Passive Defence
Vol. 15, No. 3, Autumn 2024, Serial No. 59
ISSN: 2980-8030 , E-ISSN: 2008-6849

Multilayer Modeling Framework for Hybrid Warfare Game

Yaser Khorrami*

* Assistant Professor, Imam Hossein (AS) University, Tehran, Iran

(Received: 16/04/2024, Revised: 25/05/2024, Accepted: 29/06/2024, Published: 26/10/2024)
DOR: 20.1001.1.20086849.1403.15.3.8.9

ABSTRACT

Hybrid warfare has been one of the topics of interest in recent years. This warfare has been given special importance due to the involvement of different economic, political, social, military and similar layers and its analysis has been complicated. Therefore, in addition to the need to upgrade the war game from the traditional way to the hybrid field, its multi-layer modeling and simulation is one of the necessities of today's war game. In this article, for the first time, by using the capabilities of the Python programming language and inspired by NATO's multi-layer software in simulating such wars, an agent-based framework has been developed for the development of modeling and Simulation of Hybrid warfare should be presented and implemented in the form of a computer program using the pygame library and based on the Mesa format as a modular framework for building, analyzing and visualizing agent-based models. The presentation of this framework is the basis for the development of advanced simulations in the field of Hybrid warfare as a new paradigm in modern wars and plays an effective role in creating defensive capabilities in addition to offensive capabilities in this regard. The framework considered in this article has been implemented in two general formats, first for modeling and simulating conventional warfare and then for unconventional warfare (hybrid warfare). The second format includes upgrading the framework introduced in conventional warfare to hybrid warfare by adding two layers of economic and cyber warfare. The recent format shows that the existing simulation methods based on conventional warfare can be developed into hybrid warfare simulation methods with agent-based modeling and by adding different levels of warfare as separate topic layers.

Keywords: Hybrid Warfare, Modeling Framework, War Game, Agent-Based

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Publisher: Imam Hussein University © Authors

* Corresponding Author Email: ykhorami@ihu.ac.ir

پدافند غیرعامل

سال پانزدهم، شماره ۳، پاییز ۱۴۰۳، (پیاپی ۵۹): صص ۱۰۷-۹۷

شاپای چاپی: ۶۹۴۹-۲۰۰۸ | **شاپای الکترونیکی:** ۸۰۳۰-۲۹۸۰

علمی – ترویجی

چارچوب مدل سازی چندلایهٔ بازی جنگ ترکیبی

یاسر خرّمی[1*]

DOR: 20.1001.1.20086849.1403.15.3.8.9

تاریخ دریافت: ۱۴۰۳/۰۱//۲۸ تاریخ پذیرش: ۱۴۰۳/۰۴/۰۹

تاریخ بازنگری: ۱۴۰۳/۰۳/۰۵ تاریخ انتشار: ۱۴۰۳/۰۸/۰۵

چکیده

جنگ ترکیبی یکی از مقولات مورد توجه در سال‌های اخیر بوده است. این جنگ به دلیل درگیر کردن لایه‌های مختلف اقتصادی، سیاسی، اجتماعی، نظامی و مانند آن مورد اهمیت ویژه قرار گرفته و تحلیل آن با پیچیدگی همراه شده است. بنابراین ضمن نیاز به ارتقاء بازی جنگ از شیوه سنتی به حوزه ترکیبی، الگوسازی و شبیه‌سازی چند لایهٔ آن یکی از ملزومات بازیِ جنگ امروزی است. در این مقاله سعی شده است تا برای اولین بار با استفاده از ظرفیت‌های زبان برنامه‌نویسی پایتون و با الهام از نرم‌افزار چندلایهٔ ناتو در شبیه‌سازی این گونه جنگ‌ها، چارچوبی عامل‌بنیان برای توسعه الگوسازی و شبیه‌سازی جنگ ترکیبی ارائه و در قالب یک برنامه کامپیوتری با استفاده از کتابخانه بازی‌سازِ pygame و مبتنی بر قالب Mesa به‌عنوان یک چارچوب پودمانیِ ساخت، تجزیه و تحلیل و تجسم مدل‌های مبتنی بر عامل، پیاده‌سازی شود. ارائه چارچوب فوق زمینه‌ای برای توسعه شبیه‌سازی‌های پیشرفته در حوزه جنگ ترکیبی به عنوان پارادایم جدید در جنگ‌های نوین بوده و نقش موثری در ایجاد ظرفیت‌های پدافندی علاوه بر قابلیت‌های آفندی در این خصوص ایجاد می‌نماید. چارچوب مدنظر در این مقاله در دو قالبِ کلی ابتدا برای الگوسازی و شبیه‌سازی جنگ متعارف و سپس برای جنگ غیرمتعارف (جنگ ترکیبی) پیاده‌سازی شده است. قالب دوم، شامل ارتقاء چارچوب معرفی شده در جنگ متعارف به جنگ ترکیبی از طریق اضافه کردن دو لایهٔ نبرد اقتصادی و نبرد سایبری است. قالب اخیر نشان می‌دهد که روش‌های شبیه‌سازی موجود و مبتنی بر جنگ متعارف قابل توسعه به روش‌های شبیه‌سازی جنگ ترکیبی با الگوسازی عامل‌بنیان و از طریق اضافه نمودن سطوح مختلف نبرد به‌عنوان لایه‌های موضوعی مجزا است.

کلیدواژه‌ها: جنگ ترکیبی، چارچوب الگوسازی، بازی جنگ، عامل‌بنیان

[1] استادیار مهندسی برق و الکترونیک، دانشکده پیامبر اعظم دانشگاه جامع امام حسین (ع)، تهران، ایران (ykhorami@ihu.ac.ir)- نویسنده مسئول

* این مقاله یک مقاله با دسترسی آزاد است که تحت شرایط و ضوابط مجوز Creative Commons Attribution (CC BY) توزیع شده است.

ناشر: دانشگاه جامع امام حسین (ع) © نویسندگان

۱- مقدمه

بشر همیشه به دنبال راه‌های ابتکاری و جدید برای جنگ بوده و اشکال جدیدی از جنگ را ایجاد می‌کند. این فرآیند نوآورانه که در پس‌زمینه بی‌ثباتی راهبردی جهانی تنظیم شده، در دو دهه گذشته، یک پدیده ظاهراً جدید اما پیچیده به نام جنگ ترکیبی را ایجاد نموده است. این نوع از جنگ، متمایز از سایر اشکال جنگ مدرن در چندین جنبه کلیدی، چالشی منحصر به فرد را ارائه می‌دهد که به نظر می‌رسد سیاست‌گذاران و کارشناسان امنیتی را گیج می‌کند، در حالی که به بازیگرانی که از آن استفاده می‌کنند روش جدیدی برای دستیابی به اهداف خود در مواجهه با جنگ‌های متعارف و معمول با هدف ایجاد برتری اعتقادی و راهبردی و درنهایت تسلط بر قدرت از طریق ابزارهای مختلف یا متفاوت ارائه می‌دهد [۱].

جنگ ترکیبی از طیف گسترده‌ای از ابزارهای قدرت از جمله: سیاسی، اقتصادی، نظامی، نامتقارن و غیرنظامی استفاده می‌کند. علاوه بر این، شامل ابزارهای مبتنی بر رسانه‌های داده‌محور مانند: دیپلماسی، تروریسم، نیابت‌ها و حملات اقتصادی برای متقاعد کردن مردم یا تقسیم جوامع می‌شود. جنگ ترکیبی آسیب‌پذیری‌های یک جامعه و سامانه را هدف قرار می‌دهد در حالی که عمداً از ابهام برای جلوگیری از شناسایی و باقی‌گذاردن ردپای خود، سوء استفاده می‌کند. معمولاً زمانی تشخیص داده می‌شود که کاملاً به نتیجه کاربردی برسد و بتواند آسیب عمده‌ای را وارد کند [۲].

امروزه الگوسازی و شبیه‌سازی به کمک کامپیوتر به عنوان یک ابزار علمی قدرتمند عمل می‌کند که می‌تواند برای بررسی سامانه‌های پیچیده در حوزه‌های مختلف مورد استفاده قرار گیرد.

