

ارائه طرحی برای حمله هوشمندانه ارتش به سرزمین دشمن در یک بازی استراتژی بلادرنگ

زهرا زینالپور تبریزی، بهروز مینایی بیدگلی

چکیده

بازی‌های استراتژی بر مبنای رقابت دو یا چند بازیکن در دنیای بازی، سعی در چیره شدن بر حریفان و جنگ بین رقبا است. تمام جذابیت این بازی‌ها رقابت با حریفی هوشمند است. به دلیل پیچیدگی دنیای بازی و حجم گسترده اطلاعات در این سبک از بازی‌ها، روش‌های کلاسیک هوش مصنوعی جوابگوی نیاز NPCها نیستند. یکی از بخش‌های اصلی در تولید یک NPC، طراحی تاکتیک حمله است. در این مقاله سیستمی برای کنترل ارتش در هنگام حمله به شهر دشمن پیشنهاد شده است. در این سیستم با در نظر گرفتن ویژگی‌های عناصر موجود در شهر دشمن، شرایط دنیای بازی و توازن نیروهای دو ارتش اولویت‌های حمله مشخص می‌شود. ارتش به تیم‌هایی با قدرت متناسب با خطر و هزینه دستیابی به اهداف تقسیم می‌شود. طرح پیشنهادی در بازی RTS جامع StarCraft پیاده شده و پس از انجام سی مورد آزمایش رفتار NPC هوشمندانه، مقتدرانه و طبیعی ارزیابی شد.

کلمات کلیدی

حداکثر ده کلمه به عنوان کلمات کلیدی انتخاب شود. این کلمات باید موضوعات اصلی و فرعی مقاله را نشان دهند.

A System for Controlling an Intelligent Attack to Enemy City in Real-Time Strategy Games

Abstract

Strategy games are based on competition between two or more players in a game world, who means to try to defeat other competitor in the war. All the desirability of these kinds of games is in challenging an intelligent enemy. Because of complexity of the game world, and its huge amount of information, the classic methods of IA are not sufficient in creating NPCs. One of the main parts in creating a NPC is the design of tactic for attack. In this article, we proposed a system which controls the army's attack to enemy city. In this system, by considering the specification of the enemy city, the condition of the game and power balance between two armies, we locate the priorities of the attack. Then we divide army to teams which suits the threat and goal costs. Our proposed system is implemented in the RTS Star Craft game and after thirty experimental games, the NPC behavior evaluated as intelligent, powerful and natural.

Keywords

Non Player Character (NPC), Strategy, Tactic, Real-Time Strategy (RTS), Influence Map.

۱- مقدمه

تصمیماتی که یک بازیکن در یک بازی استراتژی بلادرنگ باید اتخاذ نماید بسیار پیچیده هستند و لذا توصیه می‌شود تصمیم‌گیری به شکل سلسله مراتبی انجام شود. ما تصمیم‌گیری را به سه لایه کلی استراتژی، تاکتیک و رفتار تقسیم کردیم. شکل (۱) را مشاهده نمایید.

۲- پیشینه تحقیق

در سال‌های اخیر پژوهش‌های فراوانی بر روی هوش مصنوعی در بازی‌های استراتژی انجام شده است که در ادامه به برخی از آنها اشاره می‌کنیم. مقاله [۲] به کنترل تاکتیکی بازیکنان پرداخته و الگوریتم تکاملی تولید کرده‌اند که با ارتقاء نقشه‌های نفوذ و سپس با استفاده از الگوریتم مسیریابی A* مسیر حرکت را برای هر نیرو مشخص می‌کند. این سیستم تعدادی مدافع و مهاجم را تولید نموده و آن‌ها را با استفاده از الگوریتم ژنتیک و با بازی مقابل هم تکامل می‌دهد. تابع برازش در این الگوریتم ترکیب ساده‌ای از تعداد پیروزی‌ها و شرایط توازن نیروها در بازی است. با وجود این که این بازی نوعی بازی استراتژی محسوب می‌شود، اما بسیار ساده است. روش ارائه شده در این مقاله برای کنترل کشتی‌ها در صورت کم بودن تعداد آن‌ها رفتار غیرقابل پیش‌بینی و متنوعی از خود بروز می‌دهد. در مقاله [۳]، عامل هوشمند برای بازی‌های RTS در بستر موتور بازی ORTS توسعه داده شده است. میان‌افزار توسعه داده شده شامل بخش‌های گروه‌بندی، پیدا کردن مسیر و سیستم کنترل رفتارهای سطح پایین نیروها می‌باشد. برای هوشمندی عامل‌ها، از دانش انسانی استفاده شده و جهت کنترل رفتار هر عامل یک FSM تولید شده است.

