

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.П. АСТАФЬЕВА»
(КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики
Кафедра информатики и информационных технологий в образовании

Мамет Бинур Габидинкызы

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**Применение технологии геймификации при обучении
программированию в старшей школе**

Направление подготовки: 44.04.01 Педагогическое образование
Направленность (профиль) образовательной программы: Технология
цифровизации образовательной деятельности (с применением сетевой формы)
с Казахским национальным педагогическим университетом им. Абая



ДОПУСКАЮ К ЗАЩИТЕ

Заведующий кафедрой

д-р пед. наук., проф.

_____ Пак Н.И.
Руководитель _____ **магистерской**
программы

д-р пед. наук., проф., зав. каф. ИиИТО

_____ Пак Н.И.
Научный руководитель

канд. пед. наук, доц. каф. ИиИТО

_____ Ломаско П.С.
Обучающийся

_____ Мамет Б.Г.

Дата защиты « ____ » июня 2022 г.

Оценка (прописью) _____

Красноярск 2022

Реферат

Сегодня появляются новые модели и подходы, основанные на применении информационных и игровых технологий в образовании, таких как, например, «эдьютеймент» (от англ. edutainment - обучение через развлечение), «геймификация» (от англ. gamification - использование игровых механик и элементов в неигровом контексте), «серьёзные игры» (от англ. serious games - видеоигры, в которых основная цель - не развлекательная), «игры для обучения» (от англ. games for learning - игры, непосредственно разработанные в целях обучения или имеющие вторичную образовательную ценность), «обучение, основанное на игре» (от англ. game-based learning - подход в обучении, основанный на игре) и др. Подобные технологии позволяют учитывать особенности восприятия и обработки информации обучающихся (быстрый доступ к информации, вариативность её использования, интерактивность, визуальная подача информации), их интересы (адаптивные, индивидуальные образовательные траектории), эффективно выстраивать процессы коммуникации (механизмы быстрой обратной связи и внутригруппового общения), повышать уровень мотивации обучающихся и т. д. Однако и они имеют свои недостатки, связанные со сложностью применения и реализации.

Геймификация сегодня активно применяется для повышения уровня мотивации и вовлеченности обучающихся в образовательный процесс, который за счет этого становится более увлекательным и интересным. На протяжении последних нескольких лет геймификация довольно широко изучается и обсуждается как элемент вовлечения в области образования. Геймификация основана на использовании элементов игры, а игра, по сути, является хорошим мотиватором к той или иной деятельности. Как утверждают Г. Зикерманн и К. Каннингем, основанные на трех основных компонентах (удовольствии, наградах и времени) игры становятся одной из самых влиятельных сил всего человечества. Без сомнений, игры способны без принуждения мотивировать людей совершать определенные действия в ситуациях, когда люди не всегда знают, насколько они хотят их совершать. Геймификация основана на игре и поэтому так успешна. Как отмечают К. Вербак и Д. Хантер, «наш мозг требует решения задач, обратной связи и много другого - того, что дают нам игры. Исследования за исследованиями показывают, что игры активируют дофаминовую систему мозга, которая связана с удовольствием».

Путь комплексного решения педагогических задач мы видим в поиске и использовании в работе соответствующих образовательных технологий, позволяющих создать мотивирующее пространство для реализации личности и достижения ее успехов. Особым потенциалом для организаций и педагогов дополнительного образования детей обладает геймификация как технология

использования игровых правил для достижения педагогических целей (использование игровых механик в неигровом процессе), привлечения пользователей и потребителей, повышения их вовлеченности в решение прикладных задач.

Цель исследования: разработать и обосновать комплекс дидактических средств, демонстрирующий применение технологий геймификации при обучении программированию на уроках информатики в старшей школе.

Объект исследования: процесс обучения программированию в старшей школе.

Предмет исследования: средства реализации технологий дидактической геймификации.

Гипотеза исследования заключается в предположении о том, что повышению результативности обучения программированию в старшей школе будет содействовать комплекс дидактических средств, в котором:

- используется система разноуровневых заданий, выполнение которых предполагает получение баллов и виртуальных «наград» в качестве промежуточных достижений;
- характеристики достижений обучающихся учитываются по направлениям: работа с данными, алгоритмы, оптимизация, эргономичность кода, которые в совокупности определяют их общий уровень прогресса;
- создана среда, позволяющая публиковать рейтинг достижений и полученные виртуальные награды и вводить соревновательные моменты в образовательный процесс как частей октализа.

Основные задачи:

1. На основе анализа научно-методических источников описать теоретические основы применения технологий геймификации на уроках информатики в старшей школе.

2. Выделить виды технологий и признаки дидактической геймификации.

3. Спроектировать серию занятий по программированию в старшей школе с применением технологий геймификации.

4. Разработать учебные материалы и задания для применения технологий геймификации.

5. Провести оценку разработанных средств и проанализировать ее результаты.

Теоретическая значимость исследования заключается в том, что в работе разработана идея создания комплекса дидактических средств, демонстрирующих применение технологий геймификации при обучении программированию на уроках информатики в старшей школе; показаны примеры способов реализации геймифицированных дидактических средств с учетом когнитивных механизмов вовлечения обучающихся в учебный процесс, предложены конкретные инструменты для их разработки.

Практическая значимость исследования заключается в том, что разработанный комплекс геймифицированных дидактических средств может быть использован в практике обучения старшеклассников программированию на Python при освоении тем школьной информатики «Процедуры с параметрами», «Функции», «Рекурсивные алгоритмы», «Списки», «Статистические операции со списками», «Алгоритмы сортировки списков».

