



پرديس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

بررسی الگوریتم‌های هوش مصنوعی در بازی‌های رایانه‌ای

نگارنده

محمد رضا حسنی

استاد راهنما: دکتر هدیه ساجدی

پایان‌نامه برای دریافت درجه کارشناسی در رشته علوم کامپیوتر

بهمن ۱۴۰۱

چکیده

بازی‌های رایانه‌ای به طور فزاینده‌ای یک برنامه محبوب برای تحقیقات هوش مصنوعی (AI) هستند و هوش مصنوعی نیز باعث فروش روزافزون برای بازی‌های تجاری است. اگرچه بازی‌ها معمولاً برای سرگرمی هستند، اما کاربردهایی بسیار جدی از این بازی‌ها، از جمله کاربردهای نظامی، شرکتی و تبلیغاتی وجود دارد. همچنین بازی‌هایی به اصطلاح "انسانی" که برای آموزش پزشکی، بازی‌های آموزشی، و بازی‌هایی که بازتاب آگاهی اجتماعی یا حمایت از یک ایده می‌باشند، نیز وجود دارد. استفاده از هوش مصنوعی در بازی‌های کامپیوتری تلاشی برای فراتر رفتن از تعاملات برنامه‌نویسی شده پیچیده‌ای است که منجر به ورود به عرصه سیستم‌های تعاملی که دارای ویژگی‌های پاسخگویی، سازگاری و هوشمندی هستند، می‌شود. چنین سیستم‌هایی در حین بازی، درباره بازیکن (بازیکنان) یاد می‌گیرند. رفتارهای خود را فراتر از مجموعه‌ای از پیش برنامه‌ریزی شده که توسط نویسنده بازی ارائه شده است، تطبیق و به صورت تعاملی توسعه می‌دهند و تجربه غنی‌تری را برای بازیکن (بازیکنان) فراهم می‌کنند.

هدف بلندمدت تحقیق ما توسعه تکنیک‌های هوش مصنوعی است که می‌تواند تأثیر قابل توجهی در صنعت بازی داشته باشد. در این گزارش، فهرستی از چالش‌ها و فرصت‌های تحقیقاتی در توسعه تکنیک‌هایی که می‌توانند توسط توسعه‌دهندگان بازی‌های رایانه‌ای مورد استفاده قرار گیرند، ارائه می‌کنیم. ما سه رویکرد استدلالی مبتنی بر نمونه‌های آماری (CBR) را برای دستیابی به سازگاری در بازی‌ها مورد بحث قرار می‌دهیم.

سپاسگزاری

با تشکر از اساتید دانشگاه تهران و خانم دکتر هدیه ساجدی

پیشگفتار

بازی‌های رایانه‌ای پیشرفته، محیط‌های واقعی را با سطحی شگفت‌انگیز از جزئیات بازسازی می‌کنند. این محیط‌ها معمولاً پر از شخصیت‌های زیادی (متحدان یا دشمنان) هستند که نیاز به هوش در سطح انسانی دارند و رفتارهای قابل باوری از خود نشان می‌دهند. با این حال، حتی با وجود پیشرفت‌های عظیمی در گرافیک کامپیوتری، انیمیشن و صدا برای بازی‌ها، اکثر بازی‌ها در صورت وجود، دارای تکنیک‌های بسیار ابتدایی هوش مصنوعی (AI) هستند. تحقیقی که در این گزارش توضیح داده شد، گامی به سوی این هدف است. این تحقیق بر توسعه تکنیک‌های استدلال و یادگیری در زمینه بازی‌های رایانه‌ای مدرن متمرکز است. این تکنیک‌ها می‌توانند به متخصصان غیر AI اجازه دهند تا رفتارهایی را برای شخصیت‌ها تعریف کنند که می‌توان آن‌ها را با موقعیت‌های مختلف و بازیکنان منفرد تطبیق داد و در نتیجه تلاش برای توسعه مورد نیاز برای رسیدگی به همه موارد احتمالی در یک بازی پیچیده را کاهش داد.

فهرست مطالب

۱	مفاهیم مقدماتی	۱
۱	۱.۱ معرفی مساله	۱
۳	۲.۱ واژه‌نامه	۳
۵	۳.۱ کارهای انجام شده	۵
۱۱	۴.۱ پیشنهادها و مشکلات در بازی‌های کامپیوتری AI	۱۱
۱۶	۲ مفهوم‌سازی و تسخیر	۱۶
۱۹	۳ طراحی هوش مصنوعی مفهومی	۱۹
۲۲	۴ شناسایی مشکلات مفهومی	۲۲
۲۵	۵ ادغام هوش مصنوعی مفهومی در بازی‌های ویدیویی	۲۵
۲۹	۶ نتیجه‌گیری	۲۹

فصل ۱

مفاهیم مقدماتی

۱.۱ معرفی مساله

از آنجایی که هوش مصنوعی (AI) یک مفهوم گسترده در بازی‌های ویدیویی است، مهم است که با تعریف دامنه این کار شروع کنیم. یک بازی ویدیویی را می‌توان دارای دو جنبه اصلی در نظر گرفت، زمینه و بازی. این بازی شامل عناصری است که چالش‌های واقعی بازیکنان و مشکلاتی را که باید حل کنند، مانند قوانین و اهداف، تعریف می‌کنند. از سوی دیگر، زمینه شامل تمام عناصر تشکیل دهنده محیطی است که این مشکلات در آن ظاهر می‌شوند، مانند شخصیت‌ها و طرح. این گزارش بر روی هوش مصنوعی بازی تمرکز دارد، یعنی هوش مصنوعی که به حل مشکلات موجود در بازی مانند شکست دادن حریف در مبارزه یا حرکت در پیچ و خم می‌پردازد. برعکس، هوش مصنوعی زمینه، با وظایف زمینه‌ای، مانند وادار کردن یک شخصیت به انجام مجموعه‌ای از اقدامات برای پیشبرد طرح یا واکنش به انتخاب‌های بازیکن سر و کار دارد. بنابراین دامنه بحث در این گزارش به جنبه بازی محدود می‌شود.

از آنجایی که بازی‌های ویدیویی برای انسان‌ها طراحی شده‌اند، طبیعی است که آنها بر مهارت‌های شناختی و توانایی‌های فیزیکی خود تمرکز کنند. هر چه یک بازی غنی‌تر و پیچیده‌تر باشد،

به مهارت‌ها و توانایی‌های بیشتری نیاز دارد. بنابراین، ایجاد یک عامل واقعاً هوشمند و کاملاً مستقل برای یک بازی ویدیویی پیچیده می‌تواند به اندازه تکرار بخش بزرگی از هوش کامل انسان چالش‌برانگیز باشد. از سوی دیگر، هوش مصنوعی معمولاً برای هر بازی به طور مستقل طراحی می‌شود. لذا این امر ایجاد هوش مصنوعی کاملاً قوی را دشوار می‌کند. زیرا توسعه آن به محدوده‌ی، یک پروژه بازی فردی محدود می‌شود. اگرچه هر بازی ویدیویی منحصر به فرد است، اما بسته به ژانرشان می‌توانند چندین مفهوم را به اشتراک بگذارند. ژانرها برای دسته بندی بازی‌های ویدیویی بر اساس نحوه تعامل بازیکنان با آنها و همچنین قوانین آنها مورد استفاده قرار می‌گیرند. در سطح مفهومی، بازی‌های ویدیویی هم‌ژانر معمولاً چالش‌های مشابهی بر اساس مفاهیم یکسان دارند. سپس این چالش‌های مشابه شامل مشکلات مشترکی می‌شود که می‌توان رفتار اساسی را بدون توجه به نمونه مشکل تعریف و اعمال کرد. به عنوان مثال، در یک مسابقه تک به تک تیراندازی اول شخص، بازیکنان با مشکلاتی مانند انتخاب سلاح، پیش بینی موقعیت حریف و حرکت به سمت حریف مواجه می‌شوند. هر لحظه، یک بازیکن باید موقعیت را ارزیابی و روی مناسب ترین سلاح سوئیچ کند، پیش بینی کند که حریف احتمالاً به کجا می‌رود یا در حال حرکت است و بهترین مسیر را برای رسیدن به آنجا پیدا کند. همه این مشکلات را می‌توان در سطح مفهومی با استفاده از داده‌هایی مانند سرعت شلیک یک سلاح، وضعیت سلامت فعلی حریف و مکان بسته‌های کمک‌آی اولیه استدلال کرد. این مفاهیم در بسیاری از بازی‌های تیراندازی اول شخص مشترک است و برای تعریف رفتار موثر بدون توجه به جزئیات تفسیر آنها کافی است. چنین راه حل‌هایی، برای مثال برای برخی مشکلات حرکتی (حرکت یا navigation فرآیند یا فعالیت تعیین دقیق موقعیت و برنامه ریزی و دنبال کردن یک مسیر) وجود دارد و در بسیاری از بازی‌های ویدیویی استفاده می‌شود. علاوه بر این، بازیکنان انسانی اغلب می‌توانند بدون هیچ زحمتی از تجربه‌ای که از یک بازی ویدیویی به دست می‌آیند، در ژانر مشابه دیگری استفاده کنند. یک بازیکن با تجربه در بازی‌های تیراندازی اول شخص در اکثر موارد در یک بازی تیراندازی اول شخص جدید بهتر از بازی بدون تجربه عمل می‌کند و حتی می‌تواند عملکرد بهتری نسبت به یک بازیکن با تجربه در بازی جدید داشته باشد که نشان می‌دهد این امکان وجود دارد که برای به کار بردن رفتار آموخته شده برای یک بازی در بازی

دیگری که دارای مفاهیم مشابه است، بدون دانستن جزئیات در مورد دوم، عملکرد خوبی داشته باشد. بدیهی است که وقتی جزئیات کشف می‌شوند، می‌توان از آنها برای بهبود بیشتر رفتار مفهومی اساسی یا حتی نادیده گرفتن آن استفاده کرد. بنابراین ممکن است بتوان با شناسایی و هدف قرار دادن مشکلات مفهومی به جای نمونه‌های خاص بازی، هوش مصنوعی متقابل بازی ایجاد کرد. جدا کردن هوش مصنوعی یا بخشی از آن از توسعه بازی‌های ویدیویی، محدودیت‌های پروژه را که توسعه‌دهندگان را مجبور به محدود کردن آن می‌کند، از بین می‌برد و به آن اجازه می‌دهد فرآیند طراحی مستمر و دقیق‌تری داشته باشد.

