УДК 004.4; 004.5; 004.89
DOI: 10.26102/2310-6018/2026.58.7.001

Динамическая визуализация построения MCTS-дерева в среде GGP Base Package

Д.В. Литовкин, О.Д. Ролдугин, Г.А. Якимов, С.Д. Николаенко, Д.А. Смутин

Волгоградский государственный технический университет, Волгоград, Российская Федерация

Резюме. Статья посвящена разработке системы динамической визуализации дерева поиска в алгоритме Monte Carlo Tree Search (MCTS), реализованном в среде General Game Playing Base Package (GGP Base Package). Основная проблема существующих подходов – отсутствие средств наблюдения за эволюцией дерева поиска в процессе построения, что затрудняет отладку и анализ поведения MCTS-агентов. Методология включает: систематическую оценку восьми инструментов визуализации по 12 критериям (универсальность, динамичность, интерактивность, масштабируемость, производительность и т. д.); разработку четырёхслойной архитектуры (Java/GGP Base Package → Redis → ASP.NET Core → React + D3.js); реализацию интерактивного механизма воспроизведения эволюции дерева с пошаговым анализом. Результаты сравнительной оценки показывают, что разработанная система достигает интегральной оценки 0,752 по сформулированным критериям и, в рамках рассмотренных инструментов, демонстрирует наибольшую совокупность ключевых свойств, важных для отладки MCTS-агентов. Экспериментальные испытания на игре ConnectFour (6×8) показывают, что система обеспечивает плавную визуализацию (>50 FPS) для MCTS-деревьев до 1000 узлов, поддерживает произвольные игры в формате GDL и по экспертной оценке позволяет за считанные секунды визуального анализа выявлять отличия в поведении модификаций алгоритма MCTS, тогда как без визуализации это требует длительного ручного сопоставления финальных статистик. Сделанные выводы подтверждают, что динамическая визуализация промежуточных состояний MCTS-дерева предоставляет дополнительные возможности для обнаружения скрытых дефектов реализации и нетривиальных особенностей поведения алгоритма MCTS, остающихся неочевидными при анализе только финального состояния дерева. Предлагаемый инструмент может представлять практический интерес для исследователей, разработчиков и преподавателей в области General Game Playing.

Ключевые слова: Monte Carlo Tree Search, General Game Playing, визуализация дерева поиска, интерактивное воспроизведение, отладка алгоритмов поиска, визуализация алгоритма MCTS, воспроизведение эволюции.

Для цитирования: Литовкин Д.В., Ролдугин О.Д, Якимов Г.А., Николаенко С.Д., Смутин Д.А. Динамическая визуализация построения MCTS-дерева в среде GGP Base Package. *Моделирование, оптимизация и информационные технологии.* 2026;14(7). URL: https://moitvivt.ru/ru/journal/article?id=2340 DOI: 10.26102/2310-6018/2026.58.7.001

Dynamic visualization of MCTS-Tree construction in the GGP Base Package

D.V. Litovkin, O.D. Roldugin, G.A. Yakimov, S.D. Nikolaenko, D.A. Smutin

Volgograd State Technical University, Volgograd, the Russian Federation

Abstract. The paper presents a dynamic visualization system for the search tree of the Monte Carlo Tree Search (MCTS) algorithm implemented in the General Game Playing Base Package (GGP Base Package). The main limitation of existing approaches is the lack of tools for observing the evolution of the search tree during its construction, which complicates debugging and analysis of MCTS agents'

behaviour. The methodology includes: a systematic evaluation of eight visualization tools against 12 criteria (universality, dynamism, interactivity, scalability, performance, etc.); the design of a four-layer architecture (Java/GGP Base Package → Redis → ASP.NET Core → React + D3.js); and the implementation of an interactive mechanism for replaying tree evolution with step-by-step analysis. The comparative evaluation shows that the proposed system achieves an integral score of 0.752 according to the defined criteria and, among the considered tools, offers the most favourable combination of key properties relevant for debugging MCTS agents. Experimental studies on the ConnectFour (6×8) game demonstrate that the system provides smooth visualization (>50 FPS) for MCTS trees with up to 1000 nodes, supports arbitrary games in GDL format and, according to expert judgment, enables differences in the behaviour of MCTS algorithm modifications to be identified within seconds of visual inspection, whereas without visualization this would require time-consuming manual comparison of final statistics. The results confirm that dynamic visualization of intermediate MCTS tree states offers additional opportunities for detecting hidden implementation defects and non-trivial behavioural patterns of the MCTS algorithm that remain unobvious when only the final tree state is analysed. The proposed tool may be of practical interest to researchers, developers and educators in the field of General Game Playing.