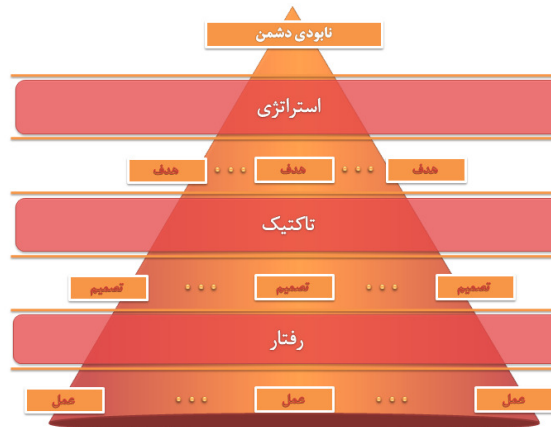
در مقاله [۴]، پیشنهاد استفاده از یکی از تکنیک‌های محاسباتی هوشمند مشهور برای بهبود رفتار AI ارائه شده است. جهت تعیین ترکیب تیم‌ها جهت جنگ در مقابل گروه‌های مشخص دشمن از SOM استفاده شده است، که برای سازگار شدن با بازی بر یک الگوریتم تکاملی متکی است. از این طریق به شکل تقریباً بهینه توانایی‌های مختلف واحدها استفاده می‌شود، که یادآور تصمیمات سردستی انسان‌هاست.

نویسندگان مقاله [۵] ابتدا با استفاده از روش‌های مقالات قبلی خود [۶] طرح بازی اولیه را ساخته، گراف وابستگی را برای طرح مطرح شده رسم می‌کنند؛ از طریق گراف وابستگی رابطه حرکت نیروها را مشخص می‌نمایند. سپس طرح بازی با استفاده از گراف وابستگی برای شرایط فعلی بازی به صورت بلادرنگ تطبیق‌پذیر می‌شود.

پس از بررسی مسائل موجود در تولید NPC برای یک بازی RTS و مرور تحقیقات انجام شده در این زمینه، روشن شد که زمینه کار بسیار زیادی در هر سه لایه تصمیم‌گیری وجود دارد. نتایج کسب شده در لایه رفتار بیش از دو لایه دیگر رضایت‌بخش هستند. در دو لایه استراتژی و تاکتیک، کارهای انجام شده غالباً بر روی بازی‌های ساده و با اعمال محدودیت‌های فراوان انجام گرفته‌اند؛ لذا خلأ کار بر روی یک

بازی‌های کامپیوتری امروزه یکی از صنایع مهم و پیشرفته در سطح دنیا محسوب می‌شود. بازی‌ها را می‌توان بر اساس ویژگی‌هایشان به سبک‌های مختلفی تقسیم کرد. جدا از بازی‌هایی که به سختی بر شبیه‌سازی فیزیکی متکی هستند، مثل بازی‌های ماشین‌رانی و مبارزه‌ای، اکثر بازی‌ها به یک رقیب هوشمند، که در مقابل ماست و با ما رقابت می‌کند، نیازمند هستند. این بازیکن‌های مصنوعی، باید رقابتی باورپذیر باشند و رفتارهای هوشمندانه‌ای در بستر بازی، به نمایش بگذارند [۱].

چالش‌های موجود در تولید بازیکن‌های مصنوعی در بازی‌های RTS، به شکل قابل توجهی از دیگر سبک‌های بازی‌های ویدئویی بیشتر است. دلیل این امر پیچیدگی بسیار زیاد دنیای بازی و گستردگی عواملی است، که تحت کنترل NPC هستند؛ همچنین در این بازی‌ها تمام تصمیمات و اعمال NPC به یکدیگر وابسته بوده و روی هم تأثیرگذار هستند. در این نوع بازی معمولاً هدف هر بازیکن نابودی کامل دشمن است. منابع محدودی در سطح دنیای بازی وجود دارد که بازیکن باید با استفاده از آنها و با پیروی از قوانین و ساختارهای بازی امکاناتی برای خود آماده نماید، در مقابل حملات دشمنان از خود دفاع کرده و در زمان مناسب به دشمن حمله کند. در واقع این بازی یک ماکت از شرایط جنگی در دنیای واقعی است.