۲.۱ واژه‌نامه

استدلال مبتنی بر نمونه‌های آماری

به‌طور کلی در هوش مصنوعی و فلسفه، استدلال مبتنی بر نمونه‌های آماری، فرآیند حل مسائل جدید بر اساس راه‌حل‌های مسائل مشابه گذشته است. در زندگی روزمره، یک مکانیک خودرو، موتور یک خودرو را با یادآوری خودروی دیگری که علائم مشابهی را نشان داده بود، تعمیر می‌کند. یعنی از استدلال مبتنی بر نمونه‌های آماری استفاده می‌کند.

استراتژی زمان واقعی

استراتژی زمان واقعی زیرشاخه‌ای از بازی‌های ویدئویی استراتژی است که به صورت نوبتی حرکت نمی‌کند. در واقع به همه بازیکنان اجازه می‌دهد تا به‌طور همزمان و در "زمان واقعی" بازی کنند. در مقابل، در بازی‌های استراتژی نوبتی، بازیکنان به نوبت بازی می‌کنند.

شخصیت غیربازیکن

شخصیت غیربازیکن یا شخصیتی که غیرقابل بازی است، به هر شخصیتی در بازی گفته می‌شود که توسط یک بازیکن کنترل نمی‌شود. این اصطلاح در بازی‌های نقش‌آفرینی رومیزی سنتی وجود دارد، که در آن به شخصیت‌هایی که توسط بازی‌مستر یا داور، به‌جای یک بازیکن دیگر کنترل می‌شود، اطلاق می‌گردد.

بازی نقش آفرینی

بازی نقش آفرینی، بازی‌ای است که در آن بازیکنان نقش شخصیت‌ها را در یک محیط تخیلی به عهده می‌گیرند. بازیکنان مسئولیت ایفای این نقش‌ها را، چه آن روایت از طریق بازی تحت اللفظی و چه از طریق فرآیند تصمیم‌گیری ساختارمند در مورد رشد شخصیت باشد، بر عهده می‌گیرند.

الگوریتم دایکسترا

الگوریتم دایکسترا (Dijkstra) الگوریتمی برای یافتن کوتاه‌ترین مسیرها، بین گره‌ها در یک نمودار وزنی است که ممکن است به عنوان مثال، شبکه‌های جاده‌ای را نشان دهد. این کار توسط دانشمند کامپیوتر Dijkstra W. Edsger در سال ۱۹۵۶ تهیه و سه سال بعد منتشر شد.

الگوریتم در انواع مختلفی وجود دارد. الگوریتم اصلی Dijkstra کوتاه‌ترین مسیر را بین دو گره داده شده پیدا می‌کند. اما یک نوع رایج‌تر وجود دارد که در آن یک گره را به عنوان گره "منبع" و ثابت می‌کنیم و کوتاه‌ترین مسیرها را از مبدا به تمام گره‌های دیگر در نمودار پیدا می‌کنیم و درخت کوتاه‌ترین مسیر را تولید می‌کنیم.

چارچوب مفهومی

چارچوب مفهومی یک ابزار تحلیلی با تنوع و زمینه‌های متفاوت است. می‌توان آن را در دسته‌های مختلف کار که در آن به یک تصویر کلی نیاز است، اعمال کرد. اساساً برای ایجاد تمایزات مفهومی و سازماندهی ایده‌ها استفاده می‌شود.

الگوریتم جستجو A^*

الگوریتم A^* (با تلفظ A استار) یک الگوریتم پیمایش گراف و جستجوی مسیر است که به دلیل کامل بودن، بهینه بودن و کارایی بهینه، در بسیاری از رشته‌های علوم کامپیوتر مورد استفاده قرار می‌گیرد. یکی از اشکالات عملی پیچیدگی فضای آن است، زیرا تمام گره‌های تولید شده را در حافظه ذخیره می‌کند. بنابراین، در سیستم‌های مسیریابی سفر عملی، به طور کلی الگوریتم‌هایی که می‌توانند نمودار را برای دستیابی به عملکرد بهتر و همچنین رویکردهای محدود به حافظه از قبل پردازش کنند، بهتر عمل می‌کند. با این حال، A^* هنوز در بسیاری از موارد بهترین راه حل است.

الگوریتم کوتاه ترین مسیر تک منبعی (گره مرکز)

مسئله کوتاهترین مسیر تک منبعی، (SSSP) شامل یافتن کوتاهترین مسیرها بین یک راس معین v و همه رئوس دیگر در نمودار است. الگوریتم هایی مانند Breadth-First-Search (BFS) برای نمودارهای بدون وزن یا Dijkstra این مشکل را حل می کنند.

مدل سازی همگانی

مدل سازی شناختی حوزه ای از علوم کامپیوتر است که به شبیه سازی حل مسئله و پردازش ذهنی انسان در یک مدل کامپیوتری می پردازد. چنین مدلی می تواند برای شبیه سازی یا پیش بینی رفتار یا عملکرد انسان در وظایف مشابه مدل سازی شده و بهبود تعامل انسان و رایانه استفاده شود.

۳.۱ کارهای انجام شده

مفهوم سازی بازی های ویدیویی فرآیندی است که شامل انتزاع است و شبیه به بسیاری از رویکردهای دیگر است که در همین هدف مشترک هستند، یعنی استفاده از هوش مصنوعی در بازی های ویدیویی. به طور کلی تر، انتزاع امکان ایجاد راه حل هایی را برای کل خانواده مشکلات فراهم می کند که در صورت حذف سطح معینی از جزئیات، اساساً یکسان هستند. به عنوان مثال، مشکل مرتب سازی یک آرایه بسته به نوع عناصر موجود در آرایه می تواند اشکال مختلفی داشته باشد، اما در نظر گرفتن نوع داده انتزاعی و تابع مقایسه به برنامه نویس اجازه می دهد تا راه حلی بنویسد که بتواند هر نوع آرایه ای را مرتب کند. این از کدنویسی غیر ضروری جلوگیری می کند و به برنامه نویسان کمک می کند تا حد امکان از راه حل های موجود استفاده کنند تا تلاش های توسعه را به حداقل برسانند. نمونه دیگری از کاربردهای انتزاعی پرکاربرد، انتزاع سخت افزاری است. اجزای فیزیکی در یک کامپیوتر را می توان به عنوان دستگاه های انتزاعی به منظور ساده سازی توسعه نرم افزار مشاهده کرد. اجزای فیزیکی مختلف که هدف یکسانی دارند، برای مثال ذخیره سازی، می تواند در یک نوع دستگاه ذخیره سازی انتزاعی خلاصه شود و به توسعه دهندگان نرم افزار اجازه دهد، تا برنامه های ذخیره سازی را بنویسند که با هر نوع مؤلفه ذخیره سازی کار کند. چنین مکانیزمی در سیستم عامل

هایی مانند NetBSD و خانواده سیستم عامل ویندوز NT استفاده می‌شود.

ایده ایجاد یک میان افزار هوش مصنوعی بازی ویدیویی یکپارچه، ایده جدیدی نیست. انجمن بین‌المللی توسعه دهندگان بازی (IGDA) کمیته استانداردهای رابط هوش مصنوعی (AISC) را در سال ۲۰۰۲ راه اندازی کرد که هدف آن ایجاد یک رابط هوش مصنوعی استاندارد برای امکان بازیافت و حتی برون سپاری کدهای هوش مصنوعی بود. این کمیته از چندین گروه تشکیل شده بود که هر گروه روی موضوع خاصی تمرکز می‌کرد. گروهی روی رابط جهانی کار می‌کردند، یکی روی هدایت، یکی روی مسیریابی، دیگری روی ماشین‌های حالت محدود، یکی روی سیستم‌های مبتنی بر قانون و دیگری روی برنامه‌ریزی اقدام هدف‌گرا، اگرچه گروهی که روی سیستم‌های مبتنی بر قانون کار می‌کردند منحل شدند. بنابراین، کمیته نه تنها به ایجاد یک رابط ارتباطی استاندارد بین بازی‌های ویدئویی و هوش مصنوعی، بلکه به ایجاد هوش مصنوعی استاندارد نیز توجه داشت. لذا پیشنهاد شد که ایجاد استانداردهای هوش مصنوعی می‌تواند منجر به ایجاد سخت افزار تخصصی هوش مصنوعی شود.

ایده ایجاد یک میان افزار هوش مصنوعی برای بازی‌های ویدئویی نیز در کارلسون مورد بحث قرار گرفته است، جایی که مسائل فنی و رویکردهای ایجاد چنین میان افزاری مورد بررسی قرار گرفته است. در میان چیزهای دیگر، استدلال می‌شود که وقتی سیستم‌های حالت در نظر گرفته می‌شوند، توسعه‌دهندگان بازی‌های ویدیویی به راه‌حلی بین ماشین‌های حالت محدود ساده و مدل‌های شناختی پیچیده نیاز دارند. بحث جالب دیگر این است که کتابخانه‌های عملکردی مناسب‌تر از راه‌حل‌های عامل جامع هستند، زیرا آنها انعطاف‌پذیری بیشتری را ارائه می‌کنند و در عین حال اجازه می‌دهند راه‌حل‌های مبتنی بر عامل ایجاد شوند. در اینجا نیز به امکان ایجاد سخت‌افزار تخصصی هوش مصنوعی اشاره شد و موازی با تأثیری که کارت‌های شتاب‌دهنده گرافیکی جریان اصلی بر تکامل گرافیک رایانه‌ای داشتند، نیز ترسیم شد.

