

## Наука и инновации в информационном обществе

**ВИДЕОИГРА КАК ИНФОРМАЦИОННАЯ СИСТЕМА:  
НАУЧНЫЕ ОСНОВАНИЯ И КЛАССИФИКАЦИЯ ЗАДАЧ**

Статья рекомендована к публикации председателем редакционного совета Ю. Е. Хохловым 01.02.2026.

**Кугуракова Влада Владимировна**

*Кандидат технических наук, доцент  
Казанский (Приволжский) федеральный университет  
Институт информационных технологий и интеллектуальных систем  
Кафедра индустрии разработки видеоигр  
Заведующая кафедрой  
Казань, Татарстан, Российская Федерация  
vlada.kugurakova@gmail.com*

**Аннотация**

Видеоигра рассматривается как дискретная динамическая информационная система, определяемая кортежем формальных компонентов  $G = \langle S, A, \delta, R, P, \omega, W \rangle$ . Показано, что существующие определения игры, принятые в гуманитарных науках, не операциональны и не пригодны для постановки инженерных задач. На основе предложенной модели сформулированы шесть классов актуальных задач технической науки: формальная верификация механик, балансировка как многокритериальная оптимизация, моделирование поведения агентов, процедурная генерация контента, тестирование пространства состояний и формальная спецификация. Для каждого класса задач установлена связь с модельными компонентами и обозначены применимые формальные методы. Результаты направлены на формирование самостоятельного направления технической науки о видеоиграх в отечественном научном пространстве.

**Ключевые слова**

*видеоигра; информационная система; формализованная модель; игровая механика; верификация; балансировка; game studies*

**1 Введение**

Современное информационное общество характеризуется не только масштабной цифровизацией производственных и коммуникационных процессов, но и формированием принципиально новых форм цифрового взаимодействия и досуга. Видеоигры стали одним из наиболее значимых феноменов этой трансформации: по своей аудитории, экономическому весу и технологической сложности они превратились в полноценный сектор цифровой индустрии. Мировой рынок видеоигр превысил 200 млрд долларов США по итогам 2023 года и продолжает демонстрировать устойчивый рост, формируя масштабный запрос на инженерные методы проектирования, анализа и верификации игровых систем [1]. Несмотря на это, в отечественных технических и математических науках видеоигра как самостоятельный объект исследования практически не фигурирует: проведенный поиск по базе диссертаций Высшей аттестационной комиссии (ВАК) РФ [2] выявил работы, относящиеся преимущественно к педагогике, культурологии и психологии. Информационно-технические же аспекты — архитектура, формальные свойства, корректность и оптимальность механик — остаются за пределами систематического изучения.

Систематический поиск по базам данных ВАК, Российской государственной библиотеки (РГБ) [3] и одной из крупнейших российских электронных научных библиотек DissertCat [4], специализирующей на каталогизации диссертаций и авторефератов, позволил выделить три характерных периода. В 1990–2000-е годы слово «игра» в диссертациях по техническим и физико-

---

© Кугуракова В. В., 2026

Производство и хостинг журнала «Информационное общество» осуществляется Институтом развития информационного общества.

Данная статья распространяется на условиях международной лицензии Creative Commons «С указанием авторства - С сохранением условий версии 4.0 Международная» (Creative Commons Attribution – ShareAlike 4.0 International; CC BY-SA 4.0). См. <https://creativecommons.org/licenses/by-sa/4.0/legalcode.ru>

[https://doi.org/10.52605/16059921\\_2026\\_04\\_105](https://doi.org/10.52605/16059921_2026_04_105)

математическим наукам (шифры научных специальностей 05.13.xx) употреблялось исключительно в значении «деловая игра» или «автоматизированная обучающая система», — то есть игра трактовалась как педагогический инструмент, но не как самостоятельный объект исследования [5, 6]. Дополнительный поиск по физико-математическим специальностям (шифры 01.01.xx, 05.13.17) не выявил ни одной диссертации, в которой видеоигра выступала бы объектом исследования: работы по данным специальностям с игровой тематикой относятся исключительно к математической теории игр фон Неймана–Моргенштерна, а не к видеоиграм как программным системам. В период 2005–2016 годов термин «видеоигра» вошел в отечественную науку через гуманитарные направления: появились диссертации по философии, искусствоведению, психологии (преимущественно в аспекте игровой аддикции (игромании) — патологического расстройства поведения, характеризуемого, в частности, приоритетом виртуального мира над реальной жизнью) и педагогике — однако ни одна из них не рассматривала видеоигру как техническую или информационную систему [7, 8]. Наконец, за период 2016–2026 годов не было защищено ни одной диссертации — ни кандидатской, ни докторской, — в которой видеоигра выступала бы самостоятельным объектом исследования в информатике или программной инженерии (единичные исключения связаны с иммерсивными средами как смежной областью, см., например, [46]). Таким образом, технические и системные аспекты проектирования игровых систем — корректность механик, балансировка, верификация — по-прежнему остаются вне систематического научного рассмотрения (сводный перечень найденных диссертаций, связанных с обозначенной тематикой, представлен в Приложении 1).

За рубежом сложилось самостоятельное направление — *game studies*, которое носит преимущественно гуманитарный характер [9, 10]. Работы инженерно-технической и информационной направленности в области видеоигр существуют, но разрознены: они публикуются в журналах по искусственному интеллекту, программной инженерии и теории формальных методов, не образуя единой дисциплины с устоявшимися терминологией и постановками задач.

Анализ англо- и франкоязычных диссертационных баз данных (ProQuest [11], HAL [12], Thèses.fr [13], репозитории ведущих университетов) показал, что зарубежные PhD-исследования также носят разрозненный характер: наиболее разработаны задачи процедурной генерации контента и автоматизированного тестирования, тогда как формальная верификация игровых механик, бисимуляционная эквивалентность реализаций и предметно-ориентированные языки спецификации игровых систем в качестве самостоятельных диссертационных тем практически не представлены (см. Приложение 2).

Отдельного упоминания заслуживает активно развивающееся направление анализа игровых данных (*game analytics*): работы по предсказанию исходов матчей, рейтингованию игроков и анализу стратегий на основе методов машинного обучения (такие как, например, [51] или [52]) демонстрируют высокий интерес к количественному изучению видеоигр. Однако данное направление принципиально отличается от формальных методов: оно оперирует статистическими закономерностями в наблюдаемых данных, не давая верифицируемых гарантий корректности игровых механик или воспроизводимости поведения при переносе на другую платформу. Формальная верификация, балансировка с доказуемыми свойствами и бисимуляционная эквивалентность реализаций остаются практически незатронутыми «игровыми аналитиками» — именно это составляет нишу настоящей статьи.

Цель статьи — предложить формальное основание для описания видеоигры как информационной системы и сформулировать на этой основе актуальные задачи, которые могут стать предметом научных и прикладных исследований.

## 2 Существующие подходы к определению видеоигры

Попытки формализовать понятие «игра» предпринимались неоднократно. Классическое определение Кроуфорда [14] трактует игру как замкнутую формальную систему с конфликтом и безопасными последствиями. Сален и Циммерман [15] описали игру через систему правил, создающую искусственный конфликт с измеримым результатом. Юул [16] характеризует любую видеоигру как «полуреальную» в том смысле, что правила игры реальны и действуют в реальном мире, тогда как её нарративный мир — вымышлен («...video games are real in that they consist of real

rules with which players actually interact. However, when winning a game by slaying a dragon, the dragon is not a real dragon but a fictional one.»).

Перечисленные определения обладают общим недостатком: они описательны, не операциональны<sup>1</sup> и не пригодны для постановки задач формального анализа. Они не отвечают на вопросы: каков вычислимый критерий баланса? как формально описать корректность механики? как верифицировать достижимость целевого состояния?

Попытки дать конструктивное формализуемое определение предпринимались в теории игр в смысле фон Неймана–Моргенштерна, однако классическая теория игр [17] рассматривает только абстрактные стратегические взаимодействия рациональных агентов с неограниченными вычислительными возможностями и не предполагает дискретного тактового времени и вычислимых ограничений агентов, характерных для цифровой среды. Методы теории автоматов применялись в ряде работ для описания отдельных игровых компонентов (например, [18]), но без обобщающей модели, охватывающей все уровни игровой системы.

В работе [46] предложена формально-логическая модель описания иммерсивной среды, описывающая функцию фиксации внимания  $g(u, s, t) = \{0, 1\}$  пользователя  $u$  на конкретных координатах  $s$  виртуального пространства  $\varphi_q$  в момент времени  $t$ , образуя маршрут по графу последовательностей событий  $G$ . Данная модель подробно описывает пространство и фиксирует взаимодействие пользователя с объектом фокуса, но не отражает основные правила и цели иммерсивной среды.

Таким образом, для постановки самостоятельных исследовательских задач в области проектирования и анализа игровых систем необходимо формальное определение видеоигры, опирающееся на аппарат теории информационных систем и строгие научные подходы.