شکل (۱): ساختار سلسله مراتبی تصمیم‌گیری یک NPC در یک بازی RTS

یک بازی استراتژی جامع شامل عناصر بسیار زیاد با فراوانی قابل توجه است. این عناصر را می‌توان به سه دسته اصلی تقسیم کرد، نقشه، منابع، عناصر. کارهایی که هر بازیکن در طول بازی باید انجام دهد شامل هشت گروه اصلی است: تحلیل نقشه، ساخت ساختمان، مدیریت منابع، مدیریت ارتش در هنگام حمله، مدیریت نیروها در زمان دفاع، کنترل رفتار نیروها، کنترل رفتار جمع‌آوری منابع و مدل کردن رفتار دشمن. در این مقاله برای ارزیابی طرح پیشنهادی از بازی RTS جامع StarCraft استفاده شده است.

اتفاق می‌افتد که یکی از تیم‌های ارسال شده در وضعیت بدی قرار گرفته و نیازمند تغییر موضع یا دریافت نیروی کمکی می‌باشد. در صورت وقوع هر کدام از این حالات، مدیر تاکتیک حمله، طراح تاکتیک حمله را فراخوانی می‌نماید تا تصمیم جدیدی که با شرایط جدید همخوانی داشته باشد، اتخاذ نماید.

۳-۲- طراح تاکتیک حمله

ماژول طراح تاکتیک حمله شامل شش بخش اصلی است که در شکل (۳) قابل مشاهده است. ابتدا تابع هیوریستیک بر اساس یک مجموعه قوانین که از بازیکنان خبره استخراج شده است، شهر دشمن را بررسی نموده و صفی از مجموعه اهداف را مشخص می‌نماید. تحلیل‌گر نقشه، دنیای بازی را از جنبه‌های مختلف تحلیل نموده و به شکل یک نقشه نفوذ در اختیار تصمیم‌گیرنده قرار می‌دهد.

ماژول اصلی این بخش ماژول تصمیم‌گیرنده است. این ماژول بر اساس نقشه نفوذ و با کمک تابع مسیریابی، اولویت‌ها را مشخص نموده و با فراخوانی ماژول مدیر نیروها، به هر یک از اهداف تعیین شده، تیمی از نیروها را منتسب می‌کند. سپس هر تیم، هدف و مسیر رسیدن به آن را به لایه رفتار ارسال می‌نماید تا حمله انجام گیرد. بازخورد عملیات هر تیم به روال یادگیری ارسال می‌شود، در این روال میزان موفق بودن تصمیم‌ها ارزیابی می‌شود و میزان انتساب نیرو به مرور با شرایط بازی وفق پیدا می‌کند.



شکل (۳): ساختار ماژول طراح تاکتیک حمله

در این مقاله ما پایه تصمیم‌گیری را تجربه و دانش بازیکن حرفه‌ای قرار داده‌ایم؛ سپس با احتساب شرایط بازی، تصمیم‌گیری نموده، یک NPC هوشمند و قدرتمند با رفتاری معقول و متناسب تولید نمودیم. بازیکنان حرفه‌ای و باتجربه بازی‌های RTS، بر اساس وضعیت شهر دشمن می‌توانند اولویت‌هایی برای نابودی اهداف تعیین کنند. خبرگی افراد خبره در زمینه تشخیص اولویت‌های حمله در بین اهداف و شرایط آن، حاصل تجربه زیاد آن‌ها در این نوع بازی‌هاست. بنابراین برای این که چارچوب اولیه تصمیمات ما هوشمندانه و طبیعی باشد، با دو نفر خبره در بازی StarCraft مصاحبه انجام داده و یک مجموعه قوانین تنظیم کردیم.

