
ТЕОРИЯ И МЕТОДИКА ОБУЧЕНИЯ И ВОСПИТАНИЯ (ПО ОБЛАСТЯМ И УРОВНЯМ ОБРАЗОВАНИЯ)/THEORY AND METHODS OF TEACHING AND UPBRINGING (BY AREAS AND LEVELS OF EDUCATION)

DOI: <https://doi.org/10.60797/IRJ.2026.170.3>

EDN: OFBLTK

СНИЖЕНИЕ КОГНИТИВНОЙ НАГРУЗКИ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ В ШКОЛЕ: ГЕЙМИФИКАЦИЯ КАК ПОВЕДЕНЧЕСКИЙ ТРИГГЕР

Научная статья

Козлов Р.А.^{1,*}, Даниелян В.А.², Горелов П.А.³^{1, 2, 3}Московский политехнический университет, Москва, Российская Федерация

* Корреспондирующий автор (roma.ko2022[at]yandex.ru)

Предложена: 07.06.2026; Принята: 28.07.2026; Опубликовано: 17.08.2026

Аннотация

В представленной статье рассматривается актуальная проблема преодоления когнитивного барьера при переходе обучающихся основной школы от начальных визуальных сред к изучению текстовых языков программирования высокого уровня.

Основная цель исследования заключается в теоретическом обосновании и математической формализации процесса того, как функциональная геймификация образовательного пространства способствует снижению внешней когнитивной нагрузки и выступает катализатором учебной активности. В отличие от традиционного восприятия игровых механик исключительно как инструмента повышения внешней мотивации, предлагается рассматривать их как эргономическое средство управления вниманием и оптимизации рабочей памяти.

В основу методологии положен синтез теории когнитивной нагрузки Дж. Свеллера и поведенческой модели Б.Дж. Фогга. В рамках работы разработана авторская математическая модель, в которой снижение нагрузки обратно пропорционально логарифму индекса геймификации, позволяющая количественно оценить влияние индекса геймификации на объем высвобождаемых ментальных ресурсов. С помощью метода аддитивной свертки критериев проанализирована архитектура обучающей платформы CodeCombat.

Результаты исследования демонстрируют, что глубокая интеграция игровых элементов (визуализация абстракций, мгновенная обратная связь, нарративный контекст) выполняет функцию контекстного скаффолдинга. Согласно расчетам, качественный педагогический дизайн способен снизить давление на рабочую память более чем в четыре раза по сравнению со стандартными консолями разработки, что в перспективе ведет к существенному уменьшению доли структурных и синтаксических ошибок. Делается вывод о том, что геймифицированная среда качественно трансформирует поведенческие паттерны: она минимизирует требования к ситуативной способности ученика и замещает дефицитный внутренний волевой ресурс внешними императивными сигналами.

Полученные результаты могут быть использованы специалистами в области педагогического дизайна для проектирования адаптивных платформ и повышения эффективности преподавания базовой информатики.

Ключевые слова: педагогический дизайн, теория когнитивной нагрузки, школьная информатика, игровые образовательные среды, модель поведения Фогга, контекстный скаффолдинг, синтаксический барьер, эргономика образовательного интерфейса, математическое моделирование в педагогике, алгоритмическое мышление.

REDUCING COGNITIVE LOAD IN SCHOOL PROGRAMMING EDUCATION: GAMIFICATION AS A BEHAVIOURAL TRIGGER

Research article

Kozlov R.A.^{1,*}, Danielyan V.A.², Gorelov P.A.³^{1, 2, 3}Moscow Polytechnic University, Moscow, Russian Federation

* Corresponding author (roma.ko2022[at]yandex.ru)

Suggested: 07.06.2026; Accepted: 28.07.2026; Published: 17.08.2026

Abstract

The article examines the relevant issue of overcoming the cognitive barrier faced by middle school pupils as they transition from basic visual programming environments to learning high-level text-based programming languages.

The main objective of the study is to provide a theoretical substantiation and mathematical formalisation of the process by which the functional gamification of the educational environment helps to reduce external cognitive load and acts as a catalyst for learning activity. In contrast to the traditional view of game mechanics as merely a tool for increasing extrinsic motivation, it is proposed that they be regarded as an ergonomic means of managing attention and optimising working memory.

