

دریافت مقاله: ۱۴۰۳/۰۹/۲۲

پذیرش مقاله: ۱۴۰۳/۱۰/۱۲

تنظیم درجه سختی در بازی‌های جورکردنی موبایلی به صورت پویا و با استفاده از سیستم‌های فازی

عماد وثوقی

فارغ‌التحصیل مهندسی کامپیوتر دانشکده کامپیوتر، دانشگاه علم و صنعت، تهران، ایران
پست الکترونیکی: e_vosoughi@comp.iust.ac.ir

مهرداد آشتیانی*

دکترای مهندسی نرم‌افزار دانشکده کامپیوتر، دانشگاه علم و صنعت، تهران، ایران
پست الکترونیکی: m_ashtiani@iust.ac.ir

چکیده

موبایلی توسعه داده شده است. نتایج ارزیابی‌ها روی ۳۰ نفری که هر دو نسخه دارا و فاقد روش پیشنهادی را بازی کردند، نشان داد که روش پیشنهادی شاخص عملکرد میانگین زمان هر جلسه بازی و نرخ تبدیل کاربران را به ترتیب به میزان ۸۶ و ۲۰۰ درصد، نسبت به وقتی که از روش پیشنهادی استفاده نشود، بالاتر می‌برد.

واژه‌های کلیدی: تنظیم پویای سختی، بازی جورکردنی، الگوریتم‌های هوش مصنوعی، سطح سختی متفاوت.

۱. مقدمه

در طول تاریخ توسعه بازی‌های دیجیتالی، یکی از چالش‌های اصلی برای طراحان و تولیدکنندگان بازی‌های ویدیویی، ایجاد تجربه‌ای است که برای طیف وسیعی از بازیکنان جذاب و متعادل باشد. این چالش با تنوع در مهارت‌ها، تجربه و توقعات بازیکنان حتی پیچیده‌تر نیز می‌شود. در این میان، یکی از ابزارهای مهم و مؤثر برای رسیدن به این هدف، تکنیکی است که به نام تنظیم پویای سطح سختی شناخته می‌شود. تنظیم پویای سطح سختی

تنظیم درجه سختی مراحل به صورت پویا در بازی‌های موبایلی و بخصوص ژانر جورکردنی در سال‌های اخیر مورد توجه بازی‌های مهم این سبک در بازار بازی‌های موبایلی بوده است. استفاده از هوش مصنوعی در جهت دقیق‌تر دسته‌بندی کردن کاربران و تنظیم سطح سختی به صورت پویا می‌تواند کمک شایانی به ترغیب بیشتر کاربران برای ادامه بازی و پیشروی در مراحل کند. در این پژوهش، تلاش بر پویاسازی فرایند تنظیم درجه سختی با استفاده از سیستم‌های فازی خواهد بود تا هر کاربر، سطح سختی متفاوتی متناسب با سطح مهارت خود در پیش‌برد بازی تجربه کند. در این پژوهش از منطق فازی برای طراحی سیستمی که بتواند سطح سختی را به صورت پویا برای کاربران تنظیم کند، استفاده شده است. در ابتدا، زیرساختی برای پیاده‌سازی سیستم‌های کنترلی فازی در موتور بازی‌سازی یونیتی پیاده شده است و سپس با استفاده از آن زیرساخت، یک سیستم کنترلی فازی برای تنظیم پویای سطح سختی در مراحل بازی‌های جورکردنی

* نویسنده مسئول

روشنی است که به بازی اجازه می‌دهد تا سطح دشواری خود را به طور خودکار و بر اساس عملکرد و رفتار بازیکن در طول بازی تنظیم کند. در بازی‌هایی که از تنظیم پویای سطح سختی استفاده می‌کنند [۱]، هدف این است که تجربه بازی برای بازیکنان چالش‌برانگیز و در عین حال انگیزه‌بخش باشد. این تکنیک می‌تواند به بهبود تجربه کاربر و افزایش مدت‌زمان تعامل با بازی منجر شود. با توجه به این‌که بازی‌ها به طور مداوم در حال تحول و پیچیدگی بیشتر هستند، تنظیم پویای سطح سختی به یکی از ابزارهای ضروری برای توسعه‌دهندگان تبدیل شده است تا اطمینان حاصل کنند که بازی‌ها قادر هستند نیازها و توقعات بازیکنان مختلف را برآورده کنند.

یکی از چالش‌های اصلی در پیاده‌سازی روش‌های تنظیم پویای سطح سختی در بازی‌های جورکردنی این است که بتوان میزان دشواری مناسب را برای هر بازیکن به درستی تشخیص داد و به کار برد. اینجا است که منطق فازی به عنوان یک ابزار قدرتمند وارد می‌شود. منطق فازی، یک روش محاسباتی است که با مفاهیم و متغیرهای نامشخص و مبهم سر و کار دارد. این منطق بر اساس نظریه مجموعه‌های فازی عمل می‌کند که در آن اشیاء می‌توانند به درجات مختلفی به یک مجموعه تعلق داشته باشند. در پژوهش‌های انجام شده در مورد تنظیم پویای سطح سختی، تصمیم‌گیری‌ها عمدتاً با منطق دودویی انجام شده است [۲]. استفاده از این منطق گرچه به راحتی قابل پیاده‌سازی است اما دقت کافی را ندارد. استفاده از منطق فازی می‌تواند به ایجاد یک سیستم تنظیم پویای سطح سختی کمک کند که به جای تکیه بر تصمیم‌گیری‌های دودویی (مثل آسان یا سخت)، بر اساس داده‌های جمع‌آوری شده از رفتار بازیکن، طیفی پیوسته و نه گسسته از دشواری را ارائه دهد. به عبارت دیگر، استفاده از منطق فازی این امکان را به ما می‌دهد که بتوانیم بر اساس مجموعه‌ای از ورودی‌ها مانند تعداد حرکت‌ها، زمان صرف شده در یک مرحله یا تعداد دفعات تلاش مجدد، تصمیمات بهینه‌تری

برای تنظیم پویای سطح سختی بگیریم. دلیل اصلی برای استفاده از منطق فازی در سیستم کنترل‌کننده تنظیم پویای سطح سختی در بازی‌های جورکردنی، انعطاف‌پذیری و قابلیت‌های آن در مدیریت داده‌های نامشخص و متغیر است. در بازی‌های جورکردنی، بازیکنان ممکن است سبک‌های بازی و سطح مهارت‌های بسیار متفاوتی داشته باشند. به همین دلیل، تنظیم دقیق سطح دشواری برای هر بازیکن نیازمند یک سیستم پویا و انعطاف‌پذیر است که به صورت پیوسته و نه گسسته و دودویی تصمیم بگیرد. منطق فازی این امکان را فراهم می‌آورد که سیستم بازی بتواند به‌طور دقیق‌تری به تنظیم دشواری بر اساس طیف گسترده‌ای از داده‌های ورودی بپردازد.

علاوه بر این، استفاده از منطق فازی می‌تواند به کاهش ناامیدی بازیکنان کمک کند. به عنوان مثال، اگر یک بازیکن به طور مداوم در یک سطح خاص گیر کند، سیستم تنظیم پویای سطح سختی مبتنی بر منطق فازی، می‌تواند به تدریج دشواری آن سطح را کاهش دهد، به‌طوری‌که بازیکن متوجه این تغییر نشود و همچنان احساس چالش و پیشرفت را داشته باشد. از سوی دیگر، اگر بازیکنی در مراحل ابتدایی با سرعت پیشرفت کند، سیستم می‌تواند با افزایش دشواری، تجربه‌ای چالش‌برانگیزتر برای او ایجاد کند تا انگیزه ادامه دادن بازی را حفظ کند. به طور کلی، پیاده‌سازی سیستم تنظیم پویای سطح سختی با استفاده از منطق فازی در بازی‌های جورکردنی می‌تواند به بهبود تجربه بازیکن و افزایش تعامل با بازی منجر شود، به‌ویژه در محیط‌های پیچیده و پویا که نیازمند تطبیق‌دهی مداوم سطح دشواری با رفتار و عملکرد بازیکن است. در واقع به‌کارگیری منطق فازی برای تنظیم پویای سطح سختی در بازی‌های جورکردنی موضوعی است که تا به حال در پژوهش‌های مرتبط کمتر به آن پرداخته شده است و قصد از انجام این پژوهش ارائه چنین راهکاری است.

در این پژوهش، با استفاده از پیاده‌سازی زیرساختی

و بخصوص، بازی‌های جورکردنی پرداخته می‌شود. همچنین تاریخچه و انواع مختلف پیاده‌سازی تکنیک تنظیم پویای سطح سختی در بازی‌های ویدیویی و به خصوص بازی‌های جورکردنی موبایلی مورد بحث خواهد بود.

۱-۲- منطق فازی^۱

منطق فازی بر چند مفهوم کلیدی استوار است:

۱. مجموعه‌های فازی: در منطق فازی، هر عنصر به جای آن‌که به طور قطعی به یک مجموعه تعلق داشته باشد یا نداشته باشد، با درجه‌ای از عضویت به آن مجموعه تعلق دارد [۳]. به عنوان مثال، یک فرد ممکن است به مجموعه‌ی «قد بلند» با درجه‌ی عضویت ۰/۷ تعلق داشته باشد، به این معنی که فرد به طور نسبی قد بلند است.
۲. توابع عضویت: تابع عضویت برای هر مجموعه‌ی فازی تعریف می‌شود و به هر مقدار از ورودی، یک درجه‌ی عضویت بین صفر و یک اختصاص می‌دهد. این توابع معمولاً به شکل‌های مختلفی مانند مثلثی، ذوزنقه‌ای، گاوسی و غیره تعریف می‌شوند و انتخاب تابع مناسب بستگی به طبیعت مسئله دارد.

۳. عملیات فازی: عملیات منطقی در منطق فازی شامل «و» (AND)، «یا» (OR) و «نه» (NOT) است که به صورت فازی تعریف می‌شوند. به عنوان مثال، عملیات «و» در منطق فازی به معنای گرفتن کمینه‌ی درجات عضویت دو گزاره است، عملیات «یا» در منطق فازی به معنای گرفتن بیشینه‌ی درجات عضویت دو گزاره است و عملیات «نه» در منطق فازی به معنای تفریق درجه‌ی عضویت گزاره از عدد یک است.