بازی RTS جامع و ارائه روش‌هایی که در عمل، در بازی‌های آینده قابل استفاده باشند، وجود دارد. در این مقاله تمرکز بر روی یکی از بخش‌های اصلی لایه تاکتیک می‌باشد. این بخش را تاکتیک حمله نامیدیم؛ که مسئول تصمیم‌گیری تاکتیکی در هنگام حمله به شهر دشمن است.

۳-۳- ماژول تاکتیک حمله

طرح پیشنهادی این مقاله، ساختاری برای کنترل تاکتیک‌های حمله ارتش به شهر دشمن است. هنگامی که لایه استراتژی تصمیم حمله به محدوده دشمن را می‌گیرد و ارتشی از نیروهای خود را برای این کار اختصاص می‌دهد، ماژول پیشنهادی هدایت حمله را بر عهده می‌گیرد. ارتش باید با توجه به ساختار شهر دشمن، آرایش نظامی دشمن، دارایی‌ها، سیستم اقتصادی و دفاعی او و عکس‌العمل دشمن در مقابل حرکات خود بهترین تصمیم‌ها را بگیرد. این تصمیمات شامل تعیین اهداف، مشخص کردن اولویت‌ها، اختصاص نیرو و تعیین راه دستیابی به این اهداف است. ساختار ارائه شده دارای چندین بخش مستقل است؛ که هر یک به شکل مستقل قابلیت تغییر یا ارتقاء دارند.

ساختار کلی ماژول تاکتیک حمله پیشنهادی در شکل (۲) مشاهده می‌شود. این ماژول از دو بخش اصلی تشکیل شده است. بخش اول "مدیر تاکتیک حمله" و بخش دوم "طراح تاکتیک حمله" است. تصمیماتی که ماژول تاکتیک حمله اتخاذ می‌کند جهت اجرا به لایه رفتار ابلاغ می‌شود.



شکل (۲): ساختار کلی ماژول تاکتیک حمله

۳-۱- مدیر تاکتیک حمله

وظیفه مدیر تاکتیک حمله، تعیین زمان‌هایی است که نیازمند یک تصمیم جدید برای حمله هستیم. این بخش، در سه حالت، فرمان تصمیم‌گیری جدید را صادر می‌کند. اولین حالت زمانی است که لایه استراتژی، ماژول تاکتیک حمله را فراخوانی نموده و دستور حمله به شهر دشمن را صادر می‌نماید. دومین حالت زمانی است که یکی از تیم‌های ارسال شده برای انجام حمله، کار خود را به پایان رسانده و آماده است که مأموریت جدیدی بر عهده بگیرد. سومین حالت هنگامی

HP) میزان سلامت گره‌ها یا هزینه نابودی آن‌ها و $D(u)$ درجه ورودی گره u را نشان می‌دهد.

$$P(v) = -HP(v) + \sum_{i=0}^n \left[\frac{1}{D(u_i)} * HP(u_i) \right] \quad (1)$$

بعد از تعیین بهینه‌ترین مجموعه ساختمان‌ها که نابودی آنها باعث از کار انداختن مجموعه ساختمان‌های مورد نظر می‌شود. خروجی تابع هیوریستیک به شکل صافی از مجموعه‌های هم اولویت اهداف به تصمیم‌گیرنده ارسال می‌گردد.

تحلیل گر نقشه، داده‌های دنیای بازی را که حجم بسیار زیادی دارند و از انواع مختلفی هستند، تحلیل نموده و به شکل قابل استفاده برای NPC در می‌آورد. خروجی این بخش یک نقشه نفوذ است. نقشه نفوذ دنیای بازی را به شکل شبکه‌ای از اطلاعات گسسته و قابل استفاده در می‌آورد. اطلاعات با عناوین هزینه، خطر ساکن و خطر متحرک تحلیل شده، تخمین زده و استخراج می‌شود.