The methodology is based on a synthesis of J. Sweller's cognitive load theory and B.J. Fogg's behavioural model. As part of this work, an original mathematical model has been developed in which the reduction in cognitive load is inversely proportional to the logarithm of the gamification index, making it possible to quantitatively assess the influence of the gamification index on the amount of mental resources freed up. The architecture of the CodeCombat learning platform was analysed using the additive convolution of criteria method.

The research results demonstrate that the deep integration of game elements (visualisation of abstract concepts, instant feedback, narrative context) serves as contextual scaffolding. According to calculations, high-quality instructional design can reduce the load on working memory by more than four times compared with standard development environments, which in the long term leads to a significant reduction in the proportion of structural and syntactic errors. It is concluded that a gamified environment qualitatively transforms behavioural patterns: it minimises the demands on the learner's situational competence and replaces deficient internal volitional resources with external imperative signals.

The results obtained can be used by specialists in the field of instructional design to develop adaptive platforms and improve the effectiveness of teaching basic computer science.

Keywords: pedagogical design, cognitive load theory, school computer science, game-based learning environments, Fogg behaviour model, contextual scaffolding, syntactic barrier, educational interface ergonomics, mathematical modelling in education, algorithmic thinking.

Введение

В условиях глобальной цифровой трансформации навыки алгоритмического мышления и программирования трансформируются из узкопрофессиональных компетенций в обязательный элемент базовой грамотности. Однако система школьного образования сталкивается с серьезным вызовом: при переходе от начальных визуальных сред к текстовым языкам высокого уровня, таким как Python или C++, наблюдается резкое снижение успеваемости и мотивации обучающихся 8–9 классов. Ключевым барьером на данном этапе становится феномен «когнитивного переполнения», когда необходимость одновременного удержания в рабочей памяти абстрактной логики алгоритма, строгого синтаксиса и сложного интерфейса профессиональных сред разработки превышает психофизиологические возможности подростков.

Степень разработанности проблемы в научной литературе остается неоднородной. Фундаментальные аспекты распределения ментальных ресурсов при обучении подробно описаны в рамках теории когнитивной нагрузки Дж. Свеллера [1]. В то же время геймификация традиционно рассматривается в педагогике преимущественно как инструмент повышения внешней мотивации через внедрение игровых элементов PBL — Points, Badges, Leaderboards [2], [3]. В существующем научном дискурсе наблюдается дефицит исследований, рассматривающих игровые механики не как мотивационную надстройку, а как эргономический инструмент оптимизации когнитивных ресурсов и управления вниманием, несмотря на доказанную эффективность геймифицированных подходов в преподавании информатики [4], [5].

Цель данной работы заключается в теоретическом обосновании и математической формализации модели, описывающей влияние функциональной геймификации на снижение внешней когнитивной нагрузки и формирование устойчивых поведенческих паттернов учебной деятельности. Научная новизна исследования состоит в предложении авторской модели, описывающей обратно пропорциональную зависимость между логарифмом индекса геймификации образовательной среды и объемом внешней когнитивной нагрузки.

Методы и принципы исследования

Методологическую основу исследования составил синтез когнитивистского подхода и математического моделирования поведенческих систем. Базовой теоретической рамкой выступает Теория когнитивной нагрузки, согласно которой общая когнитивная нагрузка (L_{total}) при обучении программированию декомпозируется на три компонента. Первый — внутренняя нагрузка (L_{int}) определяемая объективной алгоритмической сложностью задачи, например, вложенные циклы, которая является неизменной константой учебной цели. Второй — внешняя нагрузка (L_{ext}), генерируемая форматом подачи материала и интерфейсом среды. Третий — релевантная нагрузка (L_{germ}), представляющая собой ресурсы, затрачиваемые мозгом на построение ментальных схем понимания. Условием эффективного обучения является непревышение суммы этих компонентов над предельной емкостью рабочей памяти (C_{mem}):

