موتورهای بازی‌سازی چگونه کار می‌کنند؟

همانطور که در قسمت «روزی روزگاری رایانه» گفتیم، موتور بازی‌سازی نوعی نرم‌افزار است که توسعه‌دهندگان را قادر می‌سازد تا بازی‌ها را خلق کنند. این نرم‌افزار مجهز به ابزارهای ویژه‌ای است که زمان توسعه بازی را کاهش و کیفیت آن را افزایش می‌دهد.

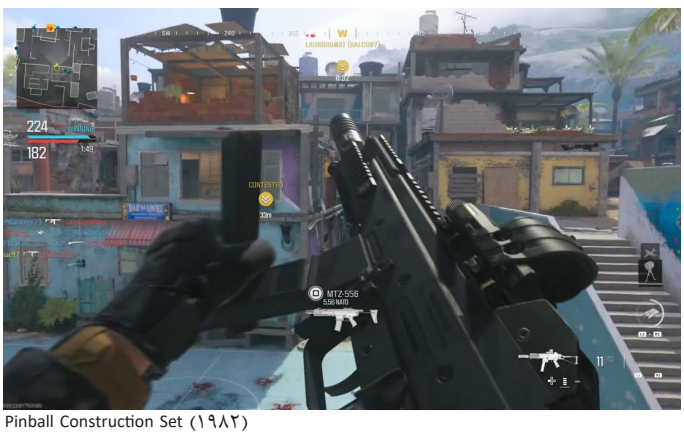
در ادامه انواع موتورهای بازی‌سازی و نحوهٔ کار کردن آن‌ها را بررسی می کنیم.

سیده‌فرگل ناظم‌زاده
دانشجو مهندسی کامپیوتر
دانشکدگان فارابی دانشگاه تهران
Fargol.nazemzadeh@ut.ac.ir

دسته‌بندی موتورها

موتورهای بازی‌سازی تا حدودی مختص به ژانر بازی‌ها هستند؛ برای مثال موتوری که برای ساخت یک بازی بوکس دو نفره به کار می‌رود، با موتور یک بازی در ژانر بازی‌های آنلاین چند نفره گسترده (MMO) یا تیراندازی اول شخص (FPS) بسیار متفاوت خواهد بود. اما صرف نظر از ژانر، همهٔ بازی‌ها نیازهای مشترکی مثل نوعی ورودی از سمت کاربر (از طریق موس، کیبورد یا موارد دیگر)، رندرینگ تصاویر بازی، (HUD Heads-Up Display)، سیستم صوتی و ... دارند.

این نیازهای مشترک باعث شده تا بتوان از موتوری مانند Unreal که برای بازی‌های تیراندازی اول شخص طراحی شده، در ساخت بازی‌های ژانر دیگر نیز استفاده شود؛ از جمله سری بازی «Batman: Arkham»از استودیو «Rocksteady» در ژانر اکشن-ماجراجویی و بازی مبارزه‌ای معروف «Tekken ۷» از استودیو «Bandai Namco».



Pinball Construction Set (۱۹۸۲)

چند مورد از معروف‌ترین ژانرهای بازی و موتورهای مورد استفاده در آن‌ها عبارتند از:

۱) بازی‌های ژانر FPS (First-Person Shooter)

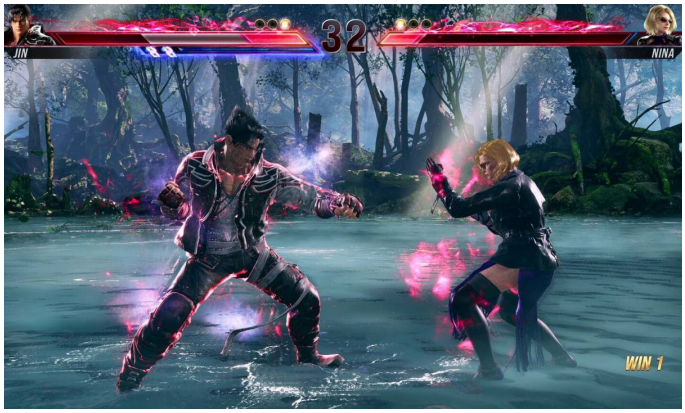
بازی‌های اول شخص معمولاً یکی از چالش برانگیزترین بازی‌ها از نظر فنی برای ساخت هستند، چون این بازی‌ها سعی می‌کنند برای بازیکن این توهم را ایجاد کنند که وارد دنیایی با جزئیات فراوان و واقع‌گرایانه شده است. بعضی از تکنولوژی‌های مورد نیاز این دسته از بازی‌ها شامل انیمیشن‌های با کیفیت بالا از بازوها و سلاح‌های بازیکن، مکانیزم دقیق کنترل و هدف‌گیری دوربین، انیمیشن‌های باکیفیت و هوش مصنوعی برای NPCها و ... می‌باشند.