Эмпирическая база исследования: комплекс заданий с применением технологии геймификации разработаны на платформе GamiLab с последующим применением для школьников старших классов при изучении языков программирования.

Abstract

Today, new models and approaches are emerging based on the use of information and gaming technologies in education, such as, for example, «edutainment» (from the English edutainment - learning through entertainment), «gamification» (from the English gamification - the use of game mechanics and elements in a non-gaming context), «serious games» (from the English serious games - video games in which the main goal is not entertaining), «games for learning» (from the English games for learning - games that are directly developed for learning purposes or have a secondary educational value), «learning based on the game» (from the English game-based learning - an approach to learning based on the game), etc. Such technologies allow taking into account the peculiarities of the perception and processing of information by students (quick access to information, variability of its use, interactivity, visual presentation of information), their interests (adaptive, individual educational trajectories), effectively build communication processes (mechanisms for quick feedback and intragroup communication), increase the level of motivation of students, etc. However, they also have their drawbacks associated with the complexity of application and implementation.

Gamification is actively used today to increase the level of motivation and involvement of students in the educational process, which, due to this, becomes more exciting and interesting. Over the past few years, gamification has been widely studied and discussed as an element of engagement in the field of education. Gamification is based on the use of game elements, and the game, in fact, is a good motivator for this or that activity. According to G. Zikermann and K. Cunningham, games based on three main components (pleasure, rewards and time) are becoming one of the most influential forces of all mankind. There is no doubt that games can motivate people without coercion to take certain actions in situations where people do not always know how much they want to do them. Gamification is based on the game and that is why it is so successful. As K. Werbach and D. Hunter note, «our brain requires problem solving, feedback, and much more - what games give us. Study after study shows that games activate the brain's dopamine system, which is associated with pleasure.»

We see the way of a comprehensive solution of pedagogical problems in the search and use in the work of appropriate educational technologies that allow creating a motivating space for the realization of the personality and the achievement of its success. Gamification has a special potential for organizations and teachers of additional education for children as a technology for using game rules to achieve pedagogical goals (using game mechanics in a non-game process), attracting users and consumers, and increasing their involvement in solving applied problems. The purpose of the study: to develop and substantiate a set of didactic tools that demonstrate the use of gamification technologies in teaching programming in computer science lessons in high school.

Object of study: the process of teaching programming in high school.

Subject of study: means of implementing didactic gamification technologies.

Research hypothesis consists in the assumption that a set of didactic tools will contribute to increasing the effectiveness of teaching programming in high school, in which:

- a system of multi-level tasks is used, the completion of which involves the receipt of points and virtual «awards» as intermediate achievements;
- characteristics of students' achievements are taken into account in the following areas: working with data, algorithms, optimization, code ergonomics, which together determine their overall level of progress;
- an environment has been created that allows publishing the rating of achievements and received virtual awards and introducing competitive moments into the educational process as parts of octalysis.

Main goals:

1. Based on the analysis of scientific and methodological sources, describe the theoretical foundations for the use of gamification technologies in computer science lessons in high school.
2. Identify types of technologies and features of didactic gamification.
3. Design a series of programming lessons in high school using gamification technologies.
4. Develop training materials and tasks for the application of gamification technologies.
5. Evaluate the developed tools and analyze its results.

Theoretical significance: The research lies in the fact that the work has developed the idea of creating a set of didactic tools that demonstrate the use of gamification technologies in teaching programming in computer science lessons in high school; examples of ways to implement gamified didactic tools are shown, taking into account the cognitive mechanisms of involving students in the educational process, and specific tools for their development are proposed.

Practical significance: research is that the developed complex of gamified didactic tools can be used in the practice of teaching high school students programming in Python when mastering the topics of school computer science «Procedures with parameters», «Functions», «Recursive algorithms», «Lists», «Statistical operations with lists», «Algorithms for sorting lists».

Empirical base of research: a set of tasks using gamification technology was developed on the GamiLab platform with subsequent use for high school students in the study of programming languages.

Оглавление

Глава 1. Теоретические основы применения технологий геймификации на уроках информатики в старшей школе.....	12
1.1. Сущность технологий дидактической геймификации и их ключевые признаки	12
1.2. Возможности геймифицированных платформ для обучения программированию	22
Выводы по первой главе.....	43
Глава 2. Практические аспекты применения технологий геймификации при обучении программированию старшеклассников	45
2.1. Методические планирования занятий по программированию на Python.....	45
2.2. Особенности разработки и использования геймифицированных средств обучения	52
2.3. Результаты оценки разработанных средств	60
Выводы по второй главе.....	65
Заключение	67
Список использованных источников	68

Введение

Сегодня появляются новые модели и подходы, основанные на применении информационных и игровых технологий в образовании, таких как, например, «эдьютеймент» (от англ. edutainment - обучение через развлечение), «геймификация» (от англ. gamification - использование игровых механик и элементов в неигровом контексте), «серьёзные игры» (от англ. serious games - видеоигры, в которых основная цель - не развлекательная), «игры для обучения» (от англ. games for learning - игры, непосредственно разработанные в целях обучения или имеющие вторичную образовательную ценность), «обучение, основанное на игре» (от англ. game-based learning - подход в обучении, основанный на игре) и др. Подобные технологии позволяют учитывать особенности восприятия и обработки информации обучающихся (быстрый доступ к информации, вариативность её использования, интерактивность, визуальная подача информации), их интересы (адаптивные, индивидуальные образовательные траектории), эффективно выстраивать процессы коммуникации (механизмы быстрой обратной связи и внутригруппового общения), повышать уровень мотивации обучающихся и т. д. Однако и они имеют свои недостатки, связанные со сложностью применения и реализации.