یکی از مشخصات رابط استاندارد هوش مصنوعی باز (OASIS) در برنندت و همکاران پیشنهاد شده است که هدف آن آسان‌تر کردن ادغام هوش مصنوعی در بازی‌های ویدیویی است. چارچوب OASIS برای پشتیبانی از بازنمایی دانش و همچنین استدلال و یادگیری طراحی شده است و شامل

پنج لایه است که هر کدام با سطوح مختلف انتزاع، مانند سطح شی یا سطح دامنه، یا ارائه خدمات مختلف مانند خدمات دسترسی، ترجمه یا داوری هدف، سروکار دارند. لایه‌های پایین‌تر به تعامل با بازی می‌پردازند در حالی که لایه‌های بالایی با بازنمایی دانش و استدلال سروکار دارند.

بدیهی است که میان‌افزار هوش مصنوعی بازی‌های ویدیویی در موتورهای بازی‌های ویدیویی نیز یافت می‌شود. موتورهای بازی ویدیویی مانند Unreal Unity، CryEngine Engine و Havok، اگرچه ممکن است تمرکز اصلی نباشد، اما به طور فزاینده‌ای نه تنها به ایجاد بلوک‌های ساختمانی برای ایجاد محیط‌های مجازی واقعی، بلکه عوامل واقعی نیز می‌پردازند.

رویکرد دیگری که گرچه به طور خاص مربوط به هوش مصنوعی نیست، اما هدف مشابهی نیز دارد، یعنی فاکتورسازی تلاش‌های توسعه در صنعت بازی‌های ویدیویی، الگوهای بازی است. الگوهای طراحی بازی به توسعه‌دهندگان بازی اجازه می‌دهد تا مشکلات و راه‌حل‌های تکراری طراحی را به گونه‌ای مستند کنند که بتوان از آن‌ها برای بازی‌های مختلف استفاده کرد و در عین حال به آنها کمک می‌کند تا انتخاب‌های طراحی مربوط به توسعه یک بازی با سبک خاص را درک کنند. کریمر، فرمالیسم (فرمالیته اداری) یک الگو را برای کمک به گسترش دانش در مورد طراحی بازی پیشنهاد می‌کند. فرمالیسم الگوهای بازی را با استفاده از چهار عنصر توصیف می‌کند که شامل، نام، مشکل، راه حل و پیامد است. مشکل، هدف و موانعی را که می‌توان با آنها روبرو شد و همچنین زمینه‌ای که در آن ظاهر می‌شود را توصیف می‌کند. راه حل، مکانیسم‌ها و موجودیت‌های انتزاعی مورد استفاده برای حل مسئله را توصیف می‌کند. در مورد پیامد، تأثیر انتخاب طراحی بر سایر بخش‌های توسعه و هزینه‌ها و مزایای آن را توصیف می‌کند.

بژورک و همکاران بین یک چارچوب ساختاری که اجزای بازی را توصیف می‌کند و الگوهای طراحی بازی که تعامل بازیکن را در حین بازی توصیف می‌کند، تفاوت قائل شدند. چارچوب ساختاری شامل سه دسته مولفه است. اینها دسته بندی مرزی هستند که شامل اجزایی می‌شوند که برای توصیف فعالیت‌هایی که در بازی مجاز هستند یا نه مانند قوانین و حالت‌های بازی استفاده می‌شوند. دسته زمانی که شامل اجزایی است که در اجرای زمانی بازی نقش دارند مانند اعمال و رویدادها و دسته هدف که شامل عناصر مشخص بازی مانند بازیکنان یا شخصیت‌ها می‌شود.

جزئیات بیشتر در مورد این چارچوب را می‌توان در بژورک و هولوپینن یافت. در مورد الگوهای طراحی بازی، آنها عناصر مسئله و راه حل را مانند کریمر شامل نمی‌شوند. آنها با استفاده از پنج عنصر که عبارتند از نام، توصیف، پیامدها، استفاده از الگو و روابط توصیف می‌شوند. عنصر پیامدها در اینجا بیشتر بر ویژگی‌های الگو تمرکز می‌کند تا تأثیر آن بر توسعه و سایر انتخاب‌های طراحی که باید در نظر گرفته شود، که این همان نقش استفاده از عنصر الگو است. عنصر روابط برای توصیف روابط بین الگوها، مانند الگوهای فرعی در الگوها و الگوهای متضاد استفاده می‌شود.

در اولسون و همکاران، الگوهای طراحی در یک مدل رابطه مفهومی ادغام می‌شوند که برای روشن کردن تفکیک نگرانی‌ها بین الگوهای بازی و مکانیک بازی استفاده می‌شود. در آن مدل، مکانیک بازی از طریق یک لایه زمینه‌سازی که نقش آن مشخص کردن آن الگوها است، از الگوهای بازی مشتق می‌شود. برعکس، الگوهای جدیدی را می‌توان از پیاده سازی خاص این مکانیک‌های بازی استخراج کرد که در مدل به صورت کد نشان داده شده است.

همچنین رویکردهایی قابل مقایسه هستند که بر حل مسائل خاص هوش مصنوعی تمرکز دارند. به راحتی می‌توان فهمید که چرا، زیرا هدف این رویکردها معمولاً ارائه راه حل‌های استاندارد برای مشکلات رایج هوش مصنوعی در بازی‌های ویدیویی است و در نتیجه شامل توسعه هوش مصنوعی می‌شود. به عنوان مثال، ایجاد مدل‌هایی برای شخصیت‌های بازی‌های ویدیویی هوشمند یک مشکل تحقیقاتی گسترده است که رویکردهای بسیاری برای آن پیشنهاد شده است. هدف زبان‌های رفتار، ارائه یک مدل طراحی عامل است که امکان تعریف رفتار را به طور شهودی و عامل فرآیندهای رایج را فراهم می‌کند. لویال و بتس یک معماری عامل واکنشی مبتنی بر هدف را ارائه می‌دهند که اجازه می‌دهد رویدادهایی که مناسب بودن رفتار فعلی را تغییر می‌دهند شناسایی و به آن واکنش نشان دهند. ABL، یک زبان برنامه ریزی واکنشی طراحی شده برای ایجاد عوامل قابل باور است که از هماهنگی چند کاراکنتری پشتیبانی می‌کند. این مطلب در متس و استرن و شرح داده شده است.

حساب موقعیت به عنوان ابزاری برای توانمندسازی استدلال و کنترل سطح بالا در فانگ پیشنهاد شد. این به شخصیت اجازه می‌دهد تا جهان را به عنوان دنباله‌ای از موقعیت‌ها ببیند و بفهمد که چگونه می‌تواند تحت تأثیر اعمال مختلف از یک موقعیت به موقعیت دیگر تغییر کند تا

بتواند تصمیم بگیرد و به اهداف برسد. یک زبان مدل‌سازی شناختی (CML) که برای مشخص کردن خطوط کلی رفتار برای شخصیت‌های مستقل استفاده می‌شود و از حساب موقعیت استفاده می‌کند و از روش‌های بازه‌ای برای فعال کردن شخصیت‌ها برای تولید برنامه‌های عمل در جهان‌های بسیار پیچیده استفاده می‌کند، نیز در فانگ و همکاران، پیشنهاد شده است.

در اورکین بحث شد که برنامه‌ریزی بلادرنگ رویکرد مناسب‌تری نسبت به اسکریپت‌نویسی یا ماشین‌های حالت محدود برای تعریف رفتار عامل است، زیرا اجازه می‌دهد موقعیت‌های غیرمنتظره به‌طور طبیعی‌تر مدیریت شوند. یک معماری برنامه‌ریزی اکشن هدف گرا مدولار، برای عوامل بازی، مشابه مورد استفاده شده در متس و استرن ارائه شده است. تفاوت اصلی آن با زبان ABL این است که بین پیاده‌سازی و داده‌ها جدایی ایجاد می‌شود. با ABL طراحان رفتار را به‌طور مستقیم پیاده‌سازی می‌کنند. اما در اینجا پیاده‌سازی توسط برنامه‌نویسان انجام می‌شود و طراحان با استفاده از داده‌ها رفتار را تعریف می‌کنند.

اندرسون زبان دیگری را برای طراحی شخصیت‌های هوشمند پیشنهاد می‌کند. زبان تعریف آواتار (AvDL) امکان تعریف رفتار قطعی و هدفمند را برای موجودیت‌های مجازی به‌طور کلی فراهم می‌کند. این زبان توسط زبان حاشیه‌نویسی موجودیت ساده (SEAL) گسترش یافته است که به تعاریف رفتار اجازه می‌دهد تا با حاشیه‌نویسی و فعال کردن کاراکترها به‌طور مستقیم در اشیا در دنیای مجازی برای تبادل اطلاعات با آنها جاسازی شوند.