۴. قواعد فازی: تصمیم‌گیری در سیستم‌های فازی بر اساس مجموعه‌ای از قواعد «اگر-آنگاه» انجام می‌شود. به عنوان مثال، این قواعد می‌توانند به شکل «اگر دمای هوا بالا است و رطوبت کم است، آنگاه سیستم تهویه روشن شود» تعریف شوند [۴]. این قواعد می‌توانند بر اساس تجربه‌ی انسانی یا به صورت خودکار از داده‌های

برای توسعه‌ی سیستم‌های کنترلی فازی در موتور بازی‌سازی یونیتی، یک سیستم کنترلی فازی برای تنظیم هوشمند سطح سختی در بازی‌های جورکردنی به صورت پویا توسعه داده شده است. این سیستم کنترلی بعد از یکپارچه‌سازی در بازی جورکردنی موبایلی «گلشیفته» باعث رشد چشم‌گیر شاخص‌های کلیدی عملکرد آن در زمینه‌هایی مثل فروش، درصد کاربران خریدکننده در بازی، میزان تماشای تبلیغ و غیره شده است. برای ارزیابی کارکرد روش پیشنهادی از ارائه‌ی پرسشنامه‌ای به ۳۰ نفر از افراد با درجات سنی مختلف که هر دو نسخه‌ی دارا و فاقد سیستم تنظیم پویای سطح سختی با استفاده از منطق فازی را به مدت حداقل یک هفته بازی کرده‌اند، استفاده شده است. در ادامه مشخص شد که ارائه‌ی این راهکار باعث رشد ۸۶ درصدی نرخ تبدیل کاربران شده است.

این پایان‌نامه در شش فصل نوشته شده است که در ادامه ساختار هر فصل به این شرح است. در فصل دوم مفاهیم پایه‌ای و اساسی موجود در این حوزه و نیز عناوین و امکانات مورد استفاده در این موضوع شرح و بسط داده شده است. در فصل سوم، کارهایی که تا کنون در این حوزه انجام شده‌اند مورد بررسی قرار گرفته است و طبقه‌بندی کامل تمامی این راهکارها به همراه نمونه‌ای از پژوهشی که در هر حوزه کار کرده‌اند و چالش‌های هر یک ارائه شده است. در فصل چهارم روش پیشنهادی به طور کامل توضیح داده شده است. در فصل پنجم نتایج حاصل از ساز و کار ارائه شده در سناریوهای مختلف و تحلیل‌های مربوط به آن آورده شده است. در فصل ششم نیز به جمع‌بندی و نتیجه‌گیری پژوهش و ارائه‌ی پیشنهادهایی برای کارهای آینده پرداخته شده است.

۲- مفاهیم اولیه

در این بخش، به منظور درک کلی از موضوع، ابتدا مفاهیم پایه‌ای پژوهش بیان می‌شود. ابتدا به معرفی و توضیح کلی منطق فازی پرداخته می‌شود. سپس به بازی‌های ویدیویی

موجود استخراج شوند.

۵. استنتاج فازی^۲: فرآیندی است که در آن قواعد فازی اعمال می‌شوند تا نتیجه‌ای از ورودی‌های داده شده استخراج شود. این فرآیند شامل سه مرحله اصلی است: فازی‌سازی ورودی‌ها، ارزیابی قواعد فازی و عکس‌فازی‌سازی^۳ خروجی‌ها به یک مقدار قطعی.

۲-۲- انواع سیستم‌های مبتنی بر منطق فازی

سیستم‌های فازی را می‌توان به چند دسته اصلی تقسیم کرد:

۱. سیستم‌های فازی نوع-۱: در این سیستم‌ها، توابع عضویت به صورت قطعی تعریف شده‌اند. این سیستم‌ها برای بسیاری از کاربردها مناسب هستند، اما ممکن است در مواجهه با عدم قطعیت زیاد یا تغییرات دینامیک کارایی لازم را نداشته باشند [۵].

۲. سیستم‌های فازی نوع-۲: این سیستم‌ها برای مدل‌سازی عدم قطعیت در توابع عضویت نیز طراحی شده‌اند. در این سیستم‌ها، توابع عضویت خود به صورت فازی تعریف می‌شوند که این ویژگی به آن‌ها امکان انعطاف‌پذیری بیشتری در مواجهه با عدم قطعیت‌های شدیدتر می‌دهد [۶].

۳. سیستم‌های فازی تطبیقی: این نوع از سیستم‌ها قادر به تطبیق و تغییر قواعد و توابع عضویت بر اساس داده‌های ورودی جدید و شرایط محیطی هستند [۷]. یکی از روش‌های مشهور در این زمینه، سیستم استنتاج عصبی فازی تطبیقی^۴ [۸] است که ترکیبی از منطق فازی و شبکه‌های عصبی را به کار می‌گیرد [۹].

۲-۳- اهمیت تنظیم پویای سطح سختی در بازی‌های رایگان

۱. تنظیم پویای سطح سختی یکی از ابزارهای کلیدی در طراحی بازی‌های رایگان است که می‌تواند تجربه بازی را برای بازیکنان مختلف شخصی‌سازی کند و در

نتیجه باعث افزایش تعامل و مشارکت آن‌ها شود. این نوع سیستم‌ها با تنظیم سطح دشواری بازی بر اساس توانایی‌ها و عملکرد هر بازیکن، از ایجاد احساس ناامیدی یا خستگی در بازیکنان جلوگیری می‌کند. این ویژگی به‌ویژه در بازی‌های رایگان اهمیت دارد زیرا حفظ تعامل بازیکنان به طور مستقیم با درآمدزایی از طریق خریدهای درون‌برنامه‌ای مرتبط است [۱۰].

۲. بازی‌هایی که از سیستم‌های تنظیم پویای سطح سختی استفاده می‌کنند، قادر هستند سطح دشواری را به گونه‌ای تنظیم کنند که بازیکنان احساس چالش‌پذیری کنند اما در عین حال از بازی ناامید نشوند. برای مثال، اگر بازیکنی در یک مرحله چندین بار شکست بخورد، سیستم ممکن است با ارائه کمک‌های اضافی مانند افزایش تعداد حرکات یا ترتیب دادن آیتم‌ها به گونه‌ای که ترکیب‌های مفیدتری ایجاد شود، او را حمایت کند. یا در مثالی دیگر اگر بازیکنی چندین مرحله پشت‌سرهم را بدون هیچ باختی رد کند، این سیستم ممکن است با کاهش تعداد حرکات، بازی را برای بازیکن چالش برانگیزتر کند تا از ایجاد حس خستگی جلوگیری کند. در ادامه به چند مورد از موارد حائز اهمیت وجود سیستم‌های تنظیم پویای سطح سختی در بازی‌های رایگان می‌پردازیم.

۳. بهبود تجربه کاربری: یکی از مهم‌ترین اهداف این سیستم‌ها در بازی‌های رایگان، بهبود تجربه کاربری است. در بازی‌هایی که سطح دشواری ثابت است، ممکن است برخی از بازیکنان احساس کنند که بازی برایشان بسیار سخت یا بسیار آسان است. این ناهماهنگی می‌تواند منجر به ناامیدی یا بی‌حوصلگی شود و در نتیجه باعث ترک بازی توسط بازیکن شود. با استفاده از سیستم‌های تنظیم پویای سطح سختی، بازی می‌تواند به طور مداوم سطح دشواری را متناسب با مهارت‌ها و عملکرد بازیکن تنظیم کند، تا تجربه‌ای متعادل و لذت‌بخش برای تمامی بازیکنان فراهم شود.

2-Fuzzy Inference

3-Defuzzification

4-ANFIS: adaptive-network-based fuzzy inference system

روبه‌رو شوند. این تطبیق‌پذیری موجب می‌شود که بازی برای همه سطوح مهارتی جذاب باقی بماند.

۷. کاهش نرخ ترک بازی در مراحل ابتدایی: یکی از مشکلات رایج در بازی‌های رایگان، ترک بازی توسط بازیکنان در مراحل ابتدایی است. اغلب بازیکنان در چند دقیقه یا چند ساعت اول بازی تصمیم می‌گیرند که به بازی ادامه دهند یا خیر [۱۳]. اگر بازی در مراحل ابتدایی بیش از حد دشوار یا بیش از حد آسان باشد، ممکن است بازیکن احساس کند که بازی برای او مناسب نیست و آن را ترک کند. سیستم تنظیم پویای سطح سختی می‌تواند با تنظیم سریع سطح دشواری بر اساس عملکرد اولیه بازیکن، این خطر را کاهش دهد و تجربه‌ای مناسب‌تر در همان مراحل ابتدایی فراهم کند.

۸. بهینه‌سازی برای درآمدزایی بلندمدت: از آن‌جا که بازی‌های رایگان به درآمدزایی بلندمدت از طریق خریدهای درون‌برنامه‌ای و تبلیغات متکی هستند، نگهداشت طولانی‌مدت بازیکنان در بازی از اهمیت بالایی برخوردار است [۱۴]. تنظیم پویای سطح سختی می‌تواند به ایجاد یک جریان درآمدی پایدار کمک کند، زیرا بازیکنانی که تجربه بازی خوبی دارند و احساس پیشرفت می‌کنند، احتمالاً بیشتر در بازی می‌مانند و در طول زمان خریدهای درون‌برنامه‌ای انجام می‌دهند. این امر به بهینه‌سازی راهبردهای درآمدزایی بلندمدت کمک می‌کند.

۹. تطابق با اهداف طراحی و برندینگ بازی: هر بازی رایگان یک هویت و هدف طراحی خاص دارد. برخی از بازی‌ها ممکن است بر روی گیم‌پلی چالشی و سخت متمرکز باشند، در حالی که برخی دیگر بر روی دسترسی‌پذیری و سرگرمی همه‌جانبه تمرکز دارند [۱۵]. سیستم تنظیم پویای سطح سختی می‌تواند به بازی‌ها کمک کند تا سطح دشواری را بر اساس این اهداف تنظیم کنند. برای مثال، یک بازی با تمرکز بر

۴. افزایش نرخ نگهداشت بازیکنان: سیستم تنظیم پویای سطح سختی می‌تواند به طور مستقیم به افزایش نرخ نگهداشت بازیکنان کمک کند. بازی‌های رایگان به شدت وابسته به تعداد کاربران فعال خود هستند، زیرا درآمد اصلی آن‌ها از طریق تعامل مداوم کاربران حاصل می‌شود. اگر بازی برای بازیکنان چالش‌های متناسب و قابل دستیابی ایجاد کند، آن‌ها بیشتر به ادامه بازی ترغیب می‌شوند. سیستم تنظیم پویای سطح سختی می‌تواند با حفظ یک چالش مناسب، از خستگی و ترک بازی جلوگیری کند و بازیکنان را برای مدت طولانی‌تری در بازی نگه دارد.

۵. تشویق به انجام خریدهای درون‌برنامه‌ای: یکی از اهداف بازی‌های رایگان تشویق بازیکنان به انجام خریدهای درون‌برنامه‌ای است [۱۱]. سیستم تنظیم پویای سطح سختی می‌تواند به این هدف کمک کند. برای مثال، زمانی که بازیکن با چالشی روبه‌رو می‌شود که به طور منطقی قابل حل است اما ممکن است زمان زیادی ببرد، سیستم تنظیم پویای سطح سختی می‌تواند سطح دشواری را کمی بالاتر ببرد تا بازیکن انگیزه پیدا کند که با خرید آیتم‌ها یا ارتقاءهای خاص، مسیر خود را تسریع ببخشد. به این ترتیب، سیستم تنظیم پویای سطح سختی می‌تواند به طور غیرمستقیم بازیکنان را به خرید بیشتر ترغیب کند [۱۲]. بدون این که آن‌ها را مجبور به این کار کند.