Геймификация сегодня активно применяется для повышения уровня мотивации и вовлеченности обучающихся в образовательный процесс, который за счет этого становится более увлекательным и интересным. На протяжении последних нескольких лет геймификация довольно широко изучается и обсуждается как элемент вовлечения в области образования. Геймификация основана на использовании элементов игры, а игра, по сути, является хорошим мотиватором к той или иной деятельности. Как утверждают Г. Зикерманн и К. Каннингем, основанные на трех основных компонентах (удовольствии, наградах и времени) игры становятся одной из самых

влиятельных сил всего человечества. Без сомнений, игры способны без принуждения мотивировать людей совершать определенные действия в ситуациях, когда люди не всегда знают, насколько они хотят их совершать. Геймификация основана на игре и поэтому так успешна. Как отмечают К. Вербах и Д. Хантер, «наш мозг требует решения задач, обратной связи и много другого - того, что дают нам игры. Исследование за исследованием показывают, что игры активируют дофаминовую систему мозга, которая связана с удовольствием».

Путь комплексного решения педагогических задач мы видим в поиске и использовании в работе соответствующих образовательных технологий, позволяющих создать мотивирующее пространство для реализации личности и достижения ее успехов. Особым потенциалом для организаций и педагогов дополнительного образования детей обладает геймификация как технология использования игровых правил для достижения педагогических целей (использование игровых механик в неигровом процессе), привлечения пользователей и потребителей, повышения их вовлеченности в решение прикладных задач.

Анализ литературы позволил выделить **следующие противоречия:**

- между высоким потенциалом технологий дидактической геймификации для повышения учебной мотивации при обучении программированию в старшей школе на уроках информатики и недостаточной разработанностью научно-методических основ для их применения;
- между потребностью практики в создании и использовании качественных геймифицированных средств обучения программированию в старшей школе и ограниченными возможностями существующего инструментария для их разработки и применения.

Проблема исследования заключается в поиске ответа на вопрос о том, каким образом можно использовать технологии геймификации при обучении программированию на уроках информатики в старшей школе.

Цель исследования: разработать и обосновать комплекс дидактических средств, демонстрирующий применение технологий геймификации при обучении программированию на уроках информатики в старшей школе.

Объект исследования: процесс обучения программированию в старшей школе.

Предмет исследования: средства реализации технологий дидактической геймификации.

Гипотеза исследования заключается в предположении о том, что повышению результативности обучения программированию в старшей школе будет содействовать комплекс дидактических средств, в котором:

- используется система разноуровневых заданий, выполнение которых предполагает получение баллов и виртуальных «наград» в качестве промежуточных достижений;
- характеристики достижений обучающихся учитываются по направлениям: работа с данными, алгоритмы, оптимизация, эргономичность кода, которые в совокупности определяют их общий уровень прогресса;
- создана среда, позволяющая публиковать рейтинг достижений и полученные виртуальные награды и вводить соревновательные моменты в образовательный процесс как частей октализа.

Основные задачи:

1. На основе анализа научно-методических источников описать теоретические основы применения технологий геймификации на уроках информатики в старшей школе.

2. Выделить виды технологий и признаки дидактической геймификации.

3. Спроектировать серию занятий по программированию в старшей школе с применением технологий геймификации.

4. Разработать учебные материалы и задания для применения технологий геймификации.

5. Провести оценку разработанных средств и проанализировать ее результаты.

Теоретическая значимость исследования заключается в том, что в работе разработана идея создания комплекса дидактических средств, демонстрирующих применение технологий геймификации при обучении программированию на уроках информатики в старшей школе; показаны примеры способов реализации геймифицированных дидактических средств с учетом когнитивных механизмов вовлечения обучающихся в учебный процесс, предложены конкретные инструменты для их разработки.

Практическая значимость исследования заключается в том, что разработанный комплекс геймифицированных дидактических средств может быть использован в практике обучения старшеклассников программированию на Python при освоении тем школьной информатики «Процедуры с параметрами», «Функции», «Рекурсивные алгоритмы», «Списки», «Статистические операции со списками», «Алгоритмы сортировки списков».

Эмпирическая база исследования: комплекс заданий с применением технологии геймификации разработаны на платформе GamiLab с последующим применением для школьников старших классов при изучении языков программирования.

Ключевые понятия: октализ (8 ключевых элементов игровых механик), дидактическая геймификация, дидактическое средство, познавательный интерес, обучение программированию.

Глава 1. Теоретические основы применения технологий геймификации на уроках информатики в старшей школе

1.1. Сущность технологий дидактической геймификации и их ключевые признаки

Идея использования игр в образовательных целях не нова. Однако глобального определения серьезных игр не существует. Например, некоторые исследователи утверждают, что все игры имеют серьезную цель (например, азартные игры). Другие подчеркивают, что серьезные игры имеют определенный результат обучения, чего нельзя сказать о обычных развлекательных играх [1]. Наиболее часто используемое определение серьезных игр звучит так: игры, основной целью которых не является развлечение, удовольствие или веселье. Это определение отличает серьезные игры от обычных видеоигр по целям их разработки. Этот аргумент вызывает фундаментальную проблему. Согласно этому определению, доступ к целям и намерениям разработчиков игр необходим, чтобы решить, является ли конкретная игра серьезной игрой или нет, что совершенно непрактично.