در نهایت، یادگیری رویکرد متفاوتی را تشکیل می‌دهد که باز هم به همان هدف منجر می‌شود. با ایجاد عواملی که قادر به یادگیری و تطبیق با محیط خود باشند، مسئله طراحی شخصیت‌های بازی‌های ویدیویی هوشمند به روشی کلی‌تر و قابل استفاده مجدد حل می‌شود. بازی‌های ویدیویی در دهه گذشته مورد توجه گسترده جامعه یادگیری ماشین قرار گرفته‌اند و تلاش‌های متعددی برای ادغام یادگیری در بازی‌های ویدیویی با درجات مختلف موفقیت انجام شده است. برخی از روش‌های مورد استفاده مشابه رویکردهای ذکر شده قبلی هستند، زیرا از انتزاع یا مفاهیم برای مقابله با تنوع زیادی که در بازی‌های ویدیویی وجود دارد، استفاده می‌کنند. تکنیک‌های استدلال مبتنی بر مورد، اطلاعات وضعیت بازی را تعمیم می‌دهند تا هوش مصنوعی در پیکربندی‌های متمایز و در عین حال

مشابه، به‌طور ثابت‌تری رفتار کند. امکان استفاده از شناسایی طرح مبتنی بر مورد، برای کاهش قابلیت پیش‌بینی بازیکنان رایانه استراتژی زمان واقعی، در چنگ و تاونماس بحث شده است. آها و همکاران یک رویکرد یادگیری موردی و انتخاب طرح را ارائه می‌دهند، که در عاملی که یاد می‌گیرد در برابر تعدادی از مخالفان هوش مصنوعی مختلف در Wargus پیروز شود، استفاده می‌شود. در اونتان و همکاران، یک چارچوب برنامه‌ریزی مبتنی بر مورد، برای بازی‌های استراتژی زمان واقعی، که به عوامل اجازه می‌دهد به‌طور خودکار دانش رفتاری را از بازپخش‌های متخصص مشروح استخراج کنند، در Wargus نیز توسعه یافته و با موفقیت آزمایش شده است. استفاده بیشتر از Wargus به عنوان یک پلتفرم آزمایشی شامل در وبر و ماتاس آمده است که نشان می‌دهد چگونه همسایه‌های مفهومی را می‌توان برای بازیابی در رویکردهای استدلال مبتنی بر مورد استفاده کرد. رویکردهای یادگیری انتقالی تلاش می‌کنند از تجربیات آموخته‌شده از برخی کارها برای بهبود رفتار در سایر وظایف استفاده کنند. در شارما و همکاران، یادگیری انتقال، با ترکیب استدلال مبتنی بر مورد و یادگیری تقویتی به دست می‌آید و برای بهبود عملکرد نسبت به بازی‌های متوالی در برابر هوش مصنوعی در MadRTS استفاده می‌شود. لی‌اوربان و همکاران، از MadRTS برای اعمال یادگیری انتقال با استفاده از معماری مدولار استفاده می‌کنند که برنامه ریزی شبکه وظایف سلسله مراتبی (HTN) و یادگیری مفهومی را ادغام می‌کند. انتقال مهارت‌ها و مفاهیم ساختار بین وظایف متفاوت با استفاده از معماری شناختی در شاپیرو و همکاران به دست آمده است.

اگرچه فناوری یادگیری ماشین ممکن است منجر به ایجاد یک هوش مصنوعی یکپارچه شود که می‌تواند در چندین بازی استفاده شود، اما در حال حاضر از عدم رشد کافی رنج می‌برد. حتی اگر برخی از تکنیک‌ها با موفقیت در چند بازی تجاری به کار گرفته شده باشند، ممکن است زمان زیادی طول بکشد تا آنها به اندازه کافی قابل اعتماد باشند تا به جریان اصلی تبدیل شوند. از سوی دیگر، موتورهای بازی‌های ویدیویی معمولاً مورد استفاده قرار می‌گیرند و رویکرد عملی تری را در فاکتورگیری فرآیندهای توسعه بازی برای بهبود کیفیت بازی‌های ویدیویی تشکیل می‌دهند. با این حال، آنها ابزارهای جامعی هستند که توسعه دهندگان باید برای طراحی کل بازی به جای صرفاً هوش مصنوعی خود استفاده کنند. علاوه بر این، آنها اجازه هیچ آزادی را در معماری بنیادی عواملی که

هدایت می‌کنند، نمی‌دهند.

رویکرد ارائه شده در این گزارش بیشترین شباهت را با ایجاد یک میان افزار یکپارچه هوش مصنوعی دارد. با این حال، به طور دقیق، یک میان افزار هوش مصنوعی نیست. در این جا از یک چارچوب مفهومی به عنوان مؤلفه اصلی استفاده می‌کند که ارتباط بین بازی‌های ویدیویی و هوش مصنوعی را امکان‌پذیر می‌کند و به توسعه‌دهندگان بازی‌های ویدیویی این امکان را می‌دهد تا از هوش مصنوعی مفهومی و مستقل از بازی در بازی‌های خود به قیمت هماهنگ‌سازی لازم بین داده‌های بازی و داده‌های مفهومی استفاده کنند. یک تفاوت کلیدی با کارهای قبلی این است که هیچ فرضی در مورد نحوه طراحی هوش مصنوعی مانند تحمیل یک مدل عامل یا ماژول‌های خاص، ندارد. راه حل‌ها را می‌توان برای هر نوع مشکل هوش مصنوعی و به هر شکلی طراحی کرد. جدایی واضحی بین توسعه چارچوب مفهومی، هوش مصنوعی و چارچوب بازی‌های ویدیویی ایجاد شده است. از آنجایی که توسعه هوش مصنوعی کاملاً از چارچوب مفهومی جدا است، پذیرش آن باید آسان‌تر باشد، زیرا آزادی کاملی را برای توسعه‌دهندگان هوش مصنوعی برای طراحی و پیاده‌سازی هوش مصنوعی به هر روشی که عادت دارند، اجازه می‌دهد. علاوه بر این، سادگی این رویکرد، ارائه یک مثال استقرار کامل را امکان‌پذیر می‌سازد که جزئیات نحوه بازنویسی کل یک بازی ویدیویی را به دنبال طرح پیشنهادی ارائه می‌دهد. به علاوه، نمونه اولیه چارچوب مفهومی محدود حاصل با موفقیت، برای استفاده مجدد از برخی از ماژول‌های هوش مصنوعی بازی در یک بازی کاملاً متفاوت مورد استفاده قرار گرفته است.

۴.۱ پیشنهادها و مشکلات در بازی‌های کامپیوتری AI

در کارهای قبلی، لارید و ونلنت (۲۰۰۰) ژانرهای مختلف بازی و چالش‌های هوش مصنوعی را که هر کدام ارائه کرده‌اند، تجزیه و تحلیل شده است. آنها در گزارش خود انواع بازی‌های اکشن، نقش آفرینی، ماجراجویی، بازی‌های استراتژیک، بازی‌های خدایی، بازی‌های ورزشی انفرادی و تیمی را در نظر گرفته‌اند. علاوه بر آن ژانرها، ما می‌خواهیم دو دسته دیگر را در نظر بگیریم، یعنی

درام تعاملی (متس و استرن ۲۰۰۳) و بازی‌های آموزشی (ریبر ۱۹۹۶). درام‌های تعاملی دارای یک طرح قوی در پشت خود هستند که نویسنده می‌خواهد با بازیکن ارتباط برقرار کند، اما این در جایی است که بازیکن ممکن است تأثیر شدیدی بر طرح داشته باشد. یک تفاوت کلیدی با ژانر ”ماجراجویی” کلاسیک این است که ماجراجویی‌ها دارای یک طرح فیلمنامه هستند، در حالی که درام‌های تعاملی پایان بازتری دارند و در طول داستان با تعامل بازیکن سازگار می‌شوند. بازی‌های آموزشی یک هدف روشن اضافی برای آموزش برخی محتوای خاص به بازیکن دارند.

با تجزیه و تحلیل دامنه کاربردهای احتمالی هوش مصنوعی در بازی‌های رایانه‌ای، در برنامه‌ها و ژانرهای مختلف بازی، دو سطح مختلف را شناسایی می‌کنیم که می‌توان هوش مصنوعی را در آن اعمال کرد: ۱) هوش مصنوعی شخصیت‌های منفرد، با هدف تولید رفتارهای هوشمندانه یا باورپذیرتر، و ۲) هوش مصنوعی جهانی که مراقب بازی یا تعامل بازی و بازیکن است و بر مسیرهایی که بازی می‌گیرد تأثیر می‌گذارد. بنابراین، می‌توانیم در مورد هوش مصنوعی در سطح شخصیت و هوش مصنوعی در سطح بازی صحبت کنیم (دومین مورد در برخی مقالات به عنوان مدیر نمایش نلسون و همکاران ۲۰۰۶) یا کارگردان (ماگرکو و همکاران ۲۰۰۴) نامیده می‌شود).

برنامه‌ها و ژانرهای مختلف بازی به ترکیب متفاوتی از این دو نوع هوش مصنوعی نیاز دارند. به عنوان مثال، بازی‌های استراتژی زمان واقعی، عمدتاً به هوش مصنوعی در سطح بازی متکی هستند که همه واحدها را کنترل می‌کند، در حالی که رفتارهای واحد فردی را می‌توان با اسکریپت نوشت. از سوی دیگر، بازی‌های نقش‌آفرینی به هوش مصنوعی در سطح شخصیت‌های باورپذیر نیاز دارند تا تجربه‌ای جالب برای بازیکن ارائه دهند. درام‌های تعاملی به ترکیبی از هر دو نوع هوش مصنوعی نیاز دارند: شخصیت‌های منفرد که باورپذیر هستند و یک مدیر درام که با هدایت شخصیت‌ها برای انجام اقداماتی که می‌تواند باعث پیشرفت درام شود، طرح را هدایت می‌کند. شبیه به مدیر نمایش، که تعامل بازی را در حین باز شدن آن نظارت می‌کند، کارها را با توجه به سطح تخصص یادگیرنده آسان یا پیچیده می‌کند، در نتیجه مطمئن می‌شود که هدف آموزشی بازی برآورده شده است.

هر ژانر بازی الزامات خاصی را برای سطح شخصیت و هوش مصنوعی سطح بازی ارائه می‌دهد. به عنوان مثال، بازی‌های خدا معمولاً به هوش مصنوعی سطح بازی برای حل مشکلات تخصیص

منابع و حل مشکلات استراتژی بلند مدت نیاز دارند. در حالی که درام تعاملی به هوش مصنوعی سطح بازی نیاز دارد تا داستان را با توجه به تعاملات بازیکن به گونه‌ای تطبیق دهد که برای بازیکن جذاب تر باشد (بنابراین، دومی نیاز به مدل سازی کاربر و برنامه ریزی داستان دارد). علاوه بر این، ماجراجویی‌ها، درام‌های تعاملی و ژانرهای دیگر با شخصیت‌های تجسم یافته معمولاً نیازمند باورپذیری و تولید زبان طبیعی هستند.