$$L_{int} + L_{ext} + L_{germ} \leq C_{mem} \quad (1)$$

В традиционных методиках обучения начинающих программистов значение внешней нагрузки часто достигает критических величин из-за сложности инструментария (IDE), что приводит к блокировке процесса усвоения знаний. Автором выдвигается гипотеза, что качественная геймификация среды разработки выступает демпфером внешней нагрузки. Для формализации этого процесса предлагается модифицированная функция зависимости внешней нагрузки от уровня геймификации (G):

$$L_{ext}(G) = \frac{L_{base}}{k \cdot \ln(G+e)} \quad (2)$$

Где L_{base} — базовый уровень внешней нагрузки традиционной среды, G — интегральный индекс геймификации, k — коэффициент качества педагогического дизайна ($0 < k \leq 1$), а e — основание натурального логарифма.

Выбор данной функциональной зависимости обусловлен необходимостью учета психофизиологических особенностей восприятия. Во-первых, использование именно натурального логарифма базируется на фундаментальном психофизиологическом законе Вебера–Фехнера, согласно которому интенсивность субъективного ощущения пропорциональна натуральному логарифму интенсивности стимула. В контексте нашей модели это отражает закон убывающей предельной полезности — внедрение первичных механик (визуализация) дает максимальный эффект разгрузки мозга, тогда как дальнейшее наращивание количества игровых элементов выводит эффект на плато насыщения. Во-вторых, введение нормировочной константы (e) обеспечивает устойчивость модели:

при полном отсутствии геймификации ($G = 0$) знаменатель равен единице, что корректно возвращает значение базовой нагрузки. В-третьих, коэффициент k выступает «штрафным» множителем, указывая на то, что высокий уровень геймификации не гарантирует результат при низком качестве UX-дизайна.

Для расчета переменной G и практического применения модели нами применен метод линейной аддитивной свертки критериев. Индекс $G \in [0, 100]$

$$G = \sum_{i=1}^n w_i \cdot x_i \quad (3)$$

Где $x_i \in \{0, 1\}$ — бинарная переменная наличия механики, а w_i — экспертный весовой коэффициент ($\sum w_i = 100$). Распределение весов в модели носит экспертно-эвристический характер и базируется на иерархии когнитивного воздействия.

Наибольший приоритет ($w_1 = 40$) присвоен механике визуализации абстракций (замещение текстовых переменных визуальными метафорами). Данное решение обосновано принципом мультимедиа Р. Майера, согласно которому двойное кодирование информации (визуальное и вербальное) наиболее эффективно расширяет пропускную способность рабочих каналов памяти, напрямую демпфируя внешнюю нагрузку. Вторую позицию по значимости занимает мгновенная обратная связь

($w_2 = 30$). Высокий коэффициент обусловлен сокращением «цикла ошибки», быстрая реакция среды предотвращает закрепление неверных ментальных схем, экономя когнитивный ресурс, который иначе был бы затрачен на переучивание. Меньший вес отведен нарративному контексту ($w_3 = 20$), так как сюжет, обеспечивая семантическую связность, требует дополнительных ресурсов внимания на обработку истории. Замыкают иерархию классические PBL-элементы (бейджи, рейтинги) с минимальным весом ($w_4 = 10$). В рамках когнитивной модели их вклад в снижение нагрузки незначителен, поскольку они воздействуют преимущественно на дофаминовую систему вознаграждения (мотивацию), не упрощая саму структуру алгоритмической задачи.

Снижение когнитивной нагрузки является необходимым условием для обучения, однако оно описывает лишь потенциальную возможность усвоения материала, но не гарантирует инициацию учебного действия. Для преодоления разрыва между «могу» и «делаю» в структуру исследования интегрирована поведенческая модель Б.Дж. Фогга ($B = M \cdot A \cdot T$) [6]. В рамках предложенной концепции геймификация выступает не просто как внешний стимул, а как системный модулятор, трансформирующий переменные Способности (А) и Триггера (Т).