۶. تطبیق‌پذیری با مخاطبان مختلف: بازی‌های رایگان معمولاً مخاطبان گسترده‌ای دارند که شامل افرادی با سطوح مختلف تجربه و مهارت در بازی می‌شوند. سیستم تنظیم پویای سطح سختی به بازی‌ها این امکان را می‌دهد که به صورت پویا با نیازهای این طیف گسترده از بازیکنان تطبیق پیدا کنند. بازیکنان تازه‌کار می‌توانند با تنظیمات آسان‌تر به بازی وارد شوند و به تدریج مهارت‌های خود را بهبود بخشند، در حالی که بازیکنان با تجربه‌تر می‌توانند با چالش‌های سخت‌تر

بازیکن به راحتی از مراحل عبور می‌کند، بازی ممکن است این آیتم‌ها را کمتر در دسترس قرار دهد.

۳. تنظیم زمان‌بندی و محدودیت‌ها: برخی از بازی‌های جورکردنی دارای محدودیت‌های زمانی برای تکمیل یک مرحله هستند. سیستم تنظیم پویای سطح سختی می‌تواند با تنظیم این محدودیت زمانی، سطح دشواری بازی را تغییر دهد. برای مثال، اگر بازیکن با یک مرحله مشکل دارد، ممکن است زمان بیشتری برای تکمیل آن مرحله دریافت کند. به همین ترتیب، برای بازیکنانی که با سرعت زیادی پیشرفت می‌کنند، زمان کمتر ممکن است به چالش اضافه شود.

۴. افزایش یا کاهش پیچیدگی موانع: در بازی‌های جورکردنی، موانعی مانند قفل‌ها، یخ‌ها یا سایر مواردی که حرکت‌های بازیکن را محدود می‌کنند، نقش مهمی در تعیین سطح دشواری دارند. سیستم پویای تنظیم سطح سختی می‌تواند با تنظیم تعداد و پیچیدگی این موانع، تجربه بازی را برای بازیکن متناسب کند. اگر بازیکن در چند مرحله با موفقیت و بدون مشکلات جدی پیشرفت کرده است، بازی می‌تواند با اضافه کردن موانع جدید یا افزایش پیچیدگی آن‌ها، چالش‌های بیشتری را ارائه دهد.

۵. تحلیل و پاسخ به الگوهای بازی بازیکن: تحلیل داده‌های مربوط به نحوه بازی بازیکن می‌تواند به بازی کمک کند تا به طور دقیق‌تری سطح دشواری را تنظیم کند. با تحلیل الگوهای بازی، مثل این‌که بازیکن در چه نوع مرحله‌ای بهتر عمل می‌کند یا کدام نوع چالش‌ها برای او سخت‌تر است، سیستم پویای تنظیم سطح سختی می‌تواند تغییرات دقیق‌تری را اعمال کند. این روش به بازی اجازه می‌دهد تا به طور شخصی‌سازی شده‌ای به نیازهای هر بازیکن پاسخ دهد و تجربه‌ای منحصر به فرد برای هر بازیکن ایجاد کند.

۶. تغییر در نحوه ارائه مراحل جدید: برخی بازی‌های جورکردنی از سیستم پویای تنظیم سطح سختی برای

روی راهبرد ممکن است از تنظیم پویای سطح سختی استفاده کند تا مطمئن شود که بازیکنان همیشه با چالش‌های راهبردی روبه‌رو هستند، در حالی که یک بازی ساده‌تر ممکن است از تنظیم پویای سطح سختی برای حفظ جریان روان و بدون استرس بازی استفاده کند.

۱-۲-۳- انواع روش‌های پیاده‌سازی تنظیم پویای سطح سختی در بازی‌های جورکردنی

روش‌های پیاده‌سازی تنظیم پویای سطح سختی در بازی‌های جورکردنی، به دلیل ساختار خاص این بازی‌ها، به طور گسترده‌ای متنوع هستند [۱۶]. در ادامه، به برخی از مهم‌ترین روش‌های پیاده‌سازی سیستم‌های تنظیم پویای سطح سختی در این نوع بازی‌ها پرداخته می‌شود:

۱. تغییر الگوریتم تولید پازل: یکی از رایج‌ترین روش‌های پیاده‌سازی سیستم تنظیم پویای سطح سختی در بازی‌های جورکردنی، تغییر الگوریتم تولید پازل است. این روش شامل تنظیم هوشمندانه ترتیب و ترکیب آیتم‌های موجود در هر مرحله بر اساس عملکرد بازیکن است. اگر بازیکن در چند مرحله پی‌درپی عملکرد ضعیفی داشته باشد، بازی می‌تواند به طور پویا ترکیب‌های آسان‌تر و فرصت‌های تطابق بیشتری ایجاد کند. برعکس، اگر بازیکن به سرعت مراحل را پشت سر می‌گذارد، پازل‌ها می‌توانند چالش‌برانگیزتر شوند تا مهارت‌های بازیکن را بهتر به چالش بکشند.

۲. تنظیم قدرت آیتم‌های تقویتی: در بسیاری از بازی‌های جورکردنی، آیتم‌هایی تقویتی وجود دارند که به بازیکن کمک می‌کنند تا موانع را سریع‌تر برطرف کند. سیستم تنظیم پویای سطح سختی می‌تواند از طریق افزایش یا کاهش فراوانی این آیتم‌ها بر اساس عملکرد بازیکن عمل کند. به عنوان مثال، اگر بازیکن در حال رد کردن یک مرحله است، بازی می‌تواند آیتم‌های تقویتی بیشتری ارائه دهد تا به او کمک کند. از طرف دیگر، اگر

بحث می‌شود که تنظیم سطح سختی بازی‌ها باید بر اساس مهارت‌ها و توانایی‌های بازیکنان به صورت بیدرنگ و پویا انجام شود تا تجربه بهتری از بازی برای کاربران ایجاد شود. این پژوهش بخصوص به بحث تطبیق چالش‌های بازی با توانایی بازیکن در زمان‌های بحرانی پرداخته و نشان داده است که این تنظیمات نه تنها می‌توانند سطح رضایت بازیکن را افزایش دهند بلکه می‌توانند میزان ماندگاری او در بازی را بیشتر کنند. ماندگاری بالاتر بازیکنان به همراه تجربه بهتر از بازی در نهایت به درآمد بیشتر و کمک بیشتر به رشد این صنعت منتج می‌شود. این تحقیق به یکی از مبانی اصلی پژوهش‌های بعدی در زمینه تنظیم هوشمند سطح سختی تبدیل شد [۱۸]. همچنین، در پژوهشی دیگر به طراحی بازی‌هایی همراه با سیستم تنظیم پویای سطح سختی در جهت ایجاد تجربه کاربری مفیدتر پرداخته شده است [۱۹]. در پژوهشی دیگر به تاثیر پیاده‌سازی درست سیستم تنظیم پویای سطح سختی در بازی‌های رایانه‌ای پرداخته شده است [۲۰]. این پژوهش بیان می‌کند که در صورت پیاده‌سازی موثر این سیستم، میزان رضایت از تجربه کاربری کلی بازی می‌تواند تا حد زیادی افزایش پیدا کند. اما ضعف این پژوهش‌ها در این است که از سیستم‌های غیرمنعطف دودویی استفاده کرده‌اند. استفاده از این سیستم‌ها دقت کافی را ندارد و می‌تواند با کمی تغییر در رفتار کاربر منجر به عوض شدن ناگهانی و زیاد رفتار سیستم تنظیم پویای سطح سختی شود. استفاده از سیستم‌های کنترلی فازی به دلیل پیوستگی در اعلام نتیجه نهایی این مشکل را حل خواهد کرد. هوش مصنوعی و یادگیری ماشین در سال‌های اخیر به عنوان ابزاری کلیدی برای تنظیم سطح سختی بازی‌ها مورد توجه قرار گرفته است. یاناکاکیس^۷ و هالم^۸ از پیشگامان استفاده از شبکه‌های عصبی و یادگیری تقویتی برای تحلیل و پیش‌بینی رفتار بازیکنان بودند. آن‌ها در مقاله خود به استفاده از این روش‌ها برای تنظیم سطح

تنظیم ترتیب مراحل استفاده می‌کنند. اگر بازی شامل مراحل متوالی با دشواری‌های مختلف است، می‌توان از سیستم پویای تنظیم سطح سختی برای ارائه مراحل به گونه‌ای استفاده کرد که بازیکن ابتدا با مراحل آسان‌تر روبه‌رو شود و سپس به تدریج با مراحل دشوارتر مواجه گردد. این روش می‌تواند به کاهش ناامیدی بازیکنان کمک کند و آن‌ها را برای مقابله با چالش‌های سخت‌تر آماده کند.

۷. ایجاد مراحل ویژه: برخی بازی‌های جورکردنی از سیستم پویای تنظیم سطح سختی برای ایجاد مراحل ویژه استفاده می‌کنند که به بازیکنان اجازه می‌دهد تا در صورت شکست‌های مکرر در یک مرحله، مراحل آسان‌تری را پشت سر بگذارند و به بازی برگردند. این مراحل می‌توانند به حفظ انگیزه بازیکنان کمک کنند و آن‌ها را از ترک بازی منصرف کنند.

۸. ارائه پاداش‌های زمانی یا چالشی: در برخی بازی‌های جورکردنی، بازیکنان می‌توانند در طی زمان‌های خاص یا پس از تکمیل چالش‌های خاص، پاداش‌هایی ویژه دریافت کنند [۱۷]. سیستم پویای تنظیم سطح سختی می‌تواند با تنظیم زمان یا شرایط دریافت این پاداش‌ها بر اساس عملکرد بازیکن، به تنظیم سطح دشواری کمک کند؛ اگر بازیکن به طور مداوم چالش‌ها را به راحتی تکمیل می‌کند، ممکن است پاداش‌ها کمتر یا دریافت آن‌ها سخت‌تر شود و برعکس.