В образовательном контексте цифровые игры имеют различные преимущества, в частности, для решения конкретных проблем или обучения определенным навыкам. В своем исследовании Аль-Азави и др. перечислили несколько преимуществ использования цифровых игр в образовании[2]:

- Игры позволяют участникам преодолеть демографические различия (например, образовательный статус, этническая принадлежность, пол и возраст);
- Игры можно использовать в качестве инструментов исследования и/или измерения, их можно легко стандартизировать и изменять;
- Игры создают веселые и стимулирующие впечатления, таким образом удерживая внимание игроков в течение длительного времени;

- Игры помогают участникам испытать новизну, любопытство и соревновательность, что стимулирует процесс обучения.

Помимо этих преимуществ, другие возможности вытекают из общих условий цифровых игр. Например, образовательные онлайн-игры могут облегчить дистанционное обучение. Онлайн-игры не ограничены местными условиями, и, таким образом, больше людей могут получить от них пользу, чем от традиционных концепций обучения (например, лекций в классе).

Впервые термин «геймификация» появился в начале XXI века, однако массовое применение этого понятия в различных сферах деятельности началось в 2010 году. В настоящий момент существует много работ, посвященных уточнению феномена геймификации. Однако в ходе анализа этих исследований замечено, что в научном сообществе пока нет единой устоявшейся трактовки этого термина.

Ценность работ И. Аттали[3] определяется тем, что им предлагается подход, в котором геймификация рассматривается как инструментальный для повышения вовлеченности пользователей. В трудах Л. П. Варениной[4] рассматривается реальный положительный опыт внедрения геймификации в образовательную среду на примере технических дисциплин. Значимость результатов Ю. П. Олейника[5] определяется тем, что он одним из первых выполнил обзор подходов к раскрытию сущности феномена «геймификация». В исследованиях О. В. Орловой[6] аргументированно обосновывается, что внедрение компьютерных игр в учебный процесс не только обогащает этап мотивации, но и «продлевает приверженность задаче, повышает вероятность достижения поставленной цели».

Термин «геймификация» в контексте образовательной деятельности появился относительно недавно, в 2011 году, благодаря немецкому ученому С. Детердингу[7]. С точки зрения ученого, геймификация – это применение игровых элементов и приемов для проектирования игр в неигровых

контекстах. Данный подход выбран потому, что в нем учитывается обязательное наличие игрового формата деятельности, возможность (способы, инструменты, приемы) проектирования пространства игры, реализация идеи в неигровых ситуациях, а конкретно – в обучении.

Когнитивный процесс представляет собой интеграцию представленной информации (из игры) с предшествующими знаниями, что приводит к созданию учащимся ментальных моделей. Игры могут способствовать ситуативному обучению, предоставляя учащемуся информацию в тот момент, когда она необходима (например, предоставляя информацию, которая необходима для решения конкретной проблемы в данный момент времени). Кроме того, игры могут способствовать переносу обучения, отражая проблемы реальной жизни. Это делается путем отработки конкретных навыков и применения ранее полученных знаний для решения новых задач. Чтобы успешно реализовать когнитивную обработку в играх, игровая механика должна быть согласована с намеченными целями обучения. Кроме того, следует сократить количество игровых элементов, не связанных напрямую с когнитивными процессами, поскольку они могут мешать обучению [10].

Аффективная основа связана с опытом обучения игрока во время игры, например, с пережитыми эмоциями, установками и убеждениями. Переживаемые эмоции включают компоненты визуального дизайна и влияют на эмоциональное состояние учащегося, что потенциально улучшает результаты обучения. Аффективные факторы должны быть хорошо интегрированы с когнитивными, мотивационными и социокультурными аспектами обучения, поскольку их нельзя разделить [11].

Элементы и приемы геймификации как в настольных, электронных и видеоиграх предлагает широкий диапазон.

Аватар в играх — это роль, которую берет на себя игрок при участии в игре (игрок меняет личность). Аватары имеют разные имена, изображения и могут иметь разные навыки в зависимости от игры.

Система вознаграждений представляет собой график с количественным описанием различных элементов вознаграждения (бонусов, значков, комбо, вознаграждений и ресурсов) и усилий, необходимых для их получения. По словам Скиннера [18], самая эффективная система поощрения — это та, в которой вознаграждения имеют переменное соотношение количества, временного интервала и усилий, которые необходимо получить.

Бонусы ожидаемые вознаграждения в виде различных типов абстрактных единиц (например, баллов). События в игре могут увеличивать или уменьшать результаты разных участников.

Значки награда, присуждаемая за успехи или заслуги в определенной области. Они могут быть физическими (небольшой металлический значок с каким-либо символом — используется в ролевых играх) или цифровыми (изображение с иллюстрацией).

Комбо (сокращение от сочетания) представляет собой совокупность действий, выполняемых последовательно, как правило, со строгими временными ограничениями, дающих значительное преимущество. Обычно дается игроку в качестве своеобразной награды за достигнутую цель.

Награда — это неожиданная награда (при определенных условиях) с различными поощрениями в игре, которые вдохновляют и мотивируют игрока.

Игровые ресурсы могут быть различные предметы, помогающие облегчить выполнение целей игры.

Таблица лидеров в игровой индустрии используется для обозначения ранга среди людей, которые играют. Игроки могут быть ранжированы на основе достижений (например, набранные очки, уровни, прогресс, время, необходимое для достижений). Таблицы лидеров могут стимулировать

игроков к совершенствованию, поскольку они дают многим из них чувство превосходства или достижения. Пользователи, находящиеся на вершине рейтинга, знают, что в данный момент находятся на вершине пирамиды, поэтому этот элемент провоцирует конкуренцию между игроками.

Уровень игры — это раздел или часть игры. Чтобы перейти на более высокий уровень, игрок обычно должен достичь определенных целей или выполнить определенные задачи. Уровни представляют собой последовательность выполнения.