Keywords: Monte Carlo Tree Search, General Game Playing, search tree visualization, interactive playback, search algorithm debugging, MCTS algorithm visualization, evolution playback.

For citation: Litovkin D.V., Roldugin O.D., Yakimov G.A., Nikolaenko S.D., Smutin D.A. Dynamic visualization of MCTS-Tree construction in the GGP Base Package. *Modeling, Optimization and Information Technology.* 2026;14(7). (In Russ.). URL: https://moitvivt.ru/ru/journal/article?id=2340 DOI: 10.26102/2310-6018/2026.58.7.001

Введение

Алгоритм Monte Carlo Tree Search (MCTS) [1, 2] и его модификации являются стандартным подходом к состязательному поиску в системах General Game Playing (GGP) [3], где универсальные агенты играют в различные игры на основе их формального описания на языке Game Description Language (GDL), а также используются как планирующий модуль в AlphaZero и подобных архитектурах [4, 5].

GGP Base Package представляет собой открытую платформу для разработки и тестирования таких агентов и включает: парсер GDL; реализацию базовых алгоритмов поиска (MCTS, minimax); средства логирования финального состояния MCTS-дерева в JSON-формате; инфраструктуру для организации турниров между агентами. При этом платформа не предусматривает логирование и анализ промежуточных состояний дерева поиска, что существенно ограничивает отладку и исследование MCTS-агентов: невозможно анализировать эволюцию дерева в процессе построения; затруднена отладка модификаций алгоритма MCTS и настройка их гиперпараметров; сложно выявлять дефекты реализации, маскируемые финальной статистикой; отсутствует наглядная демонстрация фаз Selection, Expansion, Simulation и Backpropagation в образовательных сценариях.

Цель исследования – устранить эти ограничения за счёт разработки системы динамической визуализации промежуточных состояний MCTS-дерева в среде GGP Base Package, обеспечивающей наблюдение за эволюцией дерева, поддержку отладки и оптимизации алгоритма, выявление скрытых дефектов реализации и наглядное обучение структуре и динамике алгоритма MCTS.

Monte Carlo Tree Search – стохастический алгоритм для построения дерева поиска, основанный на многократном повторении четырёх фаз [1, 2] (Рисунок 1).

1. Selection (выбор) – спуск от корня к листу в соответствии с политикой выбора, как правило, Upper Confidence Bound applied to Trees (UCT), задаваемой формулой:

$$UCT_j = X_j + C \cdot \sqrt{\frac{ln\,(N)}{n_j}}, \quad (1)$$

где X_j – среднее вознаграждение j-го узла, N – количество посещений узла родителя, n_j – количество посещений j-го узла, C – параметр баланса между исследованием и эксплуатацией.

2. Expansion (расширение) – добавление одного или нескольких дочерних узлов к выбранному листу.

3. Simulation (симуляция) – случайное разыгрывание партии от нового узла до терминального состояния.

4. Backpropagation (обратное распространение) – обновление статистики узлов на пути от терминального состояния к корню.

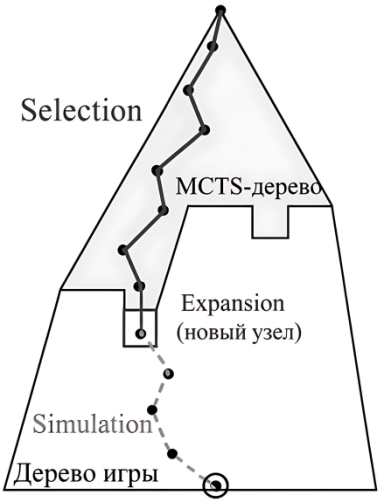


Рисунок 1 – Фазы алгоритма MCTS в рамках одной итерации
Figure 1 – Phases of the MCTS algorithm within one iteration

Ключевое свойство алгоритма MCTS – асимметричный и стохастический рост дерева: более перспективные ветви развиваются глубже, тогда как менее перспективные остаются слабо исследованными [1]. Это обеспечивает эффективность в пространствах большой размерности, но усложняет отладку и сравнение запусков алгоритма.