سپس مسیریاب با استفاده از نقشه نفوذ و اطلاعات ارتش، هزینه و خطر دستیابی به اهداف با اولویت‌های مشابه را تخمین می‌زند، این کار با اعمال الگوریتم مسیریابی A^* بر روی نقشه نفوذ، به ازای تمام اهداف و تیم‌ها انجام می‌شود. جهت تعیین اولویت اهداف تنها دو مقدار خطر ساکن و هزینه مسیر را در نظر می‌گیریم و مقدار خطر متحرک عملاً تنها در زمان اختصاص نیرو به کار می‌رود. مقدار خطر ساکن و هزینه هر یک از اهداف توسط روابط (۲) و (۳) محاسبه می‌گردند.

$$D_k = \sum_{i=1}^n \frac{d_{ik} * F(T_i)}{\sum_{j=1}^n F(T_j)} \quad (2)$$

$$C_k = \sum_{i=1}^n \frac{c_{ik} * F(T_i)}{\sum_{j=1}^n F(T_j)} \quad (3)$$

در این روابط D_k خطر متحرک رسیدن به هدف k ام است. n تعداد تیم‌هایی است که در نوبت قبل تصمیم‌گیری، تشکیل داده‌ایم. d_{ik} میزان خطر متحرکی است که تابع مسیریاب برای حرکت از مختصات تیم i ام به هدف k ام تعیین کرده است. $F(T_i)$ قدرت کل T_i را نشان می‌دهد. هزینه رسیدن به هدف k ام با C_k نشان داده شده است. c_{ik} میزان هزینه‌ای است که تابع مسیریاب برای حرکت از مختصات تیم i ام به هدف k ام تعیین کرده است.

برای تعیین ارزش اهداف بر اساس نوع ساختمان یا نیرو، از رابطه‌ای که بر اساس تعاریف آن در بازی مورد نظر به دست می‌آوریم، استفاده می‌شود. سپس با استفاده از خطر متحرک و هزینه دستیابی

خروجی تابع هیوریستیک مبنای تصمیم‌گیری تاکتیکی NPC است. به کارگیری دسته‌ای از قوانین خبره و استخراج اهداف به عنوان اولویت‌های بعدی، به سادگی انجام می‌گیرد. اما اعمال دسته‌ای از این قوانین نیازمند انجام محاسبات است، کاری که در این‌جا فرد خبره با تخمین زدن و محاسبات ذهنی انجام می‌دهد، NPC با استفاده از الگوریتم پیشنهادی انجام می‌دهد.

هنگامی که طبق قوانین خبره، نابودی نوعی از ساختمان‌ها در اولویت است، زیرمجموعه‌ای از این ساختمان‌ها و ساختمان‌های تغذیه‌کننده آن‌ها، به عنوان مجموعه اهداف تعیین می‌شود، این زیرمجموعه باید دارای دو خصوصیت باشد؛ اولاً با نابودی اعضای آن، تمامی ساختمان‌های اولیه مد نظر از کار بیفتند و دوم هزینه نابودی این زیرمجموعه کمترین مقدار ممکن باشد.

```
Build Semantic Network SN From the map,
While(SN is not empty) do
{
  Delete all Chargers with input degree = 0 from SN
  For (all Building With input degree = 1) do
  {
    if (building.DestroyCost > charger.DestroyCost) then
    {
      Add the linked charger to this building to GoalSet;
      Delete this building from SN;
    }
  }
  For all the nodes in SN calculate
  Node.Value;
  Add the node with max of Node.Value to GoalSet;
  Delete this node and its connections from SN;
}
```

شکل (۴): شبه کد تعیین بهترین مجموعه ساختمان جهت حمله

برای پیدا کردن این زیر مجموعه یک الگوریتم پیشنهاد کرده‌ایم. در این الگوریتم نقشه زمین بازی را تحلیل نموده و روابط بین ساختمان‌ها و منابع تغذیه را در قالب یک شبکه معنایی پیاده می‌کنیم. هر گره یا یک ساختمان است و یا یک منبع تغذیه و در صورتی دو گره با هم مجاور هستند که یکی از آن‌ها منبع تغذیه و دیگری ساختمانی در محدوده حمایت آن باشد. پس از استخراج شبکه معنایی بر اساس شبه‌کد (۴) عمل می‌کنیم. میزان ارزش هر یک از گره‌های این شبکه را بر اساس رابطه (۱) تعیین می‌کنیم.