Прогресс в игре используется для представления текущих достижений игрока и продвижение (например, заработанные очки, достигнутый уровень и полученные значки).

Статус в игре — это совокупная информация о текущем прогрессе (достигнутых целях), выбранном аватаре, достигнутом уровне и доступных ресурсах и инструментах.

Сотрудничество в команде — это акт совместной работы с другими игроками для достижения обоюдно желаемого и полезного результата. Это социальный аспект игр, который нравится многим игрокам. В командных играх чем больше игроки работают вместе, тем большего они могут достичь.

Время можно использовать как мотивирующий элемент активности и действий игрока. Например, *таймеры* повышают уровень стресса и мотивируют действия. Ограничения по времени — это техника, используемая для провокации игроков, они сосредотачиваются и начинают выполнять задачи, необходимые для достижения уровня или цели игры.

Коммуникация используется для доставки сообщений между двумя или более игроками с целью обмена информацией, связанной с игрой, и для стимулирования стремления к социализации между участниками.

Обратная связь получение заключения компетентной стороны о действиях выполняемые игроком, обычно используемые как своего рода подсказка или мотивация.

Выполнение поставленных задач, приключения или проблемы в играх стимулирует удовлетворение игроков. Многие игры используют этот прием для достижения различных целей, чаще всего описываемых историями и дополняемых ограничениями по времени.

Сокровища являются техникой раскрытия духа открытия игроки. Игроки должны выполнить определенные условия, чтобы разблокировать спрятанные сокровища.

Сюжет/история представляет собой описание конкретных подсцен для ролевой игры, игры и игры-рассказы.

Правила игры – одна из главных составляющих любой игры. Игра — это всего лишь набор определенных правил, которые разработаны специально для того, чтобы ограничить действия игрока и сделать игру управляемой.

В рамках системы поощрений студенты могут получать дополнительные льготы по разным правилам и в разных формах – значки (отличительный знак за достижения в различных учебных мероприятиях), поощрения (неожиданно полученные за определенные успехи, например ресурсы с интересными фактами, сертификаты и т.д.), бонусы (ожидаемое вознаграждение за выполненную учебную деятельность, например баллы), комбо (дополнительные подсказки или учебные материалы, способствующие решению определенной задачи в более короткий период времени), игровые ресурсы (разные виды учебных ресурсов) и Сокровища (табл. 1).

Успеваемость учащегося представляет собой представление процента выполненных задач обучения. Статус учитывает текущее состояние учащегося - текущий аватар, оценку учащегося, учебную активность и т.д.

Таблица 1. Взаимосвязь между игровым элементом, техникой и действием

Элемент игры	Техника игры	Игровое действие
Аватар	Изменение личности	Ролевые игры
Бонус	Система вознаграждений	Получение бонуса
Значок	Система вознаграждений	Получение награды
Комбо	Система вознаграждений	Получение преимущества
Награда	Система вознаграждений	Награждение
Ресурс	Система вознаграждений	Получение ресурсов, обмен ресурсами
Таблица лидеров	Система вознаграждений	Участие в конкурсе
Уровень	Отслеживание прогресса	Переход на следующий уровень, повышение уровня
Прогресс	Отслеживание прогресса	Получение информации о прогрессе в игре
Статус	Отслеживание текущего статуса	Получение информации о текущем состоянии
Команда	Командная Работа	Участие в групповых мероприятиях
Время	Лимит времени	Осуществление деятельности в течение определенного времени
Правила	Правила игры	Следует правилам
Ресурс, Сообщение	Обратная связь	Получение заключения компетентной стороны
Сообщение	Коммуникация	Отправка сообщения, получение сообщения
Различные элементы	Вызов / Миссия / Приключение	Завершение миссии
Ресурс, Комбо	Спрятанное сокровище	Охота за сокровищами
История	История	Создание и вход в другую реальность

Исследования ученых показывают, что применение элементов геймификации оказывает положительное влияние на обучение учащихся в контексте, в котором они были реализованы. Гистограмма на рисунке 1 (рис. 1) показывает список и частоту результатов обучения, которые были изучены и которые получают положительное влияние после применения элементов геймификации. Этот список был составлен на основе частоты каждого результата обучения из исследованных научных материалов.

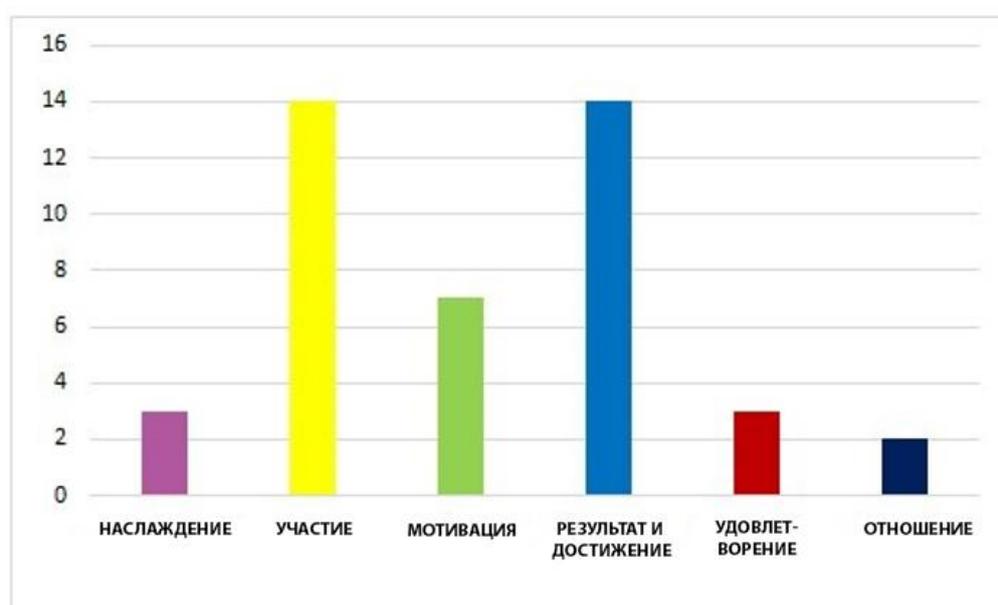


Рисунок 1. Список и частота результатов обучения, изученных в рецензируемых статьях, которые получают положительное влияние от геймификации

