌کنند. در گراف انیمیشن، این جریان نشان‌دهنده‌ی ژست‌هایی است که از یک گره به گره‌ی دیگر منتقل می‌شود. در برخی از گره‌ها مانند گره‌ی ترکیب،

[1] Sampling
[2] Blend Spaces
[3] Animation Montages
[4] Animation Sequence



شکل ۱-۱ - ساختار کلی نودهای موجود در گراف پویانمایی



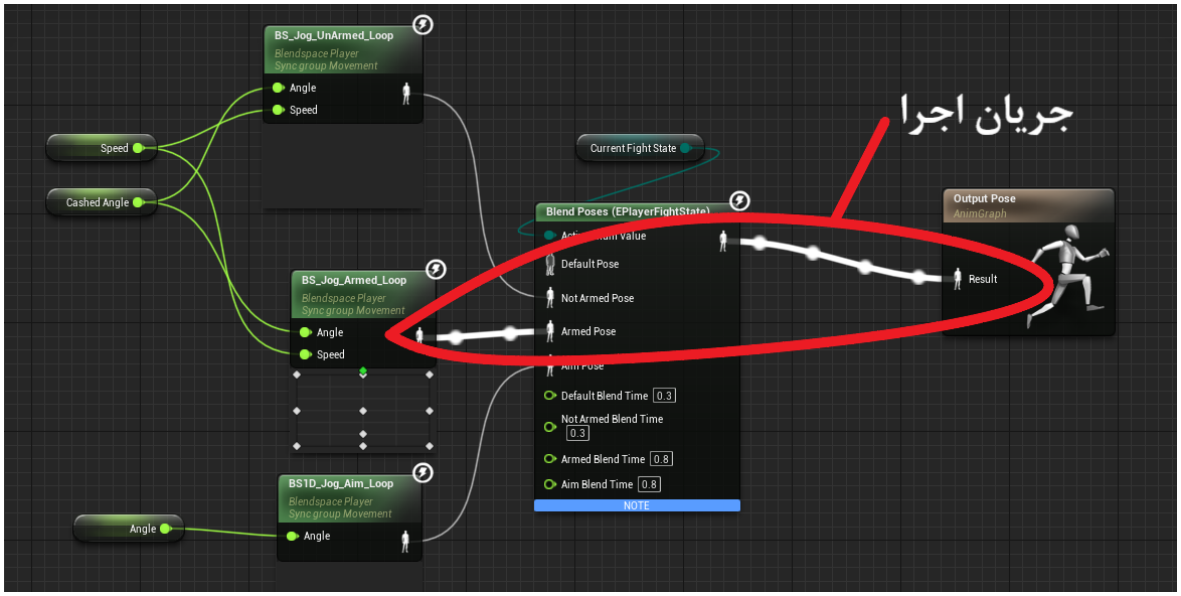
شکل ۱-۲ - پنل تنظیمات گره‌ی ترکیب

ورودی‌های متعددی وجود دارند و به صورت درونی با مقادیری که در متغیرها داریم تصمیم می‌گیرند که کدام یک از ورودی‌ها در حال حاضر بخشی از جریان اجرا است.

[۴]

۱-۱۱ دنباله‌ی پویانمایی

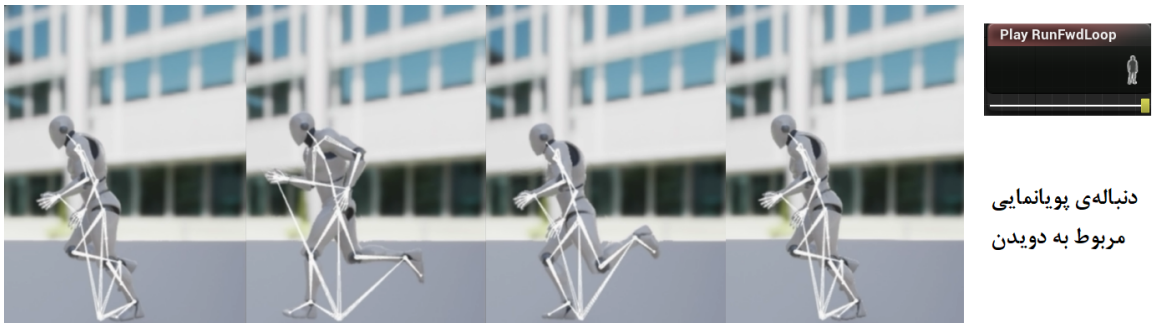
کلیپ‌های پویانمایی در آنریل به اسم دنباله‌ی پویانمایی شناخته می‌شوند. دنباله‌ی پویانمایی یک دارایی پویانمایی است که حاوی داده‌های پویانمایی است که می‌تواند روی یک مش اسکلتی پخش شود تا شخصیت مربوط به آن



شکل ۱-۳ - نمونه‌ای از جریان اجرا

اسکلت را متحرک سازد. یک دنباله‌ی پویانمایی شامل فریم‌های کلیدی هستند که این فریم‌های کلیدی بیانگر موقعیت [۱]، دوران [۲]، مقیاس [۳] اسکلت مش در نقطه‌ی خاصی از زمان است. بنابراین کلیپ‌های انیمیشنی یکی از گره‌های مهم در گراف انیمشن حساب می‌آیند.