Для сравнительного анализа существующих инструментов визуализации MCTS-дерева нами сформулированы 12 критериев оценки $k_1 - k_{12}$ и их веса w_i, отражающие относительную важность критериев с точки зрения задач отладки и исследования MCTS в контексте GGP (Таблица 1). Критически важны критерии $k_1 - k_6$ (полнота метрик MCTS, поддержка промежуточных состояний дерева, универсальность по типам игр, масштабируемость, интерактивность и воспроизведение эволюции дерева), тогда как критерии $k_7 - k_{12}$ (эстетика интерфейса, документация и др.) имеют меньший вес и не являются определяющими для целей настоящей работы.

В анализ включены восемь инструментов, пригодных для визуализации MCTS-дерева (Таблица 2). На предварительном этапе были рассмотрены более 25 инструментов визуализации алгоритмов поиска, однако в итоговый набор $S_1 - S_8$ вошли только решения с публично доступной реализацией или веб-демонстрацией, поддержкой визуализации MCTS-деревьев или деревьев схожих алгоритмов и достаточной документацией либо примерами использования. Для каждого инструмента S_p вычислялась интегральная оценка E_p по формуле, представляющая собой взвешенную среднюю оценку по всем критериям:

$$E_p = \sum_{i=1}^{12} score_{p,i} \cdot w_i/100, \quad (2)$$

где $score_{p,i}$ – экспертная оценка инструмента S_p по критерию i, w_i – вес критерия i в процентах (Таблица 1). Оценивание инструментов выполнялось двумя экспертами, имеющими опыт разработки и исследования MCTS-агентов в среде GGP Base Package. Интегральные оценки $score_{p,i}$ получены путём усреднения индивидуальных оценок экспертов по шкале 0–5 для каждого критерия. Формальные метрики согласованности оценок (например, коэффициент конкордации) не рассчитывались и относятся к перспективным задачам дальнейших исследований.

Таблица 1 – Критерии оценки инструментов для визуализации MCTS-дерева
Table 1 – Criteria for evaluating tools for visualizing the MCTS tree

k_i	Наименование критерия	Вклад w_i, %	Описание
k_1	Полнота метрик MCTS	15	Отображение метрик X_j и n_j, а также производных от них
k_2	Промежуточные состояния дерева поиска	15	Отображение эволюции дерева на разных этапах построения
k_3	Универсальность	18	Поддержка произвольных игр и платформ
k_4	Масштабируемость	8	Способность работать с большим числом узлов (1000+)
k_5	Интерактивность	14	Масштабирование, панорамирование, фильтрация, контекстные меню
k_6	Воспроизведение эволюции дерева поиска	12	«воспроизведение» / «пауза» / «запуск на 1 шаг» с управляемой скоростью
k_7	Анимация и плавность	5	Плавные визуальные переходы между состояниями дерева поиска
k_8	Эстетика интерфейса	5	Общее впечатление и удобство использования
k_9	Легкость освоения	3	Кривая обучения, интуитивность
k_{10}	Кодирование информации	2	Использование цвета, размера, формы для кодирования метрик узлов в MCTS-дереве
k_{11}	Документация	2	Наличие примеров, руководств, справок
k_{12}	Режимы для разных пользователей	1	Поддержка новичка, студента, исследователя, эксперта
Итого		100	

Полученные значения E_p приведены в Таблице 3. Результаты показывают, что ни один из рассмотренных инструментов не демонстрирует одновременно высокие значения по ключевым критериям: полноте метрик, поддержке произвольных GDL-игр в GGP Base Package, пошаговому воспроизведению эволюции MCTS-дерева, масштабируемости до 1000–5000 узлов, развитой интерактивности и управляемому воспроизведению процесса построения дерева (Таблица 3). Поскольку именно эти свойства критичны для отладки и исследования MCTS-агентов, это обосновывает необходимость разработки специализированного решения, ориентированного на динамическую визуализацию MCTS в контексте GGP.

Таблица 2 – Инструменты, пригодные для визуализации MCTS-дерева
Table 2 – Tools suitable for visualizing the MCTS tree

S_p	Наименование инструмента
S_1	MCTS-Viz[1]
S_2	AlphaZero Tic-Tac-Toe with Monte Carlo Tree Search (SagarNildas)[2]
S_3	D3.js (Data-Driven Documents)[3]
S_4	AlphaZero Educational Visualization [6]
S_5	D3 Graphviz Integration[4]
S_6	Playing with MCTS [7]
S_7	GGP Base Package + JSON Grid[5]
S_8	MCTS.jl Julia Package[6]

Таблица 3 – Сравнение инструментов, пригодных для визуализации MCTS-дерева
Table 3 – Comparison of tools suitable for visualization of the MCTS tree

