

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

تشکر و قدردانی

سپاس خدای را که سخنوران، در ستودن او بمانند ...

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.

تقدیم به



فهرست مطالب

صفحه	عنوان
هشت	فهرست مطالب
نه	فهرست شکل ها
ده	فهرست جدول ها
یازده	فهرست الگوریتم ها
۱	چکیده
۲	فصل اول: مقدمه
۲	۱-۱ موتور بازی سازی
۳	۲-۱ موتور بازی سازی آنریل
۴	۳-۱ انیمیشن های کامپیوتری
۵	فصل دوم: معماری موتور بازی سازی آنریل
۵	۱-۲ Actors و UObject
۶	۲-۲ Components
۸	پیوست ها
۹	مراجع

فهرست شکل ها

صفحه

عنوان

شکل ۱-۲: تصویر UML کلاس های پایه ی موتور بازی سازی آنریل ۷

فهرست جدول‌ها

صفحه

عنوان

فهرست الگوریتم‌ها

چکیده

انیمیشن‌های رایانه‌ای فرایندی است که برای تولید تصاویر متحرک دیجیتالی استفاده می‌شود. انیمیشن کامپیوتری مدرن معمولاً از گرافیک کامپیوتری سه‌بعدی برای ایجاد یک تصویر سه‌بعدی استفاده می‌کنند. در اکثر سیستم‌های انیمیشن کامپیوتری سه‌بعدی یک انیماتور نمایش ساده از آناتومی یک شخصیت ایجاد می‌کند که مشابه یک اسکلت یا آدمک می‌باشد. در شخصیت‌های انسان و حیوانات اکثر قسمت‌های این مدل اسکلتی با استخوان‌های واقعی مطابقت دارد.

گام اول این پروژه بررسی سیستم گراف انیمیشن در موتور بازی‌سازی آنریل می‌باشد. گراف انیمیشن برای محاسبه‌ی وضعیت نهایی یک مش اسکلتی در فریم فعلی استفاده می‌شود. به صورت کلی این گراف برای نمونه‌گیری، ترکیب و دستکاری ژست‌ها استفاده می‌شود و این ژست به مش‌های اسکلتی توسط طرح انیمیشن اعمال می‌شود. در این گام به بررسی این گراف و الگوریتم‌های به کار گرفته شده در آن خواهیم پرداخت.

در گام دوم نیز به پیاده‌سازی سیستم انیمیشن اسکلتی از پایه، با توجه به روش‌های بدست‌آمده پرداخته می‌شود. این مرحله ۴ هدف را دنبال می‌کند. هدف اول نمایش اسکلتون در یک محیط سه‌بعدی که با استفاده از OpenGL به وجود آمده است، می‌باشد. در این مرحله باید با استفاده از زبان C++ برنامه‌ای بنویسیم که در نهایت بتواند یک انیمیشن به وجود آمده به وسیله‌ی فریم‌های کلیدی را نمایش دهد. هدف بعدی نمایش دادن انیمیشن‌هایی می‌باشد که به وسیله‌ی فناوری موشن‌کیچر به وجود آمده‌اند. هدف سوم اضافه کردن یک مش به اسکلتون با استفاده از روش‌های پوسته‌سازی می‌باشد. هدف نهایی نیز پیاده‌سازی روش ترکیب انیمیشن می‌باشد تا بتوانیم انیمیشن‌های مختلف را با یکدیگر ترکیب کنیم.

کلمات کلیدی

انیمیشن‌های کامپیوتری، موتور بازی‌سازی، موتور آنریل، گراف انیمیشن، گرافیک سه‌بعدی کامپیوتری

فصل اول

مقدمه

۱-۱ 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. [۱] 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. [۲] 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: [۱]

- 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 با استفاده از دستگاه های متصل

به اینترنت است.