### 3 Видеоигра как дискретная динамическая информационная система

#### 3.1 Формальная модель

Определим видеоигру как управляемую дискретную динамическую систему. Обозначим её кортежем

$$G = \langle S, A, \delta, R, \Pi, \omega, W \rangle,$$

где компонентами кортежа являются

- $S$  — конечное или счётное множество состояний системы (состояний игрового мира), кодирующее всю информацию о текущем состоянии игрового мира. Каждое состояние  $s \in S$  представляет собой пару  $s = (s_w, t)$ , где  $s_w$  — вектор текущих значений всех характеристик всех объектов игрового мира (положение фигур на доске, очки здоровья персонажей, запасы ресурсов и т.п.),  $t \in \mathbb{N}$  — номер текущего дискретного игрового такта. Дискретность времени таким образом встроена в структуру состояния; функция перехода  $\delta$  неявно инкрементирует  $t$  при каждом шаге;
- $\Pi = \{\pi_1, \dots, \pi_n\}$  — конечное множество агентов  $\pi$  (игроков<sup>2</sup> и неигровых персонажей, NPC — Non-Player Characters);
- $A = \bigcup_{\pi \in \Pi} A_\pi$  — объединённое множество действий всех агентов, где  $A_\pi$  — действия, доступные агенту  $\pi$ ;
- $\delta: S \times A \rightarrow \Delta(S)$  — частичная функция перехода; для детерминированного случая принимает вид  $\delta: S \times A \rightarrow S$ . Частичность  $\delta$  означает, что действие  $a$  допустимо в состоянии  $s$  лишь тогда, когда пара  $(s, a)$  входит в область определения  $\delta$  — тем самым  $\text{dom}(\delta)$ <sup>3</sup> задаёт систему правил и ограничений игры. Функция  $\delta$  согласована с  $R$ :  $\delta(s, a)$  определена тогда и только тогда, когда  $\exists s': (s, a, s') \in R$ , и носитель распределения  $\delta(s, a)$  содержится в  $\{s' | (s, a, s') \in R\}$ . На уровне отдельных механик эти ограничения выражаются через предусловия  $\text{pre}_m(s)$ , определённые в §4:  $\delta(s, a)$  определена тогда и только тогда, когда выполнено предусловие соответствующей механики;

<sup>1</sup> Термин «операциональный» (от англ. operational) используется в философии науки [45] для обозначения понятий, определяемых через конкретные измеримые операции или вычислимые процедуры, а не через описательные характеристики. В данном контексте это определение, допускающее алгоритмическую проверку.

<sup>2</sup> Будем использовать  $\pi$  как мнемоническое сокращение от англ. player — игрок.

<sup>3</sup>  $\text{dom}(f)$  — область определения функции  $f$ , т.е. множество аргументов, для которых  $f$  определена:  $\text{dom}(f) = \{x | \exists y: f(x) = y\}$ .

- $R \subseteq S \times A \times S$  — отношение допустимых переходов, первичное по отношению к  $\delta$ : тройка  $(s, a, s') \in R$  означает, что переход из состояния  $s$  в состояние  $s'$  по действию  $a$  разрешён правилами игры.  $R$  задаёт декларативный уровень правил (определяет дизайнер), тогда как  $\delta$  — операциональный (реализует движок). В стохастическом случае из одной пары  $(s, a)$  возможны несколько допустимых исходов  $s'$ , все они перечислены в  $R$ ;
- $\omega : S \rightarrow \Omega$  — функция наблюдаемости, где  $\Omega$  — множество возможных представлений состояния, доступных игроку через интерфейс. Функция  $\omega$  отличает видеоигру от абстрактного автомата: игрок взаимодействует не с полным состоянием  $s \in S$ , а лишь с его проекцией  $\omega(s)$  — тем, что отображается на экране. В частном случае полной информации (шахматы, крестики-нолики)  $\omega = id$ . В общем случае («туман войны» в стратегиях (fog of war), скрытые карты соперника в покере, невидимые противники)  $\omega(s)$  существенно беднее  $s$ , что порождает задачу проектирования интерфейса как отображения неполной информации. В рамках настоящей статьи  $\omega$  фиксируется как структурный компонент модели и важна концептуально — она отличает видеоигру от абстрактного автомата, фиксируя что игрок наблюдает неполное состояние мира через интерфейс; её вычислительная роль раскрывается при постановке задач верификации и тестирования в §6;
- $W \subseteq S$  — множество терминальных состояний (условий победы, поражения или завершения сессии).

$\Delta(S)$  обозначает множество вероятностных распределений на  $S$ ; в стохастических игровых системах функция  $\delta$  принимает именно такую форму, что позволяет моделировать случайные события (броски кубика, процедурную генерацию, критические удары). Настоящая модель намеренно оставляет за скобками внутреннюю темпоральную и вероятностную структуру отдельных сущностей — временные метки  $\tau$  и вероятностные характеристики  $P$  на уровне атрибутов; их детальное развитие содержится в модели FAST-GM [43].

### 3.2 Игровая сессия

Дадим определение игровой сессии как конечной или бесконечной траектории в пространстве состояний:

$$\sigma = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \dots \xrightarrow{a_k} s_k,$$

где  $s_0 \in S$  — начальное состояние,  $a_1 \in A$  действие на шаге  $i$ ,  $s_i \in S$  — результирующее состояние, причём  $(s_{i-1}, a_i, s_i) \in R$  для всех  $i$ . Сессия является завершённой, если последнее состояние игровой сессии  $s_k \in W$ .

### 3.3 Связь со смежными формализмами

Предложенная модель является расширением ряда устоявшихся формализмов. Например, марковский процесс принятия решений (MDP) [19] соответствует частному случаю  $G$  при  $|P| = 1$  и вероятностной  $\delta$ . Конечный автомат соответствует детерминированной  $\delta$  и  $\omega = id$ . Многоагентная система (MAS) [20] воспроизводится при  $|P| > 1$  без ограничений на  $\omega$ .

Специфика видеоигры как информационной системы (далее — ИС) заключается в следующей совокупности факторов: нетривиальная функция наблюдаемости  $\omega$  создаёт задачу проектирования интерфейса как отображения неполной информации; явное множество  $W$  задаёт терминальную семантику сессии (условия её завершения); структура множества  $P$  описывает многоагентность с разнородными стратегиями. Ни один из формализмов, указанных выше, по отдельности не охватывает всех этих аспектов одновременно.

## 4 Игровые механики: формальное определение и классификация

Понятие игровой механики занимает центральное место в проектировании игровых систем, однако в литературе оно трактуется неоднозначно: от «правил взаимодействия» [21] до «атома геймплея» [22]. В рамках предложенной модели  $G$  механика получает точное формальное определение, допускающее строгий анализ.

### 4.1 Формальное определение игровой механики

Пусть  $a_m \in A$  — действие, с которым ассоциирована механика  $m$ . Определим игровую механику  $m$  как структурированную тройку

$$m = \langle pre_m, eff_m, cost_m \rangle,$$

где  $pre_m: S \rightarrow \{T, \perp\}$  — предусловие применимости<sup>4</sup>, т.е. предварительное условие, которое должно быть истинным до начала применения механики (например, условие охраны, guard condition<sup>5</sup> [47]): механика допустима в состоянии  $s$  тогда и только тогда, когда  $pre_m(s) = T$ .

Таким образом, область определения применения игровой механики:

$$dom(m) = \{s \in S | pre_m(s) = T\},$$

где

- $eff_m: S \rightarrow S$  — функция эффекта, выражающая постусловие (т.е. условие или состояние, которое должно выполняться после завершения действия или процесса): она отображает состояние до применения механики в состояние после применения в ответ на действие  $a_m \in A: s' = eff_m(s)$ ;
- $cost_m: S \rightarrow \mathbb{R} \geq 0$  — функция стоимости, задающая расход ресурсов (очки действий, запас магической энергии (маны<sup>6</sup>), время до следующего использования (откат)) при применении игровой механики в состоянии  $s$ .

Связь с моделью  $G$ :  $\delta(s, a_m) = eff_m(s)$  для всех  $s \in dom(m)$ , что обеспечивает согласованность операционального и декларативного уровней.

Игровая механика  $m$  применима в состоянии  $s$  тогда и только тогда, когда  $pre_m(s) = T$  и ресурсный бюджет агента покрывает  $cost_m(s)$ . Тройка  $\langle pre_m, eff_m, cost_m \rangle$  расширяет классический аппарат STRIPS<sup>7</sup> [23], адаптируя его к непрерывным числовым параметрам, характерным для игровых систем.

Пример. Игровая механика «атака ближнего боя» задаётся следующим образом:

$$pre_m(s) = (dist(\pi, target) \leq 1) \wedge (hp(target) > 0) \wedge (cooldown_m(s) = 0),$$

где  $\pi \in \Pi$  — атакующий агент,  $dist$  — расстояние между агентом и целью в клетках игрового поля,  $hp(target)$  — текущий запас очков здоровья цели,  $cooldown_m(s)$  — остаток тактов до следующего допустимого применения механики  $m$ . Таким образом, атака возможна, если агент находится вплотную к цели, цель ещё не побеждена, и механика не требует восстановления;  $eff_m(s)$  — состояние  $s'$ , в котором  $hp(target)$  уменьшено на величину урона  $dmg_\pi \in \mathbb{R} \geq 0$  атакующего агента  $\pi$ , а  $cooldown_m$  установлен в  $c \in \mathbb{N}$  тактов до следующего допустимого применения механики;  $cost_m(s) = ap_\pi(s)$ , где  $ap_\pi(s)$  — текущий запас очков действия агента  $\pi$  в состоянии  $s$ ; применение механики списывает одну единицу из этого бюджета, ограничивающего число операций агента за такт.

