

## چالش های پایگاه داده در بازی های کامپیوتری

فاطمه پور<sup>۱</sup>، جعفر پرتابیان<sup>۲</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد دانشگاه آزاد اسلامی واحد لامرد

<sup>۲</sup> مدیر گروه تحصیلات تکمیلی کامپیوتر، دانشگاه آزاد اسلامی، لامرد.

نام نویسنده مسئول:

فاطمه پور

تاریخ دریافت: ۱۳۹۹/۸/۴

تاریخ پذیرش: ۱۳۹۹/۱۰/۳

چکیده

امروزه بازی های رایانه ای با چالش هایی در مدیریت داده ها روبرو هستند، در این مقاله قصد داریم به این چالش ها بپردازیم. بسیاری از بازی های رایانه ای از پایگاه داده به صورت مستقیم استفاده نمیکنند، با این حال مجبورند حجم عظیمی از داده ها را پردازش کنند و از تکنولوژی های جدید در پایگاه داده سود ببرند. برخی از بازی های آنلاین به سبک چند نفره کلان باید با پایگاه داده های **commercial** ارتباط برقرار کنند و چالش های مربوط به خودشان را دارند. ما سعی کرده ایم که از تعامل با استودیو بازی های مختلف، روش های مدیریت داده های آن ها را پیدا کنیم و آنها را در این مقاله ارائه داده ایم. این مقاله فرصت های تحقیقات جدیدی را ارائه میکند.

در این مقاله نشان خواهیم داد که توسعه دهندگان بازی ها از چه ابزار هایی برای مدیریت و ایجاد محتوا استفاده میکنند، مدل طراحی چگونه بروی عملکرد تاثیر میگذارد و ساختار داده ای و تکنیک هایی که توسعه دهندگان استفاده میکنند تا اطمینان حاصل کنند که باز یهای رایانه ای پاسخگو هستند. همچنین در مورد مساله ی سازگاری بازی ها بحث میکنم و اینکه چطور بازیکنان یک **view** از فضای بازی را مشترکا استفاده میکنند. در نهایت چالش هایی را که توسعه دهندگان هنگام استفاده از پایگاه داده های قدیمی مواجه میشوند را نشان میدیم. سعی کرده ایم تا اطلاعاتی کاملی در مورد اینکه چطور تکنولوژی پایگاه داده ها و پایگاه داده به بازی های کامپیوتری مرتبطند را فراهم کنیم. این اطلاعات شرایط برای محققانی که خودشان بازی نمیکنند ولی علاقمند به حل مشکلات بازی های رایانه ای هستند بسیار سودمند است.

واژگان کلیدی: بازی، اسکریپت نویسی، شاخص گذاری، تجمیع

## مقدمه

بسیاری از بازی‌های رایانه‌ای مدرن از تکنیک توسعه داده محور استفاده میکنند. در توسعه‌ی داده محور محتویات بازی تا حد امکان از بخش نرم افزار بازی جداست و در فایل‌های داده‌ی کمکی نگهداری می‌شوند. برای بسیاری از محتویات بازی‌ها مانند موسیقی یا گرافیک این موضوع بسیار رایج است. صنعت بازی برای این نوع داده‌ها فرمت مشخصی را تعیین کرده و ابزارهایی را برای مدیریت آنها فرام کرده است. هرچند که محتویات بازی مانند خصوصیات رفتاری و محرک‌هایی برای پاسخگویی به رویدادها که ظاهراً بخش نرم افزار به نظر میرسند، در نتیجه کمپانی‌های بازی تلاش زیادی برای تولید ابزار ساخت محتویات سفارشی و زبان‌های اسکریپتی برای پشتیبانی این مدل توسعه کرده‌اند.

استودیوهای بازی به دلایل مختلفی از طراحی داده محور استقبال میکنند. اول از همه به راحتی امکان تقسیم بندی کارها را بین مهندسين نرم افزار و طراح ها فراهم میکند. مهندسين نرم افزار انتزاع موتور بازی را برنامه نویسی میکنند در حالی که طراحان بازی، محتوا و خصوصیات را برای بازی خاص تولید میکنند. طراحان معمولاً تجربه‌ی برنامه نویسی کمی دارند. بنابراین به ابزارهایی نیز دارند که به صورت طبیعی آنها را در روند طبیعی کار قرار دهد. علاوه بر این خاصیت روش‌های داده محور به کمپانی‌های بازی اجازه میدهد تا هزینه‌ی توسعه خود را تحت عناوین مختلف پخش کنند. توسعه‌ی بازی‌های جدید معمولاً محتویات جدیدی را شامل میشود اما در بخش نرم افزار بازی تغییرات خیلی کم هستند.