Во-первых, происходит функциональное переопределение Способности (А). Опираясь на фундаментальное определение Б.Дж. Фогга, мы рассматриваем данный параметр не как статический уровень навыков, а как функцию от ситуативной простоты выполнения задачи. Фогг выделяет «умственные усилия» как ключевой ограничитель простоты: если задача требует высокой когнитивной нагрузки, способность к её выполнению стремится к нулю. Применяя выведенную нами формулу (2), мы получаем возможность управления этим параметром. Геймификация, снижая визуальный шум и обеспечивая когнитивную опору, минимизирует затраты «умственных усилий» на декодирование интерфейса. Математически это выражается в том, что снижение знаменателя в формуле нагрузки ($L_{ext} \downarrow$) эквивалентно повышению показателя Способности ($A \uparrow$). Таким образом, задача смещается из зоны «Когнитивной блокировки» в зону «Выполнимости».

Во-вторых, осуществляется качественная трансформация Триггера (Т). В традиционной педагогической парадигме инициация действия опирается на внутренние триггеры - волевой ресурс ученика, который у подростков зачастую является дефицитным. Геймифицированная среда замещает необходимость волевого усилия системой внешних сигнальных триггеров. Нарративные события (например, необходимость совершить действие для продвижения сюжета) выступают естественными побудителями, встроенными в поток деятельности.

Таким образом, достигается синергетический эффект: снижение когнитивного барьера увеличивает субъективную Способность (А), а игровая механика обеспечивает своевременный Триггер (Т). Это гарантирует, что даже при умеренном уровне мотивации (М) произведение факторов $M \cdot A \cdot T$ превысит порог активации, что приведет к устойчивому целевому поведению (В) — написанию программного кода.

Основные результаты

Для оценки прогностической силы теоретической модели (2) и границ применимости метода аддитивной свертки нами был проведен структурный анализ образовательной среды CodeCombat (а также проанализированы схожие механики платформ, таких как CheckiO [7]) — платформы, обучающей текстовому программированию (Python/JavaScript) через механики RPG. В качестве объекта сравнения рассматривалась стандартная среда разработки (Python IDLE/PyCharm), а единицей анализа выступил когнитивный паттерн освоения алгоритмической конструкции «Цикл с параметром».

В контрольном сценарии, характеризующемся стремлением индекса геймификации к нулю ($G \rightarrow 0$), обучающийся взаимодействует с абстрактным кодом. Согласно положениям теории когнитивной нагрузки, в данной точке внешняя нагрузка (L_{ext}) достигает критических значений. Важно отметить, что современные профессиональные IDE, хотя и обладают функцией автодополнения, для новичка часто выступают источником «когнитивного шума». Предлагая полный список всех технических методов объекта (так называемое «техническое автодополнение»), среда создает «парадокс выбора», увеличивая нагрузку на фильтрацию информации. В сочетании с отсутствием визуальной обратной связи это блокирует активацию драйверов поведенческой модели Фогга: параметр Способности (А) падает, а инициация действия полностью зависит от дефицитного волевого ресурса (внутреннего триггера). Диаметрально противоположная архитектура наблюдается в экспериментальной среде CodeCombat, где реализован принцип изоморфности кода и игрового действия. Для количественной оценки эффективности среды был

произведен компонентный аудит на основе разработанной методики взвешенной оценки. Анализ выявил полную реализацию всех ключевых механик. Так, прямая визуализация абстракций ($w_1 = 40$) демонстрируется через буквальное исполнение команд: написание метода `hero.moveRight()` не просто изменяет координаты в памяти, а инициирует анимацию шага персонажа. Мгновенная обратная связь ($w_2 = 30$) обеспечивается физикой мира: при логической ошибке герой немедленно упирается в стену или получает урон, что позволяет диагностировать проблему без чтения логов. Удержание внимания гарантируется нарративным контекстом ($w_3 = 20$) сюжетной кампании (например, «Побег из Подземелья») и системой прогресса ($w_4 = 10$) через сбор кристаллов и апгрейд экипировки. Суммирование весовых коэффициентов этих элементов позволяет зафиксировать максимальный интегральный индекс $G = 100$. Подстановка полученного значения в базовую математическую модель (2), при допущении, что платформа-лидер рынка обладает эталонным качеством педагогического дизайна ($k \approx 1$), позволяет вывести прогнозный коэффициент редукции нагрузки:

$$K_{\text{reduction}} \approx \frac{L_{\text{base}}}{L_{\text{ext}}(100)} \approx k \cdot \ln(100 + e) \approx 4.6 \quad (4)$$

Данный расчет теоретически обосновывает, что интерфейс геймифицированной среды способен снизить давление на рабочую память в 4.6 раза по сравнению с традиционной консолью. Важную роль в этом играет механизм «педагогического автодополнения», который, в отличие от стандартных IDE, фильтрует подсказки, оставляя только контекстно-релевантные команды, тем самым устраняя парадокс выбора и выполняя функцию скаффолдинга.