بازی جنگ معمولاً به عنوان بخشی از فرآیند تصمیم‌گیری نظامی انجام می‌شود، که قبل از نهایی کردن یک طرح، دوره‌های عمل مختلف را تولید، تجزیه و تحلیل و مقایسه می‌کند [۳]. در طول بازی جنگی، افسران ستاد هر دوره عملیات را به فازهایی تقسیم می‌کنند. پس از هر مرحله، افسر اطلاعاتی با نقش آفرینی به عنوان دشمن، در مورد اینکه دشمن برای واکنش به آن مرحله چه کاری انجام خواهد داد، بحث می‌کند. این واکنش باید برای هر مرحله بعدی در نظر گرفته شود، تا زمانی که ماموریت به حالت پایانی برسد. این فرآیند برای هر اقدامی تکرار می‌شود تا فردی که بیشترین احتمال موفقیت را دارد، شناسایی شود [۴].

در این مقاله سعی شده است تا برای اولین بار، به زعم نویسنده، با استفاده از ظرفیت‌های زبان برنامه‌نویسی پایتون و با الهام از نرم‌افزار چندلایه ناتو در شبیه‌سازی این گونه جنگ‌ها، چارچوبی[1] عامل‌بنیان برای توسعه الگوسازی و شبیه‌سازی جنگ ترکیبی ارائه و در قالب یک برنامه کامپیوتری پیاده‌سازی شود. هدف از ارائه این چارچوب، نه ارائه یک نمونه بازی جنگ ترکیبی بلکه ارائه مدلی برای پیاده‌سازی نحوه تفکر جنگ ترکیبی در قالب بازی جنگ است. مسألهٔ محوری در این مقاله، توسعه روش‌های بازی جنگ از حالت معمولی (غیرترکیبی) به حالت جدید بازی جنگ ترکیبی است که برای این کار به دو مؤلفه عامل‌بنیان نمودن الگوسازی و شبیه‌سازی و نیز لایه‌لایه نمودن ابعاد موضوعی جنگ در قالب نبردهای اقتصادی، اجتماعی، سیاسی، سایبری و مانند آن توجه شده است. تحقق این مهم نگارنده را بر این داشت تا ابتدا به پیاده‌سازی قالبی متداول مبتنی بر شی‌گرایی در برنامه‌نویسی برای بازی جنگ غیرترکیبی (متعارف) پرداخته و سپس اقدام به توسعه همان قالب به قالب ارتقاء یافتهٔ بازی جنگ ترکیبی نماید.

در این بخش به ذکر مقدمه‌ای درخصوص جنگ ترکیبی پرداخته شده است. در بخش بعد مبانی نظری با رویکردی مبتنی بر علوم انسانی به‌عنوان نظرگاه مفهومی جنگ ترکیبی ارائه شده است. در بخش سوم، تنها مرجع موجود در زمینه شبیه‌سازی بازی جنگ ترکیبی در قالب پیشینه تحقیق بیان می‌شود. پیاده‌سازی بازی جنگ در دو قالب جنگ متعارف و جنگ ترکیبی در بخش چهارم بیان و مدل پیشنهادی در قالب برنامه کامپیوتری ارائه و توضیح داده خواهد شد. جمع‌بندی و ارائه پیشنهادات در بخش پنجم به عنوان فصل پایانی گنجانده شده است.

۲- مبانی نظری

جنگ ترکیبی نوعی از جنگ است که تاکتیک‌های نظامی متعارف را با سایر ابزارهای غیرنظامی مانند عملیات اطلاعاتی، حملات سایبری و فشار اقتصادی برای دستیابی به اهداف راهبردی ترکیب می‌کند.

جنگ ترکیبی ترجمه فارسی جنگ هیبریدی[2] است. کلمه هایبرید در ریشه لاتین از کلمه Hibrida ، به معنی فرزند خروس رام و گراز وحشی، گرفته شده که چنین تعبیر می‌شود که به حاصل ترکیب دو چیز ناهمگون یا از گونه‌های مختلف اطلاق می‌شود [۵]. در کنفرانس امنیتی مونیخ در سال ۲۰۱۵ [۶] ، جنگ هیبریدی را ترکیبی از چندین ابزار جنگی[3] متعارف[4] و غیر متعارف[5] تعریف کردند.

چنین بنظر می‌رسد که اصطلاح Warfare به معنای جنگ تمام عیار نیست و به نوعی مخاصمه یا تنازع تعبیر می‌شود که

[1] Framework
[2] Hybrid Warfare
[3] Tools of warfare
[4] Conventional
[5] Unconventional

شاید در ادبیات نظامی بتوان نامی محدودتر از «جنگ» نظیر «نبرد» را برای آن در نظر گرفت. بنابراین می‌توان اینگونه نتیجه گرفت که وقتی قیدی به جنگ زده شود، از War به Warfare یا به طور معادلِ فارسی از «جنگ» به «نبرد» تبدیل می‌شود. معمولاً قید یا در زمان، یا در مکان و یا در مفهوم است و بر همین اساس می‌توان نتیجه گرفت که نبرد می‌تواند از نظر زمانی با جنگ متفاوت باشد. برای مثال با تامّل در مدل جنگ/نبرد ترکیبی غرب علیه ایران می‌توان اینگونه استنباط نمود که غرب از این مدل برای پرهیز از حمله نظامی و رسیدن به برتری تدریجی یا برتری کم هزینه برعلیه ایران استفاده می‌کند، پس نبرد در مقایسه با جنگ دارای انبساط زمانی است یعنی در بازه زمانی وسیع‌تری رخ می‌دهد. حال این سوال مطرح است که تفاوت زمانی در نتیجه جنگ/نبرد ترکیبی یا هر نبرد زیرمجموعه آن به چه معنایی است؟ برای پاسخ به سوال بالا به این نکته باید توجه کرد که ترکیبی از نبردها در قالب یک کل واحد بنام جنگ قابل تداعی است. در این صورت بهتر است به Hybrid War همانطور که در فارسی به درستی ترجمه شده، واژه «جنگ ترکیبی» را اطلاق نمود که شامل جنگی با ترکیب از صحنه‌های نبرد مختلف نظیر نبرد الکترونیک[1]، نبرد سایبری[2] و مانند آن خواهد بود. بنابراین صحنه، صحنه جنگ است ولی در زمان وسعت‌یافته و دچار انبساط زمانی شده به نحویکه همان نتیجه جنگ در مدت زمان طولانی حاصل می‌شود ولی این به معنای انبساط زمانی هر نوعی از نبرد نیست بلکه برعکس، هر نبرد دارای قید زمانی به معنای محدودیت زمانی در بازه‌ای مشخص است یعنی هر نبرد به تناسب نیاز و حوزه کاربرد و میزان اثرگذاری می‌تواند در بازه‌ای از زمان فعال شده و پس از فعال شدن به خاتمه برسد. برای نمونه می‌توان به یک نبرد اجتماعی مثل موضوع بیداری اسلامی در سال ۲۰۱۱ در کشورهای عربی نظیر مصر اشاره کرد. قید دیگر در مکان است. از نظر مکانی هم یک نبرد می‌تواند شکل محدود شده جغرافیایی از یک جنگ باشد. برای مثال صحنه نبرد دریایی یا زمینی یا هوایی هرکدام یکی از صحنه‌های نبرد به تفکیک محدوده مکانی خاص است.