۳- کارهای مرتبط

یکی از اولین مطالعات مهم در این حوزه به مفهوم تنظیم وفق‌پذیر سطح سختی^۶ می‌پردازد که برای اولین بار در سال ۲۰۰۵ معرفی شد. در این پژوهش به نقش سازگاری سیستم بازی با مهارت بازیکنان پرداخته شده و توضیح داده شده که چطور تغییر سطح سختی در لحظات بحرانی بازی می‌تواند بهبود تجربه بازیکن و جلوگیری از رها کردن بازی را به همراه داشته باشد. در این مقاله

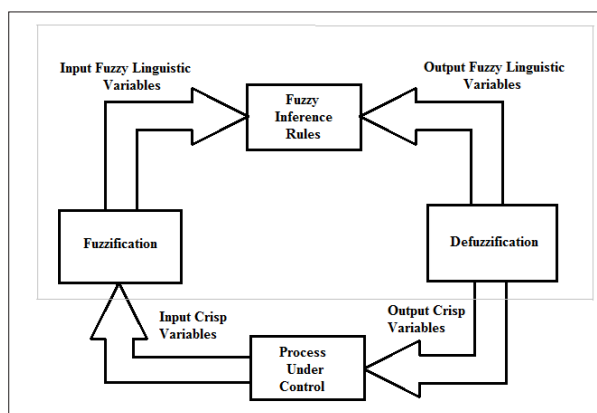
7-Yannakakis
8-Hallam

6-Adaptive Difficulty Adjustment (ADA)

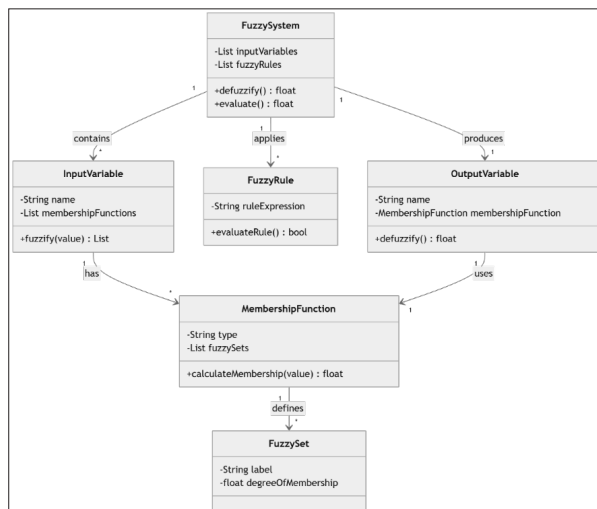
سختی در بازی‌های کامپیوتری اشاره کردند [۲۱]. در این مقاله، به استفاده از شبکه‌های عصبی و یادگیری تقویتی در تنظیم سطح سختی بازی‌ها پرداخته می‌شود. آنها سیستمی را توسعه دادند که می‌تواند رفتار بازیکنان را در طول بازی تحلیل کند و بر اساس این داده‌ها سطح سختی را به صورت پویا تنظیم نماید. یکی از نکات برجسته این مقاله، استفاده از روش‌های یادگیری ماشین برای پیش‌بینی میزان لذت و سرگرمی کاربران است که می‌تواند بر اساس داده‌های حاصل از تعاملات بازیکن، سطح سختی بازی را بهینه‌سازی کند. این پژوهش جزء اولین مقالاتی است که به کاربرد یادگیری ماشین در این حوزه پرداخته است و به عنوان مرجع بسیاری از مطالعات بعدی شناخته می‌شود. تحقیقات دیگری نیز از روش‌های پیشرفته‌تری مانند خوشه‌بندی ۹ بازیکنان با استفاده از داده‌های بزرگ بهره برده‌اند. به عنوان مثال، در مقاله‌ای، الگوریتم‌های خوشه‌بندی برای تقسیم بازیکنان به دسته‌های مختلف و تنظیم سطح سختی متناسب با رفتار آن‌ها استفاده شده است [۲۲]. در این پژوهش از داده‌های به‌دست آمده از تعاملات بازیکنان مانند مدت‌زمان بازی، تعداد شکست‌ها و پیروزی‌ها و میزان خریدهای درون بازی استفاده شده است تا بازیکنان به دسته‌های مختلف تقسیم شوند. سپس با توجه به هر دسته، سطح سختی بازی به صورت متفاوت تنظیم می‌شود. در این مقاله از تکنیک‌های خوشه‌بندی برای دسته‌بندی کاربران استفاده می‌شود و نشان می‌دهد که این روش‌ها می‌توانند تجربه کاربری را بهبود بخشند و منجر به افزایش ماندگاری بازیکنان در بازی شوند. همچنین، استفاده از تکنیک‌های پیشرفته‌تر مانند یادگیری عمیق در کارهای جدیدتر دیده می‌شود. استفاده از یادگیری عمیق برای پیش‌بینی رفتار بازیکن و تنظیم سطح سختی می‌تواند دقت تنظیمات را بهبود بخشد. در مقالات علمی به استفاده از یادگیری عمیق برای پیش‌بینی رفتار بازیکنان و تنظیم سطح سختی بازی‌ها پرداخته شده است. آنها

از شبکه‌های عصبی عمیق^{۱۰} برای تحلیل داده‌های بزرگ حاصل از رفتار بازیکنان استفاده کردند و سیستمی را توسعه دادند که می‌تواند به طور هوشمندانه سختی بازی را با توجه به سطح مهارت و سبک بازی کاربران تنظیم کند. این تحقیق نشان می‌دهد که یادگیری عمیق می‌تواند سطح دقت و شخصی‌سازی در تنظیم سختی بازی‌ها را به شدت افزایش دهد و راهی نوین برای تنظیم پویا سختی فراهم کند. اما مشکل استفاده از روش‌های مبتنی بر داده در هوش مصنوعی مانند یادگیری عمیق این است که ممکن است داده کافی در اختیار نباشد. برای مثال در اوایل یا پیش از انتشار یک بازی داده کافی از رفتار کاربران آن بازی وجود ندارد. استفاده از داده‌های دیگر بازی‌های مشابه نیز ممکن است با دقت کافی همراه نباشد زیرا هر بازی مختصات و شرایط ویژه خود را دارد. استفاده از یک سیستم کنترلی فازی می‌تواند در این شرایط بسیار کمک کننده باشد زیرا مبتنی بر داده نیست. سیستم کنترلی فازی مبتنی بر نظر متخصص است که می‌تواند طراح بازی باشد. یکی دیگر از حوزه‌های مهم پژوهشی مرتبط با این موضوع، تحلیل رفتار بازیکنان و پیش‌بینی سطح سختی مورد نیاز برای هر بازیکن است. در پژوهشی جدید، مدل‌های پیش‌بینی‌کننده برای تحلیل رفتار بازیکنان استفاده شده است که نتایج آن به تنظیم خودکار سطح سختی کمک می‌کند [۲۳]. در این مقاله به استفاده از داده‌های حاصل از تحلیل رفتاری بازیکنان برای مدل‌سازی رفتار آنها و تنظیم سطح سختی بازی‌ها پرداخته است. تحلیل رفتاری به جمع‌آوری و تحلیل داده‌های واقعی کاربران در طول بازی اشاره دارد. این مقاله به معرفی مدل‌های پیش‌بینی‌کننده‌ای می‌پردازد که از داده‌های رفتاری بازیکنان برای تعیین سطح مهارت و چالش مناسب در بازی استفاده می‌کنند. این تحقیق نشان داد که مدل‌سازی بازیکنان بر اساس رفتارشان می‌تواند به طور موثری تنظیمات پویا و دقیق‌تری را ایجاد کند. همچنین، تحقیقات دیگر به استفاده از داده‌های جمع‌آوری شده از بازیکنان و ترکیب آن با

مشاهده می‌کنید. این سیستم به صورت کلی دارای ۴ بخش است. بخش اول جمع‌آوری‌کننده داده است. بخش بعدی با استفاده از داده گرفته شده، آن را تبدیل به شکلی قابل فهم برای بخش سوم یعنی موتور اصلی یک سیستم کنترلی فازی می‌کند. در این بخش با استفاده از قوانین تعریف شده برای سیستم و ورودی داده شده، خروجی به صورت یک مجموعه فازی به دست می‌آید. بخش آخر نیز مجموعه فازی به دست آمده را تبدیل به داده خشک نهایی برای استفاده در خروجی می‌کند.



شکل ۲: معماری کلی سیستم کنترلی فازی [۲۲].



شکل ۳: نمودار کلاس سیستم پیشنهادی

۴-۳- پیاده‌سازی

ابتدا برای هر کدام از توابع عضویت هر کدام از

روش‌های هوش مصنوعی برای ایجاد یک سیستم تطبیقی سطح سختی پرداختند. آنها همچنین به تولید محتوای رویه‌ای^{۱۱} و استفاده از مدل‌سازی بازیکنان برای تنظیم سطح سختی پرداخته‌اند. آنها از شبکه‌های عصبی برای ایجاد تغییرات خودکار در محتوا و مراحل بازی و همچنین تنظیم چالش‌های بازی بر اساس سطح مهارت کاربران استفاده کردند. این روش ترکیبی از تولید خودکار محتوا و تنظیم هوشمند سختی است که می‌تواند تجربه کاربری را بهبود بخشد و نیاز به مداخله انسانی را به حداقل برساند. اما همچنان نقطه ضعفی که پیشتر نیز توضیح داده شد در این مدل‌ها دیده می‌شود. نبود داده کافی امری است که مدل‌های مبتنی بر داده را از ارائه نتایج مطلوب باز می‌دارد.

۴- روش پیشنهادی

۴-۱- مقدمه

در این بخش به بررسی روش پیشنهادی جهت تنظیم پویای سختی در بازی‌های جورکردنی با استفاده از سیستم‌های فازی پرداخته شده است. برای استفاده از تکنیک فازی در جهت توسعه سیستمی که بتواند سطح سختی بازی را کنترل کند، ابتدا باید سیستمی کلی طراحی شود که در آن امکان پیاده‌سازی هر نوع سیستم فازی وجود داشته باشد. این سیستم را می‌توان به روش‌های مختلف و با استفاده از زبان‌های مختلف برنامه‌نویسی توسعه داد. از آنجایی که هدف این پژوهش در نهایت توسعه یک سیستم فازی برای استفاده در بازی‌های موبایلی است، از زبان برنامه‌نویسی سی شارپ و موتور بازی‌سازی یونیتی استفاده شده است.