طراحی به صورت داده محور منجر می‌شود تا اشتیاق کاربران به طراحی بازی و مهارت‌های کاربر افزایش یابد. برای مثال بازی World of Warcraft شامل توصیفات زبان xml است که به کاربرها اجازه میدهد تا رابط کاربری خود مانند پنجره‌ی بازی و یا دکمه‌های بازی را به صورت دلخواه تغییر دهند. این قابلیت‌ها به کاربرها امکان میدهد تا فضای بازی را برای انجام بهترین بازی و بهره‌وری بیشتر انتخاب کنند. بعضی از بازی‌ها مثل Second Life پیشرفته‌تر شده‌اند و زبان اسکریپتی را برای کاربر برای ایجاد محتویات جدید فراهم کرده‌اند. این نوع از محتویات تولید شده توسط کاربر می‌تواند طول زمان محبوبیت بازی را افزایش دهد.

طراحی بازی‌های رایانه‌ای به صورت داده محور حجم عظیمی از داده‌ها را تولید میکند که موتور بازی باید آن‌ها را به صورت کارآمد مدیریت کند. برای فهم این چالش در نتیجه‌ی حجم داده‌ها، در ابتدا باید متوجه شویم این داده‌ها چگونه تولید و اصلاح میشوند. بنابراین این آموزش با خلاصه‌ای از ابزارهای تولید محتوا برای بازی‌ها شروع میشود. ارائه‌ی ما روی زبان‌های اسکریپتی برای بازی‌های تک نفره و چند نفری تمرکز میکند. ما روشهایی را که طراحان برای مدیریت فایل‌های xml، تعریف خصوصیات رفتاری، محرک‌های وقایع خاص و ایجاد بازی استفاده میکنند را نشان خواهیم داد.

## چالش‌های کارایی

برنامه نویسی و توسعه‌ی رفتار بازی‌ها خارج از نرم افزار کاربردی فواید زیادی به همراه دارد اما غالباً این کار منجر به کاهش کارایی میشود. اگر طراحان دقیق نباشند می‌توانند اسکریپت‌هایی بنویسند که در آن هر شی با شی دیگری در ارتباط است در نتیجه پیچیدگی ( $n^2$ ) به همراه خواهد داشت. این مساله از مشکلات جدی در طراحی بازی است و برخی از استودیوهای بازی برای جلوگیری چنین پیچیدگی‌ها پر هزینه‌ای امکان تکرار و بازگشت را از زبان‌های اسکریپت نویسی شان حذف کرده‌اند. در هر فریم متحرک اسکریپت‌ها به صورت متناوب پردازش می‌شوند، بنابراین کدهای به ظاهر بی‌ضرر می‌توانند کارایی یک بازی را فلج کنند.

توسعه دهندگان بازی‌ها برای افزایش سرعت محاسبات مانند روابط بین جفت‌شی از شاخص گذاری در کنار دیتابیس استفاده میکنند. بسیاری از بازی‌ها از شاخص گذاری‌های مکانی ۱ مرسوم مثل درخت افراز فضایی<sup>۲</sup> یا درخت هشت تایی<sup>۳</sup> استفاده میکنند. در حالیکه بسیاری از بازی‌ها ساختار داده‌های فضایی غیر معمول دارند که برای مخاطبان پایگاه داده آشنا

<sup>1</sup> Spatial indices

<sup>2</sup> Binary space partitioning

<sup>3</sup> octree

نیست. برای مثال مش ناوبری ۴ برای تعیین حرکت مجاز شی بر اساس موقعیت جغرافیایی استفاده میشود. به علاوه این مش ها اطلاعات معنایی اضافی را شامل می شود.

همچنین بررسی میکنیم که چگونه طراحان بازی ها از برنامه نویسی موازی برای افزایش کارایی استفاده میکنند. این حوزه ای است که توسعه دهندگان بازی می توانند به صورت بلقوه مسائل زیادی در مورد ارتباط با پایگاه داده بفهمند. در واقع بسیاری از تکنیک هایی که برنامه نویسان بازی برای بهینه سازی محاسبات فیزیکی و بقیه محاسبات روی واحد پردازش گرافیکی ۵ استفاده میکنند شبیه به تکنیک هایی است که پایگاه داده ها برای join processing استفاده میکنند. علاوه بر این تحقیقات در حوزه ی موتور بازی و زبان های اسکریپت نویسی برای ساده تر کردن این مدل توسعه در حال رشد است.