سون تزو (تحلیلگر نظامی چین) هزاران سال پیش بیان کرد که کارآمدترین راه برای مبارزه با دشمن، شکستن مقاومت او به طور غیرمستقیم و بدون جنگ واقعی است. جان‌مک‌کوئن، در مقاله خود با عنوان "جنگ‌های ترکیبی"، این جنگ را به عنوان «جنگ های طیفی با ابعاد فیزیکی و مفهومی[3]» توصیف کرده که اولی، مبارزه با یک دشمن مسلح و دومی، مبارزه‌ای گسترده‌تر برای کنترل و حمایت از جمعیت بومی منطقه جنگی، حمایت از جبهه‌های داخلی کشورهای مداخله‌گر و حمایت جامعه بین‌المللی است [۷]. مک‌کوئن جنگ ترکیبی را استفاده از ابزارهای گوناگون برای متقاعد کردن جمعیت داخلی یک کشور هدف می‌داند. ناتو جنگ ترکیبی را نوعی جنگ می‌داند که در آن از ابزارهای متعارف و غیر متعارف، تبلیغات، اطلاعات نادرست، نیابت‌ها و عملیات روانی در حالتی هدف‌دار استفاده می‌شود [۸]. ویلیام جی. نمت در رساله خود با عنوان «جنگ آینده و چچن: موردی برای جنگ ترکیبی» [۹]، اصطلاح جنگ ترکیبی را برای توصیف وجود تهدیدات ترکیبی بکار می‌برد، درحالیکه فرانک هافمن، پژوهشگر مرکز تهدیدات و فرصت‌های نوظهور در فرماندهی توسعه رزمی تفنگداران دریایی آمریکا، در مقاله "جنگ ترکیبی و چالش‌ها" استفاده از این اصطلاح را برای اشاره به درگیری‌های پس از جنگ سرد استفاده می‌کند [۱۰]. هافمن در رساله "درگیری در قرن بیست‌ویکم: ظهور جنگ‌های ترکیبی" معتقد است که ۱۱ سپتامبر ۲۰۰۱ مفهوم جنگ را به طرز چشمگیری تغییر داد. با این رویکرد جدید، راه‌های مرسوم برای تحلیل جنگ ناکافی به نظر می‌رسد. قدرت نظامی هنوز یک نیروی حیاتی است اما چوب جادویی برای کشورها نیست. برنامه‌ریزان باید طرز فکر خود را برای دفعات نسبی و تهدیدات مبارزه آینده اصلاح کنند. آنها همچنین باید نیازهای هزینه‌های دفاعی و نقش فناوری در راهبرد خود را مورد بازنگری قرار دهند [۱۱].

ایده غیرقابل پیش‌بینی بودن در مرکز طراحی جنگ ترکیبی قرار دارد. اگر اقدامات جنگ ترکیبی غیرقابل پیش‌بینی باشد، دشمن یا هدف قادر به جهت‌گیری و تصمیم‌گیری صحیح نخواهد بود. انقلاب‌های رنگی مدرن و جنگ‌های غیرمتعارف از این چارچوب غیرقابل پیش‌بینی استفاده می‌کنند و هیچ طرح یا برنامه‌ای از اقدامات قابل پیش‌بینی را ایجاد نمی‌کنند. این حالت غیرقابل پیش‌بینی، بیانگر غیرخطی بودن سامانه است که می‌تواند ناشی از هرج و مرج یا آشوب ایجاد شده در این سامانه باشد که با نظریه آشوب همخوانی دارد. استیون‌من در کتاب "تئوری آشوب و هنر راهبرد"، معتقد است که جنگ دارای متغیرهای غیرخطی دیگری مانند محیط زیست، توسعه فناوری، فشارهای اجتماعی، مذهب، ایدئولوژی، بازیگران غیردولتی و شرکت‌های جهانی است و درصورتیکه تنها متغیرهای ساده نظامی، اقتصادی و سیاسی را در نظر بگیریم، جنگ حالت متعارف پیدا خواهد کرد [۱۲]. این ماهیت غیرخطی، آیندهٔ سامانه را غیرقابل پیش‌بینی می‌کند و شناسایی، حمله یا از بین بردن آن توسط سامانه تقریباً غیرممکن می‌شود [۱۳].

دو شاخصه کلیدی در جنگ ترکیبی از طرف حریف نسبت به مدافع مدنظر قرار می‌گیرد: اول «سطح آستانه»[4] مدافع در مقابل

[1] Electronic warfare
[2] Cyber Warfare
[3] Conceptual
[4] Threshold

جنگ ترکیبی تاحدی که مانع از شکل‌گیری جنگ مسلحانه شود و دوم «تمایل»[1] مدافع یا جامعه بین‌الملل برای عبور از سطح آستانه جنگ ترکیبی و ورود به جنگ نظامی که بایستی این موضوع از طرف مهاجم مدیریت شود [۱۴]. به بیان دیگر، مدیریت دو موضوع بالا از طرف مهاجمِ جنگ ترکیبی برای نگه داشتن مدافع در حداقل سطح ممکنِ جنگ نظامی ضروری است. درخصوص مورد دوم، تمایل جامعه بین‌المللی برای حمایت از مدافع، دارای اهمیت زیادی است. هنگامی که تمایل جامعه بین‌المللی از آستانه فراتر رفت، جنگ ترکیبی از یک جهت پایان می‌یابد، یعنی یا مهاجم عقب‌نشینی می‌کند یا باید با یک درگیری مسلحانه با مدافعِ مورد حمایتِ جامعه بین‌المللی مواجه شود. بنابراین، برای مهاجم بهترین کار این است که آستانه را تا حد ممکن افزایش دهد، بدون اینکه کنترل تمایل را از دست دهد.

اصطلاح جنگ ترکیبی بیشتر بعد از جنگ سال ۲۰۰۶ رژیم صهیونیستی علیه لبنان توسط تحلیلگران معرفی شد اگرچه در ادبیات غربی تلاش می‌شود نقطه شروع آن را به اشغال کریمه توسط روسیه در سال ۲۰۱۴ نسبت دهند. در واقع با اتفاق سال ۲۰۱۴ این موضوع بیشتر و فشرده‌تر در سرخط موضوعات تحقیقاتی و مطالعاتی در جهان قرار گرفت [۱۵].

برای الگوسازی و شبیه‌سازی[2] جنگ ترکیبی نیاز است تا اولاً مدلی شامل چند لایه برای نشان دادن ابعاد مختلف جنگ ترکیبی ارائه شود و ثانیاً این مدل بهره‌مند از قابلیت تصادفی برای آرایش صحنه جنگ به منظور نشان دادن ایدهٔ غیرقابل پیش‌بینی بودن این نوع از جنگ باشد. در جنگ ترکیبی، دشمن از تمام ابزارهای موجود، برای سوء استفاده از آسیب‌پذیری‌های طرف مقابل و بی‌ثبات کردن آن استفاده می‌کند. ایجاد ابهام، انکار و از کار انداختن طرف مقابل در تصمیم‌گیری در هر اقدامی، هدف است و بنابراین دشمن تلاش می‌کند بدون درگیری مسلحانه و حتی بدون تغییر عمده در روابط دیپلماتیک و اقتصادی به اهداف خود دست یابد. مزایای الگوسازی و شبیه‌سازی جنگ ترکیبی عبارتند از:

- تصویرسازی صحنه نبرد بصورت یکپارچه و ایجاد حس بصری در مدیریت صحنه؛
- پشتیبانی از تصمیم‌گیرندگان کلیدی در درک پویایی جنگ ترکیبی؛
- حمایت از شناسایی کمبودها و نقاط ضعف در سازماندهی فعلی، طراحی و ارائه راه‌حل‌های فناورانه، تجهیزات و دکترین‌های نبرد؛
- پشتیبانی از توسعه یک رویکرد موثر برای واکنش به جنگ ترکیبی در سناریوهای پیچیده؛
- ارزیابی کارایی اقدامات انجام شده در شرایط بحرانی بطوریکه توانایی درک موقعیت و خطرات مربوط به صحنه‌های مختلف را داشته باشد و همچنین روند تکامل سناریو را افزایش دهد؛

۳-پیشینه تحقیق

با توجه به اینکه طرح مفهومی تحت عنوان جنگ ترکیبی بسیار نوپدید است، اطلاعات زیادی درخصوص الگوسازی و شبیه‌سازی آن یا انجام نگرفته و یا در دسترس نیست. با این وجود، در سال‌های اخیر نرم‌افزاری توسط ناتو برای شبیه‌سازی جنگ ترکیبی با نام T-REX توسعه داده شده است. هدف این نرم‌افزار همچنانکه از نام آن پیداست، شبیه‌سازی شبکه تهدید برای ایجاد ظرفیت و توانمندی واکنش به اقدامات جنگ ترکیبی است [۱۶].