در بخش بعدی، فهرستی از چالش‌های جالبی را که بازی‌های رایانه‌ای به طور کلی برای جامعه هوش مصنوعی ایجاد می‌کنند، خلاصه می‌کنیم.

اجازه دهید برخی از مسائل اصلی را که هنگام توسعه هوش مصنوعی برای بازی‌های رایانه‌ای به وجود می‌آیند، به اختصار شرح دهیم. این فهرست جامع نیست، اما در نظر گرفته شده است تا بخشی از مشکلاتی را که بازی‌های رایانه‌ای واقعی برای جامعه هوش مصنوعی ایجاد می‌کنند، نشان دهد.

- فضاهای تصمیم‌گیری پیچیده: بیشتر بازی‌های رایانه‌ای پیشرفته شامل استراتژی‌های پیچیده (بازی‌های استراتژی زمان واقعی) یا رفتارهای باورپذیر (درام‌های تعاملی) می‌شوند. هر دو نوع رفتار دارای ویژگی، داشتن فضاهای تصمیم‌گیری بزرگ هستند و بنابراین تکنیک‌های هوش مصنوعی مبتنی بر جستجوی سنتی را نمی‌توان اعمال کرد. تکنیک‌های یادگیری یا نمایش‌های سطح بالاتر برای مقابله با چنین بازی‌های پیچیده‌ای مورد نیاز است. به طور سنتی، بازی‌های رایانه‌ای از استراتژی‌های دست‌ساز استفاده می‌کنند که توسط توسعه‌دهندگان بازی کدگذاری شده‌اند، اما این استراتژی‌ها معمولاً تکراری هستند و بازیکنان به راحتی حفره‌ها را پیدا کرده و از آنها سوءاستفاده می‌کنند.

- مهندسی دانش: حتی با فرض اینکه استراتژی‌ها یا رفتارها دست‌ساز هستند، نوشتن این مجموعه‌های رفتاری در یک بازی، مستلزم تلاش عظیم مهندسی انسانی است. توسعه دهندگان بازی باید تمام دانشی را که در مورد یک دامنه دارند (چه برای دستیابی به یک رفتار استراتژیک یا یک رفتار انسانی قابل باور) در نوعی از زبان رفتار رمزگذاری کنند.

- پشتیبانی تالیف: رفتارهای دست ساز، در نهایت کدهای نرم افزاری در یک زبان برنامه نویسی پیچیده هستند که مستعد خطاهای انسانی می‌باشند. خطاهای رفتاری می‌تواند به صورت ”اشکالات (باگ)” برنامه یا عدم دستیابی به نتیجه مطلوب باشد. ابزارهایی برای حمایت از نویسندگان داستان، که معمولاً متخصص هوش مصنوعی نیستند، برای حمایت از رفتارهای نویسنده در یک زبان برنامه‌نویسی کامپیوتری مورد نیاز است.
- موقعیت‌های پیش‌بینی نشده: پیش‌بینی همه موقعیت‌های ممکن و استراتژی‌های بازیکن که در طول بازی با آن‌ها مواجه می‌شوند، امکان‌پذیر نیست. این امر ایجاد رفتارهای قابل باوری را که به شیوه‌ای مناسب به این شرایط پیش‌بینی نشده و اقدامات بازیکن واکنش نشان می‌دهند، دشوار می‌کند.
- انطباق ویژه کاربر: بازیکنان مختلف ممکن است از استراتژی‌های مختلفی برای مبارزه با آن (در مورد بازی‌های استراتژی زمان واقعی)، یا سبک‌های مختلف داستان‌گویی (در مورد درام‌های تعاملی)، انواع مختلف توسعه داستان، انواع مختلف رفتارها و تعاملات شخصیت‌ها، یا مسائل آموزشی متفاوت، لذت ببرند. همانطور که طراحان بازی شروع به گنجاندن قابلیت‌های مدل‌سازی کاربر می‌کنند، استراتژی و رفتار هوش مصنوعی باید به نوبه خود بر اساس مدل کاربر سازگار باشد.
- قابلیت پخش مجدد و تغییرپذیری: یک بازیکن ممکن است از دیدن استراتژی‌ها و رفتارهای یکسان و مکرر خسته شود. اگرچه تنوع ساده را می‌توان از طریق انتخاب تصادفی رفتارها یا استراتژی‌ها از یک منبع بزرگ به دست آورد، اما این باعث افزایش زحمت نویسنده می‌شود. علاوه بر این، انتخاب تصادفی، مسئله جالب بودن واقعی را مطرح می‌کند.
- اهداف روشن و واضح: ممکن یا حتی محتمل است، که رفتارها یا استراتژی‌های مهندسی شده توسط انسان به اندازه کافی به اهداف بازی دست پیدا نکنند، به خصوص در حوزه‌ها یا برنامه‌های کاربردی واقع‌گرایانه و مقیاس شده. این اهداف می‌تواند از سرگرمی گرفته

تا آموزش، تعلیم و غیره باشد. بنابراین، بازی باید متوجه شود که اهداف بر اساس هر بار استفاده برآورده نمی‌شوند و مطابق با آن سازگار شوند. به عنوان مثال، یک کاربر خاص ممکن است خسته شده باشد، یا درس مورد نظر را یاد نگیرد.

فصل ۲

مفهوم‌سازی و تسخیر

از آنجایی که بازی‌های ویدیویی، علی‌رغم تنوع ظاهری‌شان، مفاهیم را به‌طور گسترده به اشتراک می‌گذارند، ایجاد هوش مصنوعی که صرفاً بر اساس مفاهیم عمل می‌کند باید به توسعه‌دهندگان اجازه دهد تا از آن برای چندین بازی استفاده کنند. با این حال، این سؤال مهم یعنی در دسترس بودن تفسیر مفهومی از بازی‌های ویدیویی مطرح می‌شود. در واقعیت، برای اینکه هوش مصنوعی بتواند اشیاء مفهومی را مدیریت کند، باید به نمای مفهومی داده‌های بازی در طول زمان اجرا دسترسی داشته باشد.

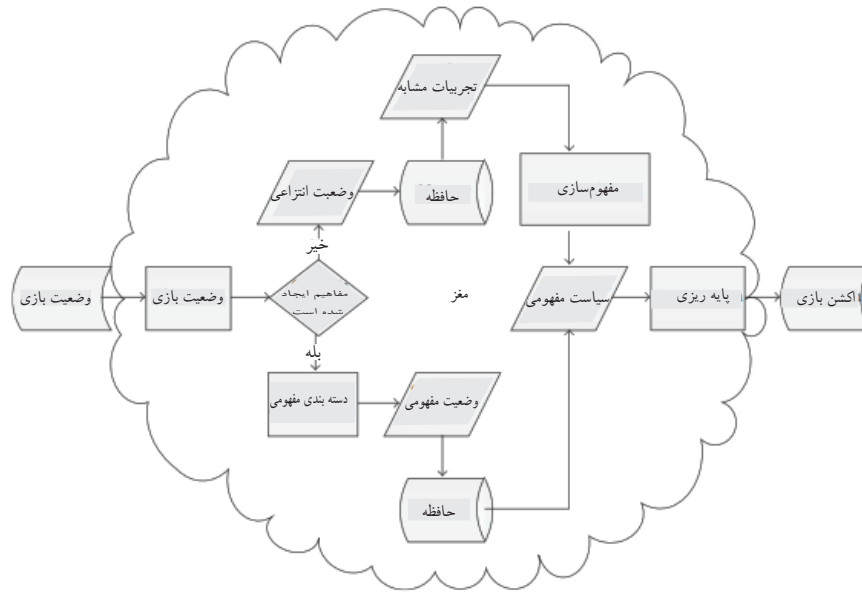
وقتی انسان‌ها یک بازی ویدیویی انجام می‌دهند، از توانایی انتزاعی خود برای تشخیص تشابهات بین بازی و بازی‌هایی که در گذشته انجام داده‌اند استفاده می‌کنند. انتزاع در این زمینه را می‌توان به عنوان فرآیندی برای کنار گذاشتن جزئیات و استخراج ویژگی‌ها از داده‌های خام در نظر گرفت. با یادآوری موارد قبلی از یک مورد مفهومی، تجربه به دست آمده از بازی‌های دیگر تعمیم می‌یابد و به یک خط مشی مفهومی (یعنی مفهوم‌سازی) تبدیل می‌شود. به عنوان مثال، یک بازیکن می‌توانست در یک بازی نقش آفرینی (RPG) یاد گرفته باشد که از حملات دامنه دار به دشمن استفاده کند و در عین حال دور از دسترس او بماند. این رفتار به نام کیتینگ شناخته می‌شود. بعداً، در یک بازی استراتژی زمان واقعی (RTS) همان بازیکن ممکن است با همان

موقعیت مفهومی، با یک واحد برد و یک دشمن مواجه شود. اگر در آن زمان، مفهوم کیتینگ به وضوح در ذهن بازیکن تثبیت نشده باشد، ممکن است تجربه کسب شده در RPG را به خاطر بیاورد و متوجه شود که با وضعیت مشابهی روبرو است، یعنی کنترل بر یک موجودیت با حمله برد و توانایی حرکت و حضور دشمن. به این ترتیب بازیکن تکنیکی را که در RPG آموخته است مفهوم سازی می‌کند و سعی می‌کند آن را در بازی RTS اعمال کند. از سوی دیگر، اگر بازیکن با مفهوم کیتینگ آشنا باشد، یک انتزاع ساده از موقعیت منجر به بازیابی خط مشی مفهومی مرتبط با آن، بدون نیاز به یادآوری موارد قبلی و تجربیات مرتبط و مفهوم سازی آنها می‌شود.