Гистограмма на рисунке 4 ясно показывает, что геймификация оказала положительное влияние на результаты обучения в различных формах, таких как удовольствие, вовлеченность, мотивация (внутренняя и внешняя), результаты и достижения, удовлетворенность и отношение. В результате в центре внимания исследования чаще всего оказывается вовлеченность, а также результаты и достижения студентов, за которыми на третьем месте следует мотивация. Высокая частота этих трех пунктов еще больше укрепила предположение о том, что геймификация может сделать учащихся более

активными и мотивированными при обучении, позволяя им достигать наилучших результатов при оценке обучения. Это также показывает, что геймификация может стимулировать вовлеченность и мотивацию (внутреннюю и внешнюю), которые необходимы для создания качественного обучения. Между тем, геймификация также оказала положительное влияние на удовольствие, удовлетворенность и отношение, но все еще проводится мало исследований для дальнейшего подтверждения этих результатов.

После изучения результатов обучения, которые получили положительное влияние от применения геймификации, так же рассмотрены элементы геймификации, которые были применены для обеспечения положительных эффектов. Для представления результатов, связанных с элементами геймификации, полученных из исследований были преобразованы их в форму столбчатой диаграммы на рисунке 2 (рис. 2), где они расположены в порядке убывания.

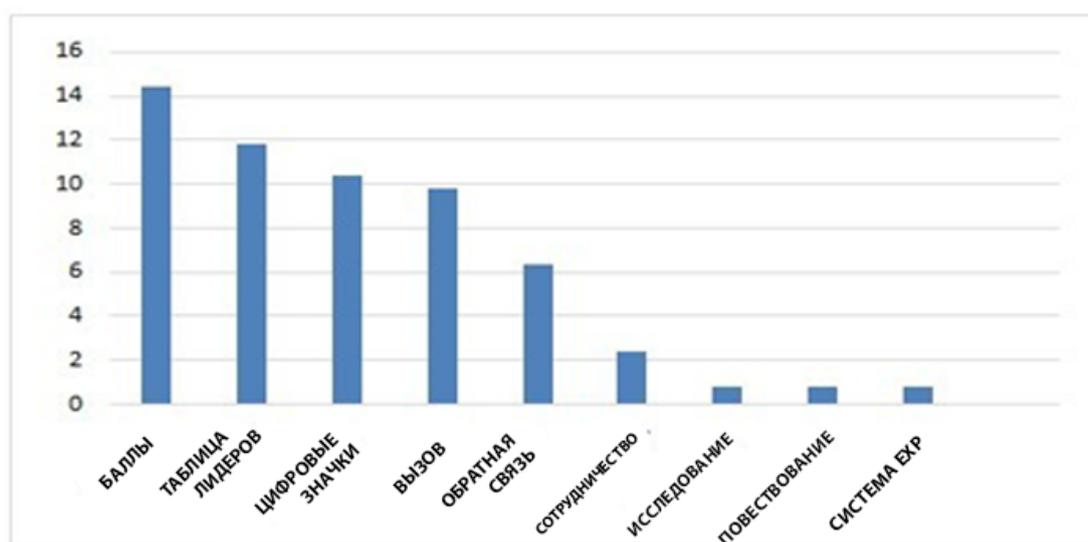


Рисунок 2. Элементы геймификации, примененные в рассмотренных исследованиях

Ссылаясь на приведенную выше гистограмму, элементами геймификации, которые применялись во многих исследованиях, являются

элементы баллов и вознаграждений (15 исследований). Баллы и награды - это то, что присуждается учащимся, которые успешно правильно ответили на вопросы, выполнили заданное задание и любое другое достижение. Кроме того, элементы таблицы лидеров, цифровые значки и задачи также зафиксировали высокую частоту использования в рассмотренных исследованиях, которые составили 19, 20 и 21 исследований соответственно. Таблица лидеров - это раздел для отображения рангов игроков, в то время как цифровой значок является одной из форм награды. С другой стороны, вызов – это набор правил игры, который делает игру более сложной.

Основываясь на цифрах, это означает, что эти четыре элемента (награды, таблица лидеров, цифровой значок и вызов) необходимы, при создании эффективные обучающие мероприятия по геймификации, как это было доказано этими исследованиями [19,20,21]. Эти элементы также рассматриваются как тесно связанные друг с другом и используемые вместе в поддержку учебной деятельности. Например, элемент «баллы» поддерживается таблицей лидеров для отображения рангов и очков, вызов использует цифровой значок в качестве награды и так далее. В то время как элементы системы опыта, повествование и исследование рассматривались исследователями как меньший выбор, но они также оказали положительное влияние на их исследования [22].