Для подтверждения корректности полученного теоретического прогноза ($K_{\text{reduction}} = 4.6$) нами определены ключевые направления эмпирической валидации. В качестве основного индикатора предлагается использовать Коэффициент синтаксических ошибок (SER). Если фундаментальный мета-анализ Brown & Altmire (2014) фиксировал частоту ошибок на уровне 15–20% для Java, то современные данные по Python (Kohn, 2019) подтверждают, что структурные ошибки остаются доминирующей причиной когнитивного отказа. На основе нашей модели для среды CodeCombat прогнозируется снижение уровня SER до значений менее 5%. Такая динамика объясняется эффектом контекстного скаффолдинга: интерфейс берет на себя рутинные операции, высвобождая ресурс учащегося для семантической работы. Для повышения достоверности выводов данный объективный показатель рекомендуется сопоставлять с физиологическими данными айтрекинга (сокращение времени зрительной фиксации, изменение паттернов саккад), которые современные глубокие нейросетевые модели используют для точной классификации когнитивной нагрузки и уровня навыков начинающих программистов [8], [9].

Совокупный анализ подтверждает, что геймификация трансформирует структуру учебного поведения согласно модели Фогга ($B = M \cdot A \cdot T$) [6]. Снижение когнитивного барьера повышает параметр Способности (A), а изменение природы Триггера (T) с внутренней на внешнюю (императивные игровые сигналы, такие как приближение врага) гарантирует устойчивый запуск учебной деятельности без значительных затрат волевого ресурса обучающегося.

Обсуждение

Несмотря на продемонстрированный в Разделе 3 высокий прогнозный потенциал геймификации ($K_{\text{reduction}} \approx 4.6$), предложенная модель не должна восприниматься как универсальное решение. Глубокий анализ механизмов когнитивной разгрузки вскрывает ряд существенных педагогических рисков, требующих критического осмысления.

В первую очередь это касается инструмента интеллектуального автодополнения. Хотя критики могут возразить, что данная функция является стандартом и для профессиональных IDE, между ними существует фундаментальный барьер. Профессиональные среды предоставляют «техническое автодополнение», выгружая полный список методов и создавая когнитивный шум. Геймифицированная среда, напротив, реализует «педагогическое автодополнение», фильтруя подсказки до узкого набора контекстно-релевантных команд. Это эффективно работает как временная опора, но одновременно порождает риск возникновения «иллюзии компетентности». Обучающийся привыкает работать в режиме узнавания команд из списка, что может затормозить формирование навыка их активного воспроизведения. Следовательно, при резком отключении игровых подсказок учащийся может испытать «синтаксический шок», что диктует необходимость стратегии постепенного угасания поддержки.

Кроме того, сама визуальная природа геймификации таит в себе опасность «эффекта соблазнительных деталей». В нашей математической модели (2) предполагается, что рост индекса геймификации (G) снижает нагрузку. Однако, как показывают исследования Р. Майера [10], при низком качестве педагогического дизайна ($k < 1$) чрезмерное насыщение интерфейса декоративными элементами начинает конкурировать за внимание обучающегося, повышая внешнюю нагрузку (L_{ext}) вместо её снижения. Геймификация оправдана лишь тогда, когда она строго функциональна, а игровые объекты служат прямыми метафорами программных сущностей.

Наконец, необходимо четко очертить границы применимости модели. Согласно «эффекту обратной экспертизы», методы визуальной поддержки, критически важные для этапа «Новичок», становятся избыточными для продвинутых учащихся. Для эксперта необходимость декодировать визуальные метафоры требует больших когнитивных затрат, чем чтение лаконичного текстового кода. Таким образом, предложенный метод аддитивной свертки валиден преимущественно для начального этапа формирования алгоритмического мышления и требует смены парадигмы по мере роста компетенций.