توجه داشته باشید که کیتینگ تنها با استفاده از مفاهیمی مانند فاصله، برد حمله و حرکت قابل تعریف است. فاصله می‌تواند چندین تفسیر متمایز داشته باشد، به عنوان مثال حیاط، کاشی یا قطعه‌ای کوچک باشد. برد حمله می‌تواند محدوده طلسم، برد سلاح گرم یا محدوده گرانس باشد. پیاده‌روی، رانندگی و تله‌پورت کردن، همه این‌ها اشکال مختلف حرکت هستند. کیتینگ یک مفهوم است، واضح است که می‌توان از یک مفهوم برای تعریف مفاهیم دیگر استفاده کرد. در واقع برای تعریف سیاست‌های مفهومی، انواع مختلفی از مفاهیم مانند اشیاء، روابط، شرایط و کنش‌ها ضروری است. اسلحه، دشمنی، تحرک (شرط متحرک بودن) و اختفا از مصادیق مفاهیم هستند.

با توجه به فرآیند نشان داده شده در شکل ۱، هوش مصنوعی مفهومی، یعنی هوش مصنوعی که کاملاً بر اساس مفاهیم عمل می‌کند، می‌تواند در بازی‌های ویدیویی با این فرض که سه الزام برآورده شده است، استفاده شود که به شرح زیر است:

- (۱) توانایی ترجمه حالت‌های بازی به حالت‌های مفهومی.
- (۲) توانایی ترجمه اقدامات مفهومی به اقدامات بازی.
- (۳) توانایی تعریف سیاست‌های مفهومی. (یک خط مشی مفهومی حالت‌های مفهومی را به اقدامات مفهومی ترسیم می‌کند.)



شکل ۱: فرآیند احتمالی تصمیم گیری انسانی در یک بازی ویدیویی با استفاده از سیاست های مفهومی، همانطور که در بالا توضیح داده شد.

فصل ۳

طراحی هوش مصنوعی مفهومی

از منظر فنی، نوشتن هوش مصنوعی مفهومی شبیه به نوشتن هوش مصنوعی معمولی است. یعنی توسعه‌دهندگان آزادند که هوش مصنوعی خود را به هر نحوی که مناسب می‌دانند طراحی کنند. هوش مصنوعی مفهومی به فرم خاصی نیاز ندارد. تنها تفاوت بین هوش مصنوعی مفهومی و هوش مصنوعی معمولی این است که اولی به استفاده از داده‌های مفهومی محدود می‌شود. به جای عملیات بر روی اجسام بتنی، مانند شوالیه‌ها، تفنگ‌های رعد و برق یا طلسم‌های گلوله آتشین، با مفاهیمی مانند غوغا (برعکس محدوده، فقط می‌تواند در فاصله دست و پنجه نرم حمله کند)، واحدهای تانک (یک واحد تانک یا تانک، واحدی است که می‌تواند در برابر آسیب‌های زیادی مقاومت کند و نقش اصلی آن ترسیم حملات دشمن به منظور اطمینان از بقای واحدهای متحد ضعیف‌تر است.)، سلاح‌های هیت‌اسکن دوربرد (سلاح ضربه‌ای اسلحه‌ای است که در هنگام شلیک فوراً به هدف اصابت می‌کند (یعنی بدون پرتابه مسافرتی))، و توانایی‌های پرتابه آسیب ناحیه اثر تایپ شده، سروکار دارد. (قابلیت‌های منطقه اثر، یک منطقه را به جای یک واحد مورد هدف قرار می‌دهند.) به همین ترتیب، اقدامات شامل اشیاء و ویژگی‌های مفهومی به‌جای عناصر مشخص بازی می‌شود و می‌تواند شامل تولید یک واحد ضد هوا یا تجهیز لوازم جانبی کاهش آسیب باشد. این تفاوت در الگوریتم‌های ۱ و ۲ نشان داده شده است.

الگوریتم ۱ گزیده‌ای از کد رزمی Defender، Fortress یک شخصیت غیربازیکن (NPC) در یک بازی RPG را نشان می‌دهد. یک مدافع قلعه می‌تواند دشمنان را بی حرکت کند، یک توانایی مفید در برابر حریفانی که ممکن است سعی کنند آن کیت کنند. قبل از فرمان دادن به NPC برای حمله به دشمنی که با آن مواجه شده، کد بررسی می‌کند که آیا نوع حریف یکی از آن‌هایی است که از سلاح‌های ترکیب شده استفاده می‌کنند یا خیر و با استفاده از قابلیت تثبیت‌پذیری در مورد آن شروع به کار می‌کند.

الگوریتم ۲ یک مفهوم سازی احتمالی از همان کد را نشان می‌دهد. توجه داشته باشید که چگونه طراحی یکسان باقی می‌ماند و تنها تغییر در جایگزینی عناصر بازی با مفاهیم مفهومی است. در نتیجه، کد جدید از استدلال مفهومی بیشتر تقلید می‌کند. به منظور جلوگیری از ضربه زدن به NPC غوغا، توسط دشمن، دومی بررسی می‌کند که آیا توانایی مختل‌کنندگی حرکت در دسترس است یا خیر و قبل از حرکت به سمت هدف از آن استفاده می‌کند. اینکه آیا توانایی واقعی برای کند کردن، بی حرکت کردن یا کاملاً بی حرکت کردن حریف مشخص شود، تا زمانی که هدف مفهومی جلوگیری از کیت کردن NPC محقق شود، اهمیت کمی دارد. اگرچه این امر مستلزم آن است که توسعه‌دهندگان به شیوه‌ای انتزاعی‌تر فکر کنند، اما آن‌ها آزادی طراحی هوش مصنوعی خود را هر طور که عادت دارند حفظ می‌کنند.

```
void handle_enemy(pc.t & enemy)
{
    ...
    if (enemy.type() == pc.t::cleric || enemy.type()
        == pc.t::sorcerer || enemy.type() == pc.t::ranger)
        queue_action(use_skill(Skill::root, enemy));
        queue_action(attack(enemy));
    ...
}
```

الگوریتم ۱: قطعه کد رزمی Defender، Fortress

```
void handle_enemy(pc.t & enemy)
{
    ...
    if (enemy.ranged() && can_impair_movement())
        queue_action(use_skill(get_skill(SkillType::
            disable_move), enemy));
    queue_action(attack(enemy));
    ...
}
```

الگوریتم ۲: قطعه کد رزمی مفهومی.

فصل ۴

شناسایی مشکلات مفهومی

مشکلات مفهومی قلب مدل توسعه هوش مصنوعی این بازی ویدیویی است. در واقع، اگر مفاهیم به دست آمده برای شناسایی مشکلاتی که در چند بازی مشترک هستند، به کار نرود، مفهوم سازی بازی‌های ویدیویی چندان کاربردی ندارد. تکرار مشکل در بازی‌های ویدیویی دلیل وجود چنین مدلی است و دلیل این که چرا فاکتورگیری هوش مصنوعی بازی ویدیویی ارزش پیگیری دارد، است. میزان فاکتورسازی که می‌توان به دست آورد بستگی به این دارد که چگونه مشکلات تکرار شونده در بازی‌های ویدیویی نه تنها از همان سبک، بلکه در هر سبکی جدا شده‌اند. این می‌تواند به عنوان معیاری برای کارایی مدل مورد استفاده قرار گیرد، همانطور که می‌تواند میزان افزونگی در راه حل‌های هوش مصنوعی برای مشکلات ناهمگون باشد. بنابراین شناسایی و سازماندهی واضح مشکلات مفهومی یکی از ابعاد مهم این مدل توسعه است.

مسائل و راه حل‌های آنها می‌تواند ابتدایی یا ترکیبی باشد. مسائل ابتدایی مشکلاتی هستند که تجزیه آنها به مشکلات کمتر منجر به فاکتورگیری هیچ هوش مصنوعی نمی‌شود. آنها بلوک‌های سازنده مشکلات ترکیبی هستند. اما دومی چندین مشکل ابتدایی یا مرکب را در یک بسته واحد ترکیب می‌کند تا با یک راه حل کامل هوش مصنوعی قابل رسیدگی باشد. به عنوان مثال، یک عامل برای مسابقه مرگ در عرصه FPS می‌تواند به عنوان راه حلی برای مشکل کنترل یک شخصیت در

آن تنظیمات دیده شود. این مشکل را می‌توان به مشکلات کوچک‌تری تقسیم کرد که می‌توان آن‌ها را در تنظیمات مختلف مانند جنگ و حرکت در زمان واقعی یافت.

حرکت یک مشکل محبوب و مورد مطالعه است که در بسیاری از بازی‌های ویدیویی یافت می‌شود. پیمایش در دنیای مجازی اغلب شامل مسیریابی است. تعاریف مشترک و همچنین راه حل‌های بهینه از قبل برای مسائل مسیریابی وجود دارد. به عنوان مثال می‌توان به الگوریتم جستجوی A^* اشاره کرد که مشکل کوتاه‌ترین مسیر تک-جفتی را حل می‌کند (مسیر کم هزینه را بین گره مبدا و گره مقصد در یک نمودار بیابید). و الگوریتم Dijkstra، که مشکل کوتاه‌ترین مسیر تک منبعی را حل می‌کند (مسیر کم هزینه را بین گره ریشه و سایر گره‌ها در یک نمودار پیدا کنید). اشاره کرد. اگرچه پیاده‌سازی‌های استاندارد را می‌توان در چارچوب‌های توسعه‌دهنده و جعبه‌های ابزار یافت، این غیرعادی نیست که توسعه‌دهندگان، به پیاده‌سازی خود برای سفارشی‌سازی مبتنی بر محیط متعهد باشند.