## 4.2 Классификация механик

По структуре предусловия и функции эффекта разделим игровые механики на следующие три класса (эта классификация основана на формальном аппарате STRIPS [23]; см. также [24]):

**Атомарные механики** — примитивные правила взаимодействия, предусловие и эффект которых задаются непосредственно и не разделяются на взаимодействия. Примеры: перемещение на клетку, нанесение урона, сбор ресурса, наложение временного эффекта (статуса)<sup>8</sup>. Атомарные

<sup>4</sup>  $T$  (истина, от лат. *verum*) и  $\perp$  (ложь, от лат. *falsum*) — стандартные обозначения булевых значений в математической логике и теории формальных методов;  $\{T, \perp\} = \{true, false\}$ .

<sup>5</sup> Термин «условие охраны» (guard condition, охраняемая команда) введён Дейкстрой [47] для описания логических условий, управляющих выполнением команд в недетерминированных программах: команда выполняется только при истинности своего охранного предиката. В контексте игровых механик роль охранного предиката играет предусловие  $pre_m(s)$  — механика применима тогда и только тогда, когда  $pre_m(s) = T$ .

<sup>6</sup> Здесь и далее используется устоявшаяся игровая терминология: мана (mana) — условный ресурс, расходуемый при активации способностей; откат (cooldown) — интервал недоступности механики после применения; статус (status effect) — временный эффект, изменяющий параметры персонажа или системы.

<sup>7</sup> STRIPS (Stanford Research Institute Problem Solver) — формальный язык описания задач планирования, предложенный Файксом и Нильссоном в 1971 году [23]. Задаёт предусловия и постусловия действий в виде множеств атомарных высказываний.

<sup>8</sup> Здесь и далее использована устоявшаяся игровая терминология: статус (status effect) — временный эффект, изменяющий параметры персонажа или системы; откат (cooldown) — интервал недоступности механики после применения; мана (mana) — условный ресурс, расходуемый при активации способностей.

механики образуют базисное множество  $M_0 \subset M$ , из которого конструируются более сложные механики  $M_C$  (от *composite*); полное множество механик игры  $M = M_0 \cup M_C$ .

**Составные механики** — механики, определяемые как упорядоченная композиция атомарных или других составных механик:  $m = m_n \circ \dots \circ m_1$ , где каждая  $m_i \in M_0 \cup M_C$ . Составные механики образуют множество  $M_C \subset M$ ; полное множество механик игры  $M = M_0 \cup M_C$ . Предусловие составной механики учитывает, что каждый следующий шаг применяется к уже изменённому состоянию:

$$pre_m(s) = pre_1(s) \wedge pre_2(ef f_1(s)) \wedge pre_3(ef f_2(ef f_1(s))) \wedge \dots$$

Примеры:

- «ход с взятием» = перемещение на клетку  $\circ$  удаление фигуры противника;
- «рокировка» = перемещение короля  $\circ$  перемещение ладьи (при выполнении совокупности предусловий обеих атомарных механик одновременно).

**Системные (эмерджентные) механики** — поведенческие паттерны, возникающие при совместном применении нескольких механик и не заданные явно ни одной из них [48]. Примеры: последовательности ударов с суммарным эффектом, превышающим сумму отдельных ударов («комбо-цепочки» в файтингах), нелинейное нарастание преимущества лидера («снежный ком» в стратегических играх), тактика удержания дистанции для атаки без получения ответного урона («кайтинг»<sup>9</sup>). Системные механики не определяются единственным оператором, но допускают описание через темпоральные свойства наблюдаемых траекторий, т.е. свойства, характеризующие их поведение, изменения или взаимосвязи во времени.

#### 4.3 Взаимодействие механик: композиция и эмерджентность

Введём три формальных отношения для описания взаимодействия игровых механик.

*Последовательная композиция.* Механики  $m_1$  и  $m_2$  применяются последовательно:

$$(m_2 \circ m_1)(s) = ef f_{m_2}(ef f_{m_1}(s)), \text{ при условии } pre_{m_1}(s) \wedge pre_{m_2}(ef f_{m_1}(s)).$$

*Конфликт механик.* Механики  $m_1$  и  $m_2$  конфликтуют в состоянии  $s$ , если обе применимы, однако применение одной нарушает предусловие другой:

$$conflict(m_1, m_2, s) \Leftrightarrow pre_{m_1}(s) \wedge pre_{m_2}(s) \wedge \neg pre_{m_2}(ef f_{m_1}(s)).$$

Выявление конфликтов игровых механик является частным случаем задачи верификации (раздел 6.1) и допускает автоматическую проверку при конечном  $S$ .

*Эмерджентность.* Темпоральное свойство траектории  $\varphi$  является эмерджентным [15] относительно набора механик  $M = \{m_1, \dots, m_k\}$ , если оно не выполняется ни на одной траектории, порождённой отдельной механикой  $m_i$ , но выполняется на траекториях, возникающих при совместном применении механик из  $M$ :

$$\exists \sigma \in \Sigma(M): \sigma \models \varphi, \text{ при этом } \forall i: \sigma_i \not\models \varphi,$$

где  $\Sigma(M)$  — множество траекторий, порождённых совместным применением механик из  $M$ , а  $\sigma_i$  — траектории, порождённые отдельной механикой  $m_i$ . Иными словами,  $\varphi$  это наблюдаемый эффект взаимодействия механик, который не предсказывается ни одной из них по отдельности.

Пример: Правила движения (суть игровые механики) шахматных фигур сами по отдельности не позволяют поставить мат, он возникает только при совместном применении нескольких механик на конкретной траектории и является эмерджентным свойством их взаимодействия. Шах представляет собой эмерджентное состояние, возникающее как следствие взаимодействия правил движения фигур и правила неприкосновенности короля. Король не может быть оставлен под ударом и не может быть взят как обычная фигура, поэтому возникновение угрозы ему порождает особый системный режим: игрок обязан на следующем ходу устранить эту угрозу. Если ни одно допустимое действие не позволяет этого сделать, система переходит в терминальное состояние мата, которое фиксирует проигрыш. Следовательно, шах является не самостоятельной механикой, а производным состоянием, возникающим на уровне целостной организации правил.

<sup>9</sup> Кайтинг (англ. kiting), комбо-цепочки и «снежный ком» — устойчивые термины игровой инженерии, обозначающие эмерджентные поведенческие паттерны, т.е. незапланированные конструктором системные взаимодействия механик.

Эмерджентность делает полное тестирование поведения игры как ИС принципиально сложным: даже при верифицированных атомарных механиках свойства их взаимодействия требуют самостоятельной проверки на уровне траекторий — именно здесь применяются методы верификации моделей, рассматриваемые в §6.

## 5 Таксономия компонентов игровой информационной системы

На основе модели  $G$  выделим четыре уровня абстракции компонентов игровой ИС.

*Механики реализуют элементарные допустимые переходы из  $R$* : это атомарные правила, задающие, как действие агента (игрока или NPC) изменяет состояние системы (атака, перемещение, сбор ресурса). Формально игровая механика  $m$  определена в §4.1 как тройка  $m = \langle pre_m, eff_m, cost_m \rangle$ ; на уровне таксономии существенно, что  $eff_m: S \rightarrow S$  частичная функция, определённая лишь при  $pre_m(s) = T$ , что соответствует атомарному допустимому переходу из  $R$ .

*Правила* — это ограничения на допустимые траектории  $\sigma$ : инварианты системы (формальное определение см. §6.1; пример: здоровье  $\geq 0$ ), темпоральные ограничения (действие доступно не более одного раза в ход), ресурсные зависимости.

Правила игры задаются на двух уровнях. На уровне переходов — отношением  $R \subseteq S \times A \times S$  (§3.1), определяющим допустимые тройки  $(s, a, s')$ . На уровне траекторий — формулами темпоральной логики LTL<sup>10</sup> [25] / CTL [26]: инварианты системы (здоровье  $\geq 0$  в каждом состоянии), темпоральные ограничения (действие доступно не более одного раза в ход), ресурсные зависимости.

Как различать механики и правила? Содержательно: механики активны — каждая механика  $m$  вызывается агентом действием  $a_m \in A$  и изменяет состояние через  $eff_m$ ; правила пассивны — они проверяются системой и фиксируют допустимые переходы в  $R$ , не требуя явного вызова агентом. Формально: механика  $m$  ассоциирована с конкретным действием  $a_m \in A$ , где  $dom(m) \subseteq S$ ,  $eff_m: S \rightarrow S$ ; правило есть тройка  $(s, a, s') \in R$ , фиксирующая допустимый переход безотносительно к конкретному вызывающему агенту.