## چالش های سازگاری

در باز یهای آنلاین چند نفری کلان ۶، سازگاری ۷ به اندازه ی کارایی مهم است. بازی ها به داده های خودشان که معمولا کل دنیای بازی است به صورت سازگار نیاز دارند. فراهم کردن این سازگاری بسیار سخت و دشوار است زیرا بازیکن ها به صورت همزمان عملیات مختلفی را انجام میدهند. این موضوع به این معنی است که روش هایی مانند قفل گذاری برای بازی ها بسیار کند هستند. علاوه بر این زبان های اسکریپت نویسی از امکاناتی مانند همزمانی پشتیبانی نمیکند و طراحان بازی ها و بازیکن ها از این خصوصیت به صورت کارآمد نمیتواند بهره ببرند. در واقع همزمانی در زبان های اسکریپت نویسی یکی از بزرگترین منابع باگ در بازی های آنلاین چند نفری کلان است. در این آموزش ما روش های مختلفی را برای مقابله با این چالش ها ارائه میکنیم و این که چطور می توانند برای تحقیقات آینده استفاده شوند.

یکی از موفق ترین راه حل ها causality bubbles است. این اصطلاح یعنی تکنیک های مختلفی را برای پیش بینی اینکه کدام یک از بازیکن ها حرکات متضادی باهم خواهند داشت و به صورت پویا پایگاه داده ها را برای کاهش بار سرور جدا میکند. بسیاری از بازی های تجاری از متد های مختلفی برای محیط های مختلف استفاده میکنند. برای مثال EVE معادلات دیفرانسیل مختلفی را برای تخمین سرعت سفینه ها در منظومه ی شمسی اجرا میکند.

این معادلات دیفرانسیلی به آنها اجازه میدهد تا برای بازه ی زمانی مختلف تخمین بزند که کدام سفینه میتواند در محدوده ی دیگری حرکت کند. با استفاده از این روش می توانند نقشه را به قسمت های مناسب تقسیم بندی کنند. تحقیقات اخیر زیادی سعی کرده اند که این ایده را توسعه دهند.

روش دیگر که برای مقابله با همزمانی این است که سازگاری کمتری در محیط بازی تضمین شود. در برخی مواقع این یعنی سازگاری در سطح درشت دانه در بازی رعایت شود. انیمیشن یا فعالیت های بی دسر و بی رقیب در بازی بین کامپیوتر ها همگام نیست ولی یک حالت پایدار مشابه دارند. برخی از بازی های دیگر به صورت پیش تر رفته اند و به بازیکن ها اجازه میدهند که غیر سازگار باشند اما حالت های بازی بسیار شبیه به هم دارند. برای مثال مدیریت aggro تکنیکی است که در World of Warcraft استفاده میشود برای هدف قرار دادن مخاطبان و مبارزه.

## چالش های مهندسی

بسیاری از بازی های آنلاین از پایگاه داده استفاده میکنند و بنابراین مشکلات رایج پایگاه داده را خواهند داشت. برای مثال باید اطمینان حاصل کنند که پل ارتباطی میان نرم افزار کلاینت و کد های SQL به اندازه ی کافی برای رسیدگی به تغییرات موجود در هریک به اندازه ی کافی قوی است. اگرچه چالش هایی هستند که مربوط به یک بازی می شوند و انگیزه ای برای کارهای آینده خواهند بود. در این آموزش در صورت زمان داشتن برخی از این مسایل را آزمایش میکنیم.

<sup>4</sup> navigational meshes

<sup>5</sup> Graphics Processing Unit

<sup>6</sup> MMOS (massively multiplayer online games)

<sup>7</sup> consistency

یکی از مسائل مهم مهندسی مساله‌ی نقطه بازرسی های ۸ هوشمند است. بازی های چند نفره‌ی کلان از پایگاه داده commercial برای تداوم و بازیابی از مشکلات سرور استفاده میکنند. در حالی که بازیکن ها با سرعت بسیار زیادی در حال تعامل با بازی هستند و هر عمل با پایگاه داده هزینه سنگینی دارد. بسیاری از بازی ها یک لایه‌ی پایگاه داده‌ی داخل حافظه دارند که تمام عملیات را پردازش میکند و فقط روی پایگاه داده به صورت دوره ای مینویسد. در بعضی بازی ها این نقطه بازرسی ها هر ۱۰ دقیقه یک بار از هم فاصله دارند. بازیابی ممکن است یک بازیکن را مجبوره تکرار جنگ سخت و یا از دست دادن پاداش و جایزه کند بنابراین بازی ها به راهی نیاز دارند که نقطه بازرسی ها را به صورت هوشمند تعیین شود و هر موقع اتفاق مهمی در بازی تمام میشود نقطه بازرسی انجام شود نه در بازه‌ی زمانی خاص.