- مولفه منطق یا مکانیک بازی: این مولفه قوانین حاکم بر دنیای مجازی، ویژگی‌های شخصیت‌های بازیکنان، هوش مصنوعی و اشیاء موجود در دنیای مجازی و همچنین وظایف و اهداف بازیکنان را تعریف می‌کند.
- ابزارهای نرم‌افزاری: وظیفه اصلی این ابزارها افزایش راندمان و سرعت تولید بازی با موتور بازی‌سازی اسن. آن‌ها توانایی اضافه کردن بسیاری از عناصر مختلف را به بازی‌ها، از انیمیشن و جلوه‌های صوتی گرفته تا الگوریتم‌های هوش مصنوعی، را فراهم می‌کنند.

۲-۱ موتور بازی‌سازی آنریل

اولین نسل موتور بازی‌سازی آنریل توسط تیم سوینی، بنیانگذار اپیک گیمز^۱، توسعه یافت. سوینی در سال ۱۹۹۵ شروع به نوشتن این موتور برای تولید بازی تیراندازی اول شخصی به اسم غیرواقعی^۲ کرد. نسخه‌ی دوم موتور بازی‌سازی آنریل در سال ۲۰۰۲ منتشر شد. نسخه سوم نیز در سال ۲۰۰۴ پس از اینکه حدود ۱۸ ماه در حال توسعه بود منتشر شد. در این نسخه، معماری پایه‌ای موجود در نسخه‌ی اول مانند طراحی شی گرا، اسکریپت‌نویسی مبتنی بر داده و رویکرد نسبتاً ماژولار نسبت به زیرسیستم‌ها وجود داشت. اما برخلاف نسخه دوم که از یک خط لوله با عملکرد ثابت^۳ استفاده می‌کرد، این نسخه به صورتی طراحی شده بود تا بتوان قسمت‌های سایه‌زنی سخت‌افزاری^۴ را برنامه‌نویسی کرد. موتور بازی‌سازی آنریل ۴ در سال ۲۰۱۴ در کنفرانس توسعه‌دهندگان بازی^۵ منتشر شد. این نسخه با طرح کسب و کار 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 و این نسخه از تمام سیستم‌های موجود از جمله کنسول‌های نسل بعدی پلی‌استیشن ۵^۶ و ایکس‌باکس سری X/S^۷ 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 و به طور رسمی در سال ۲۰۲۲ نسخه‌ی کامل این موتور برای توسعه‌دهندگان انتشار یافت. [۳]

^۱Epic Games

^۲Unreal

^۳fixed-function pipeline

^۴shader hardware

^۵GDC

^۶PlayStation 5

^۷Xbox Series X/S

۳-۱ انیمیشن‌های کامپیوتری

فصل دوم

معماری موتور بازی سازی آنریل

در این فصل ابتدا به معماری موتور آنریل پرداخته شده و پس از آن درباره ی ابزار هایی 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 زیر شیء^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. [۴، ۵]

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 غیرفیزیک مفید هستند. این نوع از اجزاء هیچ گونه مکان فیزیک 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.

[۵]