*Нарратив* — концептуальное разграничение нарратива и игровых правил восходит к [16]: нарратив является интерпретационным слоем, не влияющим на функцию перехода  $\delta$ , и относится в архитектурном смысле к слою представления. Формализовав это разграничение в рамках модели  $G$ , представим нарратив как отображение  $N: \sigma \rightarrow \mathbb{S}$ , где  $\mathbb{S}$  множество возможных нарративных структур игровой сессии (последовательностей событий и эпизодов, выводимых из траектории).

*Интерфейс* есть реализация функции  $\omega$ : пользователь взаимодействует не непосредственно с состоянием  $s \in S$ , а с его наблюдаемой проекцией  $\omega(s) \in \Omega$ . Если  $\omega$  не является тождественным отображением (то есть игрок видит не всё состояние системы, а лишь его часть, а  $\omega(s)$  несёт меньше информации, чем полное состояние  $s$ ), возникает задача проектирования интерфейса: какую информацию показывать, в каком объёме и когда.

## 6 Актуальные задачи исследования видеоигр как информационных систем

На основе предложенной формальной модели выделим следующие классы задач, каждый из которых представляет самостоятельное направление исследований.

### 6.1 Формальная верификация игровых механик

*Формальная верификация* — автоматическая проверка соответствия модели системы заданной спецификации свойств корректности методами исчерпывающего анализа пространства состояний. Задача состоит в проверке того, что игровая система  $G$  удовлетворяет заданным свойствам корректности — то есть при любом допустимом действии в ходе игры не возникает нежелательных ситуаций. Задача состоит в автоматической проверке того, что игровая система  $G$  удовлетворяет заданным свойствам корректности, т.е. при любом допустимом действии в ходе игры не возникает нежелательных ситуаций. Выделим следующие типовые классы таких свойств:

<sup>10</sup> LTL (Linear Temporal Logic) — линейная темпоральная логика, описывающая свойства бесконечных трасс исполнения [25]. CTL (Computation Tree Logic) — ветвящаяся темпоральная логика для описания свойств деревьев вычислений [26]. Основные операторы:  $\Box$  (всегда),  $\Diamond$  (когда-либо),  $U$  (до тех пор, пока),  $X$  (следующий шаг).

- достижимость целевого состояния:  $\exists \sigma : s_0 \rightarrow^* s_1, s_1 \in W$  (игра завершима)<sup>11</sup>;
- отсутствие «дедлоков»<sup>12</sup>:  $\nexists s \in S \setminus W : A(s) = \emptyset$ , где  $A(s)$  множество действий, допустимых в  $s$ ;
- инвариантность:  $AG(p(s))$ , где  $p$  — инвариант состояния (например, «здоровье всегда неотрицательно»);
- темпоральные свойства траекторий, специфицируемые в LTL [25] или CTL [26].

Задача решается инструментами model checking<sup>13</sup> (такими как SPIN<sup>14</sup>, NuSMV<sup>15</sup>, UPPAAL<sup>16</sup>) [26] при условии, что множество состояний  $S$  конечно, т.е. перебор всех возможных ситуаций в игре завершим. При росте числа переменных состояния полный перебор становится практически невозможным (число состояний растёт экспоненциально). В этом случае в практике model checking применяют символьные методы (BDD<sup>17</sup>-based verification [27]) и абстракцию состояний — стандартные подходы в данной области.

## 6.2 Балансировка как задача многокритериальной оптимизации

Под балансировкой игровой модели  $G(\theta)$  понимаем настройку числовых параметров  $\theta \in \Theta \subseteq \mathbb{R}^n$  таким образом, чтобы игровой процесс удовлетворял желаемым характеристикам. Формально задача балансировки — это задача оптимизации:

$$\theta^* = \operatorname{argmin}_{\{\theta \in \Theta\}} F(G(\theta), \varphi),$$

где  $\theta^*$  — оптимальный вектор параметров,  $F: \Theta \times \Phi \rightarrow R$  — функция качества баланса, оценивающая степень соответствия игрового процесса при параметрах  $\theta$  желаемым свойствам  $\varphi$ ,  $\varphi \in \Phi$  — вектор целевых значений показателей (коэффициент побед, время до завершения сессии, эффективность использования ресурсов и др.),  $\Theta$  — допустимая область параметров.

Задача, как правило, многокритериальна:  $F = (f_1, \dots, f_m)$ , что переводит её в класс задач многокритериальной оптимизации с построением Парето-фронта<sup>18</sup> [28]. Сложность здесь обусловлена тем, что вычисление  $F$  требует многократного моделирования траекторий  $G(\theta)$ , а само моделирование может быть вычислительно затратным. Методы, применяемые для решения этой задачи: генетические алгоритмы, байесовская оптимизация, имитационное моделирование. Данная задача является центральной в ряде отечественных разработок, в частности, в инструментарию Tula [29, 30].

## 6.3. Моделирование поведения агентов

Введём стратегию агента  $\pi$  как отображение  $\beta_\pi: \Omega \rightarrow A_\pi$  — отображением наблюдения в действие. В стохастическом случае  $\beta_\pi: \Omega \rightarrow \Delta(A_\pi)$ . Задачи данного класса:

- автоматическое построение (синтез) стратегий поведения агентов, гарантированно удовлетворяющих заданным темпоральным или вероятностным условиям (спецификация в PCTL<sup>19</sup>, верификация в PRISM<sup>20</sup> [31]);

<sup>11</sup> Звёздочка в записи  $\rightarrow^*$  означает *транзитивное замыкание* отношения перехода, то есть «достижимо за ноль или более шагов». Формально:  $s_0 \rightarrow^* s_1$  означает, что существует траектория  $\sigma = s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_1$  произвольной длины.

<sup>12</sup> Дедлок (намертво заблокировано, от англ. Dead Lock) — ошибочно достижимое терминальное состояние игры, не предусмотренное разработчиком на траектории  $\sigma$ .

<sup>13</sup> Model checking (проверка на моделях) — метод автоматической верификации систем путём исчерпывающего обхода пространства состояний и проверки выполнимости темпоральных свойств [26].

<sup>14</sup> SPIN (Simple Promela INterpreter) — инструмент верификации параллельных систем на основе языка PROMELA. Проверяет свойства, заданные в LTL, методом явного обхода пространства состояний.

<sup>15</sup> NuSMV — символьный верификатор моделей, поддерживающий BDD-based и ограниченную верификацию (bounded model checking). Поддерживает свойства LTL и CTL.

<sup>16</sup> UPPAAL — инструмент верификации систем реального времени на основе сетей временных автоматов (timed automata). Проверяет темпоральные свойства с количественными временными ограничениями.

<sup>17</sup> BDD (Binary Decision Diagram) — бинарная диаграмма решений; компактное символьное представление булевых функций, используемое для эффективного обхода пространства состояний при model checking без явного перечисления всех состояний [27].

<sup>18</sup> В русскоязычной математической литературе по многокритериальной оптимизации наряду с термином «Парето-фронт» (от англ. Pareto front) используются эквивалентные обозначения: «множество Парето», «граница Парето», «граница эффективности» [28].

<sup>19</sup> PCTL (Probabilistic Computation Tree Logic) — вероятностная ветвящаяся темпоральная логика, расширяющая CTL количественными вероятностными операторами вида  $P \geq p[\varphi]$  («с вероятностью не менее  $p$  выполнится  $\varphi$ ») [42].

<sup>20</sup> PRISM (Probabilistic Symbolic Model Checker) — верификатор вероятностных и стохастических систем; поддерживает PCTL, CSL, LTL и цепи Маркова с непрерывным временем.

- верификация поведения NPC на соответствие спецификации:  $\beta\pi \models \varphi\pi$ ;
- обеспечение разнообразия поведения, формализуемого через метрику расстояния между траекториями:  $div(\sigma_1, \sigma_2) = d(\beta_\pi(\sigma_1), \beta_\pi(\sigma_2))$ , где  $\omega(\sigma) = (\omega(s_0), \omega(s_1), \dots)$  последовательность наблюдений вдоль траектории  $\sigma$ , а в качестве  $d$  может выступать редакционное расстояние (расстояние Левенштейна [49]) — минимальное число операций вставки, удаления и замены действий для преобразования  $\sigma_1$  в  $\sigma_2$ ; расстояние бисимуляции, согласованное с отношением  $\sim$  из §5; либо метрика Канторовича [50] на распределениях действий  $\beta_\pi(\omega(s))$  в стохастическом случае. Выбор конкретной метрики определяется жанром игры и задачей анализа.

Особую сложность представляет задача проектирования противника — неигрового персонажа (NPC), выступающего оппонентом игрока и управляемого алгоритмами искусственного интеллекта (ИИ)<sup>21</sup>. Противник должен сформировать вызов, соразмерный уровню навыка конкретного игрока: слишком простой противник снижает вовлечённость в игру, слишком сложный — вызывает фрустрацию (разочарование, тревогу, раздражение или апатию, возникающие из-за невозможности достичь поставленной цели). Такая ситуация формализуется как задача динамической адаптации сложности (DDA — Dynamic Difficulty Adjustment [32]): параметры противника корректируются в реальном времени на основе наблюдаемых метрик сессии, чтобы удерживать игрока в зоне оптимального потока [32].

#### 6.4. Процедурная генерация контента и тестирование пространства состояний

Здесь две задачи объединены в один класс в силу их принципиальной взаимозависимости: генерация порождает новые состояния и траектории, а тестирование оценивает степень их охвата и корректность.