Таким образом, геймификация способна оказывать положительное влияние на обучение учащихся с различных точек зрения, особенно на их достижения, вовлеченность и мотивацию в учебную деятельность. Лишь немногие исследователи сообщили о негативных последствиях геймификации в своих исследованиях. Элементами геймификации, которые часто применялись в преподавании и обучении в рассмотренных исследованиях, были баллы и награды, таблицы лидеров и цифровые значки. Основываясь на этом, можно сделать вывод, что при использовании определенных игровых

элементов геймификация может принести положительный результат в обучении студентов, и она должна широко применяться учителями или преподавателями в процессе преподавания и обучения [25].

1.2. Возможности геймифицированных платформ для обучения программированию

Существует множество программ и сайтов для геймификации. Некоторые из них являются сетевыми (облачными сервисами) и не требуют установки специального программного обеспечения и обеспечивают доступ в любое время и из любого места. Среди самых популярных инструментов геймификации: Socrative, Kahoot!, FlipQuiz, Duolingo, Ribbon Hero, ClassDojo и Goalbook. BadgeOS™ и его надстройка BadgeStack — это бесплатный плагин для WordPress, который автоматически создает различные типы достижений и страницы, необходимые для настройки системы значков.

Mozilla Open Badges Project — это проект, цель которого — обеспечить идентификацию и признание приобретенных знаний и навыков учащихся за пределами классной комнаты — результатов неформального обучения. Через проект Mozilla Open Badges любой может выдавать выигрыши и отображать значки через общую техническую инфраструктуру (Mozilla Open Badges).

Чтобы ограничить количество рассмотренных обучающих игр и платформ, была применена стратегия систематического поиска с критериями включения и исключения. На первом этапе был создан длинный список онлайн-платформ и игр с использованием двух разных поисковых ключевых слов («изучение программирования онлайн» и «онлайн-игра по программированию») в поисковой системе Google. Еще один поиск был проведен в Play Market Store для Android, чтобы включить мобильные приложения в обзор. Чтобы еще больше сократить количество платформ для

рассмотрения и исключить нерелевантные платформы, были применены критерии исключения и включения.

В ходе этого исследования было определено несколько игровых сред, чтобы провести сравнительный анализ и выбрать идеальный инструмент для включения во вводный курс программирования (табл.2). Критерии учитывают несколько аспектов, таких как: тип платформы, среды (мобильная или веб); поддерживаемые языки программирования; реализованные игровые элементы.

Таблица 2. Сравнительная матрица игровых сред для компьютерного программирования

Название платформы	Тип платформы	Поддерживаемые языки программирования	Геймифицированные элементы	Деятельность за пределами класса
SoloLearn	Мобильный и веб	Python, C++, Java, JavaScript, C#, PHP, Swift, Ruby, jQuery C, HTML, CSS, SQL	Очки, трофеи, уровни, соревнование сверстников, области социального взаимодействия, лидерство, доски и значки	Дискуссионные форумы
CheckiO	веб	Python, JavaScript	Уровни, испытания, таблицы лидеров, социальные сети; Зоны взаимодействия, очки	Блоги и форумы
Codin-Game	веб	Java, Python, JavaScript, Swift, Ruby, Rust, C#, C++, C	Испытания, списки лидеров, социальное взаимодействие, очки, миссии и награды	Блоги и форумы
ViLLe	Мобильный и веб	Различный	Неограниченное количество попыток, визуальные подсказки и обучающие материалы для начального развития навыков.	Блоги и форумы
Stack Overflow	веб	Различный	Статистика выступлений, значки, очки опыта, помощь других участников, каналы связи	

EasyLogic	веб	Javascript, графика, блоки	Очки, Трофеи, Неограниченное количество попыток, Визуальные подсказки и подсказки обратной связи	
Funprog	веб	Кодовые блоки и опросы	Уровни обучения, награды, очки, игровые ограничения, списки лидеров и очки, Составление отчетов	Дискуссионные форумы
Code Combat	веб	Python, JavaScript, HTML и CSS	Уровни, Аватары, Виртуальные миры, Очки, Испытания, Несколько попыток, Таблицы лидеров, Клань, Значки, Миссии, Награды, Прогресс, Монеты, Персонализация, Социальная зона и рейтинги	Дискуссионные форумы
GamiLab	веб	Различный	Статистика, значки, очки опыта, помощь других участников, каналы связи, неограниченное количество попыток, визуальные подсказки и подсказки обратной связи	Дискуссионные форумы
CodeGym	веб	Java	Испытания, игровые ограничения, способы заработать или потерять очки, персонажи, сюжетные сценарии, неограниченное количество попыток, значки, команды, статистика производительности, списки лидеров, области социального взаимодействия, уровни. доступ к заблокированному контенту	Поиск работы, дискуссионные форумы и блоги с соответствующими статьями

CodeHS	веб	Java, Javascript, HTML, Python, CSS, SQL, Karel и React	Очки, значки, индикаторы прогресса и уровни	Планы уроков, руководства по проблемам, руководства по решениям и интерактивные упражнения без программирование
Co-decademy	веб	Java, Python, PHP, JavaScript, Ruby, HTML, CSS, C# и C++	Значки или медали за выполнение упражнений, Запись общего балла пользователя, доска лидеров и очки	Дискуссионные форумы, Глоссарий по курсу
Code Avengers	веб	Python, Javascript и HTML	Видео, Значки, Очки, Уровни, доска Лидеров и интерактивные игры	Учебные материалы и планы уроков
Ozaria	Веб	Python	Уровни, Аватары, Виртуальные миры, Очки, Испытания, Несколько попыток, Таблицы лидеров, Значки, Миссии, Награды, Прогресс, Персонализация, Социальная зона и рейтинги	Учебные материалы и планы уроков, дискуссионные форумы,

Codecademy является онлайн-платформой для обучения программированию, которая интегрирует геймификацию. Их самопровозглашенная миссия состоит в том, чтобы переосмыслить образование «снизу вверх», используя больше подсказок от современных технологических новаторов в создании привлекательного образовательного опыта. Платформа поставляется в бесплатной и профессиональной версиях. Зарегистрированные пользователи могут выбирать из множества курсов, в первую очередь ориентированных на изучение различных языков программирования (например, Javascript, Python 3, Java). Каждый курс состоит

из нескольких заданий, в которых последовательно и структурированно излагается конкретное содержание обучения. Панель мониторинга предоставляет обзор запущенных курсов, а также информацию о прогрессе пользователя в них (рис. 3).