تجزیه مسئله، اغلب در طراحی هوش مصنوعی یک بازی ویدیویی منعکس می‌شود. به عنوان مثال، هوش مصنوعی در یک بازی RTS ممکن است به دو جزء اصلی تقسیم شود. یک جزء به مسئله رفتار واحد و تعریف رفتار واحدها در حالت‌های مختلف، مانند بیکار بودن یا پیروی از دستورات خاص، می‌پردازد. این مؤلفه هوش مصنوعی به نوبه خود می‌تواند شامل اجزای فرعی برای مسائل فرعی مانند مسیریابی باشد. تعریف رفتار واحد خودمختار شامل عناصری مانند طراحی پاسخ واحد به تهدید، حمله یا حضور متحد است و مشکلی است که در بازی‌های دیگر مانند RPG و FPS یافت می‌شود. جزء اصلی دیگر با مسئله بازی RTS سروکار دارد تا امکان رویارویی یک بازیکن انسانی با حریفان را بدون نیاز به سایر بازیکنان انسانی فراهم کند. این جزء می‌تواند در تعدادی مازول سازماندهی شود تا با وظایف مختلفی که یک بازیکن باید در یک بازی RTS انجام دهد، سروکار داشته باشد. یک مدیر استراتژی می‌تواند تصمیماتی مانند زمان حمله و تولید واحدها را مدیریت کند. یک مدیر تولید می‌تواند وظایف ساخت و ساز را انجام دهد و کارگران را به جوخه‌های معدن اختصاص دهد. یک مدیر رزمی می‌تواند مکان‌های حمله را تعیین کند و واحدهای نظامی را به جوخه‌های رزمی اختصاص دهد. مدیران جوخه می‌توانند وظایفی مانند تشکیل واحدها

و هماهنگی و انتخاب هدف را انجام دهند. این مؤلفه‌های هوش مصنوعی می‌توانند بینشی در مورد مشکلات مفهومی مختلفی که سعی در حل آن‌ها دارند و سازمانشان دارند، ارائه دهند. هماهنگی بین گروهی از واحدها برای انتخاب یک هدف مشترک یا توزیع اهداف بین واحدها و واحدهای مانور را می‌توان در مشکل بزرگتر نبرد بلادرنگ که مختص ژانر RTS نیست، گنجانند. از سوی دیگر، تصمیمات مربوط به تولید را می‌توان بر اساس داده‌های عمومی مانند کل قدرت آتش هوایی یا کل زره زمینی اتخاذ کرد، که استفاده از سیاست مفهومی یکسانی را برای هر بازی RTS امکان پذیر می‌کند که نمای مفهومی از طریق آن داده‌ها را ارائه دهد، را می‌توان محاسبه کرد.

فصل ۵

ادغام هوش مصنوعی مفهومی در بازی‌های ویدیویی

از آنجایی که هوش مصنوعی مفهومی مستقل از بازی‌ها، طراحی شده است، مکانیزم یکپارچه سازی برای استفاده از آن توسط توسعه دهندگان بازی ضروری است. توسعه دهندگان بازی باید بتوانند راه حل‌های هوش مصنوعی را انتخاب کرده و به یک بازی متصل کنند. این با ثبت کنترلرهای هوش مصنوعی با اشیاء مفهومی به دست می‌آید. برای تخصیص کنترل جزئی یا کامل یک موجودیت در بازی به یک هوش مصنوعی خاص، کنترل کننده مربوطه باید نمونه سازی شده و با طرح ریزی موجودیت در CDS ثبت شود. سپس هوش مصنوعی موجودیت مفهومی را کنترل می‌کند و به طور موثر موجودیت موجود در بازی را کنترل می‌کند. برای مثال، یک توسعه‌دهنده بازی می‌تواند از دو راه‌حل هوش مصنوعی برای یک بازی مسابقه‌ای استفاده کند، یکی برای کنترل حریفان رایانه‌ای در مسیر و دیگری برای تنظیم پویای دشواری مسابقه با عملکرد بازیکن. هر بار که یک حریف کامپیوتری به مسابقه اضافه می‌شود، یک نمونه جدید از هوش مصنوعی رانندگی ایجاد و با طرح مفهومی آن ثبت می‌شود. در مورد سختی هوش مصنوعی، می‌توان آن را در ابتدای مسابقه ایجاد کرد و با یک شی، ارزیابی عملکرد بازیکن در زمان واقعی را ثبت کرد.

برای هر شی مفهومی قابل کنترل تعریف شده توسط توسعه دهندگان، CF یک رابط کنترل کننده همراه با آن تعریف می‌شود. این رابط عملکردهایی را توصیف می‌کند که هوش مصنوعی باید اجرا کند تا بتواند به درستی کنترل شی مفهومی را در اختیار بگیرد. اینها نباید با کنترل‌های مفهومی، که توسط توسعه‌دهندگان CF نیز تعریف شده‌اند، یعنی این که هوش مصنوعی می‌تواند از آنها برای کنترل شی مفهومی استفاده کند و توسط توسعه‌دهندگان بازی اجرا می‌شود، اشتباه گرفته شوند.

این امکان وجود دارد که چندین کنترلر، کنترل یک شی را به اشتراک بگذارند. به عنوان مثال، یک NPC بسته به وضعیت آن می‌تواند توسط راه‌های هوش مصنوعی مختلف کنترل شود. ممکن است هوش مصنوعی، رزمی پیچیده‌ای داشته باشد که تنها زمانی وارد عمل می‌شود که NPC وارد حالت نبرد می‌شود و در غیر این صورت در حالت آماده باش باقی می‌ماند. در حالی که وقتی NPC در حالت بیکار است از هوش مصنوعی دیگری برای پرسه زدن، سرگردانی یا استراحت استفاده می‌شود. با این حال، کنترل‌کننده‌های متعدد ممکن است در مواردی که کنترل همپوشانی دارند، منجر به درگیری شود. یکی از راه‌های حل تعارض‌ها این است که کنترل‌کننده‌های هوش مصنوعی جدولی داشته باشند که سطح اولویت را برای هر کنترل مفهومی نشان دهد. سپس فراخوان‌های کنترل مفهومی توسط کنترل‌کننده با اولویت‌های مربوطه خود صادر می‌شود و البته در صف داوری قرار می‌گیرد. هنگامی که چندین کنترل‌کننده هوش مصنوعی در یک راه حل کامل ادغام می‌شوند، این موضوع می‌تواند توسط مرجع راه حل به هر روشی که آنها انتخاب کنند، رسیدگی شود و تنها کنترل‌کننده کامل می‌تواند نیاز به ارائه یک جدول اولویت برای کنترل‌های مفهومی داشته باشد.

نشان می‌دهیم که چگونه چندین کنترلر را می‌توان با یک شی مفهومی ثبت کرد. ابتدا یک شی در بازی به نام Peon Undead ساخته می‌شود. به دنبال آن، طرح ریزی آن در CDS، یک NPC ایجاد می‌شود و به Peon Undead متصل می‌شود و در نهایت، چندین کنترلر هوش مصنوعی مستقل، یکی برای ایجاد رفتار بیکار در زمانی که Peon Undead بیکار است، یکی دیگر برای ایجاد رفتار اجتماعی زمانی که Peon Undead در اطراف دیگر Peon‌ها و سایر انواع

NPC است و دیگری برای ایجاد رفتار رزمی زمانی که Peon ارواح با دشمنان روبرو می‌شود، ایجاد شده و در NPC در CDS ثبت می‌شود. در این مورد، هیچ همپوشانی در کنترل NPC توسط راه حل‌های مختلف هوش مصنوعی وجود ندارد. با استفاده از این مکانیسم ثبت، یک کنترل‌کننده هوش مصنوعی همچنین می‌تواند تأیید کند که وابستگی‌هایش در حال اجرا هستند و از طریق شی مفهومی به آنها دسترسی داشته باشد.

نمونه‌هایی از توابع موجود در رابط‌های کنترلر، یک تابع به روزرسانی و کنترل‌کننده رویداد است. یک تابع به روزرسانی، برای به روزرسانی وضعیت داخلی هوش مصنوعی استفاده می‌شود و می‌توان آن را در هر چرخه بازی یا با نرخ به روزرسانی دلخواه تنظیم کرد. توجه داشته باشید که چگونه NPC در CDS هیچ چرخه به روزرسانی حالت داخلی ندارد. این به این دلیل است که هیچ دینامیکی در CDS وجود ندارد. اشیاء در CDS پیش بینی اشیاء بازی هستند و فقط در نتیجه تغییر در اشیاء بازی اصلاح می‌شوند. کنترل‌کننده‌های رویداد برای اطلاع دادن به کنترل‌کننده‌های هوش مصنوعی از رویدادهای بازی، مانند کشته شدن یک واحد توسط دیگری، استفاده می‌شوند. هنگامی که یک رویداد در بازی رخ می‌دهد، یک طرح مفهومی در طرح ریزی شی بازی درگیر شلیک می‌شود. رویدادهایی که می‌توانند شامل یک شی مفهومی باشند توسط توسعه دهندگان CF تعیین می‌شوند و برای ایجاد رابط کنترل استفاده می‌شوند. یک کنترل‌کننده هوش مصنوعی لزوماً نیازی به مدیریت همه رویدادها ندارد. این برای کنترلرهای جزئی واضح است. بنابراین، این امکان برای کنترلرهای هوش مصنوعی وجود دارد که برخی رویدادها را نادیده بگیرند. نمونه‌های دیگر توابعی برای تعلیق و از سرگیری کنترلر هستند.

وقتی توسعه‌دهندگان بازی راه‌حل‌های هوش مصنوعی را به بازی‌های خود پیوند می‌دهند، می‌توانند آنها را در زمان ساخت به صورت ایستا پیوند دهند یا در زمان اجرا به صورت پویا بارگذاری کنند. بارگیری هوش مصنوعی در زمان اجرا آزمایش راه‌حل‌های مختلف هوش مصنوعی را آسان تر می‌کند و همچنین می‌تواند به بازیکنان اجازه دهد تا هوش مصنوعی خود را به بازی متصل کنند. به طور معمول، هوش مصنوعی در فرآیند بازی ویدیویی اجرا می‌شود، اگرچه می‌تواند، اجرای آنها به صورت جدا جالب هم باشد. استقرار هوش مصنوعی در یک فرآیند جداگانه به این معنی است

که می‌تواند روی دستگاه دیگری اجرا شود. دومی می‌تواند برای پردازش هوش مصنوعی بهینه شود یا حتی می‌تواند در اینترنت باشد و این امکان را برای توسعه دهندگان هوش مصنوعی فراهم می‌کند تا هوش مصنوعی را به عنوان یک سرویس ارائه دهند.