۴-۲- توضیح کلی معماری سیستم

معماری کلی یک سیستم کنترلی فازی را در شکل (۲) مشاهده می‌کنید. نمودار کلاس این سیستم بر مبنای قواعد ترسیم UML را نیز در شکل (۳) مشاهده می‌کنید. همچنین شبکه‌کد سیستم فازی طراحی شده را در شکل (۴)

```

# For each input variable
FOR each input_variable IN input_variables:
    # Evaluate the degree of membership for each membership function
    FOR each membership_function IN input_variable.membership_functions:
        degree = evaluate_membership(input_variable, membership_function)

# For each rule
FOR each rule IN rules:
    firing_strength = 1
    # Apply fuzzy logic operators (AND, OR) on the membership degrees of input variables
    FOR each condition IN rule.conditions:
        IF condition.operator == 'AND':
            firing_strength = min(firing_strength, condition.membership_degree)
        ELSE IF condition.operator == 'OR':
            firing_strength = max(firing_strength, condition.membership_degree)

    # Determine the output membership function based on the rule's consequent
    output_membership_function = rule.consequent.output_membership_function

    # Calculate contribution of each output membership function using firing strength
    output_contribution = firing_strength * output_membership_function.degree

# Aggregate contributions of all output membership functions using fuzzy logic (MAX, SUM)
aggregated_result = aggregate_output_contributions(output_membership_functions)

# Defuzzify the aggregated result to obtain a crisp output value
crisp_output = defuzzify(aggregated_result)

```

شکل ۴: شبه کد سیستم کنترلی فازی

به تعداد لازم ورودی جدید وارد کنیم. همچنین هر یک از ورودی‌ها باید بتوانند به تعداد لازم مجموعه مشخص داشته باشند. همچنین هر یک از مجموعه‌ها باید قابلیت تعریف نمودار مخصوص خود را داشته باشند. در آخر ما می‌خواهیم سازوکاری داشته باشیم که به کمک آن بتوانیم مقداری آزمایشی در نمودار ورودی ایجاد شده وارد کنیم و ببینیم که اگر ورودی جدید این مقدار را داشته باشد، درصد عضویت آن در هر کدام از مجموعه‌هایی که برای این ورودی مشخص کرده‌ایم چه قدر است. از آنجایی که قرار است متخصصین این حوزه یعنی طراحان بازی این ورودی‌ها را برای سیستم‌های جدید وارد کنند، پس بهتر است پنجره‌ای درست کنیم که افراد غیر برنامه‌نویس (مثل طراحان بازی) با آن راحت باشند و بتوانند سیستم‌های جدیدی را به راحتی و داخل محیط گرافیکی موتور یونیتی و بدون نیاز به کدنویسی طراحی کنند که در شکل (۵) نشان داده شده است.

بخش بعدی که باید آن را تعریف کنیم، بخش تعریف قوانین سیستم است. نیازمندی‌های ما در این بخش، وجود پنجره‌ای است که بتوان در آن قوانین سیستم فازی در حال توسعه را بر اساس ورودی‌های تعریف شده، تعریف کرد.

ورودی‌ها با توجه به ورودی قطعی داده شده، مقدار آن تابع عضویت را به دست می‌آوریم. سپس برای هر کدام از قوانین فازی که تعریف کرده‌ایم، درجه درستی آن قانون را با توجه به مقادیری که برای توابع عضویت در بخش قبلی مشخص شد، به دست می‌آوریم. اگر قانون نوشته شده، دارای «و» منطقی بود، از روش کمینه‌گیری استفاده می‌کنیم. اگر قانون نوشته شده دارای «یا» منطقی بود، از روش بیشینه‌گیری بین دو مقدار استفاده می‌کنیم. اگر قانون نوشته شده دارای نقیض منطقی بود، از روش تفریق مقدار تابع عضویت از عدد یک استفاده می‌کنیم. با ترکیب نمودار هر کدام از توابع عضویت خروجی که از هر کدام از آن‌ها به اندازه صحت قانون مربوطه به آن برداشته شده، به یک شکل واحد می‌رسیم. در بخش آخر مرکز ثقل این شکل را محاسبه می‌کنیم و مکان این نقطه روی محور عرض‌ها ما را به خروجی قطعی می‌رساند. برای پیاده‌سازی سیستم فازی در موتور بازی‌سازی یونیتی، باید این سیستم را ابتدا به بخش‌های مختلف تقسیم کنیم. ابتدا بخش مشخص کردن نوع هر یک از ورودی‌ها و توابع عضویت متعلق به هر یک از ورودی‌ها را پیاده‌سازی می‌کنیم. نیازمندی‌های ما در این بخش وجود پنجره‌ای است که از طریق آن بتوانیم

دیگر رابطه شرطی باید یکی از مجموعه‌های خروجی که در نهایت، آن را هم تعریف می‌کنیم، وارد شود. پنجره ایجاد شده در نهایت به شکل (۶) است.

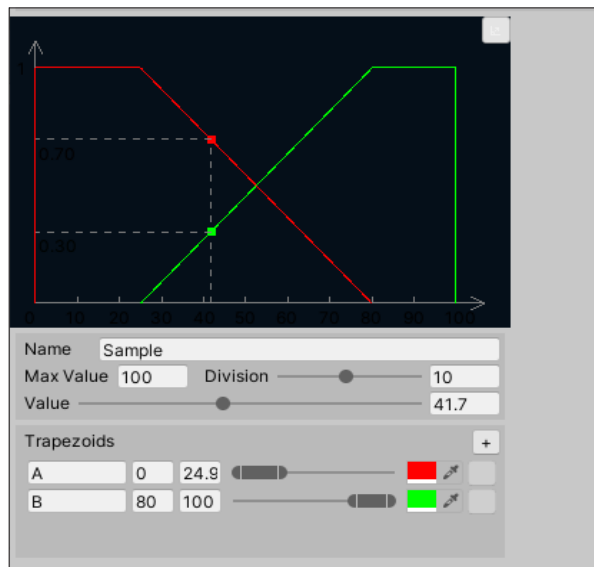


شکل ۶: پنجره طراحی شده برای تعریف قوانین سیستم فازی

برای بخش تعریف خروجی، نیازمندی‌های ما در این بخش بسیار مشابه بخش اول یعنی تعریف ورودی است. تفاوت در این است که ما فقط یک خروجی داریم و فقط باید برای خروجی اسم انتخاب کنیم و تعیین کنیم که خروجی ما دقیقاً چه باید باشد. اما باید قابلیت تعریف تعداد دلخواهی از مجموعه‌ها را برای خروجی خود داشته باشیم. به دلایل مشابه که در دو بخش قبلی توضیح داده شد به سراغ پیاده‌سازی این بخش در پنجره‌های قابل تغییر در موتور یونیتی می‌رویم. یکی از مثال‌های عملی برای این بخش یعنی خروجی سیستم فازی می‌تواند مدت زمان محدود برای رد کردن یک مرحله در یک بازی جورکردنی باشد. پنجره کدنویسی شده در نهایت به شکل (۷) است.

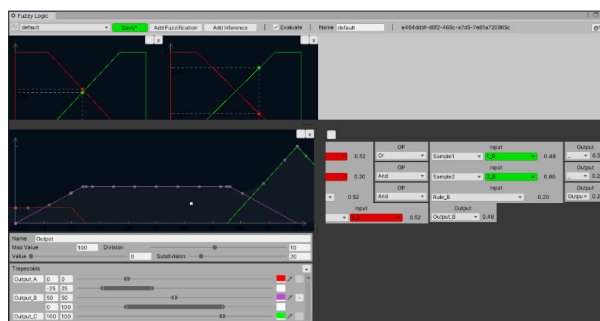
در نهایت باید بخش‌های مختلف ساخته شده را در کنار هم قرار دهیم تا در یک سیستم کلی قابل تغییر و ساخت باشد. در این پنجره کلی ما باید بتوانیم سیستم فازی جدید طراحی کنیم و سیستم‌های قبلی را در صورت نیاز تغییر دهیم. همچنین باید بتوانیم در هر سیستم، ورودی جدید تعریف کرده یا ورودی‌های قبلی را حذف کنیم یا تغییر دهیم. همچنین باید بتوانیم خروجی خود را با هر تعداد مجموعه مورد نیاز که هر کدام ممکن است توابع خاص خود را داشته باشند، تعریف کنیم یا تغییر دهیم. در نهایت باید بتوانیم قوانین فازی دلخواه خود را تعریف کرده و ورودی‌های سیستم خود را به خروجی ربط دهیم. همچنین امکانی می‌خواهیم تا با مشخص

باید بتوانیم هر کدام از مجموعه‌های هر کدام از ورودی‌ها را به هر کدام از مجموعه‌های هر ورودی دیگری ترکیب کنیم.



شکل ۷: پنجره طراحی شده برای تعریف ورودی سیستم

نوع این ترکیب می‌تواند اشتراکی (و) یا اجتماعی (یا) باشد. همچنین باید قابلیت نقیض کردن هر کدام از آن‌ها را نیز داشته باشیم. قابلیت دیگری که به آن نیاز داریم، توانایی ایجاد گزاره‌های شرطی ترکیبی با هر تعدادی که می‌خواهیم است. یعنی ما باید بتوانیم شرطی داشته باشیم که در آن از تمام مجموعه‌های ورودی‌ها و از تمام ورودی‌ها استفاده کرده باشیم. همان‌طور که در بخش قبل نیز گفته شد، محیط تعریف این قوانین فازی باید گرافیکی باشد زیرا متخصصی که با این سیستم کار می‌کند الزاماً برنامه‌نویس نیست. پس دوباره به سراغ استفاده از امکانی که موتور یونیتی در ساخت پنجره‌های قابل تغییر به ما می‌دهد می‌رویم. برای داشتن قابلیت شرط‌های ترکیبی، به سراغ تکنیک زنجیر کردن عبارت‌های شرطی کوچک و دوتایی یا تکی (در صورت استفاده از نقیض) می‌رویم. به این صورت که هر دو عبارت با هم ترکیب شده و سپس عبارت‌ها، خود با یکدیگر ترکیب می‌شوند و این روند زنجیری می‌تواند تا بی‌نهایت ادامه داشته باشد. در سمت



شکل ۸: پنجره کلی تعبیه شده برای توسعه یک سیستم کامل فازی.

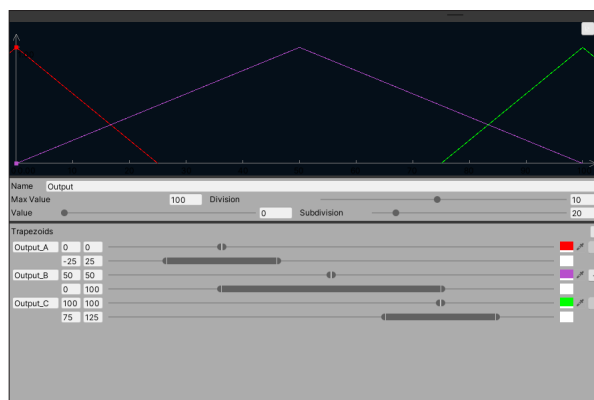
۴-۴- پیاده سازی سیستم تنظیم سطح سختی با استفاده از منطق فازی

معیارهای زیادی می تواند در تنظیم سطح سختی بازیکن مورد استفاده باشد که ما شش مورد از آن ها را که بیشتر مورد استفاده هستند انتخاب کرده ایم. برای تعریف هر کدام از آن ها از هفت مجموعه استفاده کرده ایم که به ترتیب «خیلی خیلی زیاد»، «خیلی زیاد»، «زیاد»، «متوسط»، «کم»، «خیلی کم» و «خیلی خیلی کم» هستند. توابع هر کدام از مجموعه ها را به صورت مثلثی و با توزیع یکسان و با تقارن بین کمترین مقدار و بیشترین مقدار تعریف شده برای هر ورودی تعریف می کنیم. در هر کدام از ورودی ها اگر مقدار وارد شده بیش از بالاترین حد یعنی حد «خیلی خیلی زیاد» باشد، ما آن را صد در صد عضو مجموعه خیلی خیلی زیاد و اگر مقدار وارد شده کمتر از پایین ترین حد تعریف شده یعنی «خیلی خیلی کم» باشد، ما آن را صد در صد عضو مجموعه خیلی خیلی کم در نظر می گیریم. در ادامه به توضیح تک به تک هر یک از ورودی ها پرداخته می شود.