Инстру-мент	Критерии												E_i
	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	
S_1	5	4	0	2	2	4	2	2	4	2	0	2	0,530
S_2	4	4	2	3	4	3	3	4	4	3	4	1	0,668
S_3	3	2	5	4	4	1	3	4	3	4	5	1	0,656
S_4	3	2	4	3	3	1	2	3	3	2	3	2	0,542
S_5	3	2	4	4	3	1	2	4	2	4	5	1	0,576
S_6	3	3	1	2	3	2	2	3	3	2	2	1	0,466
S_7	5	1	3	5	0	1	0	1	2	3	0	3	0,432
S_8	2	1	5	3	2	1	2	3	2	2	3	1	0,482

Подход к использованию дерева поиска как центрального объекта визуального анализа подтверждён и в смежных областях. В системе generAItor [8] дерево beam search является основным интерфейсным элементом для отладки генерации текста языковыми моделями, а TreeVersity [9] реализует интерактивное сравнение пар деревьев по структурным и метрическим изменениям. В отличие от этих инструментов, предложенная система обеспечивает динамическое воспроизведение процесса построения MCTS-дерева в контексте GGP с поддержкой произвольных GDL-игр.

Материалы и методы

Для устранения отмеченных ограничений сформулированы функциональные (FR) и нефункциональные (NFR) требования к системе визуализации MCTS-дерева в среде GGP Base Package.

Функциональные требования: FR1 – визуализация дерева поиска для произвольных игр, описываемых на языке GDL в GGP Base Package, без модификации кода визуализации; FR2 – интерактивное воспроизведение процесса построения MCTS-дерева с операциями «воспроизведение», «пауза», «шаг вперёд», «шаг назад» и переходом к произвольной итерации алгоритма; FR3 – сохранение контекста наблюдения (масштаб, позиция камеры, раскрытые узлы, выбранный узел) при переходе

[1] Garcia V. *mcts-viz.* GitHub. URL: https://github.com/vgarciasc/mcts-viz (дата обращения: 20.03.2026).
[2] Das S. *AlphaZero-Tic-Tac-Toe-App.* GitHub. URL: https://github.com/sagarnildass/AlphaZero-Tic-Tac-Toe-App (дата обращения: 20.03.2026).
[3] d3. *D3.js.* GitHub. URL: https://github.com/d3/d3 (дата обращения: 20.03.2026).
[4] magjac. *d3-Graphviz.* GitHub. URL: https://github.com/magjac/d3-graphviz (дата обращения: 20.03.2026).
[5] *Json Grid Viewer.* URL: https://jsongrid.com/json-grid/ (дата обращения: 20.03.2026).
[6] JuliaPOMDP. *MCTS.jl.* GitHub. URL: https://github.com/SISL/MCTS.jl (дата обращения: 20.03.2026).

между состояниями дерева и анимация этих переходов; FR4 – фильтрация узлов по максимальной глубине и минимальной относительной частоте посещений по отношению к узлу-родителю; FR5 – отображение в контекстном меню базовых метрик узла (среднее вознаграждение, число посещений, глубина, предшествующее действие); FR6 – визуальное кодирование метрик: размер узла в зависимости от числа посещений (с использованием логарифмической шкалы) и цвет в зависимости от относительной интенсивности исследования.

Нефункциональные требования: NFR1 – интеграция с GGP Base Package без существенных модификаций ядра платформы; NFR2 – плавная визуализация (не менее 50 FPS) для деревьев до 1000 узлов и поддержка работоспособности при деревьях до 5000 узлов; NFR3 – работа во всех современных браузерах (Chrome 120+, Firefox 121+, Safari 17+) под управлением Windows, Linux и macOS.

Для выполнения этих требований разработана система с четырёхслойной микросервисной архитектурой (Рисунок 2). Компонент «GGP Base Package (Java)» представляет собой модифицированную версию стандартного решения и дополнительно формирует снимки (snapshots) MCTS-дерева через каждые T_m итераций алгоритма, причём интервал T_m изменяется по экспоненциальному закону (увеличивается по мере роста дерева). Снимки сохраняются в хранилище Redis, используемом как NoSQL-база данных типа key–value. Компонент «Backend (ASP.NET Core)» предоставляет REST API для доступа к снимкам MCTS-дерева и служебной информации. Компонент «Frontend (React + D3.js)» реализует интерактивную визуализацию эволюции дерева, включая панорамирование, масштабирование, фильтрацию узлов и управление воспроизведением процесса построения MCTS-дерева.