معروف‌ترین موتورهای مورد استفاده در این ژانر، موتورهای Unreal و CryEngine هستند که هر دو به زبان C++ نوشته شده‌اند.

۲) بازی‌های ژانر Fighting

بازی‌های مبارزه‌ای معمولاً دو نفره و در یک فضا با اندازهٔ ثابت اتفاق می‌افتند، به همین خاطر کاراکترها در این نوع بازی، حرکت محدودی دارند. از تکنولوژی‌های مورد نیاز این بازی‌ها می‌توان به تشخیص دقیق ضربه، مجموعه‌ای غنی از انیمیشن‌های مبارزه‌ای، سیستمی که قادر به تشخیص دقیق و منظم ورودی‌های کاربر باشد و ... اشاره کرد.

موتورهای Unreal و Unity که به زبان C++ و C# نوشته شده، از معروف‌ترین موتورهای مورد استفاده در این ژانر از بازی‌ها هستند.



Tekken ۸ (۲۰۲۴)

۳) بازی‌های ژانر Racing

این ژانر شامل همه بازی‌هایی است که هدف اصلی در آن‌ها رانندگی است و در آن بازیکن ماشین یا وسایل نقلیه دیگری را در نوعی مسیر هدایت می‌کند. بازی‌های این ژانر معمولاً تمام جزئیات گرافیکی را روی ماشین، مسیر و محیط اطراف متمرکز می‌کنند. برخی از تکنیک‌های به کار رفته در این بازی‌ها عبارتند از: دنبال کردن ماشین از پشت سر برای دید سوم شخص، تلاش برای برخورد نداشتن دوربین با فضای اطراف در مسیرهایی که شامل تونل یا فضاهای بستهٔ دیگر است و ... .

از موتورهای معروفی که در ساخت این ژانر از بازی‌ها به کار می‌روند می‌توان به Frostbite که به زبان C++ و C# و EGO که به زبان C++ نوشته شده است، اشاره کرد.

اجزای مختلف یک موتور

یک موتور بازی‌سازی شامل بخش‌های مختلفی است که هر کدام وظایف

مشخصی دارند. در این بخش به بررسی چند مورد از آن‌ها می‌پردازیم.

۱) هستهٔ بازی (Game Core)

هستهٔ بازی حاوی منطق مربوط به بازی است. در این بخش الگوریتم‌ها، گیم‌پلی و اقدامات اصلی بازی تعریف می‌شوند. مدیریت منابع، مکانیزم‌ها و قوانین مربوط به بازی در این قسمت پیاده‌سازی می‌شوند.

۲) موتور گرافیکی (Graphics Engine)

موتور گرافیکی یا Rendering Engine، یکی از بزرگ‌ترین و پیچیده‌ترین قسمت‌های هر موتور بازی‌سازی است. رندررها را می‌توان به روش‌های مختلفی طراحی کرد اما اکثر موتورهای رندریگ مدرن، در فلسفه‌های اساسی طراحی، اشتراک دارند. این بخش وظیفه رندر دو بعدی یا سه بعدی گرافیک بصری بازی را بر عهده دارد و شامل تکنولوژی‌ها و الگوریتم‌هایی برای نورپردازی و افکت‌های بصری است.

موتورهای رندریگ می‌توانند از تکنیک‌های مختلفی برای نزدیک‌تر کردن بازی به واقعیت استفاده کنند. برخی از این تکنیک‌ها عبارتند از:

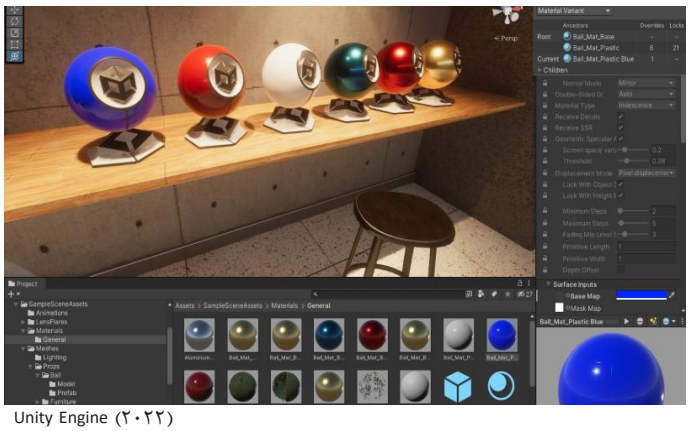
تکنیک Particle System: با استفاده از این تکنیک می‌توانیم تعداد بسیار زیادی از ذرات کوچک را تحت کنترل خود بگیریم و به هر کدام از آن‌ها یک مدل سه بعدی نسبت دهیم. این تکنیک برای ایجاد آتش، دود، پاشیدن آب، آبشار و ... استفاده می‌شود.