۴-۱-۴- تعداد تلاش برای رد کردن مرحله

این معیار نشان دهنده این است که کاربر چندبار در آن مرحله باخته است. این معیار بعد از هر بار بردن یک مرحله صفر می شود. اگر تعداد باخت کاربر در یک مرحله به خصوص زیاد شود، باید بازی را برای کاربر آسان کنیم چون بیش از حد در یک مرحله گیر کردن می تواند باعث

کردن مقدار آزمایشی هر کدام از ورودی ها مقدار نهایی خروجی را مشاهده کنیم. در نهایت پنجره کلی ساخته شده در موتور یونیتی به شکل (۸) است که مشاهده می کنید.



شکل ۷: پنجره طراحی شده برای تعریف خروجی سیستم فازی

در این پنجره مقدار خروجی با در نظر گرفتن شکل کلی ساخته شده از مجموعه های خروجی مشخص شده است. در نهایت باید منطق اصلی یک سیستم فازی که در پشت این پنجره کار می کند را بنویسیم. برای این کار از شکل کلی یک سیستم مبتنی بر منطق فازی که در بخش دوم توضیح داده شد استفاده می کنیم. شایان ذکر است که برای تعیین شرط «یا» از روش بیشینه گیری و برای تعیین شرط «و» از روش کمینه گیری استفاده می کنیم. برای تعیین خروجی نیز از روش یافتن مرکز ثقل استفاده می کنیم یعنی مرکز ثقل شکل نهایی به دست آمده پس از مرحله تفهیم را به دست می آوریم و خروجی نهایی سیستم ما مختصات مرکز ثقل در محور افقی خواهد بود. این سیستم در نهایت قبل از زمان اجرا از پیش آماده^{۱۲} خواهد بود و زمان دسترسی به خروجی در آن در زمان اجرا از 0(1) پیروی می کند. برای یکپارچه سازی سیستم طراحی شده درون بازی، ما باید ورودی های تعریف شده را در زمان اجرا یک به یک مقداردهی کنیم و سیستم خروجی را به ما برمی گرداند.

۴-۳-۴- مقدار کل خرید انجام شده توسط کاربر

این معیار نشان‌دهنده کل مبلغ خرید انجام شده توسط کاربر در طول مدتی که به بازی پرداخته است می‌شود. مشابه قسمت قبلی، اینجا هم اگر کاربر خرید بیشتری داشته باشد، باید بازی را برای او آسان کنیم تا در ناخودآگاه کاربر این تفکر به وجود بیاید که اگر پول بیشتری در بازی خرج کند، منجر به قوی‌تر شدن و راحت‌تر رد کردن مراحل می‌شود. برعکس آن برای کمتر خرج کردن و سخت‌تر شدن بازی نیز با همان منطق صادق است.

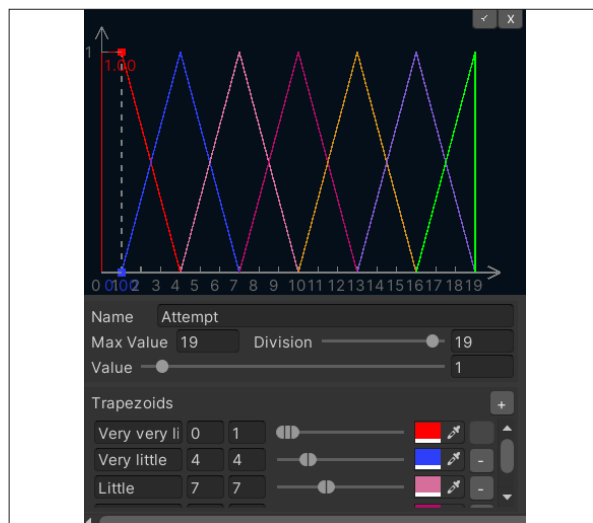
۴-۳-۵- فاصله زمانی بین دو جلسه بازی کاربر

واحد این معیار را ساعت در نظر می‌گیریم. اگر این عدد زیاد باشد باید بازی را برای کاربر آسان کنیم زیرا زیاد بودن این عدد می‌تواند نشان‌دهنده این حقیقت باشد که علاقه کاربر به بازی به دلیل پیشروی بیشتر و سخت‌تر شدن تدریجی مراحل بالاتر کم شده است. ما می‌توانیم با آسان‌تر کردن مراحل از حس خستگی و ناامیدی ناشی از باخت زیاد کم کنیم و کاربر را دوباره به بازی علاقه‌مند کنیم. در آن سمت، اگر این عدد پایین باشد نشان از علاقه زیاد کاربر به بازی است. در این شرایط ما می‌توانیم با سخت‌تر کردن بازی، اجازه پیشروی آسان در مراحل را به او ندهیم و به همین خاطر باعث ترغیب بیشتر او به خرید کردن برای پیشروی شویم.

۴-۳-۶- میانگین زمانی مدت هر جلسه از بازی کاربر

واحد اندازه‌گیری این معیار را دقیقه در نظر می‌گیریم. این معیار نشان‌دهنده این است که هر جلسه از بازی کاربر چه مدت طول کشیده است. کم بودن این معیار، می‌تواند نشان‌دهنده این باشد که کاربر هر بار که شروع به بازی می‌کند، خیلی سریع می‌تواند به پیشروی مورد نیاز خود برسد و از نظر روانی تخلیه شده و بازی را ببندد. در این حالت ما باید بازی را برای کاربر سخت کنیم تا تعادل برقرار شود. دقیقاً برعکس این اتفاق در حالتی که این معیار زیاد باشد وجود دارد و ما در آن حالت باید بازی را برای

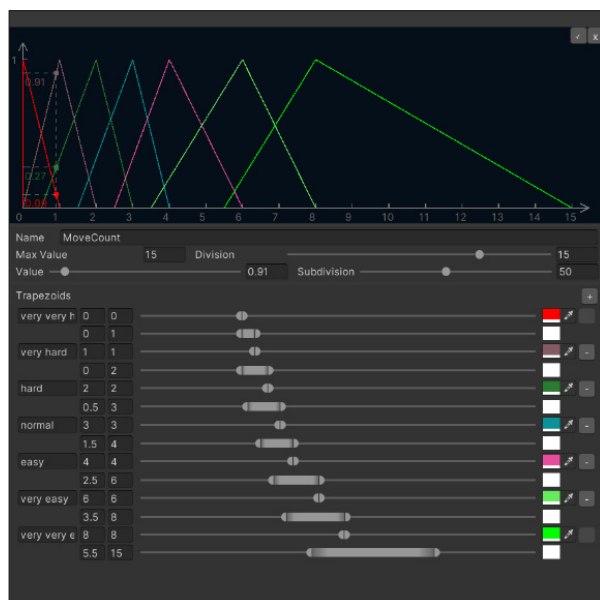
سرخوردگی و رها کردن بازی شود. همچنین اگر تعداد باخت کاربر در یک مرحله به‌خصوص کم باشد، باید بازی را برای کاربر سخت کنیم، زیرا باید کاربر را ترغیب به خرید بیشتر برای توانا شدن در رد کردن مرحله کنیم. شکل (۹) حالت نهایی ورودی تعداد تلاش برای رد کردن مرحله را نشان می‌دهد.



شکل ۹: تعریف ورودی «تعداد تلاش برای رد کردن مرحله»

۴-۲-۴- مقدار سکه خرج شده در مرحله

این معیار در حقیقت نشان‌دهنده این است که کاربر از زمانی که به یک مرحله به‌خصوص رسیده است، چقدر سکه خرج کرده است. اگر سکه خرج شده در یک مرحله زیاد شود، باید بازی را برای کاربر آسان کنیم زیرا این کار باعث این می‌شود که در ناخودآگاه کاربر این تفکر به وجود بیاید که خرج سکه بیشتر می‌تواند باعث رد کردن مرحله شود. در آن سمت، اگر مقدار سکه خرج شده توسط کاربر در یک مرحله کم باشد، باید بازی او سخت شود تا به صورت مشابه این تفکر برای کاربر به وجود بیاید که بدون خرج کردن سکه نمی‌توان در بازی پیشرفت کرد پس خرید سکه و سپس خرج آن تبدیل به یک نیاز می‌شود. محدوده خرج سکه را بین صفر تا ۱۲۹۵ در نظر می‌گیریم.



شکل ۱۰: تصویر خروجی تعریف شده بر روی تعداد حرکات مجاز برای هر مرحله.

می‌گذاریم. در آن سمت هر کدام از شرط‌ها نیز یکی از مجموعه‌های خروجی قرار می‌گیرد. پس هفت شرط داریم که در هر سمت از آن یکی از حالت‌های خروجی قرار دارد. مثلاً یکی از حالت‌ها «خیلی خیلی سخت» است. با توجه به توضیحاتی که در قسمت تعریف ورودی‌ها داده شد، برای هر کدام از ورودی‌ها یکی از حالت‌های «خیلی خیلی سخت» یا «خیلی خیلی آسان» را انتخاب می‌کنیم. توجه به این نکته مهم است که صفت قبل از هر مجموعه که از هر ورودی انتخاب می‌شود باید متناظر با صفت انتخابی آن مجموعه از خروجی باشد که در سمت نتیجه عبارت شرطی قرار دارد. یعنی «خیلی خیلی»، «خیلی» یا بدون صفت («متوسط»). نمونه‌ای از شروط ایجاد شده به شرح زیر است:

اگر تعداد باخت، خیلی خیلی کم و سکه خرج شده در مرحله خیلی خیلی کم و مقدار کل پول خرید در بازی خیلی خیلی کم و میانگین مدت فاصله بین جلسات بازی خیلی خیلی کم و میانگین طول مدت هر جلسه از بازی خیلی خیلی کم و مدت زمان گیر کرده در آن مرحله خیلی خیلی کم باشد، آنگاه بازی باید خیلی خیلی سخت شود. همان‌طور که مشاهده می‌شود شرط‌های نوشته شده کل

کاربر آسان کنیم تا تعادل برقرار شود.

۷-۳-۴- مدتی که کاربر در یک مرحله بخصوص گیر کرده است

واحد اندازه‌گیری این معیار را روز در نظر می‌گیریم. این معیار در حقیقت نشان‌دهنده این است که کاربر از زمان ورود به این مرحله چند روز نتوانسته آن را رد کند و به اصطلاح در آن مرحله گیر کرده است. محدوده این معیار را بین صفر تا ۱۰ روز در نظر می‌گیریم.