فصل ۶

نتیجه گیری

سهم اصلی این تحقیق رویکردی برای توسعه هوش مصنوعی برای بازی‌های ویدیویی بر اساس استفاده از یک چارچوب مفهومی یکپارچه برای ایجاد یک لایه مفهومی بین بازی و هوش مصنوعی است. (هوش مصنوعی که در اینجا به آن اشاره می‌شود مربوط به بازی است و هوش مصنوعی مرتبط با زمینه را که در ابتدای این کار مشخص شده است شامل نمی‌شود.) این رویکرد از تفسیری از رفتار انسان الهام گرفته شده است. بازیکنان انسانی توانایی تشخیص قیاس بین بازی‌ها و تعمیم یا مفهوم سازی دانش کسب شده در یک بازی و به کارگیری آن در بازی دیگر را دارند. با مفهوم سازی بازی‌های ویدیویی و درخواست از توسعه دهندگان بازی برای ایجاد نماهای مفهومی از بازی‌های خود با استفاده از یک چارچوب یکپارچه، ایجاد راه حل‌هایی برای مشکلات مفهومی رایج و استفاده از آنها در چندین بازی ویدیویی امکان پذیر می‌شود. توسعه راه‌حل‌هایی برای مشکلات مفهومی به جای بازی‌های ویدیویی خاص به این معنی است که طراحی هوش مصنوعی دیگر محدود به محدوده پروژه‌های بازی‌های فردی نیست و می‌تواند در طول زمان به طور کارآمدتر اصلاح شود.

کتابنامه

- [1] The NetBSD Foundation, “Portability and supported hardware platforms,” <http://netbsd.org/about/portability.html>.
- [2] Microsoft, Windows NT Hardware Abstraction Layer (HAL), <http://support.microsoft.com/kb/99588>.
- [3] A. Nareyek, N. Combs, B. Karlsson, S. Mesdaghi, and I. Wilson, “The 2003 report of the IGDA’s artificial intelligence interface standards committee,” Tech. Rep., International Game Developers Association, 2003, <http://www.igda.org/ai/report2003/report-2003.html>, <http://archive.org/web/>.
- [4] A. Nareyek, N. Combs, B. Karlsson, S. Mesdaghi, and I. Wilson, “The 2004 report of the IGDA’s artificial intelligence interface standards committee,” Tech. Rep., International Game Developers Association, 2004, <http://www.igda.org/ai/report2004/report-2004.html>.
- [5] A. Nareyek, N. Combs, B. Karlsson, S. Mesdaghi, and I. Wilson, “The 2005 report of the IGDA’s artificial intelligence interface standards committee,” Tech. Rep., International Game Developers Association, 2005, <http://www.igda.org/ai/report2005/report-2005.html>, <http://archive.org/web/>.
- [6] B. Yue and P. de Byl, “The state of the art in game AI standardisation,” in Proceedings of the 2006 International Conference on Game Research and Development, pp. 41–46, Murdoch University., 2006.
- [7] B. F. F. Karlsson, “Issues and approaches in artificial intelligence middleware development for digital games and entertainment products,” CEP 50740:540, 2003.

- [8] C. Berndt, I. Watson, and H. Guesgen, “OASIS: an open AI standard interface specification to support reasoning, representation and learning in computer games,” in Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games (IJCAI ’05), pp. 19–24, 2005.
- [9] Unity Technologies, “Unity—Game Engine,” <http://unity3d.com/>.
- [10] Epic Games, Unreal Engine Technology — Home, <https://www.unrealengine.com/>.
- [11] Crytek, CRYENGINE: The complete solution for next generation game development by Crytek, <http://cryengine.com/>.
- [12] Havok, <http://www.havok.com/>.
- [13] B. Kreimeier, The case for game design patterns, 2002, http://www.gamasutra.com/view/feature/132649/the_case_for_game_design_patterns.php?print=1.
- [14] S. Björk, L. Sus, and H. Jussi, “Game design patterns,” in Proceedings of the Level Up-1st International Digital Games Research Conference, Utrecht, The Netherlands, November 2003.
- [15] S. Björk and J. Holopainen, “Describing games—an interaction-centric structural framework,” in Level Up: Proceedings of Digital Games Research Conference, 2003.
- [16] C. M. Olsson, S. Björk, and S. Dahlskog, “The conceptual relationship model: understanding patterns and mechanics in game design,” in Proceedings of the DiGRA International Conference (DiGRA ’14), 2014.
- [17] A. B. Loyall and J. Bates, “Hap: a reactive, adaptive architecture for agents,” Tech. Rep. CMU-CS-97-123, Carnegie Mellon University, School of Computer Science, 1991.
- [18] M. Mateas and A. Stern, “A behavior language for story-based believable agents,” IEEE Intelligent Systems and Their Applications, vol. 17, no. 4, pp. 39–47, 2002.

- [19] M. Mateas and A. Stern, “A behavior language: joint action and behavioral idioms,” in *Life-Like Characters, Cognitive Technologies*, pp. 135–161, Springer, Berlin, Germany, 2004.
- [20] J. D. Funge, “Making them behave: cognitive models for computer animation,” 1998.
- [21] J. Funge, X. Tu, and D. Terzopoulos, “Cognitive modeling: knowledge, reasoning and planning for intelligent characters,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp.29–38,ACM Press/Addison-Wesley, 1999.
- [22] J. Funge, “Representing knowledge within the situation calculus using interval-valued epistemic fluents,” *Reliable Computing*, vol.5,no.1,pp.35–61,1999.
- [23] J. Orkin, “Symbolic representation of game world state: toward real-time planning in games,” in *Proceedings of the AAAI Workshop on Challenges in Game Artificial Intelligence*,2004.
- [24] J. Orkin, “Agent architecture considerations for real-time planning in games,” in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE '05)*, pp. 105–110, 2005.
- [25] E. F. Anderson, “Scripting behaviour—towards a new language for making NPCs act intelligently,” in *Proceedings of the zfxCON05 2nd Conference on Game Development*,2005.
- [26] E. F. Anderson, “SEAL—a simple entity annotation language,” in *Proceedings of zfxCON05-2nd Conference on Game Development*,pp.70–73,StefanZerbst,Braunschweig,Germany,2005.
- [27] E. F. Anderson, “Scripted smarts in an intelligent virtual environment,” in *Proceedings of the Conference on Future Play: Research, Play, Share*, pp. 185–188, ACM, 2008.
- [28] D. C. Cheng and R. Thawonmas, “Case-based plan recognition for real-time strategy games,” in *Proceedings of the 5th International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE '04)*,pp.36–40,2004.

- [29] D.W.Aha,M.Molineaux,and M. J. V. Ponsen,“Learning to win: case-based plan selection in a real-time strategy game,” in Proceedings of the 6th International Conference on Case-Based Reasoning (ICCBR ’05),pp.5–20,August2005.
- [30] S. Ontañón,K.Mishra,N.Sugandh,andA.Ram,“Case-based planning and execution for real-time strategy games,” in Case-Based Reasoning Research and Development: 7th International Conference on Case-Based Reasoning, ICCBR 2007 Belfast, NorthernIreland,UK,August13–16,2007Proceedings,vol.4626, pp. 164–178, Springer, Berlin, Germany, 2007.
- [31] B. Weber and M. Mateas, “Conceptual neighborhoods for retrieval in case-based reasoning,” in Proceedings of the 8th International Conference on Case-Based Reasoning (ICCBR ’09), pp. 343–357, 2009.
- [32] B. G. Weber and M. Mateas, “Case-based reasoning for build order in real-time strategy games,” in Proceedings of the 5th Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE ’09), pp. 106–111, October 2009.
- [33] M.Sharma, M. Holmes,J.Santamaria, A. Irani, C. Isbell,and A. Ram, “Transfer learning in real-time strategy games using hybrid CBR/RL,” in Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI ’07), pp. 1041–1046, January 2007.
- [34] S. Lee-Urban, H. Muñoz-Avila, A. Parker, U. Kuter, and D. Nau, “Transfer learning of hierarchical task-network planning methods in a real-time strategy game,” in Proceedings of the 17th International Conference on Automated Planning & Scheduling (ICAPS ’07), Workshop on AI Planning and Learning (AIPL), 2007.
- [35] D. Shapiro, T. Konik, and P. O’Rorke, “Achieving far transfer in an integrated cognitive architecture,” in Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI ’08), pp.1325–1330,July2008.
- [36] M. Buckland, Programming Game AI by Example, Jones & Bartlett Learning, 2004.
- [37] B. Schwab, AI Game Engine Programming, Cengage Learning, 2008.

- [38] I. Millington and J. Funge, *Artificial Intelligence for Games*, CRC Press, Boca Raton, Fla, USA, 2009.
- [39] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2009.
- [40] S. Rabin, *AI Game Programming Wisdom*, CharlesRiverMedia, 2002.
- [41] S. Rabin, *AI Game Programming Wisdom 2*, Cengage Learning, 2003.
- [42] S. Rabin, *AI Game Programming Wisdom 3*, Cengage Learning, Boston, Mass, USA, 2006.
- [43] S. Rabin, *AI Game Programming Wisdom 4*, CharlesRiver Media Group, 2008.
- [44] R. Straatman and A. Beij, “Killzone’s AI: dynamic procedural combat tactics,” in *Proceedings of the Game Developers Conference*, 2005.
- [45] D. Pottinger, “Implementing coordinated movement,” *Game Developer Magazine*, pp. 48–58, 1999.
- [46] bwapi—An API for interacting with Starcraft: Broodwar (1.16.1)—Google Project Hosting, <https://code.google.com/p/bwapi/>.



College of Science
School of Mathematics, Statistics, and Computer Science

Reviewing AI algorithms in computers games

Author name
Mohammad Reza Hassani

Supervisor: Dr.Hedie Sajedi

A thesis submitted in partial fulfillment of the requirements for
the degree of B.Sc. in Computer Science

2023