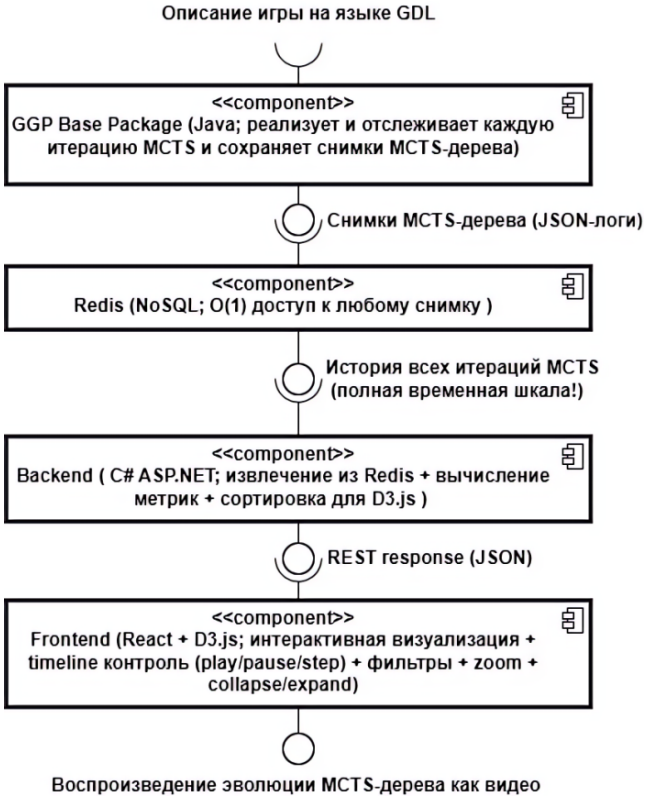
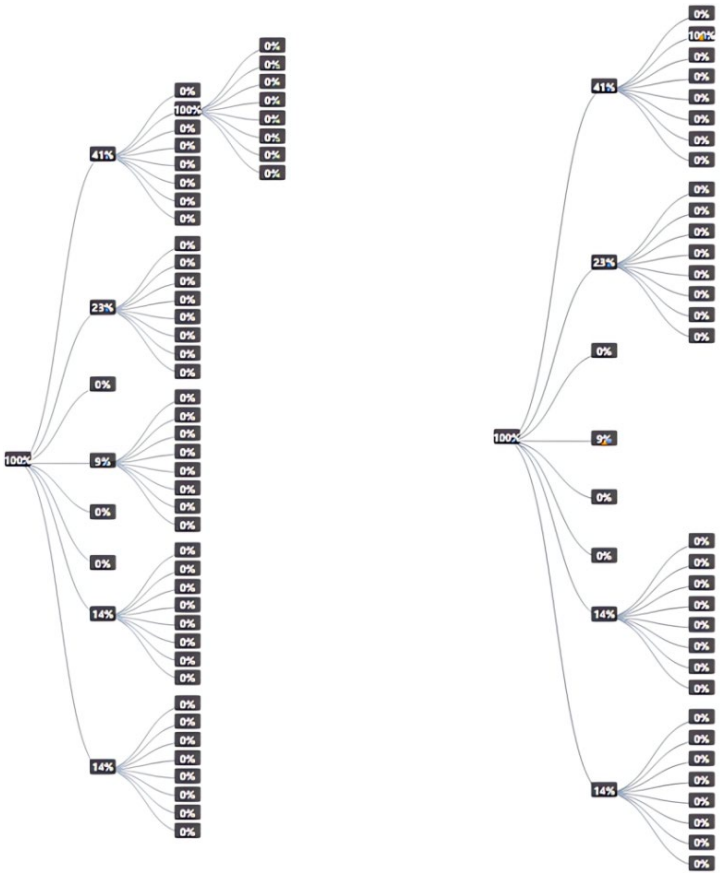


Рисунок 2 – Архитектура системы динамической визуализации MCTS-дерева для произвольных игр

Figure 2 – The architecture of the MCTS-tree dynamic visualization system for arbitrary games

Результаты и обсуждение

Процесс визуального воспроизведения эволюции MCTS-дерева реализован как последовательное отображение снимков состояния дерева, соответствующих различным итерациям алгоритма (Рисунки 3 и 4). Пользователь может управлять воспроизведением с помощью панели, которая обеспечивает операции «воспроизведение/пауза», пошаговый переход вперёд/назад и изменение скорости проигрывания (Рисунок 5). Дополнительно поддерживаются панорамирование, масштабирование и фильтрация узлов по глубине и частоте посещений, что позволяет анализировать как глобальную структуру дерева, так и локальные поддеревья.