در نرم‌افزار T-REX امکان الگوسازی و شبیه‌سازی با قابلیت تعامل و بازی جدّی دیده شده که به بازتولید صحنه جنگ ترکیبی و ترکیب شدن با عناصر دیگر برای ارزیابی تأثیر این اقدامات اختصاصی می‌پردازد. این نرم‌افزار تحت برنامه NATO MSG ET-43 Hybrid Warfare توسط آگوستینو بروزونه[3] در تیم شبیه‌سازی دانشگاه جنوا[4] ایتالیا و تحت پروژه SIM4Future انجام گرفته است. در این نرم‌افزار از روش شبیه‌سازی رویداد گسسته تصادفی[5] استفاده می‌شود. یک شبیه سازی فرضی در این نرم‌افزار می‌تواند شامل لایه‌های متعددی از جمله جمعیت (به عنوان مثال، افراد و/یا خانواده‌ها) و همچنین گروه‌های مرتبط (به عنوان مثال، بخش‌های صنعتی، گروه‌های مذهبی، طبقات اجتماعی) باشد. این عناصر در شبکه‌های اجتماعی ساختاریافته و با روابط متقابلی که توسط متغیرهای فازی از نظر نگرش و شدت بیان می‌شوند، قابل تنظیم است. T-REX همچنین شامل لایه‌های دیگری است که با شبکه‌های اجتماعی تعامل دارند، به ویژه لایه سایبری و هویت‌ها و واحدها[6] (نظیر زیرساخت‌های نظامی و غیرنظامی) که واحدها و دارایی‌های نظامی را بازتعریف می‌کنند. برای مثال، اهداف پهپادهای نظامی می‌توانند زیرساخت‌های حیاتی‌ای باشند که بر تکامل سناریوی جنگ ترکیبی و ایجاد رفتارهای جمعیتی تأثیر می‌گذارند [۱۷]. در شکل (۱) ساختار چندلایه این نرم‌افزار برای شبیه‌سازی جنگ ترکیبی دیده می‌شود [۱۸]:

[1] Willingness
[2] Modeling and Simulation (M&S)
[3] Agostino Bruzzone
[4] Genoa University
[5] Stochastic discrete event simulation
[6] Entity & Units (E&U)

شکل (۱): ساختار چندلایه شبیه‌سازی جنگ ترکیبی در نرم‌افزار T-REX [۱۸]

در شکل (۱) ، نیرو یا عوامل مهاجم به رنگ قرمز و عوامل مهارکننده یا مدافع به رنگ آبی نشان داده شده‌اند. همانطور که در شکل دیده می‌شود، بستر سایبری به‌عنوان بستر جامع (به دلیل ماهیت مجازی) برای تعامل با کلیه لایه‌ها در نظر گرفته شده است.

۴- الگوسازی چارچوب بازی جنگ

در این بخش به بیان مشخصات کلی چارچوب لازم برای الگوسازی و شبیه‌سازی بازی جنگ خواهیم پرداخت و در ادامه چارچوب مدنظر را در دو قالب کلی یکی برای جنگ متعارف و دیگری برای جنگ غیرمتعارف (جنگ ترکیبی) پیاده‌سازی خواهیم کرد. قالب دوم، شامل ارتقاء چارچوب معرفی شده برای جنگ متعارف (قالب اول) به حالت جنگ ترکیبی از طریق اضافه کردن دو لایه اقتصادی و سایبری به قالب اول می‌باشد.

۴-۱- مشخصات کلی

برای الگوسازی و شبیه‌سازی جنگ ترکیبی با استفاده از زبان برنامه‌نویسیِ پیش‌فرضِ پایتون توجه به نکات زیر ضروری است:

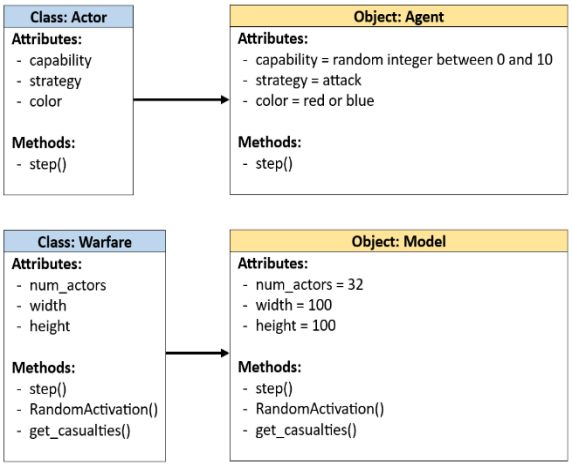
- تعریف مسأله: قبل از شروع به کدنویسی، مهم است که مسأله و اهداف شبیه‌سازی را تعریف نمود. این شامل شناسایی بازیگران درگیر در مناقشه، توانایی‌ها و راهبردهای آنها و نتایج احتمالی است.
- انتخاب چارچوب شبیه‌سازی: چندین چارچوب شبیه‌سازی از جمله Mesa، SimPy و PyCX در پایتون وجود دارد. هر چارچوب دارای نقاط قوت و ضعف خاص خود است، بنابراین انتخاب صحیح بهترین چارچوب متناسب با نیازهای مدّ نظر شبیه‌سازی از اهمیت بالایی برخوردار است [۱۹].
- پیاده‌سازی مدل: پس از انتخاب چارچوب شبیه‌سازی، می‌توان پیاده‌سازی مدل را شروع کرد. اینکار شامل تعریف بازیگران، توانایی‌ها و راهبردهای آنها و قوانین بازی است. همچنین ممکن است نیاز به ترکیب عواملی مانند زمین، آب و هوا و پشتیبانی در برنامه باشد.
- اجرای شبیه‌سازی: پس از پیاده‌سازی مدل، می‌توان شبیه‌سازی را اجرا و داده‌ها را در مورد نتایج جمع‌آوری نمود. ممکن است برای محاسبه تصادفی و عدم قطعیت، نیاز به اجرای تکرار شبیه‌سازی در چندین مرحله باشد.
- تجزیه و تحلیل نتایج: پس از اجرای شبیه‌سازی، می‌توان نتایج را تجزیه و تحلیل و در مورد نتایج احتمالی جنگ، نتیجه‌گیری نمود. این امر مستلزم شناسایی نقاط قوت و ضعف هر بازیگر و ارزیابی اثربخشی راهبردهای مختلف آنها است.
- ورودی و خروجی: این امر به جزئیات خاص شبیه‌سازی بستگی دارد. ورودی معمولاً شامل مولفه‌های شبیه‌سازی، مانند قابلیت‌ها و راهبردهای بازیگران می‌شود، در حالی که خروجی شامل نتایج شبیه‌سازی، مانند تعداد تلفات، قلمرو کنترل‌شده توسط هر بازیگر و نتیجهٔ کلی درگیری است.

چارچوب پیشنهادی ما برای الگوسازی و شبیه‌سازی، برخلاف نرم‌افزار T-REX که به روش رویداد تصادفی گسسته انجام گرفته بود در قالب روش مبتنی بر عامل[1] انجام شده است. روش‌های شبیه‌سازی مبتنی بر عامل دارای مزیت‌های عمده زیر نسبت به روش‌های رویداد محور هستند [۲۰]:

- روش عامل بنیان در پیاده‌سازی کلان زیرساختارها از جزءترین زیرسامانه شروع و به سامانهٔ کلان می‌رسد ولی در روش‌های رویدادمحور اینکار از بالا به پایین است.
- در روش عامل بنیان، هویت‌ها مستقل بوده و امکان توسعه مدل یا پیشگیری از غافلگیری وجود دارد ولی در روش رویدادمحور، رویدادها متکی بر مشاهده خارجی هستند.
- در روش عامل‌بنیان، عملکرد برنامه‌ریزی شده عوامل را شاهدیم درحالیکه در روش رویدادمحور، شاهد پاسخ برنامه‌ریزی شده به رویدادهای گسسته خواهیم بود. بنابراین، در روش عامل بنیان، رویدادها قابل استخراج خواهند بود ولی در روش رویدادمحور، رفتار درونی ناشناخته است.
- در روش عامل‌بنیان، سادگی در قوانین الگوسازی وجود دارد ولی در روش رویدادمحور در الگوسازی ورودی‌ها، حالت‌ها و خروجی‌ها شاهد سادگی هستیم.