Процедурную генерацию определим как алгоритмическое построение компонентов  $G$  — пространств уровней, наборов механик, нарративных структур — с сохранением заданных свойств корректности. Алгоритм процедурной генерации (PCG<sup>22</sup>)  $Gen: P \rightarrow G$  сопоставляет каждому параметрическому вектору  $\rho \in P$  игровую систему  $G' = Gen(\rho)$ . Задача PCG состоит в нахождении  $\rho^* \in P$  такого, что  $Gen(\rho^*) \models \psi$ , где  $\psi$  — формула, задающая требования к корректности и играбельности порождённого контента (достижимость победного состояния, отсутствие тупиков, баланс ресурсов и т.п.). Ключевая проблема — обеспечение верифицируемости генерируемых состояний: не всякий автоматически сгенерированный контент допустим с точки зрения модели  $G$  [33]. Генерация должна сопровождаться проверкой формальных свойств или гарантировать их конструктивно (constraint-based generation, grammar-based generation) [34]. Перспективное направление исследований в этом направлении — применение больших языковых моделей (LLM<sup>23</sup> — Large Language Models) для генерации нарративных структур (квестов, диалогов) с последующей верификацией порождаемых траекторий [35].

Тестирование игровых ИС осложняется взрывным ростом пространства состояний  $|S|$  и вероятностным поведением агентов, что делает полное перечисление траекторий практически неосуществимым. Задача формулируется как достижение заданного уровня покрытия пространства состояний:  $cov_k(\sigma_1, \dots, \sigma_m) \geq \alpha$ , где  $\alpha$  — минимально требуемая доля охваченных состояний или переходов,  $m$  — число запусков игровой симуляции (тестовых прогонов),  $\sigma_i$  — траектории, полученные при каждом запуске. Критерии покрытия [36]: покрытие переходов (T-coverage), покрытие пар «состояние–действие», покрытие сценариев. Перечисленные критерии и методы покрытия применяются в тестировании программных систем в целом [36] и адаптируются для игровых ИС с учётом их специфики: случайное тестирование, тестирование на основе свойств

<sup>21</sup> ИИ (искусственный интеллект; англ. AI — Artificial Intelligence) — в контексте игровых систем: совокупность алгоритмов, управляющих поведением NPC, адаптирующих сложность игрового процесса и генерирующих контент.

<sup>22</sup> PCG (Procedural Content Generation) — процедурная генерация контента; автоматическое создание игрового содержимого (уровней, персонажей, квестов, нарративных структур) с помощью алгоритмов без ручной разработки каждого элемента [34].

<sup>23</sup> LLM (Large Language Models — большие языковые модели) — нейросетевые модели, обученные на больших текстовых корпусах. В контексте игровых ИС применяются для генерации нарративных структур, диалогов и описаний с последующей формальной верификацией согласованности порождённого контента с моделью  $G$ .

(property-based testing), «фаззинг»<sup>24</sup> последовательностей действий, поиск по дереву игры (MCTS<sup>25</sup>) [37].

Связь между подзадачами двунаправленная: генерация расширяет покрываемое пространство состояний, тестирование выявляет неохваченные области и состояния, нарушающие свойства корректности модели  $G$ , направляя последующую генерацию.

### 6.5. Бисимуляционная эквивалентность реализаций при переносе между платформами

При переносе игры как информационной системы на другие платформы (портировании) или смене движка реализация игровой системы изменяется, тогда как игровая модель  $G$  должна оставаться инвариантной [38]. Поэтому возникает задача верификации того, что две реализации  $G_1$  и  $G_2$  являются наблюдаемо-эквивалентными, т.е. ни игрок, ни внешний наблюдатель не могут их различить по порождаемым траекториям. Такое отношение эквивалентности называется бисимуляцией: две системы бисимильны, если каждый шаг одной воспроизводится другой с точностью до наблюдаемого результата [39].

Формально,  $G_1$  и  $G_2$  бисимильны ( $G_1 \sim G_2$ ), если существует такое отношение бисимуляции  $B \subseteq S_1 \times S_2$ , что для любой пары  $(s_1, s_2) \in B$  и любого действия  $a \in A$ :

- если  $\delta_1(s_1, a) = s'_1$ , то  $\exists s'_2 : \delta_2(s_2, a) = s'_2$  и  $(s'_1, s'_2) \in B$ ;
- то же самое выполняется в обратную сторону: любой переход  $G_2$  воспроизводится в  $G_1$  (симметричное условие для  $G_2$ ).

Данная задача решается методами бисимуляционной верификации, реализованными в специализированных инструментах, в частности, в CADP<sup>26</sup> [39]. Практическая значимость определяется тем, что портирование игр является широко распространённой практикой разработки, однако формальные гарантии сохранения игрового поведения при таком переносе в игровой индустрии на сегодняшний день отсутствуют.

### 6.6. Формальная спецификация на языках описания игровых систем

Проектирование игровой ИС требует языков спецификации, поддерживающих описание механик, темпоральных свойств и вероятностных аспектов в единой нотации. Существующие общецелевые языки (TLA+<sup>27</sup> [40], Alloy [41]) пригодны для решения части задач, однако не учитывают специфику игровых объектов: агентов с ограниченной наблюдаемостью, игрового времени как дискретного параметра, целевой семантики сессии через  $W$ .

Перспективным направлением является разработка предметно-ориентированных языков спецификации игровых систем, интегрирующих операторы темпоральной (LTL [25] / TCTL) и вероятностной (PCTL [42]) логик. Такие языки позволяют задавать не только структуру механик, но и формальные критерии их корректности, непосредственно пригодные для верификации.

Примером отечественной реализации подходов к решению задач, названных выше, служат отечественные платформа Tula и язык FAST-GM [29, 43], разработанные на базе Казанского федерального университета. Платформа Tula включает визуальный редактор игровых механик, модуль балансировки на основе генетических алгоритмов и подсистему формальной верификации, осуществляющую трансляцию спецификаций в промышленные верификаторы. Таким образом, здесь Задача 1 (верификация механик), Задача 2 (балансировка) и Задача 6 (формальная спецификация) реализованы в виде единого инструментария.

Конечно, перечисленными классами круг исследовательских задач в области игровых ИС не исчерпывается. В число задач, требующих отдельного исследования, можно включить: обеспечение консистентности состояния в многопользовательских системах (синхронизация распределённых

<sup>24</sup> Фаззинг (fuzzing) — метод тестирования, при котором система подвергается воздействию случайных, в том числе некорректных или непредвиденных входных данных с целью выявления сбоев, исключений и нарушений инвариантов. Применительно к игровым ИС фаззинг генерирует случайные последовательности действий агентов для обнаружения состояний, нарушающих свойства корректности модели  $G$ .

<sup>25</sup> MCTS (Monte Carlo Tree Search) — поиск по дереву Монте-Карло) — алгоритм эвристического поиска в пространстве состояний, использующий случайное моделирование (симуляцию до терминального состояния) для оценки узлов; широко применяется в игровом ИИ и автоматическом тестировании игровых систем.

<sup>26</sup> CADP (Construction and Analysis of Distributed Processes) — инструмент для формальной верификации распределённых систем на основе исчисления CCS/LOTOS; включает поддержку бисимуляционной проверки.

<sup>27</sup> TLA+ (Temporal Logic of Actions) — язык формальной спецификации параллельных и распределённых систем Лэмпорта; спецификации проверяются инструментом TLC model checker.

копий  $S$  при наличии сетевых задержек); обнаружение аномального поведения — «читерства»<sup>28</sup> — как верификация нарушений инвариантов  $R$  в наблюдаемых траекториях; формальное определение показателей качества игрового опыта (engagement, flow) в терминах модели  $G$ ; верификация нарратива как проверка согласованности отображения  $N: \sigma \rightarrow S$  с правилами  $R$ ; верификация эквивалентности патчей<sup>29</sup> как частный случай бисимуляции  $G$  до и после обновления. Перечисленное не претендует на полноту — область находится в стадии формирования, и постановка новых задач в терминах модели  $G$  представляет самостоятельный научный интерес.

## 7 Заключение

Предложена формализованная модель видеоигры как дискретной динамической информационной системы, определяемой кортежем  $G = \langle S, A, \delta, R, \Pi, \omega, W \rangle$ . Показано, что эта модель обобщает ряд устоявшихся формализмов (MDP, конечный автомат, многоагентная система), фиксируя специфику игровой ИС: функцию наблюдаемости, целевую семантику сессии и многоагентность.

На основе модели сформулированы шесть классов актуальных задач: формализованная верификация механик, балансировка как многокритериальная оптимизация, моделирование поведения агентов, процедурная генерация контента и тестирование пространства состояний, бисимуляционная эквивалентность реализаций при переносе между платформами, а также формальная спецификация. Каждая из этих задач имеет аналоги в смежных областях, однако специфика игровых систем требует достаточно сложной адаптации и развития существующих методов.