а) *б*)

Рисунок 4 – Снимок MCTS-дерева после 30-ти итераций: *а* – без фильтрации узлов; *б* – с фильтрацией узлов
Figure 4 – Snapshot of the MCTS tree after 30 iterations: *a* – without node filtering; *b* – with node filtering

С точки зрения удовлетворения сформулированных критериев визуализации (Таблица 1) реализованная система получает интегральную оценку 0,752, что является наибольшим значением среди рассмотренных инструментов (Таблица 3) и отражает её преимущество по совокупности ключевых свойств. В отличие от MCTS-Viz и образовательных AlphaZero-визуализаций, ориентированных на конкретные игры и фиксированные сценарии, предложенная система поддерживает произвольные GDL-игры в GGP Base Package и предоставляет общий механизм пошагового воспроизведения MCTS-дерева с фильтрацией и управляемым уровнем детализации. В отличие от универсальных библиотек визуализации (D3.js, D3-Graphviz) и инструментов работы с JSON-структурами, она интегрирует MCTS-специфичные метрики и процессы

и снимает необходимость ручной разработки связующего кода. Тем самым система закрывает сочетание требований по универсальности, динамичности, интерактивности и интеграции с GGP Base Package, которое не реализовано ни одним из рассмотренных решений (Таблицы 2, 3).

Рисунок 5 – Панель для воспроизведения эволюции MCTS-дерева
Figure 5 – Panel for reproducing the evolution of the MCTS tree

Производительность системы оценивалась на компьютере с процессором Intel i7-8700K, 32 GB оперативной памяти и браузером Chrome 120 под управлением Windows 10. Тестирование выполнялось на MCTS-деревьях, построенных при анализе Connect Four на доске 6×8. Для стандартной доски 6×7 число легальных состояний оценивают как порядка $4{,}5\times10^{12}$, а для доски 6×8 это число значительно больше. Поэтому Connect Four удобно использовать как тестовый стенд для алгоритмов, работающих с большим пространством состояний [10, 11]. Результаты (Таблица 4) показывают, что для деревьев до 1000 узлов система обеспечивает частоту кадров не ниже 50 FPS и время загрузки менее 1 секунды, что соответствует требованию NFR2. При размере дерева 2000–5000 узлов частота кадров снижается до 18–35 FPS, время загрузки возрастает до 1,5–3,2 секунды, но интерфейс остаётся работоспособным и поддерживает панорамирование, масштабирование и управление воспроизведением без критичных задержек.

Таблица 4 – Производительность системы визуализации MCTS-дерева
Table 4 – Performance of the MCTS-tree visualization system

Размер дерева	FPS	Время загрузки	Память	Статус
100 узлов	60 FPS	200 мс	45 MB	Плавно
500 узлов	58 FPS	450 мс	95 MB	Плавно
1000 узлов	52 FPS	850 мс	145 MB	Плавно
2000 узлов	35 FPS	1,5 сек	210 MB	Приемлемо
5000 узлов	18 FPS	3,2 сек	380 MB	Медленно

Чтобы продемонстрировать практическую полезность динамической визуализации эволюции MCTS-дерева, мы провели эксперимент с двумя версиями алгоритма MCTS для ConnectFour (6×8). В обеих версиях на фазе Expansion к выбранному узлу добавляются все дочерние состояния, но в фазе Simulation первая версия за итерацию симулирует один случайно выбранный дочерний узел, а вторая – все добавленные узлы. При лимите в 10 секунд первая версия выполнила 1611 итераций, вторая – 1392, то есть объём вычислений остаётся сопоставимым. В Таблице 5 приведена финальная статистика узлов первого уровня MCTS-дерева, которые соответствуют возможным первым ходам. Версия 1 – симуляция одного добавленного узла за итерацию, Версия 2 – симуляция всех добавленных узлов за итерацию. Обе версии алгоритма выбирают один и тот же ход как наилучший, но распределения средних вознаграждений (X) и числа посещений (n) по остальным ходам заметно различаются. Эти различия могут быть связаны как со стохастической природой MCTS, так и с различиями в реализации фазы Simulation, что отражается в профилях роста поддеревьев: в первой версии дерево

заметно смещено в сторону отдельных ветвей, а во второй альтернативные ходы исследуются более равномерно. Последовательный просмотр снимков MCTS-дерева позволяет увидеть эти закономерности за несколько секунд, тогда как анализ одной только финальной статистики требует значительно больше времени. Динамическая визуализация, таким образом, служит инструментом «быстрого обзора» поведения MCTS-агентов и помогает оперативно обнаруживать аномалии и неочевидные паттерны поиска. При этом данный эксперимент выполнен для одного набора параметров и фиксированного начального состояния генератора случайных чисел. Систематический анализ статистической устойчивости наблюдаемых закономерностей по множественным запускам и различным параметрам алгоритма остаётся задачей для дальнейших исследований.