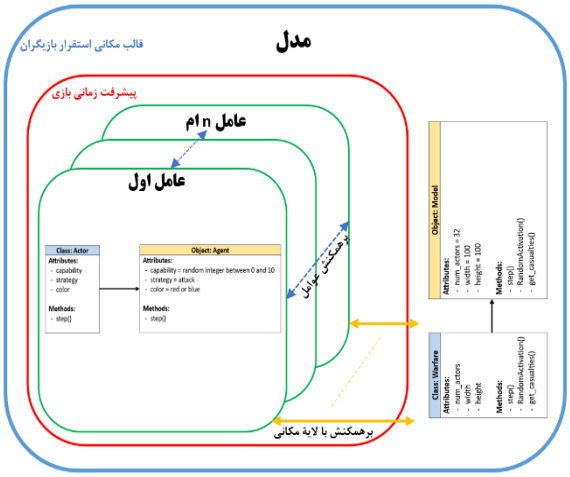
چارچوب پیشنهادی برای الگوسازی بازی جنگ متعارف، در قالب طرحِ کلاس[2] شکل (۲) بصورت شی‌گرا نشان داده شده است:

[1] Agent-based model
[2] Class Diagram



شکل (۲): طرحِ کلاس پیشنهادی برای الگوسازی بازی جنگ متعارف

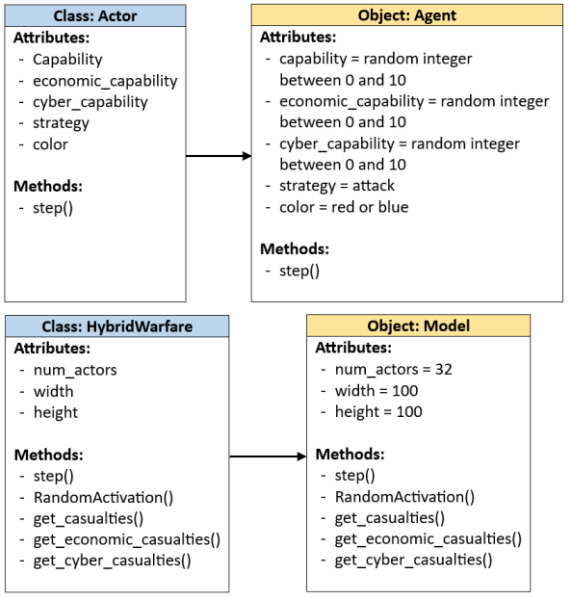
در ادامه چارچوب پیشنهادی عامل‌بنیان به منظور الگوسازی بازی جنگ در شکل (۳) بصورت ساختار مفهومی ارائه شده است.



شکل (۳): ساختار مفهومی چارچوب الگوسازی بازی جنگ متعارف

همانطور که در شکل (۳) دیده می‌شود، هر عامل دارای دو سطح تعاملات است. سطح اول شامل لایهٔ بیرونی برای تعامل با مدل و سایر عامل‌ها است و سطح دوم، لایهٔ درونی برای حل فرآیندهای داخلی است. همانطور که قبلاً هم بیان شد، هر عامل از طریق کلاس Actor مطابق شکل (۲) الگوسازی خواهند شد. این کلاس به همراه شیءِ مربوطه یعنی Agent در شکل (۳) به‌عنوان فرآیند درونی هر عامل و نیز در تعامل با شیءِ Model از کلاس Warfare به‌عنوان فرآیند بیرونی نشان داده شده است.

ارتقای چارچوب ارائه شده در شکل (۳) از جنگ متعارف به جنگ ترکیبی مستلزم اضافه کردن لایه‌های جدید نبرد (در اینجا برای نمونه دو لایهٔ اقتصادی و سایبری) به سطح فرآیندهای درونی هر عامل و نیز اصلاح مدل در قالب کلاس جدیدی با عنوان HybridWarfare است. طرحِ کلاس برای حالت جنگ ترکیبی بصورت شکل (۴) خواهد بود.



شکل (۴): طرحِ کلاس پیشنهادی برای الگوسازی بازی جنگ ترکیبی

بدیهی است که با ارتقای شکل (۲) به کلاس‌های جدید در شکل (۴)، چارچوب ارائه شده در شکل (۳) براحتی با جایگزینی کلاس‌های اخیر به‌عنوان چارچوب نهاییِ الگوسازی بازی جنگ ترکیبی تعریف خواهد شد.

با توجه به توضیحات ارائه شده، در بخش بعدی به پیاده‌سازی ساختار مفهومی چارچوب ارائه شده برای جنگ متعارف خواهیم پرداخت.

۴-۲- پیاده‌سازی چارچوب الگوسازی جنگ متعارف

برای شروع، یک مدل سادهٔ مبتنی بر عامل با استفاده از چارچوب Mesa در پایتون برای صحنه نبرد متعارف تشکیل می‌دهیم [۲۱]. برای اینکار ابتدا کتابخانه‌های اساسی ذیل را فراخوانی می‌کنیم:

```
import pygame
import matplotlib.pyplot as plt
from mesa import Agent, Model
from mesa.space import MultiGrid
from mesa.time import RandomActivation
from tqdm import tqdm
import time
```

همانطور که در خطوط برنامه فوق دیده می‌شود، کتابخانه بازی‌سازِ pygame به منظور نمایش تجسمی بازی جنگ و در ادامه کتابخانه Mesa برای پیاده‌سازی مدل عامل‌بنیان فراخوانی می‌شود. همچنین از matplotlib برای نمایش نمودارها و ترسیم تعداد تلفات (بازیگران شکست خورده) در طول زمان استفاده می‌شود. کتابخانه tqdm نیز برای نمایش نوار پیشرفت در طول شبیه‌سازی استفاده شده است. همچنین برای سهولت در استفاده از رنگ‌بندی به تعریف رنگ برای نمایش بازیگران در قالب زیر اقدام شده است:

```
# Define colors for the actors
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
```

در ادامه چارچوب کلی برنامه را در دو کلاس بازیگر و نبرد با مدل برنامه‌نویسی شی‌گرا و به ترتیب با اشیاءِ عامل[1] و مدل مشخص می‌کنیم:

```
class Actor(Agent):

class Warfare(Model):
```

این یک مثال بسیار اولیه است که تعدادی عامل (در این مورد بازیگران[2]) ایجاد می‌کند، آنها را روی یک شبکه[3] قرار می‌دهد و تابع گام[4] آنها را در هر تکرار[5] شبیه‌سازی اجرا می‌کند. تابع گام همان جایی است که می‌توان راهبرد بازیگر را بر اساس توانایی‌ها و قوانین بازی اجرا نمود. اولین کلاس را بصورت کلاس بازیگر یا Actor تعریف می‌کنیم:

```
class Actor(Agent):
    def __init__(self, unique_id, model):

    def step(self):
```

کلاس Actor نشان‌دهنده یک بازیگر فردی در شبیه‌سازی مانند یک واحد نظامی، یک ایالت یا یک کشور است.

هر بازیگر دارای ویژگی‌های مختلفی از جمله شناسه منحصر به فرد، قابلیت انتخاب تصادفی از میان بازیگران، نوع راهبرد بازیگر و درنهایت رنگ تخصیص داده شده به آن برای نمایش در صفحه بازی جنگ است. در اینجا ذکر این نکته ضروری است که انتخاب بازیگر اگرچه بصورت تصادفی و با استفاده از قابلیت متد random نوشته شده است ولی امکان محدود کردن انتخاب‌های ممکن به شرایط ورودی از طرف برنامه‌نویس وجود داشته و لذا میزان تصادفی بودن آن قابل کنترل خواهد بود. ویژگی دیگری تحت عنوان capability نیز برای کلاس بازیگر در نظر گرفته شده که توانایی نظامی بازیگر را نشان می‌دهد و یک مقدار تصادفی مابین ۱ و ۱۰ در هنگام مقداردهی اولیه به آن تخصیص می‌یابد:

```
class Actor(Agent):
    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)
        # Assign random capabilities to each actor
        self.capability = self.random.randint(1, 10)
        # Set strategy to "attack" for all actors
        self.strategy = "attack"
        self.color = None
```