قابل ذکر است این گره مانند بقیه‌ی گره‌ها دارای تنظیماتی است که در پنل تنظیمات قابل مشاهده هستند. به عنوان مثال می‌توان سرعت حرکت کلیپ را در این تنظیمات مشخص کرد یا اینکه کلیپ به صورت حلقه‌وار تکرار شود یا خیر.



شکل ۱-۴ - نمونه‌ای از گره‌ی دنباله‌ی پویانمایی

[1] Position
[2] Rotation
[3] Scale

فصل دوم
معماری موتور بازی‌سازی آنریل

در این فصل ابتدا به معماری موتور آنریل پرداخته شده و پس از آن درباره‌ی ابزارهایی که این انجین در اختیار ما می‌گذارد صحبت می شود.

۲-۱ UObject و Actors

کلاس پایه برای تمامی کلاس‌های دیگر در موتور آنریل UObject است. شئ‌ها [1] نمونه‌هایی از کلاس‌هایی هستند که از UObject ها ارث می‌برند. بازیگران [2] نمونه‌هایی از کلاس‌هایی هستند که از. AActor ارث برده اند. کلاس AActor کلاس پایه برای تمامی اشیائی است که می‌توانند در جهان بازی قرار گیرند. به صورت کلی، بازیگران را می‌توان به عنوان یک کل یا موجودیت در نظر گرفت و اشیاء را قطعات تخصصی‌ای درنظر گرفت که در این موجودیت به کار می‌روند که به آن ها جزء [3] می‌گویند. بنابراین اجزاء یک نوع خاصی از اشیاء هستند که بازیگران می‌توانند آن ها را به عنوان یک زیرشئ [4] به خود متصل کنند.

به عنوان مثال اگر یک ماشین را درنظر بگیریم. ماشین به عنوان یک موجودیت کلی به عنوان بازیگر درنظر گرفته

[1] Objects
[2] Actors
[3] Component
[4] sub-object

می‌شود. در صورتی که قسمت‌های مختلف این ماشین مانند درِ ماشین یا چرخ ماشین اجزای آن ماشین درنظر گرفته می‌شوند. در ادامه این مثال، اگر کاربر قرار باشد که این ماشین را کنترل کند، یک جزء دیگر می‌تواند مسئولیت تغییر سرعت و جهت ماشین بر اساس ورودی کاربر را داشته باشد. [۵، ۶]

۲-۲ Components

همانطور که گفته‌شد، اجزاء نوع خاصی از اشیاء هستند که بازیگران می‌توانند به عنوان اشیاء فرعی به خود متصل کنند. کلاس پایه برای تمامی اجزاء، کلاس UActorComponent است. از آنجایی که استفاده از اجزاء تنها راه ممکن برای پرداخت [۱] مش‌ها [۲]، تصاویر ،پخش صداو درواقع هرچیزی که بازیکن هنگام بازی در جهان مشاهده یا تعامل می‌کنند هستند، بنابراین در نهایت از انواعی از این نوع اجزاء در توسعه‌ی بازی استفاده می‌شود.