Таблица 5 – Статистика узлов первого уровня MCTS-дерева для возможных первых ходов в ConnectFour, полученная двумя версиями алгоритма
Table 5 – Statistics of nodes of the first level of the MCTS tree for possible first moves in ConnectFour, obtained by two versions of the algorithm

Узлы первого уровня MCTS-дерева	Возможные ходы первого игрока	Версия 1	Версия
Node1	drop(1)	X = 50,00, n = 58	X = 56,46, n = 209
Node2	drop(2)	X = 49,02, n = 307	X = 54,02, n = 137
Node3	drop(3)	X = 55,66, n = 106	X = 52,58, n = 97
Node4	drop(4)	X = 62,13, n = 338	X = 54,97, n = 161
Node5	drop(5)	X = 62,91, n = 399	X = 59,60, n = 401
Node6	drop(6)	X = 62,87, n = 307	X = 44,90, n = 137
Node7	drop(7)	X = 44,74, n = 38	X = 51,55, n = 97
Node8	drop(8)	X = 46,15, n = 58	X = 54,90, n = 153

Несмотря на показанную практическую полезность динамической визуализации для анализа поведения MCTS-агентов, текущее решение имеет ряд ограничений. Производительность заметно падает при размере дерева свыше 5000 узлов, хранение полных снимков требует много памяти, нет раздельной визуализации фаз MCTS, а деревья с десятками тысяч узлов трудно воспринимать. В дальнейшей работе планируется перейти к Canvas/WebGL-рендерингу, использовать дельта-кодирование снимков, визуализировать отдельные фазы алгоритма, применить методы семантического масштабирования, а также провести формализованные пользовательские исследования, дополняющие текущую экспертную оценку качества визуализации.

Заключение

В работе представлена система динамической визуализации MCTS-дерева с четырёхслойной архитектурой (GGP Base Package на Java → Redis → ASP.NET Core → React + D3.js), интегрированная в GGP Base Package и удовлетворяющая сформулированным функциональным и нефункциональным требованиям. Система обеспечивает интерактивное воспроизведение эволюции дерева поиска, отображение ключевых метрик узлов и управление уровнем детализации структуры дерева. С точки зрения сформулированных критериев визуализации (Таблица 1) реализованная система достигает интегральной оценки 0,752 – наибольшего значения среди рассмотренных инструментов (Таблицы 2, 3). Это означает, что по совокупности критически важных для GGP-сценариев свойств (поддержка произвольных GDL-игр, динамическая

визуализация эволюции дерева, интерактивное воспроизведение и масштабируемость до нескольких тысяч узлов) система обеспечивает более полную поддержку отладки и исследования MCTS-агентов по сравнению с рассмотренными решениями.

Экспериментальные результаты на игре ConnectFour (6×8) демонстрируют, что предложенное решение обеспечивает плавную визуализацию деревьев до 1000 узлов и приемлемую интерактивность при размере до 5000 узлов (Таблица 4). Пошаговая визуализация процесса построения дерева позволяет выявлять различия в поведении модификаций алгоритма MCTS, которые неочевидны при анализе только финальной статистики узлов (Таблица 5). В частности, потенциальные аномалии и нетривиальные особенности реализации могут быть обнаружены в течение нескольких секунд визуального анализа вместо десятков минут ручного сопоставления финальных распределений.