تکنیک Ray Tracing: این تکنیک با ردیابی اشعه در زمان واقعی، می‌تواند نورپردازی درون بازی را بسیار واقعی‌تر جلوه دهد و با دقت بیشتری نحوهٔ تعامل اشیا موجود در محیط را شبیه‌سازی کند. این تکنیک برای شبیه‌سازی جلوه‌های نوری مانند بازتاب و شکست نور استفاده می‌شود.

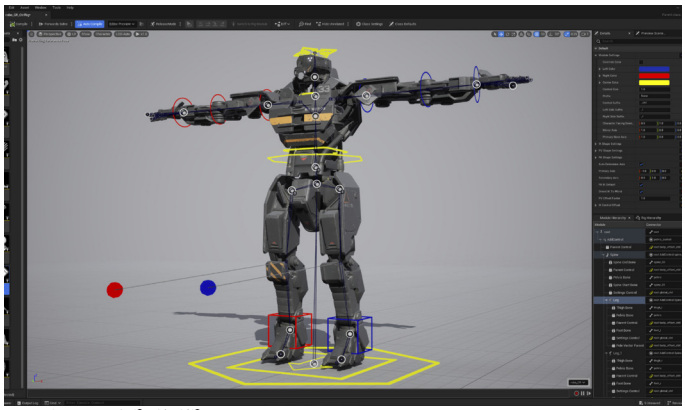
۳) موتور فیزیکی (Physics Engine)

موتور فیزیکی مسئول محاسبه و شبیه‌سازی رفتار فیزیکی اشیا در بازی، با استفاده از قوانین فیزیک است. بدون موتورهای بازی که فیزیک بازی را برای ما فراهم می‌کنند، حرکت‌ها در بازی غیرعادی و غیرطبیعی به نظر می‌رسند. این بخش به توسعه‌دهندگان امکان مدلسازی واکنش‌های طبیعی مانند گرانش، برخورد، اصطکاک و سیالیت را می‌دهد.

یکی از کارهای مهمی که موتور فیزیکی انجام می‌دهد، تشخیص برخورد اشیا در بازی است. اگر این تشخیص اتفاق نیافتد، اشیا به یکدیگر نفوذ می‌کنند و تعامل در بازی به هر روش معقولی غیرممکن می‌شود.



Unity Engine (۲۰۲۲)



Unreal Engine ۵,۴ (۲۰۲۴)

۴) موتور صوتی (Audio Engine)

موتور صوتی بخشی است که وظیفهٔ ایجاد و مدیریت صداها را برعهده دارد. توسعه‌دهندگان این امکان را دارند تا صداهای بازی را از پایه طراحی کنند. این صداها شامل موسیقی، صداهای محیطی، افکت‌های صوتی، موسیقی پس‌زمینه و صدای کاراکترها می‌شوند.

فایل‌های صوتی با استفاده از فناوری به کار رفته در موتور بازی، در کد ادغام می‌شوند و توسط اقدامات خاصی که با گیم‌پلی بازی مطابقت دارند، فعال می‌شوند.

۵) موتور هوش مصنوعی (Artificial Intelligence Engine)

در حال حاضر هوش مصنوعی بیش از هر زمان دیگری در ساخت بازی و گیم‌پلی آن نقش دارد. هوش مصنوعی به بازی‌ها این امکان را داده است تا علاوه بر طراحی نرم‌افزاری موتورهایشان، داستان سرایی خود را نیز ارتقا دهند.

از موتور هوش مصنوعی برای پیاده‌سازی الگوریتم‌ها و منطق AI استفاده می‌شود. به عنوان مثال، این بخش مسئول حرکات، تصمیم‌گیری هوشمندانه و تعامل با محیط اطراف یا به صورت خلاصه، منطق NPCها و سایر عناصر بازی مانند دشمنان و متحدان است.

۶) سیستم اسکریپت‌نویسی (Scripting System)

بسیاری از موتورهای بازی‌سازی از یک زبان برنامه‌نویسی استفاده می‌کنند تا بتوان قوانین و گیم‌پلی بازی را آسان‌تر و سریع‌تر توسعه داد. بدون زبان اسکریپت، بعد از هر تغییر در منطق یا ساختار داده‌های استفاده شده، باید بازی خود را مجدداً کامپایل کرده و لینک کنید، اما زمانی که یک زبان برنامه‌نویسی در موتور شما ادغام شود، می‌توان با اصلاح و بارگذاری مجدد کد اسکریپت، تغییراتی در منطق و داده‌های بازی ایجاد کرد.

زبان‌های Python، Lua، JavaScript و AWK از جمله زبان‌های اسکریپت‌نویسی هستند.

منابع:

Jason Gregory, Game Engine Architecture (Third Edition)