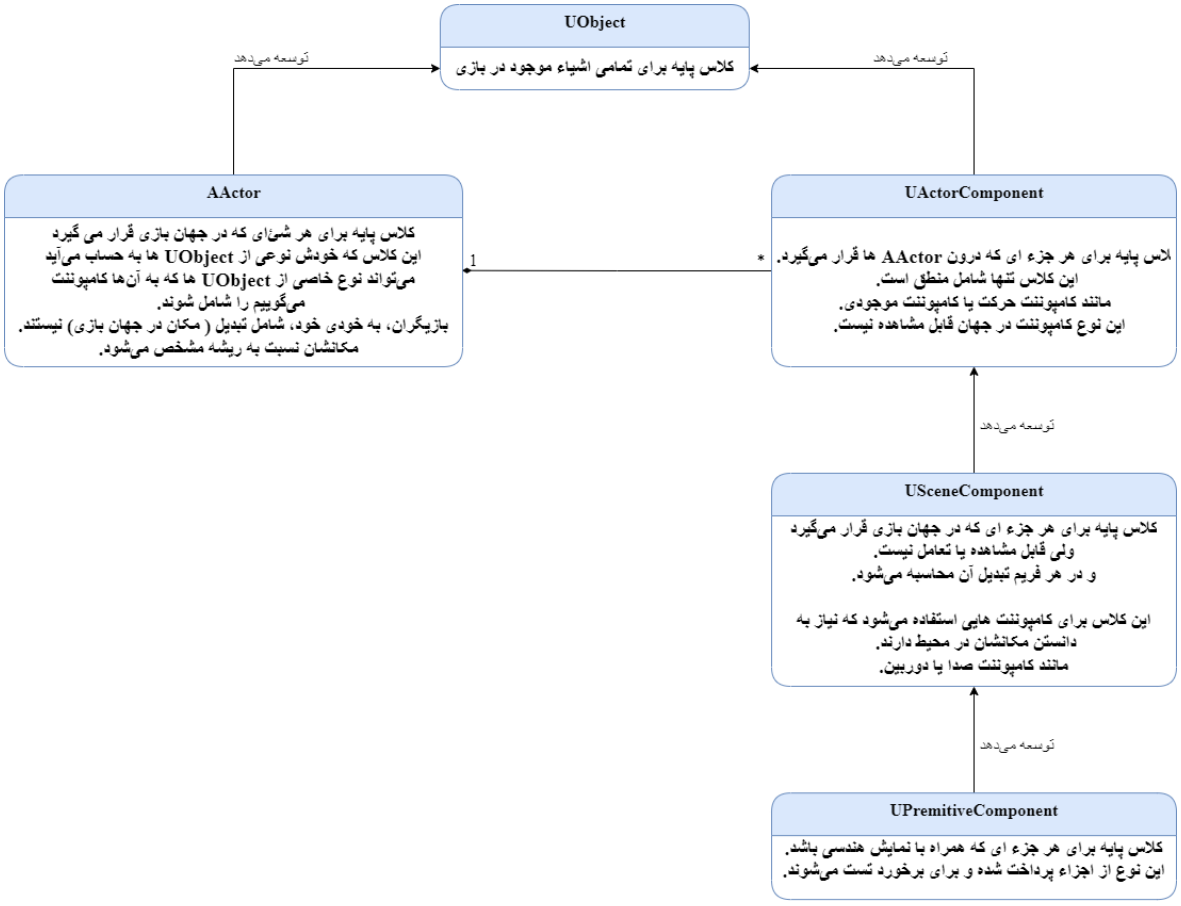
برای ساخت اجزاء، چند کلاس اصلی وجود دارد که در هنگام ایجاد اجزاء باید به آن توجه کرد.

- اجزای بازیگر[۳]: این کلاس بیشتر برای رفتارهای انتزاعی مانند حرکت، مدیریت موجودی یا ویژگی و سایر مفاهیم غیرفیزیکی مفید هستند. این نوع از اجزاء هیچ گونه مکان فیزیکی یا چرخشی در جهان ندارد.
- اجزای صحنه [۴]: این کلاس فرزند کلاس اجزای بازیگر است و از رفتارهای مبتنی بر مکان پشتیبانی می‌کند که به نمایش هندسی نیاز ندارند. این کلاس می‌تواند شامل بازوهای فنری، دوربین‌ها، نیروهای فیزیک و حتی صدا شود.
- اجزای اولیه [۵]: این کلاس فرزند کلاس اجزای صحنه است. درواقع این کلاس همان کلاس اجزای صحنه، همراه با نمایش هندسی است که عموما برای نمایش عناصر بصری و برخورد [۶] یا همپوشانی [۷] با اشیاء فیزیکی استفاده می‌شود. این کلاس می‌تواند شامل مش‌های استاتیک [۸] یا اسکلتی [۹]، اسپرایت‌ها یا بیلبوردها، سیستم‌های ذرات [۱۰] و همچنین حجم برخورد [۱۱] جعبه، کپسول و کره شود.

[۶]

---

[1] Render
[2] Meshes
[3] ActorComponent
[4] SceneComponent
[5] Primitive Components
[6] collide
[7] overlap
[8] static mesh
[9] skeletal mesh
[10] particle systems
[11] collision volumes



شکل ۲-۱ - تصویر UML کلاس‌های پایه‌ی موتور بازی‌سازی آنریل

پیوست‌ها

مراجع

[1] “Unreal engine wikipedia”, https://en.wikipedia.org/wiki/Unreal_Engine.

[2] “Unreal engine animation”, https://docs.unrealengine.com/5.0/en-US/animating-characters-and-objects-in-unreal-engine/.

[3] “Unreal engine event graph”, https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/AnimBlueprints/EventGraph/.

[4] “Unreal engine animation graph”, https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/AnimBlueprints/AnimGraph/.

[5] “Unreal engine architecture”, https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/.

[6] “Unreal engine components”, https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/Actors/Components/.

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.