Полученные результаты подтверждают, что динамическая визуализация процесса построения MCTS-дерева предоставляет качественно новые возможности для отладки, анализа и обучения алгоритмам поиска в контексте GGP. Дальнейшее развитие работы связано с масштабированием визуализации на деревья с числом узлов порядка 10^5 за счёт использования Canvas/WebGL, внедрением дельта-кодирования снимков для снижения требований к памяти, а также интеграции методов семантического масштабирования для повышения удобства анализа крупных деревьев.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Browne C.B., Powley E., Whitehouse D., et al. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games.* 2012;4(1):1–43. https://doi.org/10.1109/TCIAIG.2012.2186810
2. Świechowski M., Godlewski K., Sawicki B., et al. Monte Carlo tree search: a review of recent modifications and applications. *Artificial Intelligence Review.* 2022;56(5):2497–2562. https://doi.org/10.1007/s10462-022-10228-y
3. Genesereth M., Thielscher M. *General game playing.* Cham: Springer; 2014. 213 p.
4. Scheiermann J., Konen W. AlphaZero-inspired game learning: faster training by using MCTS only at test time. *IEEE Transactions on Games.* 2023;15(4):637–647. https://doi.org/10.1109/TG.2022.3206733
5. Trudeau A., Bowling M. Targeted search control in AlphaZero for effective policy improvement. In: *AAMAS '23: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 29 May – 02 June 2023, London, United Kingdom.* New York: ACM; 2023. P. 842–850.
6. Li R., Mo A., Su G., et al. *AlphaZero-Edu: Democratizing Access to AlphaZero.* arXiv. URL: https://doi.org/10.48550/arXiv.2504.14636 [Accessed 20th March 2026].
7. Zhao Y., Hu Ch., Liu J. Playing with Monte-Carlo Tree Search [AI-eXplained]. *IEEE Computational Intelligence Magazine.* 2024;19(1):85–86. https://doi.org/10.1109/MCI.2023.3328150
8. Spinner Th., Kehlbeck R., Sevastjanova R., et al. -generAItor: tree-in-the-loop text generation for language model explainability and adaptation. *ACM Transactions on Interactive Intelligent Systems.* 2024;14(2):14. https://doi.org/10.1145/3652028
9. Guerra-Gómez J.A., Buck-Coleman A., Plaisant C., et al. TreeVersity: Comparing tree structures by topology and node's attributes differences. In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST), 23–28 October 2011, Providence, RI, USA.* IEEE; 2011. P. 275–276. https://doi.org/10.1109/VAST.2011.6102471

10. Taylor H., Stella L. *An evolutionary framework for Connect-4 as test-bed for comparison of advanced Minimax, Q-learning and MCTS.* arXiv. URL: https://doi.org/10.48550/arXiv.2405.16595 [Accessed 20th March 2026].
11. Ala'anzy M.A., Madiyarova A., Aigeldiyev A., et al. Connect-4 AI: A comprehensive taxonomy and critical review of methods and metrics. *Symmetry*. 2026;18(2):293. https://doi.org/10.3390/sym18020293

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Литовкин Дмитрий Васильевич, кандидат технических наук, доцент кафедры «Программное обеспечение автоматизированных систем», Волгоградский государственный технический университет, Волгоград, Российская Федерация.
email: litd@mail.ru
ORCID: 0009-0006-9206-1290

Dmitry V. Litovkin, Candidate of Engineering Sciences, Associate Professor at the Department of Software for Automated Systems, Volgograd State Technical University, Volgograd, the Russian Federation.

Ролдугин Олег Денисович, студент, Волгоградский государственный технический университет, Волгоград, Российская Федерация.
email: olroldugin@yandex.ru
ORCID: 0009-0005-2382-2968

Oleg D. Roldugin, Student, Volgograd State Technical University, Volgograd, the Russian Federation.

Якимов Григорий Алексеевич, аспирант, Волгоградский государственный технический университет, Волгоград, Российская Федерация.
email: greg.yakimov@gmail.com
ORCID: 0009-0003-6216-9805

Gregory A. Yakimov, Postgraduate, Volgograd State Technical University, Volgograd, the Russian Federation.

Николаенко Софья Денисовна, студентка, Волгоградский государственный технический университет, Волгоград, Российская Федерация.
email: nikolaenko.sofya@mail.ru
ORCID: 0009-0004-2287-7548

Sofya D. Nikolaenko, Student, Volgograd State Technical University, Volgograd, the Russian Federation.

Смутин Даниил Александрович, студент, Волгоградский государственный технический университет, Волгоград, Российская Федерация.
email: danilsmutin@gmail.com
ORCID: 0009-0004-0775-6705

Daniil A. Smutin, Student, Volgograd State Technical University, Volgograd, the Russian Federation.

Статья поступила в редакцию 08.04.2026; одобрена после рецензирования 10.06.2026; принята к публикации 24.06.2026.

The article was submitted 08.04.2026; approved after reviewing 10.06.2026; accepted for publication 24.06.2026.