ویژگی strategy نشان دهنده راهبرد فعلی بازیگر است که برای همه بازیگران روی حالت attack یا «حمله» تنظیم شده است ولی این امکان وجود دارد که راهبرد را بر اساس نیاز مجریان بصورت چندگانه تعریف و امکان انتخاب هر بازیگر با یک راهبرد متفاوت را ایجاد نمود. متد step از کلاس Actor راهبرد بازیگر را پیاده‌سازی می‌کند:

```
def step(self):
    # Implement actor's strategy
    if self.strategy == "attack":
```

اگر راهبرد "حمله" باشد، بازیگر انتخابی، یک بازیگر همسایه تصادفی را به عنوان هدف انتخاب می‌کند و بر اساس توانایی آن آسیبی را به او وارد می‌کند:

```
if self.strategy == "attack":
    # Find neighboring cells with other actors
    neighbors = self.model.grid.get_neighbors(self.pos,...
    moore=True, include center=False)
```

سعی شده تا کلیه خطوط کد در درون برنامه توضیح داده شود. در این کلاس ابتدا بازیگران بصورت تصادفی انتخاب و سپس برای پیاده‌سازی راهبرد بازیگران، پس از تعریف گام، اقدام به تعریف راهبرد حمله می‌نماییم. قابلیت هر بازیگر با یک عدد تصادفی انتخاب می‌شود ولی راهبرد به صورت پیش‌فرض در حالت حمله تعریف شده که این حمله برعلیه یک هدف تصادفی از کلاس بازیگرِ مدافع که در همسایگیِ مهاجم قرار دارد، فرض می‌شود. اگر در این حمله، قابلیت هدف به صفر یا کمتر از آن نزول کند، هدف از بازی حذف می‌شود:

```
# Check if the target is defeated
if target.capability <= 0:
    self.model.grid.remove_agent(target)
    self.model.schedule.remove(target)
```

در کلاس بازیگر، بازیگران اقدام به یافتن همسایگان خود در میان بازیگران می‌نمایند و بصورت تصادفی به اهداف حمله می‌کنند. با توجه به اینکه این برنامه یک چارچوب کلی را ارائه می‌دهد از بیان شرایط خاص اهداف، بازیگران و صحنهٔ بازی پرهیز شده است. بدیهی است با توجه به مسأله کاربردی هریک از این موارد قابلیت اضافه شدن به کلاس فوق را دارند. در ادامه کلاس نبرد یا Warfare پیاده‌سازی خواهد شد:

```
class Warfare(Model):
    def __init__(self, num_actors, width, height):

    def step(self):

    def get_casualties(self):
```

در این کلاس ابتدا طول و عرض صحنه نمایشِ نبرد مشخص و براساس تعداد بازیگران انتخابی، تخصیص رنگ آبی و قرمز به ترتیب برای مدافع و مهاجم انجام و بازیگران در صفحه بصورت تصادفی با مربع‌های کوچک توزیع می‌شوند. سپس در هر گامِ اجرای شبیه‌سازی، شرایط پیروزی یا شکست رقم خورده و تعداد بازیگران کشته‌شده[6]، پیروز و در حال بازی مشخص می‌شود. کلاس Warfare مدل شبیه‌سازی را نشان می‌دهد. این مدل شامل ویژگی‌هایی مانند num_actors، grid، schedule

[1] Agents
[2] Actors
[3] Grid
[4] Step function
[5] Iteration
[6] Casualties

و casualties است. ویژگی num_actors تعداد بازیگران در شبیه‌سازی را تعیین می‌کند. ویژگی grid فضای چند شبکه‌ای را نشان می‌دهد که بازیگران در آن قرار می‌گیرند و با هم تعامل دارند. صفت schedule ترتیب فعال شدن بازیگران را مدیریت می‌کند. ویژگی casualties فهرستی است که تعداد بازیگران (تلفات) را در هر تکرارِ شبیه‌سازی پیگیری می‌کند. متد init از کلاس Warfare مدل شبیه‌سازی را مقداردهی اولیه می‌کند. تعداد مشخصی از بازیگران را ایجاد می‌کند، رنگ‌ها را بر اساس id منحصر به فرد آنها اختصاص می‌دهد، آنها را به طور تصادفی در شبکه قرار می‌دهد و ویژگی در حال اجرا را روی True تنظیم می‌کند.

```
def __init__(self, num_actors, width, height):
    self.num_actors = num_actors
    self.grid = MultiGrid(width, height, torus=True)
    self.schedule = RandomActivation(self)
    self.casualties = []

    # Create actors
    for i in range(self.num_actors):
        actor = Actor(i, self)
        #Assign colors to actors
        actor.color = RED if i % 2 == 0 else BLUE
        self.schedule.add(actor)
        x = self.random.randrange(self.grid.width)
        y = self.random.randrange(self.grid.height)
        self.grid.place_agent(actor, (x, y))

    self.running = True
    self.winner = None
```

متد step از کلاس Warfare شبیه‌سازی را یک مرحله پیش می‌برد:

```
def step(self):
  self.schedule.step()
  # Check for victory or defeat conditions
  num_alive = sum([1 for agent in ...
  self.schedule.agents if isinstance(agent, Actor)])
  if num_alive == 0:
  elif num_alive == 1:
  self.casualties.append(self.num_actors - len(self.schedule.agents))
```

حلقه شرطی زیر، بدنه اصلی برنامه را تشکیل می‌دهد که در ابتدای آن، بازی راه‌اندازی اولیه شده و صفحه بازی جنگ با عنوان Warfare Simulation پدیدار می‌شود. در ادامه مدل بازی بر اساس تعداد فرضی ۳۲ بازیگر در شبکه‌ای به طول و عرض ۱۰۰ در ۱۰۰ پیکسل ساخته می‌شود. این اعداد بنابه نیاز قابل تغییر هستند. پس از انجام مراحل فوق، حلقه اصلی برنامه تا رسیدن به حالت پایدار که حذف کامل بازیگران مدافع یا مهاجم ادامه خواهد یافت. با تعریف یک حلقه، مدل با شروطی برای برنده یا بازنده شدن شروع به اجرا خواهد نمود و در نهایت خروجی در قالب گزارش ارائه می‌شود.

```
if __name__ == "__main__":
    # Initialize Pygame
```

همه بازیگران راهبرد یکسانی دارند، یعنی حمله به بازیگران دیگر. شبیه‌سازی در مراحل زمانی گسسته پیش می‌رود و در طول هر مرحله زمانی، هر بازیگر به‌طور تصادفی یک بازیگر همسایه را برای حمله انتخاب می‌کند و بر اساس قابلیت‌های خود، آسیبی را به‌صورت تصادفی به آن وارد می‌کند. اگر توانایی یک بازیگر به صفر یا کمتر برسد، از شبکه حذف می‌شود و شبیه‌سازی تا زمانی ادامه می‌یابد که تنها یک بازیگر باقی بماند یا همه بازیگران شکست بخورند.

در پایان شبیه‌سازی، کد منجر به نمایش برنده شده و تعداد تلفات را نشان می‌دهد. همچنین گزارشی متنی از شبیه‌سازی حاوی اطلاعات مشابه تولید می‌کند که در ذیل نشان داده شده است:

```
# Print the outcome of the simulation
if model.winner:
else:

print("Casualties:", model.get_casualties())

# Plot the casualties over time
plt.plot(model.casualties)
plt.title("Casualties over time")
plt.xlabel("Iteration")
plt.ylabel("Casualties")
plt.show()

# Create a report on the simulation
report = f"""Simulation Outcome:

print(report)

# Quit Pygame
pygame.quit()
```

شکل (۵) نمونه‌ای از خروجی در حال اجرای برنامه را در تکرار شماره ۱۶۰۰۸ در ثانیه ۲۲ ام از اجرای برنامه را که شامل بازیگران تصادفیِ مهاجم و مدافع است، نشان می‌دهد:

شکل (۵): نمونه‌ای از صحنه اجرای برنامه بازی جنگ متعارف شامل بازیگران مهاجم (قرمز) و مدافع (آبی)

۴-۳- پیاده‌سازی چارچوب الگوسازی بازی جنگ ترکیبی

در این قسمت با اضافه کردن دو لایه نبرد اقتصادی و نبرد سایبری به صحنه نبرد متعارف، صحنه جنگ ترکیبی به صورت نمونه در قالب شی‌گرایی الگوسازی و نهایتاً با ابزارهای برنامه‌نویسی شبیه‌سازی خواهد شد. نظیر بخش قبل، برنامه شامل