Авторы рассматривают настоящую статью как основу для формирования в отечественном научном пространстве самостоятельного исследовательского направления — формальных методов анализа и проектирования видеоигр. Приоритетами дальнейших исследований представляются: разработка языков формальной спецификации, ориентированных на игровые системы; создание инструментария верификации игровых механик; формализация задачи балансировки в классе многокритериальных задач с симуляционными ограничениями; разработка методов верификации бисимуляционной эквивалентности игровых реализаций при кроссплатформенном переносе.

Сформулированные задачи в целом соответствуют паспорту научной специальности 2.3.8 «Информатика и информационные процессы» (технические науки, группа 2.3) [44]. Задачи балансировки (Задача 2), процедурной генерации (Задача 4) и формальной спецификации (Задача 6) прямо отвечают следующим направлениям паспорта этой специальности: «разработка компьютерных методов и моделей описания, оценки и оптимизации информационных процессов», «автоматизированные информационные системы по областям применения», «системы проектирования объектов и процессов». Задача моделирования поведения агентов (Задача 3) соотносится с направлением «системы принятия решений». Вместе с тем Задача 1 (формальная верификация механик) и Задача 5 (бисимуляционная эквивалентность) тяготеют и к специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем», паспорт которой явно включает «верификацию и тестирование программных систем». Такие теоретические аспекты работ, как темпоральные логики, теория автоматов и марковские процессы, ближе к специальности 1.2.3 «Теоретическая информатика, кибернетика». Таким образом, научные исследования в области видеоигр носят междисциплинарный характер и могут быть реализованы в диссертационных исследованиях по научным специальностям 2.3.8, 2.3.5 и 1.2.3 в зависимости от акцентов каждой конкретной работы.

## Благодарности

Работа выполнена за счёт гранта, предоставленного Академией наук Республики Татарстан образовательным организациям высшего образования, научным и иным организациям на поддержку планов развития кадрового потенциала в части стимулирования их научных и научно-

<sup>28</sup> Читерство (от англ. cheating) — использование игроком запрещённых приёмов, программного обеспечения или эксплуатация незапланированных уязвимостей для получения нечестного преимущества. Формально: нарушение инвариантов отношения допустимых переходов  $R$  в наблюдаемых траекториях сессии.

<sup>29</sup> Патч (от англ. patch — заплатка) — обновление программного обеспечения, вносящее изменения в параметры, механики или правила игровой системы. В терминах модели  $G$  патч представляет собой замену одного или нескольких компонентов кортежа  $\langle S, A, \delta, R, \Pi, \omega, W \rangle$ , как правило  $\delta$  и  $R$ , при сохранении остальных.

педагогических работников к защите докторских диссертаций и выполнению научно-исследовательских работ (Соглашение от 22.12.2025 № 12/2025-ПД-КФУ).

## Литература

1. Newzoo Global Games Market Report 2023 [Электронный ресурс]. URL: <https://newzoo.com> (дата обращения: 15.11.2024).
2. Объявления о защитах диссертаций [Электронный ресурс] // Высшая аттестационная комиссия при Министерстве науки и высшего образования Российской Федерации. URL: [https://vak.minobrnauki.gov.ru/adverts\\_list](https://vak.minobrnauki.gov.ru/adverts_list) (дата обращения: 01.04.2026).
3. Электронная библиотека диссертаций Российской государственной библиотеки [Электронный ресурс]. URL: <https://dvs.rsl.ru> (дата обращения: 01.04.2026).
4. Dissercat: электронная библиотека диссертаций [Электронный ресурс]. URL: <https://www.dissercat.com> (дата обращения: 01.04.2026).
5. Маркова В. В. Разработка и исследование моделирующих автоматизированных деловых игр на основе анализа проблемно-ориентированных программных машин: дис. ... канд. техн. наук: 05.13.18. Рязань, 2001. 170 с.
6. Гучапшева А.Х. Автоматизация проектирования программных игровых обучающих систем на основе алгебраического анализа: дис. ... канд. техн. наук: 05.13.12. [Б. м.], 2004.
7. Гутман И. Е. Компьютерные виртуальные игры: культурно-антропологические аспекты анализа: дис. ... канд. филос. наук: 09.00.13. СПб., 2009. 192 с.
8. Рыбалтович Д. Г. Психологические особенности пользователей онлайн-игр с различной степенью игровой аддикции: дис. ... канд. психол. наук: 05.26.02. СПб., 2012. 172 с.
9. Aarseth E. Computer Game Studies, Year One // Game Studies. 2001. Т. 1. № 1.
10. Frasca G. Ludology meets Narratology: Similitude and Differences between (Video) Games and Narrative // Parnasso. 1999. Т. 3. С. 365–371.
11. ProQuest Dissertations & Theses Global [Электронный ресурс]. URL: <https://www.proquest.com/products-services/pqdtglobal.html> (дата обращения: 01.04.2026).
12. HAL: archive ouverte pluridisciplinaire [Электронный ресурс]. URL: <https://hal.science> (дата обращения: 01.04.2026).
13. Thèses.fr: le portail des thèses françaises [Электронный ресурс]. URL: <https://www.theses.fr> (дата обращения: 01.04.2026).
14. Crawford C. The Art of Computer Game Design. Berkeley: Osborne/McGraw-Hill, 1984. 110 с.
15. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. Cambridge: MIT Press, 2004. 688 с.
16. Juul J. Half-Real: Video Games between Real Rules and Fictional Worlds. Cambridge: MIT Press, 2005. 233 с.
17. von Neumann J., Morgenstern O. Theory of Games and Economic Behavior. Princeton: Princeton University Press, 1944. 625 с.
18. de Alfaro L., Henzinger T. A. Interface Automata // Proceedings of FSE. 2001. С. 109–120.
19. Puterman M. L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York: Wiley, 1994. 649 с.
20. Shoham Y., Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge: Cambridge University Press, 2009. 532 с.
21. Hunicke R., LeBlanc M., Zubek R. MDA: A Formal Approach to Game Design and Game Research // Proceedings of AAAI Workshop on Challenges in Game AI. 2004.
22. Sicart M. Defining Game Mechanics // Game Studies. 2008. Т. 8. № 2.
23. Fikes R., Nilsson N. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving // Artificial Intelligence. 1971. Т. 2. С. 189–208.
24. Шубин А. В., Кугуракова В. В. Проектирование инструментария для создания игрового процесса через систематизацию игровых механик // Электронные библиотеки. 2024. Т. 27, № 5. С. 774–795.
25. Pnueli A. The Temporal Logic of Programs // Proceedings of FOCS. 1977. С. 46–57.
26. Clarke E. M., Grumberg O., Peled D. Model Checking. Cambridge: MIT Press, 1999. 314 с.
27. Bryant R. E. Graph-based algorithms for boolean function manipulation // IEEE Transactions on Computers. 1986. Т. 35. № 8. С. 677–691.
28. Ehrgott M. Multicriteria Optimization. 2nd ed. Berlin: Springer, 2005. 323 с.

29. Рахманкулова В. Р., Кугуракова В. В. Онлайн-инструмент Tula для балансировки видеоигр // Электронные библиотеки. 2025. Т. 28, № 4. С. 903–930.
30. Сахибгареева Г. Ф., Кугуракова В. В. Практики балансирования компьютерных игр // Программные системы: теория и приложения. 2022. Т. 13. № 3(54). С. 255–273.
31. Kwiatkowska M., Norman G., Parker D. PRISM 4.0: Verification of Probabilistic Real-Time Systems // Lecture Notes in Computer Science. 2011. Т. 6806. С. 585–591.
32. Hunicke R. The Case for Dynamic Difficulty Adjustment in Games // Proceedings of ACM CHI Workshop on Designing for User Experiences. 2004.
33. Большаков Э. С., Кугуракова В. В. Генеративная симуляция игрового окружения в реальном времени // Электронные библиотеки. 2025. Т. 28, № 2. С. 188–212.
34. Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Springer, 2016. 244 с.
35. Трофимчук В. Т., Кугуракова В. В. Разработка адаптивной системы генерации игровых квестов и диалогов на основе больших языковых моделей // Электронные библиотеки. 2025. Т. 28, № 4. С. 953–993.
36. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken: Wiley, 2012. 240 с.
37. Browne C. et al. A Survey of Monte Carlo Tree Search Methods // IEEE Transactions on Computational Intelligence and AI in Games. 2012. Т. 4. № 1. С. 1–43.
38. Бондарь А. О., Кугуракова В. В. Проблемы, решения и перспективы автоматизированного переноса игровых сцен между игровыми движками // Электронные библиотеки. 2025. Т. 28. № 2. С. 213–243.
39. Garavel H. et al. CADP 2011: A Toolbox for the Construction and Analysis of Distributed Processes // Lecture Notes in Computer Science. 2013. Т. 7795. С. 233–240.
40. Lamport L. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Boston: Addison-Wesley, 2002. 364 с.
41. Jackson D. Software Abstractions: Logic, Language, and Analysis. Cambridge: MIT Press, 2006. 344 с.
42. Hansson H., Jonsson B. A Logic for Reasoning about Time and Reliability // Formal Aspects of Computing. 1994. Т. 6. № 5. С. 512–535.
43. Kugurakova V. V. A Formal Approach to Spatio-Temporal Modeling of Game Systems // Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki. 2024. Т. 166. № 4. С. 532–554.
44. Паспорт научной специальности 2.3.8 «Информатика и информационные процессы» [Электронный ресурс]: утв. приказом Минобрнауки России от 24.02.2021 № 118; в ред. от 24.07.2023 № 730. URL: <https://vak.minobrnauki.gov.ru> (дата обращения: 01.04.2026).
45. Bridgman P. W. The Logic of Modern Physics. New York: Macmillan, 1927. 228 с.
46. Александрова М. В. Управление глубиной погружения в иммерсивных интерактивных средах: автореф. дис. ... канд. техн. наук: 2.3.1 / Александрова Маргарита Владимировна. Самара, 2024. 23 с.
47. Dijkstra E. W. Guarded Commands, Nondeterminacy and Formal Derivation of Programs // Communications of the ACM. 1975. Т. 18. № 8. С. 453–457.
48. Fabricatore C. Gameplay and Game Mechanics Design: A Key to Quality in Videogames // Proceedings of the OECD-ENLACES Expert Meeting on Videogames and Education. Santiago de Chile, 2007. URL: <https://eprints.hud.ac.uk/id/eprint/20927/> (дата обращения: 01.04.2026).
49. Levenshtein V. I. Binary codes capable of correcting deletions, insertions, and reversals // Soviet Physics Doklady. 1966. Т. 10. № 8. С. 707–710.
50. Канторович Л. В. О перемещении масс // Доклады АН СССР. 1942. Т. 37. № 7–8. С. 227–229.
51. Makarov I., Savostyanov D., Litvyakov B., Ignatov D. I. Predicting Winning Team and Probabilistic Ratings in "Dota 2" and "Counter-Strike: Global Offensive" Video Games // Analysis of Images, Social Networks and Texts (AIST 2017). — Cham : Springer, 2017. — P. 183–196.
52. Semenov A., Romov P., Korolev S., Yashkov D., Neklyudov K. Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2 // Analysis of Images, Social Networks and Texts (AIST 2016). — Cham : Springer, 2016. — P. 26–37.