شمای میراثی یکی دیگر از مسائل مهم در بازی ها است. برخی از بازی های چند بازیکنی چند نفره‌ی کلان مثل venerable Everquest برای چند دهه فعال هستند. به دلیل مسائل رقابتی این بازی ها به صورت مکرر ویژگی ها و قابلیت های زیادی را به بازی های خود اضافه میکنند. این ویژگی های جدید نیازمند تغییراتی در پایگاه داده خواهد شد. تغییرات پایگاه داده روی سیستم آنلاین می تواند برای توسعه دهندگان بسیار سخت باشد.

### نتیجه گیری

در این مقاله ما نشان دادیم که چطور پایگاه داده با بازی های کامپیوتری انطباق پیدا می کند و همچنین حوزه های تحقیقاتی را که طراحان پایگاه داده می توانند به آن کمک کنند را نشان دادیم. به عنوان حرف آخر، جدای از جذاب بودن این حوزه برای مسائل تحقیقاتی جدید ما فکر می کنیم دو دلیل اساسی وجود دارد که چرا جامعه پایگاه داده باید بازی ها را در بر بگیرد.

دلیل اول باید با معماری های چند هسته ای انجام شود. به عنوان مثال، سرعت های حافظه نسبتا پایین به وسیله کش ها (حافظه های نهان)، اجرای خارج ترتیب، و اجرای پیش بینی گرایانه (با تفکر و اندیشه)، مورد بررسی قرار گرفته است. بهره وری پایین از منابع پردازنده با چند نخه همزمان مورد بررسی قرار گرفته و به همین ترتیب. با این حال، اخیراً رشد فرکانس کلاک متوقف شده است، چرا که روند مصرف توان غیر قابل کنترل شده است.

دوم اینکه سال های اخیر شاهد یک افت شدید در تاریخ علوم کامپیوتر در دانشگاه بوده است. بسیاری از دانشگاه ها اخیرا برنامه ها را در بازی های کامپیوتری توسعه می دهند. یک گروه دانشجویی معمول شامل هنرمندان، نویسندگان، موسیقی دانان، و برنامه نویسان هستند که همگی با هم برای ایجاد یک بازی اصلی کار می کنند. تمام دانشجویان درگیر فرایند طراحی بازی، طرح ریزی و خالص کردن سیستم های قوانین بازی، و واسطه ها هستند. جذب دانشجویان در حوزه علوم کامپیوتر از طریق بازی های کامپیوتری یک روش برای انتقال هیجان زمینه ما می باشد و ما اعتقاد داریم که طراحان پایگاه داده برای کمک آماده است.

## منابع و مراجع

- [1] E. Coumans. Physics tutorial: Parallel game physics for spu. In Proc. GDC, San Francisco, CA, 2008.
- [2] B. Dalton. Online gaming architecture: Dealing with the real-time data crunch in MMOs. In Proc. Austin GDC, Austin, TX, September 2007.
- [3] M. DeLoura, editor. Game Programming Gems, volume 1. Charles River Media, 2000.
- [4] Halldor Fannar Gugj\_onsson. The server technology of EVE Online: How to cope with 300,000 players on one server. In Proc. Austin GDC, 2008.
- [5] N. Gupta, A. Demers, J. Gehrke, P. Unterbrunner, and W. White. Scalability for virtual worlds. In Proc. ICDE, 2009.
- [6] T. Keating. Dupes, speed hacks and black holes: How players cheat in MMOs. In Proc. Austin GDC, Austin, TX, September 2007.
- [7] Linden Labs. Linden Scripting Language. <http://wiki.secondlife.com/wiki/LSL> Portal.
- [8] J. Lee, R. Cedeno, and D. Mellencamp. The latest learning - database solutions. In Proc. Austin GDC, Austin, TX, September 2007.
- [9] Sun Microsystems. Project darkstar. <http://www.projectdarkstar.com/>.
- [10] S. Posniewski. Massively modernized online: MMO technologies for next-gen and beyond. In Proc. Austin GDC, Austin, TX, September 2007.
- [11] B. Sowell, A. Demers, J. Gehrke, N. Gupta, H. Li, and W. White. From declarative languages to declarative processing in computer games. In CIDR, 2009.
- [12] P. Tozour. Building a near-optimal navigation mesh. In AI Game Programming Wisdom, volume 1, pages 298{304. Charles River Media, 2002.
- [13] W. White, A. Demers, C. Koch, J. Gehrke, and R. Rajagopalan. Scaling games to epic proportions. In Proc. SIGMOD, pages 31{42, 2007.
- [14] WoWWiki. XML user interface. <http://www.wowwiki.com/XML> user interface.