

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

صفحه	عنوان
شش	فهرست مطالب
هفت	فهرست شکل‌ها
هشت	فهرست جدول‌ها
نه	فهرست الگوریتم‌ها
۱	چکیده
۲	فصل اول: سیستم گراف پویانمایی در موتور بازی‌سازی آنریل
۳	۱-۱ بازیگران، پیاده‌ها و شخصیت‌ها
۳	۲-۱ اجزاء
۴	۳-۱ شخصیت‌ها
۴	۴-۱ مولفه‌ی مش اسکلتی
۵	۵-۱ طرح پویانمایی
۵	۶-۱ گراف رویداد
۵	۷-۱ گراف پویانمایی
۶	۸-۱ معماری سیستم پویانمایی آنریل
۷	فصل دوم: معماری موتور بازی‌سازی آنریل
۷	۱-۲ UObject و Actors
۸	۲-۲ Components
۱۰	پیوست‌ها
۱۱	مراجع

فهرست شکل ها

صفحه

عنوان

شکل ۱-۲: تصویر UML کلاس های پایه ی موتور بازی سازی آنریل ۹

فهرست جدول‌ها

صفحه

عنوان

فهرست الگوریتم‌ها

چکیده

کلمات کلیدی

فصل اول

سیستم گراف پویانمایی در موتور بازی سازی آنریل

موتور آنریل مجموعه کاملی از ابزارهای تولید محتوا برای توسعه ی بازی، مصورسازی معماری و خودرو، ایجاد محتوا برای فیلم و تلویزیون، پخش رویدادهای زنده، آموزش، شبیه سازی و سایر برنامه های بلادرننگ است. این موتور برای اولین بار برای توسعه ی بازی "غیرواقعی" در سال ۱۹۹۸ توسعه پیدا کرد. پس از آن نسخه های متعددی از این موتور منتشر شده است. [۱]

موتور آنریل مانند تمامی موتورهای بازی سازی دارای مولفه های فراوانی است که برای تولید بازی به کار می رود. مولفه های انیمیشن، هوش مصنوعی، رندر، رابط کاربری تنها تعداد اندکی از مولفه هایی است که می توان در آنریل استفاده کرد.

آنریل طیف گسترده ای از ابزارهای قدرتمند را برای مدیریت شخصیت ها، ایجاد محتوای سینمایی و انیمیشن سازی را ارائه می دهد. با استفاده از سیستم پویانمایی مش اسکلتی، کاربران می توانند شخصیت ها، اسکلت ها و کلیپ های پویانمایی وارد شده خود را مدیریت کنند. سپس این محتوا می تواند برای ایجاد گیم پلی تعاملی پویا شده با استفاده از ویژگی های مختلف مانند فضاهای ترکیبی^۱، طرح های پویانمایی^۲ و ماشین های حالت^۳ استفاده شود. سکانس های

^۱Blend Spaces

^۲Animation Blueprint

^۳State 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 مدل‌های اولیه شبکه و دریافت ورودی را پیاده‌سازی می‌کنند. اگر بخواهیم شخصیت درون بازی از انیمیشن‌های اسکلتونی استفاده کند، باید از این کلاس بهره ببریم.

۲-۱ 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 مثال یک مولفه نور نقطه‌ای باعث می‌شود که بازیگر مانند یک نور نقطه‌ای، نور ساطع کند. یا یک مولفه

^۱Actors

^۲Pawns

^۳Characters

^۴Components

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. در واقع 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 در نظر گرفته می‌شوند در حالی که خود آن ماشین، بازیگر است.

۳-۱ 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.

۴-۱ 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^۱ است، انتخاب می‌شود.

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

^۱ Animation Blueprint

مش‌های اسکلتی از دو قسمت ساخته شده‌اند. 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.

۵-۱ 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، مستقیماً 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، فضاها و ترکیب و بقیه‌ی نودهایی که در گراف پویانمایی استفاده می‌شوند، به کار روند.

۶-۱ گراف رویداد

درون هر طرح انیمیشن، یک گراف رویداد وجود دارد. این گراف برای دریافت مقادیر منطقی از بخش گیمپلی و منطق بازی به کار 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]

۷-۱ 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^۱ 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.

[۴]

۸-۱ معماری سیستم پویانمایی آنریل

سیستم انیمیشنی در آنریل در دو بخش جدا از هم کار می‌کند. بخش اول مربوط به مش اسکلتی است. این بخش نشان دهنده‌ی

^۱Sampling

فصل دوم

معماری موتور بازی سازی آنریل

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

۱-۲ UObject و Actors

کلاس پایه برای تمامی کلاس های دیگر در موتور آنریل UObject است. شیء ها^۱ نمونه هایی از کلاس هایی هستند که از UObject ها ارث می برند. بازیگران^۲ نمونه هایی از کلاس هایی هستند که از AActor ارث برده اند. کلاس AActor کلاس پایه برای تمامی اشیائی است که می توانند در جهان بازی قرار گیرند. به صورت کلی، بازیگران را می توان به عنوان یک کل یا موجودیت در نظر گرفت و اشیاء را قطعات تخصصی ای در نظر گرفت که در این موجودیت به کار می روند که به آن ها جزء^۳ می گویند. بنابراین اجزاء یک نوع خاصی از اشیاء هستند که بازیگران می توانند آن ها را به عنوان یک زیرشیء^۴ به خود متصل کنند.