## ПРИЛОЖЕНИЕ 1. Диссертационные исследования, связанные с видеоиграми и игровыми системами, защищённые в Российской Федерации (1990–2026)

Таблица составлена на основе баз ВАК, РГБ, DisserCat и tekhnosfera.com. Звёздочкой (\*) отмечены работы, в которых видеоигра или игровая система выступает непосредственным объектом технического исследования; двойной звёздочкой (\*\*) — работы, связанные с игровыми механиками в контексте образовательных или тренажёрных систем. Работы без пометки относятся к гуманитарным, экономическим или смежным направлениям.

|    | Автор             | Название                                                                                                                                             | Ученая степень | Специальность | Год  | Тематика |
|----|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|------|----------|
| 1  | Маркова В.В.      | Разработка и исследование моделирующих автоматизированных деловых игр на основе анализа проблемно-ориентированных программных машин                  | к.т.н.         | 05.13.18      | 2001 | **       |
| 2  | Гучапшева А.Х.    | Автоматизация проектирования программных игровых обучающих систем на основе алгебраического анализа                                                  | к.т.н.         | 05.13.12      | 2004 | **       |
| 3  | Першин А.А.       | Методы создания интерактивных онлайн-курсов на основе игровых механик                                                                                | к.т.н.         | 05.13.06      | 2014 | **       |
| 4  | Катаев А.В.       | Программно-информационная поддержка процесса разработки обучающих компьютерных игр                                                                   | к.т.н.         | 05.13.01      | 2012 | *        |
| 5  | Матлин А.О.       | Автоматизация процесса создания виртуальных тренажеров                                                                                               | к.т.н.         | 05.13.12      | 2012 | **       |
| 6  | Захаров А.Ю.      | Структурно-параметрический синтез физических моделей изолирующих дыхательных аппаратов для тренажерных комплексов виртуальной/дополненной реальности | к.т.н.         | 05.13.01      | 2020 | **       |
| 7  | Александрова М.В. | Управление глубиной погружения в иммерсивных интерактивных средах                                                                                    | к.т.н.         | 2.3.1         | 2024 | *        |
| 8  | Аббаси М.М.       | Лингвистический интерфейс с эмоциональным анализом текста в компьютерных системах                                                                    | к.т.н.         | 2.3.1         | 2022 |          |
| 9  | Гутман И.Е.       | Компьютерные виртуальные игры: культурно-антропологические аспекты анализа                                                                           | к.ф.н.         | 09.00.13      | 2009 |          |
| 10 | Рыбалтович Д.Г.   | Психологические особенности пользователей онлайн-игр с различной степенью игровой аддикции                                                           | к. психол. н.  | 05.26.02      | 2012 |          |
| 11 | Сигал А.В.        | Моделирование процессов оптимального распределения ресурсов на основе решения антагонистических игр                                                  | д.э.н.         | 08.00.13      | 2015 |          |
| 12 | Кутуракова В.В.   | Математическое и программное обеспечение многопользовательских тренажеров с погружением в иммерсивные виртуальные среды                              | к.т.н.         | 05.13.11      | 2019 | *        |

## ПРИЛОЖЕНИЕ 2. Зарубежные PhD-диссертации по тематике видеоигровых систем (англо- и франкоязычные, 2015–2024)

Таблица составлена на основе баз ProQuest Dissertations & Theses, HAL Open Science (theses.hal.science) и theses.fr. Приведены диссертации, в которых видеоигра или компонент игровой системы выступает основным техническим объектом. Колонка «Задача» отсылает к классификации раздела 6 настоящей статьи.

| № | Автор       | Название                                                                              | Университет      | Год  | Задача(и) |
|---|-------------|---------------------------------------------------------------------------------------|------------------|------|-----------|
| 1 | Constant T. | Éléments de game design pour le développement d'une attitude réflexive chez le joueur | CNAM, Париж (FR) | 2017 | 3.1, 3.2  |
| 2 | Chauvin S.  | Un modèle narratif pour les jeux vidéo émergents                                      | CNAM, Париж (FR) | 2019 | 3.4 (PCG) |

| № | Автор                         | Название                                                                                             | Университет              | Год   | Задача(и)   |
|---|-------------------------------|------------------------------------------------------------------------------------------------------|--------------------------|-------|-------------|
| 3 | Jothimurugan K.               | Specification-Guided Reinforcement Learning                                                          | U of Pennsylvania (USA)  | ~2022 | 3.3, 3.6    |
| 4 | Roohi S.                      | Advances in AI-assisted Game Testing                                                                 | Aalto University (FIN)   | ~2022 | 3.4 (тест.) |
| 5 | Wen J.                        | Optimization-based Approaches for Structured Content Creation in Game Design                         | U of Minnesota (USA)     | ~2022 | 3.2, 3.4    |
| 6 | Bailly R.                     | Génération procédurale de niveaux de jeu alliant approche constructive et optimisation (Genetic-WFC) | CNAM, Париж (FR)         | 2022  | 3.4 (PCG)   |
| 7 | Le Pelletier de Woillemont T. | Data and AI in Video Games: Reinforcement Learning for NPC Play-Styles and Automated Testing         | Sorbonne Université (FR) | 2023  | 3.3, 3.4    |
| 8 | Taesiri M. R.                 | Leveraging Foundation Models for Video Game Quality Assurance                                        | U of Alberta (CAN)       | 2024  | 3.4 (тест.) |

*Примечание: ~ – год определен приблизительно; 3.1–3.6 – задачи из разделов 1-6; PCG – процедурная генерация контента; тест. – тестирование. Диссертации по бисимуляционной эквивалентности игровых реализаций (3.5) и формальной верификации механик (3.1) в доступных базах не выявлены.*

# VIDEO GAME AS AN INFORMATION SYSTEM: SCIENTIFIC BASIS AND CLASSIFICATION OF TASKS

**Kugurakova Vlada Vladimirovna**

*Candidate of Technical Sciences, associate professor*

*Kazan (Volga Region) Federal University*

*Institute of Information Technologies and Intelligent Systems*

*Department of video game development*

*Head of the department*

*Kazan, Republic of Tatarstan, Russian Federation*

*vlada.kugurakova@gmail.com*

## Abstract

*A video game is viewed as a discrete dynamic information system defined by a tuple of formal components  $G = \langle S, A, \delta, R, \Pi, \omega, W \rangle$ . It is shown that existing definitions of a game, as adopted in the humanities, are not operational and are unsuitable for formulating engineering problems. Based on the proposed model, six classes of relevant problems in technical science are formulated: formal verification of mechanics, balancing as multi-criteria optimization, agent behavior modeling, procedural content generation, state space testing, and formal specification. For each class of problems, a connection to model components is established, and applicable formal methods are identified. The results are aimed at establishing an independent branch of technical science on video games within the domestic scientific community.*

## Keywords

*video game; information system; formalized model; game mechanics; verification; balancing; game studies*