¹Render

²Meshes

³ActorComponent

⁴SceneComponent

⁵Primitive Components

⁶collide

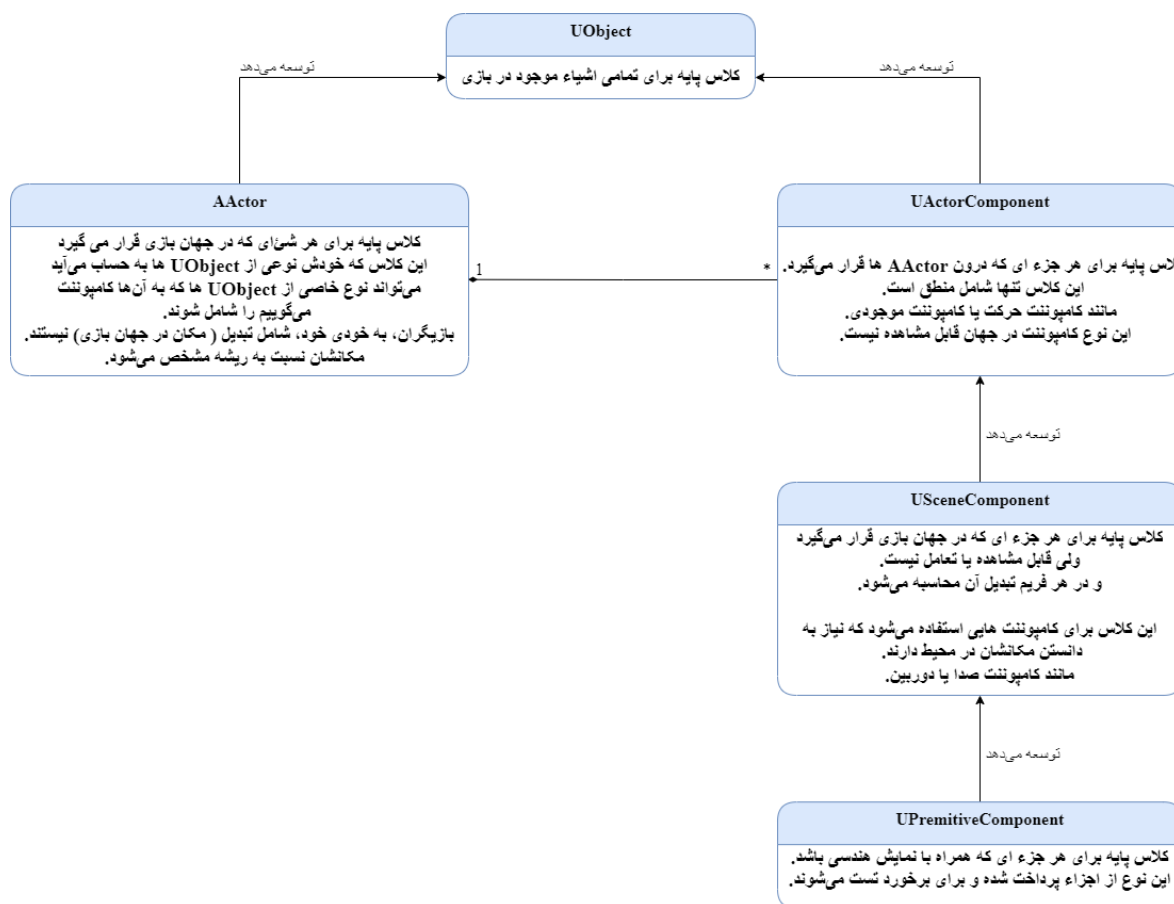
⁷overlap

⁸static mesh

⁹skeletal mesh

¹⁰particle systems

¹¹collision volumes



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

پیوست‌ها

مراجع

- [1] 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.
- [2] Gregory, J., *Game Engine Architecture*, A K Peters/CRC Press, 3rd ed. , 2018.
- [3] “Unreal engine wikipedia”, https://en.wikipedia.org/wiki/Unreal_Engine.
- [4] “Unreal engine architecture”, <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/ProgrammingWithCPP/UnrealArchitecture/>.
- [5] “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

Computer animation is the process used for digitally generating animated images. Modern computer animation usually uses 3D computer graphics to generate a three-dimensional picture. In most 3D computer animation systems, an animator creates a simplified representation of a character's anatomy, which is analogous to a skeleton or stick figure. In human and animal characters, many parts of the skeletal model correspond to the actual bones.

In the first phase, the animation graph will be examined in Unreal Engine. The animation graph is used to evaluate a final pose for the Skeletal Mesh for the current frame. This graph is used to sample, blend, and manipulate poses to be applied to Skeletal Meshes by the Animation Blueprints. We will examine the graph and the algorithm it uses at this stage.

In the second phase, the skeletal animation system is implemented from the ground up, based on the methods obtained. This step has four objectives. The first objective is to render a skeleton in a 3D environment created using OpenGL. As part of this step, we will write a program in C++ that will display the animation created by key frames. The next step will be to show motion-capture animations. As a third objective, skinning methods are used to add a mesh to the skeleton. Ultimately, we want to use the animation blending method to blend together different animations.

Keywords

Computer Animation, Game Engine, Unreal Engine, Animation Graph, 3D Computer Graphics



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.