۸-۳-۴- تعریف تابع خروجی

با زیاد کردن تعداد حرکات مجاز، بازی آسان شده و با کم کردن این عدد بازی سخت می‌شود. هر مرحله خود یک عدد پایه برای تعداد حرکات مجاز دارد که در صورت نبود سیستم تنظیم پویای سختی همیشه و برای همگی کاربران استفاده می‌شود. خروجی ما به عدد پایه اضافه یا از آن کم می‌شود و در نهایت باعث سخت‌تر یا آسان‌تر شدن بازی می‌شود. مشابه ورودی‌ها، خروجی خود را نیز به هفت قسمت تقسیم می‌کنیم که به ترتیب «خیلی خیلی سخت»، «خیلی سخت»، «سخت»، «متوسط»، «خیلی آسان» و «خیلی خیلی آسان» هستند. محدوده خروجی خود را بین منفی هشت و مثبت هشت در نظر می‌گیریم. منفی هشت سخت‌ترین حالت ممکن و مثبت هشت آسان‌ترین حالت ممکن است. شکل (۱۰) نسخه نهایی خروجی تعریف شده برای بازی را نشان می‌دهد.

۹-۴-۳- تعریف قوانین سیستم کنترلی فازی

در این بخش باید ورودی خود را به خروجی ربط دهیم و قوانینی پایه‌گذاری کنیم که بتواند ایده کلی که در تعریف ورودی‌ها و خروجی خود داشتیم را پیاده‌سازی کند. از آنجایی که هر کدام از ورودی‌های ما هفت مجموعه و خروجی ما نیز هفت مجموعه دارد، پس هفت قانون جدید اضافه می‌کنیم که در هر کدام از آن‌ها یکی از هفت مجموعه هر کدام از ورودی‌ها انتخاب شده و بین آن‌ها «و» منطقی

Rule_A	Input	OP	Input	Output
x	Attempt	And	Coin Sink	0.70
Rule_B	Input	OP	Input	Output
x	Rule_A	And	Revenue	0.70
Rule_C	Input	OP	Input	Output
x	Rule_B	And	Session Diff	0.70
Rule_D	Input	OP	Input	Output
x	Rule_C	And	Session Length	0.70
Rule_E	Input	OP	Input	Output
x	Rule_D	And	Age Diff	0.70

شکل ۱۱: یکی از قوانین تعریف شده در سیستم کنترلی نهایی که به بازی خیلی خیلی سخت منتج می‌شود.

کنید. نتایج این پرسش‌نامه مشخص خواهد کرد که عملکرد سیستم هوشمند تنظیم پویای سطح سختی در مقایسه با حالتی که این سیستم وجود ندارد، در یک محیط یکسان (بازی گلشیفته)، چگونه است. معیارهای مختلفی مورد مقایسه و پرسش قرار گرفته‌اند؛ مقایسه میزان کل وقتی که کاربران در هر یک از نسخه‌ها گذرانده‌اند، مقایسه «میزان درگیر شدن کاربران با جهان بازی» در هر یک از نسخه‌ها، مقایسه «میزان خستگی کاربران از ادامه‌ی بازی» هر کدام از نسخه‌ها، مقایسه «میزان تمایل برای تماشای تبلیغ» در هر کدام از نسخه‌ها و مقایسه «میزان تمایل به خرید از درگاه پرداخت» در هر کدام از نسخه‌ها از معیارهای مورد بررسی در این آزمایش بوده‌اند. همچنین برخی شاخص‌های کلیدی مهم به صورت خودکار پایش شده و در یک پایگاه داده ذخیره شده است که نتایج آن را در ادامه مشاهده خواهید کرد. شایان ذکر است که شرکت‌کنندگان از این که کدام یکی از نسخه‌ها دارای سیستم تنظیم سطح سختی و کدام یک فاقد آن بوده است، اطلاعی نداشته‌اند.

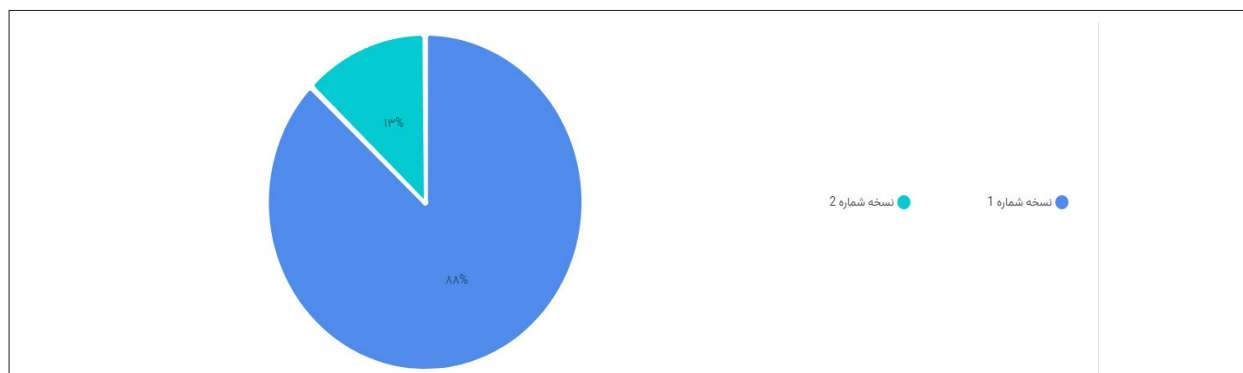
۵-۱- افراد شرکت‌کننده در آزمایش

دو نسخه متفاوت از بازی در اختیار ۳۰ شرکت‌کننده قرار گرفت. بعد از گذشت یک هفته، هر کدام از شرکت‌کننده‌ها حداقل چهار ساعت در هر کدام از بازی‌ها وقت گذرانده‌اند. محیط نسخه‌هایی که در اختیارشان قرار گرفته است، محیط محصولی واقعی که در بازار منتشر می‌شود بوده است و به درگاه پرداخت واقعی متصل بوده است. از نظر جنسیت، ۲۲ نفر از شرکت‌کنندگان آقا و بقیه خانم بوده‌اند. همچنین شش نفر بین ۱۲ تا ۱۸ سال، ۲۱ نفر

طیف را از خیلی خیلی سخت تا خیلی خیلی آسان پوشش می‌دهند. برای تعریف کردن این شرط‌ها، قوانین را با هم زنجیر می‌کنیم. یک نمونه از قوانین تعریف شده را در شکل (۱۱) مشاهده می‌کنید.

۵- ارزیابی رویکرد پیشنهادی

از سیستم پیاده‌سازی شده در بخش‌های قبلی در بازی گلشیفته (<https://cafebazaar.ir/app/info.me-> drick.golshifteh) استفاده خواهیم کرد. بازی گلشیفته یک بازی موبایلی جورکردنی است که در سال‌های اخیر جزو بازی‌های پرفروش و موفق بازار ایران بوده است. این بازی را می‌توانید از کافه بازار با پیوند داده شده بارگیری کنید. راه‌های زیادی برای بررسی عملکرد یک سیستم تنظیم پویای سختی وجود دارد. یکی از آنها این است که کاربران را به دو دسته تقسیم کنیم. یک دسته آنهایی که بازی‌ای که در اختیار دارند، سیستم تنظیم پویای سطح سختی را ندارد و مراحل را به صورت عادی بازی می‌کنند و دسته‌ی دیگر کاربرانی که بازی‌ای که در اختیار دارند، دارای سیستم تنظیم هوشمند سطح سختی است و به همین دلیل تعداد حرکات مجاز در هر مرحله برای آن‌ها طبق رفتاری که از خود نشان می‌دهند تغییر می‌کند. برای این منظور، دو نسخه متفاوت از بازی گلشیفته آماده شد که در اختیار گروهی از افراد با جنسیت، سن و شغل مشخص قرار گرفت. سپس پرسش‌نامه‌ای آماده شده است که بعد از یک هفته پیشروی در بازی به آن جواب خواهند داد. این پرسش‌نامه را می‌توانید از این پیوند (<https://survey.porsline.ir/s/cw8irDDO>) مشاهده



شکل ۱۲: نمودار دایره‌ای خرید انجام شده کاربران در نسخه شماره ۱ (دارای سیستم تنظیم سطح سختی) و نسخه شماره ۲ (نسخه بدون سیستم تنظیم سطح سختی)

جدول ۱: مقایسه شاخص‌های مهم کلیدی عملکرد در دو نسخه مورد آزمایش قرار گرفته.

شاخص مورد بررسی	میانگین زمان هر جلسه بازی	میانگین میزان پیشروی (تعداد مراحل)	نرخ تبدیل کاربران (نسبت کابرائی که خرید داشته‌اند به کل کاربران)	میزان خرید کل (هزار تومان)	تعداد تبلیغات کل دیده شده (میانگین در روز)
نسخه عادی	۱۴ دقیقه	۳۴ مرحله	۳٪	۲۹	۳۶
نسخه دارای سیستم تنظیم سطح سختی	۲۶ دقیقه	۴۱ مرحله	۹٪	۲۰۷	۷۳

بین ۱۸ تا ۳۰ سال و ۳ نفر بالای ۳۰ سال سن داشته‌اند. از نظر شغل، چهار نفر شغل خود را مرتبط با بازی، ۱۵ نفر شغل خود را نامرتبط با بازی، شش نفر دانش‌آموز و پنج نفر خود را دانشجوی معرفی کرده‌اند. شش نفر از این افراد خود را گیمر معرفی کرده‌اند. همان‌طور که مشاهده می‌شود پراکندگی شرکت‌کنندگان تا حد خوبی از جنبه‌های مورد نیاز در آزمایش تعیین شده‌است.

۲-۵- بررسی نتایج

طبق نتایج به دست آمده، ۷۵ درصد از کاربران با نسخه‌ای که دارای سیستم تنظیم سطح سختی به صورت پویا با استفاده از سیستم‌های فازی بوده‌است، ارتباط بیشتری برقرار کرده‌اند و وقت بیشتری در آن نسخه گذرانده‌اند که نشان از برتری این سیستم نسبت به نسخه عادی دارد. ۶۳ درصد از کاربران در نسخه عادی بدون سیستم تنظیم سطح سختی، احساس خستگی کم‌تری کرده‌اند که می‌تواند ناشی از سختی بیش از حد سیستم طراحی شده باشد. ۷۵ درصد از کاربران در نسخه‌ای که دارای سیستم تنظیم هوشمند سطح سختی بوده‌است،

بیشتر به دیدن تبلیغات برای به دست آوردن آیتم‌های درون بازی ترغیب شده‌اند. یکی از راه‌های درآمدزایی بازی‌های رایگان طبق توضیحات داده‌شده در فصل دوم، نشان دادن تبلیغات است که در این مورد نیز، اضافه شدن سیستم هوشمند تنظیم سطح سختی کمک شایانی به بیشتر دیده شدن تبلیغات کرده‌است. مهمترین معیار برای بررسی عملکرد این سیستم، میزان پرداختی و خرید کاربران در بازی است. طبق نتایج به دست آمده، ۷۵ درصد کاربران در نسخه دارای این سیستم، تمایل بیشتری نسبت به خرید نسبت به نسخه دیگر داشته‌اند. همچنین ۸۸ درصد کاربران در عمل خرید بیشتری در نسخه دارای سیستم هوشمند سطح سختی نسبت به نسخه دیگر داشته‌اند. نمودار دایره‌ای این معیار را در شکل (۱۲) مشاهده می‌کنید.