Заключение

Проведенное исследование позволило формализовать влияние инструментов геймификации на когнитивные процессы при обучении программированию и предложить количественный метод оценки образовательных сред.



Во-первых, разработанная математическая модель $L_{ext}(G)$, представляющая собой зависимость, обратно пропорциональную логарифмической функции, теоретически обосновывает, что качественная геймификация выполняет роль «когнитивного экзоскелета». Расчеты показывают, что при максимальном индексе геймификации ($G = 100$) давление на рабочую память снижается в 4.6 раза. Это подтверждает гипотезу о том, что визуальная инкапсуляция синтаксиса позволяет преодолеть барьер входа, критический для новичков.

Во-вторых, анализ кейса CodeCombat через призму поведенческой модели Фогга выявил механизм трансформации учебного поведения. Сочетание изоморфности кода (снижение требований к Способности) и замещения ненадежной внутренней мотивации императивными внешними сигналами (Триггеры, Т) создает устойчивый паттерн вовлеченности. Прогнозируемое снижение коэффициента синтаксических ошибок (SER) с 15–20% до значений менее 5% служит объективным маркером перехода обучающегося из зоны «Тревоги» в состояние «Потока» [11].

В-третьих, выявлено, что геймификация не является статичным решением. Для предотвращения «иллюзии компетентности» и эффекта обратной экспертизы инструменты поддержки (такие как педагогическое автодополнение) должны применяться динамически. Оптимальная стратегия предполагает постепенное «угасание» визуальной поддержки, обеспечивающее плавный трансфер навыков в профессиональные среды разработки.

Практическая значимость работы заключается в том, что предложенный метод аддитивной свертки может быть использован не только для аудита существующих платформ, но и как инженерная методика при проектировании образовательного ПО нового поколения, где баланс между игровой увлекательностью и когнитивной эффективностью рассчитывается математически, а не выбирается интуитивно.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Sweller J. Cognitive load during problem solving: Effects on learning / J. Sweller // Cognitive Science. — 1988. — Vol. 12. — № 2. — P. 257–285. — DOI: 10.1207/s15516709cog1202_4
2. Deterding S. The Lens of Intrinsic Skill Atoms: A Method for Gameful Design / S. Deterding // Human-Computer Interaction. — 2015. — Vol. 30. — № 3-4. — P. 294–335. — URL: <https://doi.org/10.1080/07370024.2014.993471> (accessed: 10.07.26). — DOI: 10.1080/07370024.2014.993471
3. Sailer M. The Gamification of Learning: A Meta-Analysis / M. Sailer, L. Homner // Educational Psychology Review. — 2020. — Vol. 32. — P. 77–112. — URL: <https://link.springer.com/article/10.1007/s10648-019-09498-w> (accessed: 10.07.26). — DOI: 10.1007/s10648-019-09498-w
4. Файзиева Д.Х. Обучение программированию с помощью геймификации / Д.Х. Файзиева, Б.Н. Тахиров, З.М. Адизова // Вестник науки и образования. — 2022. — № 1. — С. 45–48.
5. Быстрова Н.В. Геймификация в современном образовательном процессе / Н.В. Быстрова, Н.А. Бакулина, А.В. Гнездин и др. // Проблемы современного педагогического образования. — 2022. — № 76-3. — С. 45–47.
6. Fogg B.J. A behavior model for persuasive design / B.J. Fogg // Proceedings of the 4th International Conference on Persuasive Technology. — Claremont: ACM (Association for Computing Machinery), 2009. — P. 1–7. — URL: <https://dl.acm.org/doi/10.1145/1541948.1541999> (accessed: 10.07.26). — DOI: 10.1145/1541948.1541999
7. Образовательная платформа для изучения программирования // CheckIO. — 2023. — URL: <https://checkio.org> (дата обращения: 10.07.26)
8. Wibirama S. Classification of Cognitive Load Using Deep Learning Based on Eye Movement Indices / S. Wibirama, S.A.I. Alfarozi, P.I. Santosa // IEEE Access. — 2024. — Vol. 12. — URL: <https://ieeexplore.ieee.org/document/10534241> (accessed: 10.07.26). — DOI: 10.1109/ACCESS.2024.3401569
9. Ahsan Z. Spatial Temporal Eye-Tracking Indicators of Novice Programmers' Skill / Z. Ahsan, U. Obaidellah // IEEE Access. — 2024. — Vol. 12. — URL: <https://ieeexplore.ieee.org/document/10423067> (accessed: 10.07.26).
10. Mayer R.E. The Cambridge Handbook of Multimedia Learning / R.E. Mayer, L. Fiorella. — Cambridge: Cambridge University Press, 2021. — 550 p.
11. Csikszentmihalyi M. Flow: The Psychology of Optimal Experience / M. Csikszentmihalyi. — New York: Harper & Row, 1990. — 303 p.