در این رابطه، ارزش هر گره بر اساس میزان هزینه نابودی آن و میزان تأثیر این نابودی بر گره‌های مجاور آن، محاسبه شده است. تأثیری که نابودی یک گره بر ساختمان‌های مجاور آن دارد، وابسته به درجه ورودی و هزینه نابودی آن‌ها است. در این رابطه $P(v)$ میزان ارزش نابودی گره v است. n تعداد گره‌های مجاور گره v است و

۷	۸۸	۷۸۴	%۰	%۰	۰	عقب‌نشینی	-
۸	۲۰۰	۱۸۰	%۲۲.۵۵	%۱۰۰	۱۰۵۰	پیروزی	۹.۵۵
۹	۳۶۴	۵۰۰	%۲۶.۲۰	%۱۰۰	۲۰۰۰	پیروزی	۹.۵۵
۱۰	۸۸	۸۸	%۶۳.۳۶	%۸۴.۲۵	۷۳۳	پ / ا / ع	۹.۸۸

جهت ارزیابی NPC پیشنهاد شده، از سه نفر بازیکن انسانی استفاده شد. انتخاب این سه بازیکن با هدف بررسی عملکرد هر بازیکن در ده شرایط طراحی شده در ده بازی جداگانه با NPC روبرو می‌شود. در جدول (۱) میانگین نتایج کسب شده در مقابل سه بازیکن انسانی ارائه شده است.

برای ارزیابی میزان هوشمندی یا طبیعی بودن رفتار NPC از سه بازیکن خواسته شد که یک عدد در بازه [۰، ۱۰] را به عنوان میزان هوشمندی کاراکتر تعیین کنند؛ میانگین این سه مقدار، عددی است که در جدول آورده شده است. میانگین کل این امتیازات در همه بازی‌ها، معادل عدد ۹.۴۹۲ شد؛ که بسیار رضایت‌بخش است.

بر اساس این نتایج می‌توان رفتار NPC طراحی شده را تحلیل کرد. در زمانی که قدرت NPC از دشمن بسیار کوچک‌تر باشد، NPC با اجتناب از درگیری، سعی در پیدا کردن راه نفوذ می‌نماید، تا حداکثر خسارتی که می‌تواند به دشمن وارد نماید؛ هنگامی که خسارت زدن به دشمن معادل از دست دادن چندین برابر نیرو باشد، کاراکتر اقدام به عقب‌نشینی می‌نماید. در زمان مساوی بودن قدرت دو ارتش، بسته به ساختار شهر دشمن، نتایج مختلفی حاصل شد، عملکرد NPC در این حالت قوی و هوشمند تحلیل می‌شود. در بازی‌هایی که ارتش NPC قدرتی بسیار بیشتر از حریف داشته باشد، به سرعت بازی را به نفع خود به پایان می‌رساند.

۵- نتیجه

در این مقاله بر روی یکی از بخش‌های اصلی لایه تاکتیک، یعنی بخش هدایت‌گر حمله متمرکز شدیم. به این منظور، ساختاری برای انجام یک حمله قوی و هوشمند به شهر دشمن، با ارتشی مشخص پیشنهاد کردیم. این بخش از لایه تاکتیک را ماژول تاکتیک حمله نامیدیم. طرح ارائه شده دارای دو بخش نظارت و طراحی است. بخش نظارت، رفتار کاراکتر و شرایط دنیای بازی را تحت نظر گرفته، زمان‌هایی که نیازمند تصمیم‌گیری جدیدی هستیم تشخیص می‌دهد. بخش تصمیم‌گیری، شرایط دنیای بازی را بررسی نموده و تاکتیک حمله را طراحی می‌نماید. سپس NPC پیشنهادی در محیط بازی StarCraft پیاده‌سازی و رفتار NPC قدرتمند و هوشمند ارزیابی شد.

مراجع

- [1] J. Hagelback and S.J. Johansson, "A study on human like characteristics in real time strategy games", IEEE Symposium, pp. 139-145, 2010.

به اهداف، معیار اولویت هر هدف را از طریق رابطه (۴) بدست می‌آوریم:

$$Me_k = V_k - \omega D_k - \delta C_k \quad (4)$$

در این رابطه معیار اولویت هدف Me_k ، V_k ارزش آن هدف و ω و δ ضرایب خطر و هزینه رسیدن به هدف هستند. اهدافی که تابع هیوریستیک در مجموعه‌هایی دسته‌بندی و مرتب کرده است، بر اساس معیار اولویت نیز مرتب می‌گردند. بدین شکل مهم‌ترین، تاثیرگذارترین و قابل تحقق‌ترین اهداف انتخاب می‌شوند.