## References

1. Newzoo Global Games Market Report 2023 [Electronic resource]. URL: <https://newzoo.com> (accessed at: 15.11.2024).
2. Ob'yavleniya o zashchitakh dissertatsii [Electronic resource] // Vysshaya attestatsionnaya komissiya pri Ministerstve nauki i vysshego obrazovaniya Rossiiskoi Federatsii. URL: [https://vak.minobrnauki.gov.ru/adverts\\_list](https://vak.minobrnauki.gov.ru/adverts_list) (accessed at: 01.04.2026) [in Russian].
3. Elektronnaya biblioteka dissertatsii Rossiiskoi gosudarstvennoi biblioteki [Electronic resource]. URL: <https://dvs.rsl.ru> (accessed at: 01.04.2026) [in Russian].
4. DissersCat: elektronnaya biblioteka dissertatsii [Electronic resource]. URL: <https://www.disserscat.com> (accessed at: 01.04.2026) [in Russian].
5. Markova V. V. Razrabotka i issledovanie modeliruyushchikh avtomatizirovannykh delovykh igr na osnove analiza problemno-orientirovannykh programmnykh mashin: dis. ... kand. tekhn. nauk: 05.13.18. Ryazan', 2001. 170 p. [in Russian].
6. Avtomatizatsiya proektirovaniya programmnykh igrovyykh obuchayushchikh sistem na osnove algebraycheskogo analiza: dis. ... kand. tekhn. nauk: 05.13.12. [B. m.], 2003. [in Russian].
7. Gutman I. E. Komp'yuternye virtual'nye igry: kul'turno-antropologicheskie aspekty analiza: dis. ... kand. filos. nauk: 09.00.13. SPb., 2009. 192 p. [in Russian].
8. Rybaltovich D. G. Psikhologicheskie osobennosti pol'zovatelei onlain-igr s razlichnoi stepen'yu igrovoi addiktzii: dis. ... kand. psikhol. nauk: 05.26.02. SPb., 2012. 172 p. [in Russian].
9. Aarseth E. Computer Game Studies, Year One // Game Studies. 2001. Vol. 1. № 1.
10. Frasca G. Ludology meets Narratology: Similitude and Differences between (Video) Games and Narrative // Parnasso. 1999.
11. ProQuest Dissertations & Theses Global [Electronic resource]. URL: <https://www.proquest.com/products-services/pqdtglobal.html> (accessed at: 01.04.2026).
12. HAL: archive ouverte pluridisciplinaire [Electronic resource]. URL: <https://hal.science> (accessed at: 01.04.2026).
13. Thèses.fr: le portail des thèses françaises [Electronic resource]. URL: <https://www.theses.fr> (accessed at: 01.04.2026).

14. Crawford C. The Art of Computer Game Design. Berkeley: Osborne/McGraw-Hill, 1984. 110 p.
15. Salen K., Zimmerman E. Rules of Play: Game Design Fundamentals. Cambridge: MIT Press, 2004. 688 p.
16. Juul J. Half-Real: Video Games between Real Rules and Fictional Worlds. Cambridge: MIT Press, 2005. 233 p.
17. von Neumann J., Morgenstern O. Theory of Games and Economic Behavior. Princeton: Princeton University Press, 1944. 625 p.
18. de Alfaro L., Henzinger T. A. Interface Automata // Proceedings of FSE. 2001. P. 109–120.
19. Puterman M. L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York: Wiley, 1994. 649 p.
20. Shoham Y., Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge: Cambridge University Press, 2009. 532 p.
21. Hunicke R., LeBlanc M., Zubek R. MDA: A Formal Approach to Game Design and Game Research // Proceedings of AAAI Workshop on Challenges in Game AI. 2004.
22. Sicart M. Defining Game Mechanics // Game Studies. 2008. Vol. 8. № 2.
23. Fikes R., Nilsson N. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving // Artificial Intelligence. 1971. Vol. 2. P. 189–208.
24. Shubin A. V., Kugurakova V. V. Proektirovanie instrumentariya dlya sozdaniya igrovogo protsessa cherez sistematizatsiyu igrovyykh mekhanik // Elektronnye biblioteki. 2024. T. 27. № 5. P. 774–795 [in Russian].
25. Pnueli A. The Temporal Logic of Programs // Proceedings of FOCS. 1977. P. 46–57.
26. Clarke E. M., Grumberg O., Peled D. Model Checking. Cambridge: MIT Press, 1999. 314 p.
27. Bryant R. E. Graph-based algorithms for boolean function manipulation // IEEE Transactions on Computers. 1986. Vol. 35. № 8. P. 677–691.
28. Ehrgott M. Multicriteria Optimization. 2nd ed. Berlin: Springer, 2005. 323 p.
29. Rakhmankulova V. R., Kugurakova V. V. Onlain-instrument Tula dlya balansirovki videoigr // Elektronnye biblioteki. 2025. T. 28. № 4. P. 903–930 [in Russian].
30. Sakhibgareeva G. F., Kugurakova V. V. Praktiki balansirovaniya komp'yuternykh igr // Programmnye sistemy: teoriya i prilozheniya. 2022. T. 13. № 3(54). P. 255–273 [in Russian].
31. Kwiatkowska M., Norman G., Parker D. PRISM 4.0: Verification of Probabilistic Real-Time Systems // Lecture Notes in Computer Science. 2011. Vol. 6806. P. 585–591.
32. Hunicke R. The Case for Dynamic Difficulty Adjustment in Games // Proceedings of ACM CHI Workshop on Designing for User Experiences. 2004.
33. Bol'shakov E. S., Kugurakova V. V. Generativnaya simulyatsiya igrovogo okruzheniya v real'nom vremeni // Elektronnye biblioteki. 2025. T. 28. № 2. P. 188–212 [in Russian].
34. Shaker N., Togelius J., Nelson M. J. Procedural Content Generation in Games. Springer, 2016. 244 p.
35. Trofimchuk V. T., Kugurakova V. V. Razrabotka adaptivnoi sistemy generatsii igrovyykh kvestov i dialogov na osnove bol'shikh yazykovyykh modelei // Elektronnye biblioteki. 2025. T. 28. № 4. P. 953–993 [in Russian].
36. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken: Wiley, 2012. 240 p.
37. Browne C. et al. A Survey of Monte Carlo Tree Search Methods // IEEE Transactions on Computational Intelligence and AI in Games. 2012. Vol. 4. № 1. P. 1–43.
38. Bondar' A. O., Kugurakova V. V. Problemy, resheniya i perspektivy avtomatizirovannogo perenosa igrovyykh stszen mezhdu igrovymi dvizhkami // Elektronnye biblioteki. 2025. T. 28. № 2. P. 213–243 [in Russian].
39. Garavel H. et al. CADP 2011: A Toolbox for the Construction and Analysis of Distributed Processes // Lecture Notes in Computer Science. 2013. Vol. 7795. P. 233–240.
40. Lamport L. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Boston: Addison-Wesley, 2002. 364 p.
41. Jackson D. Software Abstractions: Logic, Language, and Analysis. Cambridge: MIT Press, 2006. 344 p.
42. Hansson H., Jonsson B. A Logic for Reasoning about Time and Reliability // Formal Aspects of Computing. 1994. Vol. 6. № 5. P. 512–535.

43. Kugurakova V. V. A Formal Approach to Spatio-Temporal Modeling of Game Systems // Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki. 2024. Vol. 166. № 4. P. 532–554.
44. Passport nauchnoi spetsial'nosti 2.3.8 "Informatika i informatsionnye protsessy" [Electronic resource]: utv. prikazom Minobrnauki Rossii ot 24.02.2021 No. 118: v red. ot 24.07.2023 No. 730. URL: <https://vak.minobrnauki.gov.ru> (accessed at: 01.04.2026) [in Russian].
45. Bridgman P. W. The Logic of Modern Physics. New York: Macmillan, 1927. 228 p.
46. Aleksandrova M. V. Upravlenie glubinoi pogruzheniya v immersivnykh interaktivnykh sredakh: avtoref. dis. ... kand. tekhn. nauk: 2.3.1. Samara, 2024. 23 p. [in Russian].
47. Dijkstra E. W. Guarded Commands, Nondeterminacy and Formal Derivation of Programs // Communications of the ACM. 1975. Vol. 18. № 8. P. 453–457.
48. Fabricatore C. Gameplay and Game Mechanics Design: A Key to Quality in Videogames // Proceedings of the OECD-ENLACES Expert Meeting on Videogames and Education. Santiago de Chile, 2007. URL: <https://eprints.hud.ac.uk/id/eprint/20927/> (accessed at: 01.04.2026).
49. Levenshtein V. I. Binary codes capable of correcting deletions, insertions, and reversals // Soviet Physics Doklady. 1966. Vol. 10. № 8. P. 707–710.
50. Kantorovich L. V. O peremeshchenii mass // Doklady AN SSSR. 1942. T. 37. № 7–8. P. 227–229 [in Russian].
51. Makarov I., Savostyanov D., Litvyakov B., Ignatov D. I. Predicting Winning Team and Probabilistic Ratings in "Dota 2" and "Counter-Strike: Global Offensive" Video Games // Analysis of Images, Social Networks and Texts (AIST 2017). — Cham : Springer, 2017. — P. 183–196.
52. Semenov A., Romov P., Korolev S., Yashkov D., Neklyudov K. Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2 // Analysis of Images, Social Networks and Texts (AIST 2016). — Cham : Springer, 2016. — P. 26–37.