فراخوانی کتابخانه‌های اصلی، تعریف رنگ‌بندی، تعریف دو کلاس اصلی مبتنی بر برنامه‌نویسی شی‌گرا و حلقه اصلی برنامه است با این تفاوت که در پیاده‌سازی این برنامه، کلاس دوم از warfare به hybrid warfare تغییر نام داده می‌شود.

```
import pygame
import matplotlib.pyplot as plt
from mesa import Agent, Model
from mesa.space import MultiGrid
from mesa.time import RandomActivation
from tqdm import tqdm
import time

# Define colors for the actors
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)

class Actor(Agent):

class HybridWarfare(Model):

if __name__ == "__main__":
```

در تعریف کلاس بازیگر، برخلاف قبل، دو قابلیت نبرد اقتصادی و نبرد سایبری نیز به صورت تصادفی به هریک از بازیگران اختصاص داده شده است. ویژگی‌های Economic_capability و cyber_capability به‌ترتیب توانایی‌های اقتصادی و سایبری بازیگر را نشان می‌دهند و همچنین مقادیر تصادفی بین ۱ و ۱۰ در طول مقداردهی اولیه به آنها اختصاص داده می‌شود. مجدداً تاکید می‌شود که با توجه به قصد مقاله در تعریف چارچوب کلی بازی جنگ ترکیبی، از ذکر شرایط و قابلیت‌های نبرد اقتصادی و سایبری صرفنظر شده است. بدیهی است در حالت ایده‌آل می‌توان بجای انتخاب بازیگران بصورت تصادفی، اقدام به تعریف مشخصات هر نبرد و تخصیص آن به بازیگران بر اساس اطلاعات تجربی حاصل از نبردهای مشابه نمود.

```
class Actor(Agent):
    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)
        self.capability = self.random.randint(1, 10)
        self.strategy = "attack" # Set strategy to "attack"
        self.color = None
        self.economic_capability = self.random.randint(1, 10)
        self.cyber_capability = self.random.randint(1, 10)

    def step(self):
```

در کلاس فوق، براساس تعریف دو لایه نبرد جدید، حلقه‌های شرطی ناشی از آسیب‌های اقتصادی و سایبری نیز به برنامه اضافه شده است بطوریکه در متد step از کلاس Actor ، اگر توانایی هدف به صفر یا کمتر از آن برسد، از شبیه‌سازی حذف می‌شود و قابلیت‌های اقتصادی و سایبری هدف از تعداد تلفات مربوطه کم خواهد شد. بصورت مشابه، کلاس HybridWarfare نیز بصورت زیر نوشته می‌شود:

```
class HybridWarfare(Model):
    def __init__(self, num_actors, width, height):

    def step(self):

    def get_casualties(self):

    def get_economic_casualties(self):

    def get_cyber_casualties(self):
```

همانطوریکه در برنامه فوق دیده می‌شود، ویژگی‌های economic_casualties و cyber _casualties به کلاس HybridWarfare و نیز ویژگی winner به متد step اضافه شده است. ویژگی‌های تلفات اقتصادی و تلفات سایبری مجموع قابلیت‌های اقتصادی و سایبری از دست رفته در طول شبیه‌سازی را دنبال می‌کنند. ویژگی برنده نیز نتیجه شبیه‌سازی را نشان می‌دهد و در ابتدا بر روی None تنظیم می‌شود. تغییرات ایجاد شده در کد اصلی شامل اضافه شدن قابلیت‌های اقتصادی و سایبری برای هر بازیگر، اجرای تلفات اقتصادی و سایبری و تولید خروجی‌های جدید و ارائه گزارش در پایان می‌باشد.

در نهایت حلقه اصلی برنامه نظیر بخش قبل نوشته می‌شود با این تفاوت که علاوه بر نقش دو لایه نبرد اقتصادی و سایبری در بروزرسانی برنامه بعد از هر بار تکرار، دولایه هر یک با صفحه مجزا و گزارش‌دهی اضافی علاوه بر آنچه در جنگ متعارف مشاهده شد، در خروجی ظاهر خواهند شد:

```
if __name__ == "__main__":
    # Initialize Pygame
    pygame.init()
    screen = pygame.display.set_mode((500, 500))
    pygame.display.set_caption("Hybrid Warfare Simulation")

    # Create model
    model = HybridWarfare(num_actors=32, width=100, height=100)

    # Main simulation loop
    with tqdm(total=100) as pbar:

            # Optional delay to slow down the simulation
            # time.sleep(0.1)

    # Print the outcomes of the simulation
    print("Overall simulation results:")
    if model.winner:
    else:

    print("Overall casualties:", model.get_casualties())
    print("Economic casualties:", model.get_economic_casualties())
    print("Cyber casualties:", model.get_cyber_casualties())
```

در چارچوب اخیر هر بازیگر دارای یک قابلیت تصادفی برای موضوع اقتصادی و سایبری است و زمانی که بازیگری شکست می‌خورد، توانایی‌های اقتصادی و سایبری بازیگرِ شکست‌خورده به‌ترتیب از تعداد تلفات اقتصادی و سایبری جداگانه کسر می‌شود. پس از تکمیل شبیه‌سازی، برنامهٔ در حال اجرا، تلفات کلی و همچنین تلفات اقتصادی و سایبری را به طور جداگانه منتشر می‌کند. این برنامه، سه خروجی جدید نیز تولید می‌کند: یکی برای تلفات کلی در طول زمان و دو مورد دیگر برای تلفات اقتصادی و سایبری در طول زمان. در نهایت، گزارشی متنی در مورد شبیه‌سازی با کد برنامه کامل‌تری تولید خواهد شد که شامل همان اطلاعات خروجی چاپ شده است. به طور کلی، این مواردِ اضافه نسبت به بخش قبلی، نمای دقیق‌تری از شبیه‌سازی

و نتایج آن ارائه می‌دهند و امکان تجزیه و تحلیل و درک بیشتر عوامل مختلف را فراهم می‌کنند.

۵- نتیجه‌گیری و کارهای آینده

با طرح موضوع جنگ ترکیبی، به تدریج شبیه‌سازی بازی جنگ به شکل مرسوم، که در آن محیط بازی بصورت جنگ تمام‌عیار نظامی دو یا چند طرفه تعریف می‌شود، بایستی مورد بازنگری قرار گیرد. جنگ ترکیبی اغلب بر بی‌ثبات کردن زنجیره‌های حکمرانی یا فرماندهی و پیچیده کردن تصمیم‌گیری برای تصمیم‌گیرندگان در سطوح مختلف تمرکز می‌کند. این اقدامات به ویژه در برابر سازمان‌هایی که به دلیل ماهیت دموکراتیک یا چند ملیتی خود در فرآیند تصمیم‌گیری کُند هستند، مؤثر است.
در این مقاله سعی شد تا برای اولین بار چارچوبی برای الگوسازی و شبیه‌سازی بازی جنگ ترکیبی ارائه شود. به‌منظور نشان‌دادن سِیر گذار از بازی جنگ معمول به بازی جنگ ترکیبی، ابتدا قالبی برای حالت معمول ارائه و سپس با اضافه کردن لایه‌های مختلف، قالب ارتقاء یافته برای حالت ترکیبی پیاده‌سازی شد. در این مدل از روش عامل‌بنیان مبتنی بر عاملِ جنگ اعم از مدافع و مهاجم بهره گرفته شد. بدلیل پیچیدگی و حجم کار تنها به وارد کردن دو لایه اقتصادی و سایبری در مدل بسنده شد تا صرفاً نشان‌دهنده نحوه اضافه شدن لایه‌های درگیر در جنگ ترکیبی باشد. بدیهی است در شرایط کاری متناسب با نیاز بایستی لایه-های مختلف به همین شیوه به قالب اضافه شوند. نکته دیگر اینکه بدلیل عدم دسترسی به اطلاعات نظامی و نیز با توجه به تفاوت شرایط در کاربردها و صحنه‌های مختلف، از بیان راهبرد عاملان و لایه‌های نبرد، صرفنظر گردید ولی راهبرد بصورت یک ویژگی در چارچوب مبتنی بر شی‌گرایی در ذیل کلاس بازیگر با پیش‌فرض راهبرد حمله و با قابلیت وارد کردن سایر راهبردهای مدنظر کاربران نهایی گنجانده شده است.