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

¹ Objects

² Actors

³ Component

⁴ sub-object

می‌شود. در صورتی که قسمت‌های مختلف این ماشین مانند در ماشین یا چرخ ماشین اجزای آن ماشین در نظر 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 گفته شد، 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 صدا و در واقع هر چیزی 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 فتری، دوربین‌ها، نیروهای فیزیک و حتی صدا شود.

□ اجزای اولیه^۵: این کلاس فرزند کلاس اجزای صحنه است. در واقع این کلاس همان کلاس اجزای صحنه، همراه با نمایش هندسی است که عموماً 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 بیلبردها، سیستم‌های ذرات^{۱۰} و همچنین حجم برخورد^{۱۱} جعبه، کپسول و کره شود.

[۶]

¹Render

²Meshes

³ActorComponent

⁴SceneComponent

⁵Primitive Components

⁶collide

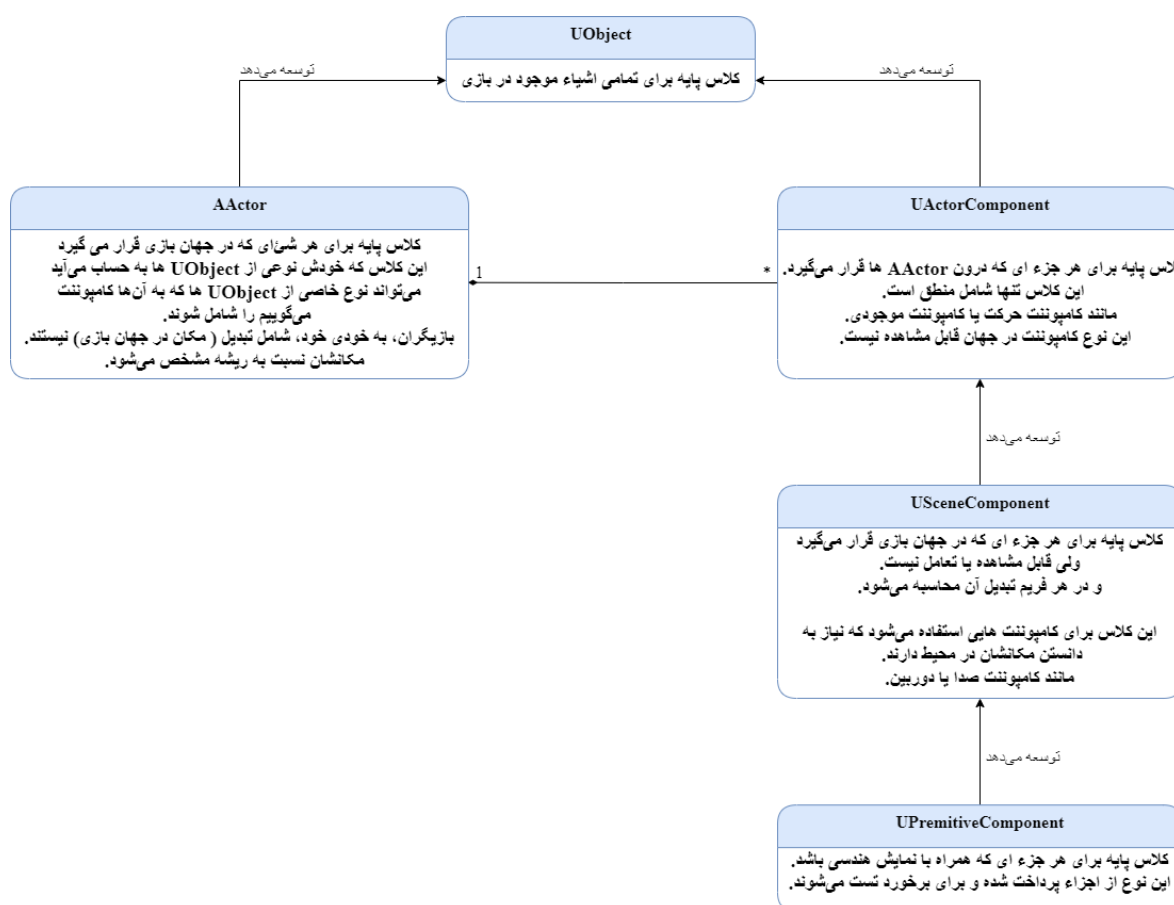
⁷overlap

⁸static mesh

⁹skeletal mesh

¹⁰particle systems

¹¹collision volumes



شکل ۱-۲ - تصویر UML کلاس های پایه ی موتور بازی سازی 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.