مدیر نیروها، برای هر هدف تیمی با میزان قدرت متناسب از نیروهایی با انواع مختلف تشکیل و اختصاص می‌دهد، جزئیات این کار در مقاله پیشین ما [۷] شرح داده شده است. تیم‌ها، اهداف و مسیرهای حرکت آنها به لایه رفتار ارسال می‌گردد و حمله انجام می‌گیرد. روال یادگیری نتایج حمله را در هر حمله بررسی نموده و در صورت نیاز، ضریب قدرت لازم برای تشکیل تیم‌ها را تصحیح می‌نماید. این کار با بررسی میزان تلفات تیم‌ها و میزان بیکاری نیروها در طول انجام عملیات حاصل می‌گردد.

۴- ارزیابی

در این پروژه NPC وظیفه حمله به شهر دشمن را دارد و نیازمند حریفی است که از این شهر دفاع کند؛ سه راه‌کار برای ارزیابی وجود دارد: مبارزه با بازیکن انسانی، تولید NPCهای ساده با عملکرد ثابت در دفاع و مبارزه با NPC پیش‌فرض بازی StarCraft. به عنوان نقاط قوت استفاده از روش ارزیابی اول می‌توان گفت که بازیکن انسانی در شرایط مختلف عکس‌العمل‌های متنوعی نشان می‌دهد و در صورتی که عملکرد NPC قابل پیش‌بینی باشد و یا نقطه ضعفی داشته باشد، بازیکن از آن‌ها برای شکست NPC استفاده می‌نماید. علاوه بر این هدف نهایی این NPC بازی در مقابل انسان است و عملکرد آن در مقابل یک موجود هوشمند ملاک ماست.

جدول (۱): جدول میانگین نتایج ۳۰ بازی آزمایشی

بازی	قدرت ارتش	قدرت دفاعی	درصد تلفات ارتش	درصد تلفات قدرت دفاعی دشمن	ارزش نابود شده از ساختمان‌های دشمن	نتیجه	میزان طبیعی بودن رفتار
۱	۳۰۰	۵۰۴	%۴۲.۷۰	%۶۷.۷۰	۱۵۵۸	پ / ا / ع	۸.۶۶
۲	۶۸۰	۴۸۴	%۱۶.۹۴	%۱۰۰	۱۹۵۰	پیروزی	۹.۸۸
۳	۶۰	۷۲	%۶۴.۰۲	%۸۸.۸۴	۹۶۶	عقب‌نشینی	۸.۸۸
۴	۳۵۶	۴۴۰	%۳۵.۶۳	%۱۰۰	۱۷۰۰	پیروزی	۹.۵۵
۵	۸۸۰	۹۱۶	%۲۲.۵۸	%۱۰۰	۳۰۵۰	پیروزی	۹.۸۸
۶	۳۲۰	۳۳۶	%۷۰.۶۳	%۱۰۰	۱۲۰۰	پیروزی	۹.۵۵

- [2] P. Avery, S. Louis and B. Avery, "Evolving coordinated spatial tactics for autonomous entities using influence maps", IEEE Symposium, pp. 341-348, Milano, 2009.
- [3] S. Wintermute, J. Xu, and J. E. Laird, "SORTS: A Human-Level Approach to Real-Time Strategy AI", Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference, pp. 55-60, Stanford, California, USA, 2007.
- [4] M. Preuss, N. Beume, H. Danielsiek, T. Hein, B. Naujoks, N. Piatkowski, R. Stüker, A. Thom and S. Wessing, "Towards Intelligent Team Composition and Maneuvering in Real-Time Strategy Games", IEEE Transactions, vol. 2, no. 2, pp. 82-98, Jun. 2010.
- [5] N. Sugandh, S. Ontañón and A. Ram, "On-line case-based plan adaptation for real-time strategy games" in AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence, vol. 2, 2008.
- [6] S. Ontañón, K. Mishra, N. Sugandh and Ram, A. "Case-based planning and execution for real-time strategy games", In Proceedings of ICCBR, pp. 64-178, 2007.
- [7] Zeinalpour-Tabrizi Z, Minaei-Bidgoli B, "Intelligent Decision Making Process for an Attack in RTS Games", Third International Conference on Advances in Computer Engineering, Amsterdam, The Netherlands, 2012.