Список литературы на английском языке / References in English

1. Sweller J. Cognitive load during problem solving: Effects on learning / J. Sweller // Cognitive Science. — 1988. — Vol. 12. — № 2. — P. 257–285. — DOI: 10.1207/s15516709cog1202_4
2. Deterding S. The Lens of Intrinsic Skill Atoms: A Method for Gameful Design / S. Deterding // Human-Computer Interaction. — 2015. — Vol. 30. — № 3-4. — P. 294–335. — URL: <https://doi.org/10.1080/07370024.2014.993471> (accessed: 10.07.26). — DOI: 10.1080/07370024.2014.993471



3. Sailer M. The Gamification of Learning: A Meta-Analysis / M. Sailer, L. Homner // *Educational Psychology Review*. — 2020. — Vol. 32. — P. 77–112. — URL: <https://link.springer.com/article/10.1007/s10648-019-09498-w> (accessed: 10.07.26). — DOI: 10.1007/s10648-019-09498-w
4. Fajzieva D.X. Obuchenie programmirovaniyu s pomoshh'yu gejmifikacii [Teaching programming using gamification] / D.X. Fajzieva, B.N. Taxiroy, Z.M. Adizova // *Bulletin of Science and Education*. — 2022. — № 1. — P. 45–48. [in Russian]
5. By'strova N.V. Gejmifikaciya v sovremennom obrazovatel'nom processe [Gamification in the modern educational process] / N.V. By'strova, N.A. Bakulina, A.V. Gnezdin et al. // *Problems of Modern Pedagogical Education*. — 2022. — № 76-3. — P. 45–47. [in Russian]
6. Fogg B.J. A behavior model for persuasive design / B.J. Fogg // *Proceedings of the 4th International Conference on Persuasive Technology*. — Claremont: ACM (Association for Computing Machinery), 2009. — P. 1–7. — URL: <https://dl.acm.org/doi/10.1145/1541948.1541999> (accessed: 10.07.26). — DOI: 10.1145/1541948.1541999
7. Obrazovatel'naya platforma dlya izucheniya programmirovaniya [Educational platform for learning programming] // CheckiO. — 2023. — URL: <https://checkio.org>. (accessed: 10.07.26) [in Russian]
8. Wibirama S. Classification of Cognitive Load Using Deep Learning Based on Eye Movement Indices / S. Wibirama, S.A.I. Alfarozi, P.I. Santosa // *IEEE Access*. — 2024. — Vol. 12. — URL: <https://ieeexplore.ieee.org/document/10534241> (accessed: 10.07.26). — DOI: 10.1109/ACCESS.2024.3401569
9. Ahsan Z. Spatial Temporal Eye-Tracking Indicators of Novice Programmers' Skill / Z. Ahsan, U. Obaidellah // *IEEE Access*. — 2024. — Vol. 12. — URL: <https://ieeexplore.ieee.org/document/10423067> (accessed: 10.07.26).
10. Mayer R.E. *The Cambridge Handbook of Multimedia Learning* / R.E. Mayer, L. Fiorella. — Cambridge: Cambridge University Press, 2021. — 550 p.
11. Csikszentmihalyi M. *Flow: The Psychology of Optimal Experience* / M. Csikszentmihalyi. — New York: Harper & Row, 1990. — 303 p.