همچنین جدول تفاوت شاخص‌های کلیدی عملکرد در دو نسخه مورد آزمایش را طبق اطلاعات جمع‌آوری شده از ۳۰ فردی که مورد آزمایش قرار گرفته‌اند را در جدول (۱) مشاهده می‌کنید.

۶- نتیجه‌گیری و کارهای آینده

در این پژوهش سعی شد با استفاده از طراحی سیستمی که سطح سختی در بازی‌های جورکردنی را مبتنی بر سیستم‌های کنترلی فازی تنظیم می‌کند، به رشد شاخص‌های کلیدی عملکرد در بازی‌های جورکردنی برسیم. برای رسیدن به این هدف، ابتدا زیرساختی برای توسعه سیستم‌های کنترلی فازی در موتور بازی‌سازی یونیتی به صورت کلی توسعه داده شد. سپس با استفاده از زیرساخت آماده شده، یک سیستم کنترلی مبتنی بر منطق فازی که بتواند میزان سختی مراحل بازی را به صورت پویا و هوشمند و طبق رفتار کاربر تنظیم کند توسعه داده شد. در توسعه این سیستم از نظر متخصصین این حوزه یعنی طراحان بازی استفاده شده است. در این سیستم از شش ورودی تعداد تلاش (باخت)، مقدار خرید کاربر، سکه از دست داده شده در مرحله، میانگین مدت زمان جلسات بازی کاربر، میانگین زمانی فاصله بین جلسات کاربر و مدت زمانی که کاربر در آن مرحله گیر کرده است، استفاده شد. خروجی این سیستم تعداد حرکات مجاز اضافه یا کم شده از تعداد حرکات مجاز اصلی آن مرحله خواهد بود. سپس این سیستم در بازی گلشیفته یکپارچه‌سازی شده و دو نسخه از این بازی تهیه شد که یکی دارای سیستم تنظیم هوشمند میزان سختی مراحل بازی بوده و دیگر نسخه دارای این سیستم نبود. این دو نسخه در اختیار داوطلبان آزمایش قرار گرفت و بعد از انجام بازی توسط داوطلبان، پرسشنامه‌ای در اختیار آنان قرار گرفت تا عملکرد سیستم طراحی شده مورد بررسی قرار گیرد. طبق بررسی‌های انجام شده، سیستم هوشمند مبتنی بر منطق فازی برای تنظیم سطح سختی در بازی‌های جورکردنی باعث رشد خرید کاربران، بیشتر دیده شدن تبلیغات در بازی، درگیر شدن بیشتر کاربران با بازی و بیشتر شدن زمان بازی آن‌ها خواهد شد.

یکی از مشکلات سیستم کنترلی فازی طراحی شده برای تنظیم پویای سطح سختی در بازی‌های جورکردنی

این است که در حال حاضر کیفیت سیستم بستگی به نظر متخصص یا متخصصینی که آن را طراحی می‌کنند دارد. اولین مشکلی که این موضوع دارد این است که سیستم به اندازه کافی به صورت خودکار کار نمی‌کند و میزان تخصص و مهارت فرد طراح می‌تواند تاثیر مستقیم در میزان عملکرد آن بگذارد. این نکته برای بازی‌های کوچکتر که تازه منتشر شده‌اند به دلیل نبودن داده کافی از رفتار کاربران می‌تواند نکته‌ای مثبت باشد زیرا می‌توان با تکیه بر تجربه متخصصین، سیستمی قابل قبول برای شروع کار تولید کرد. اما در مورد بازی‌های بزرگتر که تعداد کاربر زیادی دارند و زمان نسبتاً زیادی از عرضه آنها می‌گذرد نکته‌ای منفی به شمار می‌رود زیرا می‌توان از ظرفیت داده کلان کاربران که در بازی وجود دارد استفاده کرد و از آن در جهت بهبود مداوم سیستم و بالاتر بردن دقت آن استفاده کرد. همچنین با بالاتر رفتن تعداد ورودی‌ها و قوانین فازی استفاده شده، سیستم با شدت زیادی شروع به پیچیده‌تر شدن می‌کند و نگهداری و تنظیم کردن عملکرد آن را سخت‌تر و سخت‌تر می‌کند. در آینده می‌توان با استفاده از تکنیک‌های مبتنی بر داده در هوش مصنوعی مانند یادگیری ماشین یا یادگیری عمیق و استفاده از داده‌های واقعی کاربران در حال استفاده از بازی، انسان را از روند طراحی این سیستم‌ها حذف کرد و دید که آیا مشکلات گفته شده حل می‌شوند یا خیر. از آنجا که رفتار کاربران در هر بازی با توجه به نوع مخاطب هدف بازی و بسیاری موارد دیگر می‌تواند تا حد زیادی متفاوت باشد، می‌توان از داده هر بازی برای یادگیری سیستم هوشمند تنظیم سطح سختی استفاده کرد و در نهایت سیستمی شخصی‌سازی شده برای همان بازی را به عنوان تجربه‌ای میزان شده از نظر سختی هر مرحله را به هر کاربر ارائه داد. این روش ممکن است شاخص‌های کلیدی ارائه شده را بهبود ببخشد. یکی دیگر از روش‌هایی که می‌توان با استفاده از آن سیستمی هوشمند برای تنظیم سطح سختی در بازی‌های جورکردنی توسعه داد، کنار گذاشتن سیستم‌های کنترلی فازی و

- pp. 373-379.
11. Lin, H. & Sun, C.-T. 2007. "Cash trade within the magic circle: Free-to-play game challenges and massively multiplayer online game player responses", O'Reilly Media, Inc. 2007.
 12. Hamari, J. & Lehdonvirta, V. 2010. "Game design as marketing: How game mechanics create demand for virtual goods," *International Journal of Business Science & Applied Management*, pp. 14-29.
 13. Alha, K., Koskinen, E., Paavilainen J. & Hamari, J. 2019. "Why do people play location-based augmented reality games: a study on Pokemon GO," *Computers in Human Behavior*, pp. 114-122.
 14. Lehdonvirta, V. 2009. "Virtual item sales as a revenue model: Identifying attributes that drive purchase decisions," *Electronic Commerce Research*, pp. 97-113.
 15. Drachen, A., Canossa, A. & Yannakakis, G. N. 2009. "Player modeling using self-organization in Tomb Raider: Underworld," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Games*, Milano, Italy, pp. 1-8.
 16. Nicholas, F. & Kulshreshth, Arun K. 2024. "Exploring Dynamic Difficulty Adjustment Methods for Video Games." In *Virtual Worlds*, vol. 3, no. 2, pp. 230-255. MDPI.
 17. Koster, R. 2013. "Theory of fun for game design", O'Reilly Media, Inc.
 18. Kose, U. & Deperlioglu, O. 2015. "Design and development of an easy-to-use, interactive fuzzy logic control software system," *Applied Mathematics & Information Sciences*, vol. 9, no. 12, pp. 883-884.
 19. Christian, E. & McCoy Jr, M. 2021. "Designing with the Player Experience in Mind: Dynamic Difficulty Adjustment." Final Thesis, University of Edinburg.
 20. Chineng, V. 2022. "The impact of dynamic difficulty adjustment on player experience in video games." *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal* 9, no. 1.
 21. Yannakakis, G. & Hallam, J. 2007. "Towards optimizing entertainment in computer games," *Applied Artificial Intelligence*, pp. 933-971.
 22. Seufert, E. B. 2013. "Freemium economics: Leveraging analytics and user segmentation to drive revenue", Elsevier Publishing.
 23. Hunicke, R. 2005. "The case for dynamic difficulty adjustment in games", in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, Valencia, Spain, pp. 429-433.
 24. Kose, U. & Deperlioglu, O. 2015. "Design and development of an easy-to-use, interactive fuzzy logic control software system," *Applied Mathematics & Information Sciences*, vol. 9, no. 12, pp. 883-884.

استفاده کامل از یک روش هوش مصنوعی مانند یادگیری عمیق است. به این صورت که خروجی شبکه عصبی دیگر قوانین فازی نبوده بلکه خروجی نهایی سیستم یعنی تعداد حرکات مجاز اضافه‌شونده یا کم‌شونده از حرکات مجاز مراحل بازی هستند. در صورت استفاده از این روش، سیستم به صورت کاملاً خودکار عمل می‌کند و دیگر نیاز به تنظیم توسط طراحان بازی نیست. همچنین می‌توان از یادگیری تقویتی^{۱۳} یا حتی ترکیب آن با یادگیری عمیق که در واقع از آن با عنوان یادگیری عمیق تقویتی^{۱۴} یاد می‌شود نیز استفاده کرد. استفاده از این روش‌ها ممکن است باعث بهبود شاخص‌های کلیدی عملکرد شوند. امید است این روش‌ها در پژوهش‌هایی دیگر مورد آزمایش قرار گرفته تا کارایی عملکرد آنها مورد بررسی قرار گیرد.

۷- مراجع

1. Andrade, G. 2005. "Challenge-sensitive action selection: an application to game balancing," in *Proceeding of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Compiegne, France, pp. 194-200.
2. Su, X. et al., 2017. "Dynamic difficulty adjustment for maximized engagement in digital games." In *Proceedings of the 26th International Conference on World Wide Web Companion*, Perth, Australia, pp. 465-471.
3. Zadeh, L. A. 2001. "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.
4. J. M. Mendel, "Uncertain rule-based fuzzy logic systems: Introduction and new directions", Prentice Hall PTR.
5. Wang, L. X. 1996. "A course in fuzzy systems and control", Prentice-Hall, Inc.
6. Ross, T. J. 2005. "Fuzzy logic with engineering applications", John Wiley & Sons.
7. Mamdani, E. H. & Assilian, S. 1975. "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13.
8. Jang, J. S. 1993. "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685.
9. Juul, J. 2012. "A casual revolution: Reinventing video games and their players", MIT Press.
10. Flunger, R., Mladenow, A. & Strauss, C. 2017. "The free-to-play business model," in *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services*, Salzburg, Austria,

13-Reinforcement learning

14-Deep reinforcement learning