در پایان پیشنهاد می‌گردد در محیط بازی جنگ ترکیبی ارائه شده، برای نمونه از ظرفیت‌های شبیه‌سازی نظیر بندر، پایانه نفت، مجموعه مخازن، انواع نیروگاه، واحدها و سامانه‌های امنیتی، افراد، دولت‌های همسو و متخاصم، شبکه‌های تهدید، انواع رسانه-ها، شیوه‌های ارتباطات، دارایی‌های سنتی و همچنین سامانه‌های مستقل مختلف و سایر زیرساخت‌های حیاتی نظیر ترمینال‌ها، متروها، فرودگاه‌ها و پایانه‌های نفتی که توسط سامانه‌های مختلف حفاظت از شبکه در برابر تهدید، محافظت می‌شوند، استفاده شود. همچنین لایه سایبری شامل رایانه‌ها، لپ‌تاپ‌ها و اینترنت اشیاء متحرک[1] و نیز فایروال‌ها و رویه‌های متداول در شبکه، متناسب با نیاز کاربران عملیاتی به نرم‌افزار مستخرج از این چارچوب افزوده گردد. شبکه عاملین تهدید نیز می‌تواند به انواع عوامل تروریستی توسعه یابد که می‌توانند با استفاده از حملات متعارف و سکوهای سنتی و مستقل (مانند پهپاد، کشتی‌ها، هواپیماها، وسایل نقلیه زمینی، موشک‌ها و غیره)، در لایه‌های مختلف (یعنی دریا، زمین، هوا، فضا، سایبر) با لحاظ پویایی سامانه و با استفاده از محیط نرم‌افزاری اقدام به انجام عملیات‌های گوناگون نمایند. یکی دیگر از لایه‌های قابل استفاده در الگوسازی، می‌تواند تاثیر هوش مصنوعی بر محیط جنگ ترکیبی باشد. بسیاری از محققان ادعا می‌کنند که جنگ ترکیبی تنها تا آنجایی جدید است که از فناوری‌های نوین به عنوان سلاح جنگی استفاده می‌کند. از این رو هوش مصنوعی پیامدهای سطح راهبردی برای هدایت و دفاع در برابر جنگ ترکیبی خواهد داشت [۲۲].

با توجه به موارد گفته شده و نظر به اهمیت جنگ ترکیبی و گسترش طرح و بکارگیری عملی آن توسط دولت‌ها در سال‌های اخیر، ارائه چارچوب اخیر تلاشی است برای آغاز استخراج نرم-افزارهای بازی جنگ ترکیبی مبتنی بر چارچوب پیاده‌سازی شده فوق که قابلیت استفاده در کاربردهای مختلف، جغرافیاهای متفاوت و در برابر تهدیدات متنوع را دارا باشد. بدیهی است پیاده‌سازی نرم‌افزار کامل و سفارشی‌شده متناسب با نیاز کاربران و شرح خدمات درخواستی و با تشکیل تیم‌های موضوعی در هر لایهٔ جنگ ترکیبی به‌منظور استحصال مولفه‌های تخصصی و لحاظ تجارب میدانی امکان‌پذیر بوده و توصیهٔ موکّد است.

در پایان قابل ذکر است که کلیه کدهای برنامه‌نویسی بکار رفته در این مقاله، که بدلیل محدودیت نمایش متنی تنها بخشی از آن در این مقاله گنجانده شد، در سایت گیت‌هاب به آدرس https://github.com/ykhorrami در قالب دو مخزن مجزا مطابق با بخش‌بندی مقاله همراه با توضیحات تکمیلی جهت هرگونه بهره‌برداری به‌صورت منبع‌باز[2] بارگذاری شده است. ارائه منبع‌باز کدهای برنامه این قابلیت را فراهم آورده است تا امکان ویرایش و توسعه نرم‌افزار به‌صورت برخط و بدون هیچگونه محدودیتی در فضای وب و بصورت اشتراکی برای اعضای جامعهٔ علمی وجود داشته باشد.

۶- مراجع

[1] J. R. Davis, "The Hybrid Mindset and Operationalizing Innovation: Toward a Theory of Hybrid", Political Science, 2014.
[2] E. Cayirci, et al., "A Model to Describe Hybrid Conflict Environments", 13th International Multidisciplinary Modeling & Simulation Multiconference, pp. 52-60, 2016.
[3] Headquarters Department of the Army, Field Manual 6.0: Commander and Staff Organization and Operations. Washington, DC, USA: Department of the Army, 2016. (https://litpolukrbrig.wp.mil.pl/u/12_FM_6-0_C2._Commander_and_Staff_Organization_and_Operations._22_April_2016.pdf)

[1] Mobile IoT

[2] Open-source

[4] V. Mittal and A. Davidson, “Combining Wargaming With Modeling and Simulation to Project Future Military Technology Requirements”, IEEE Transactions on Engineering Management, vol. 68, no. 4, pp. 1195-1207, 2021.
[5] Oxford Learner's Dictionaries. Definition of hybrid adjective [Online], Available: https://www.oxfordlearnersdictionaries.com
[6] MSC (2015). Munich Security Conference 2015 [Online], Available: https://securityconference.org/msc-2015/
[7] J. J. McCuen, “Hybrid Wars”, Military Review, vol. 108, 2008.
[8] OTAN-Allied Command Transformation. Multiple Futures Project. Navigating Towards 2030. Norfolk: OTAN, pp. 55, 2009.
[9] W. J. Nemeth, “Future war and Chechnya: a case for hybrid warfare”, Monterey, Naval Postgraduate School, 2006.
[10] F. G. Hoffman, “Hybrid warfare and challenges”, vol. 52, pp. 34-39, 2009.
[11] F. G. Hoffman, “Conflict in the 21st Century: The Rise of Hybrid Wars,” Potomac Institute for Policy Studies, Arlington, Virginia. 2007
[12] J. A. Warden, “The Air Campaign, Planning for Combat,” DIANE Publishing Company. 1994
[13] M. Myronova, “The EU-NATO Cooperation: Prospectives for More Autonomous Europe.UA: Ukraine Analytica”, vol. 14, 2018. Available: http://ukraine-analytica.org/the-eu-nato-cooperation-perspectives-for-more-autonomous-europe/
[14] E. Çayırcı, “Modelling and Simulation for Hybrid Environments”, STO-MP-MSG-143, vol. 143, pp. 8-14, 2018.
[15] S. Antonenko, “Functioning Features of the Armed Forces of Ukraine Management System in a Hybrid Warfare. Information dimension of hybrid warfare: the experience of Ukraine, materials of the international scientific-practical”, conference of The National Defence University of Ukraine, pp. 10-16, 2017.
[16] Project: T-REX [Online]. Available: http://www.liophant.org/projects/t-rex.html.
[17] S. Mittal, U. Durak, and T. Ören, “Guide to Simulation-Based Disciplines. Springer International Publishing, 2017.
[18] T. E. Culton, D. W. Parkes and J. T. Walrond, “Future Construct/Architecture for Modeling and Simulation Support to Joint and Collective Training Across the Continuum of Military Operations”, STO-MP-MSG-143, p. 20, 2019. Available: https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/STO-MP-MSG-143/MP-MSG-143-10P.pdf.
[19] J. Kazil, D. Masad and A. Crooks, “Utilizing Python for Agent-Based Modeling: The Mesa Framework”, Lecture Notes in Computer Science, vol. 12268. Springer, Cham. 2020. Available: https://doi.org/10.1007/978-3-030-61255-9_30
[20] W. C. Baldwin, B. Sauser and R. Cloutier, “Simulation Approaches for System of Systems: Events-Based versus Agent Based Modeling”, Conference on Systems Engineering Research, Procedia Computer Science, vol. 44, pp. 363 – 372, 2015.
[21] D. Masad, J. Kazil, “Mesa: An Agent-Based Modeling Framework”, Python in Science Conference, 2015.
[22] B. Bankov, “Revolutionising Hybrid Warfare: The Role of Artificial Intelligence”, Annual Conference of Crisis Management and Disaster Response Centre, 2023.