Рисунок 3. Панель управления Codecademy «Мои курсы» - Обзор курса и прогресс

Задачи в Codecademy разделены на два или три раздела, в зависимости от содержания сессии. На рис. показан интерактивный сеанс. Пользователям предоставляется обучающий контент и инструкции в левой части экрана. Правая часть экрана используется для визуализации вводимых пользователем данных. На рис. 5 показана задача курса, в которую в дополнение к визуализации и инструкциям встроен редактор кодирования. В этом примере концепция циклов в программировании преподается путем визуализации ранее закодированной пользователем циклической программы с помощью цветных плиток.

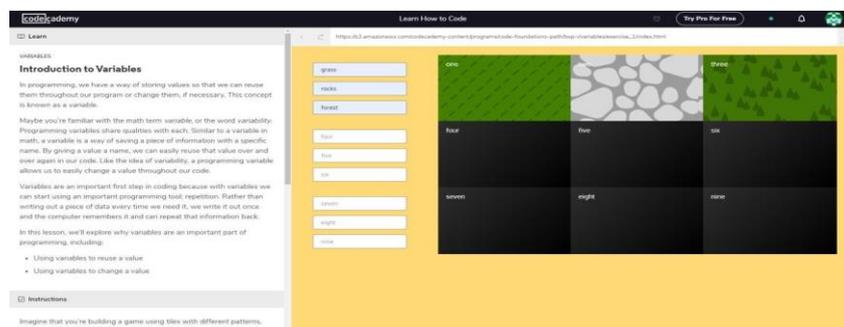


Рисунок 4. Интерактивный урок в Codecademy

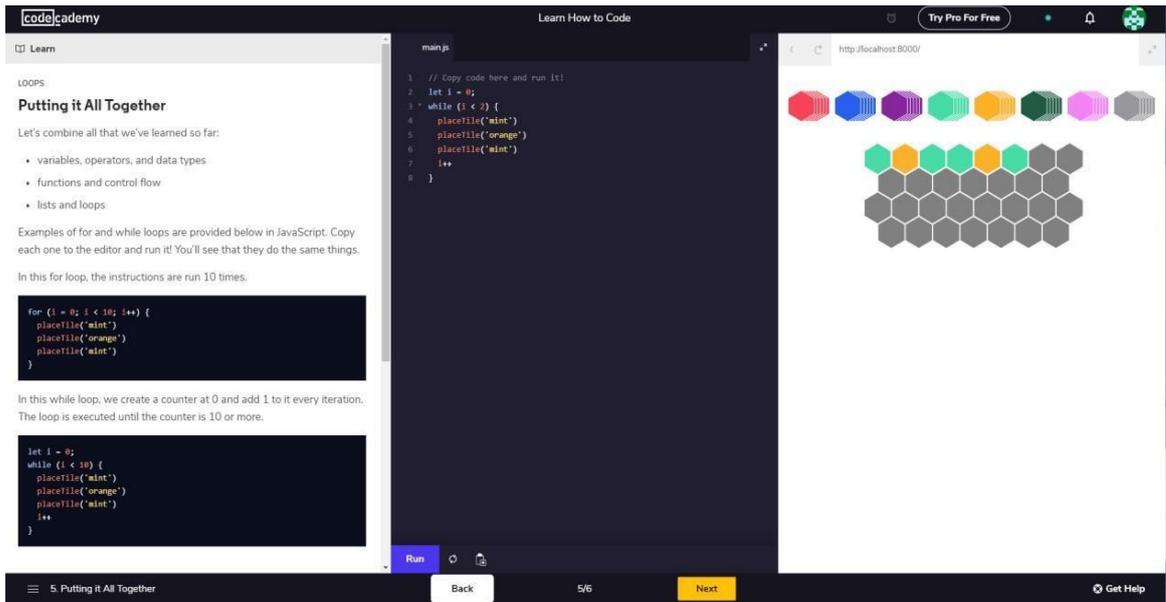


Рисунок 5. Интерактивный урок с редактором кодирования в Codecademy

В отличие от Codecademy и Codewars, CodeCombat использует полноценный игровой подход к обучению навыкам программирования. Игра объединяет компоненты, обычно используемые в ролевых играх (RPG): пользователи выбирают предпочитаемого персонажа из набора из персонажей с разной квалификацией, так называемых «героев» (рис. 6).



Рисунок 6. Персонажи в CodeCombat

Такой дизайн игры способствует балансу между навыками игрока и задачей, что особенно подчеркивается в обучающих играх. Чтобы пройти подземелье, игроки должны ввести функциональный код, который управляет персонажем игрока для выполнения определенных задач(рис. 7).



Рисунок 7. Цели подземелий CodeCombat

В начале каждого подземелья игроку предоставляются четкие цели и примеры, обладающие развитыми функциями автозаполнения и автозамены. Таким образом, обеспечивается мгновенная обратная связь и непрерывная поддержка во время обучения и опыта игры (рис. 8).

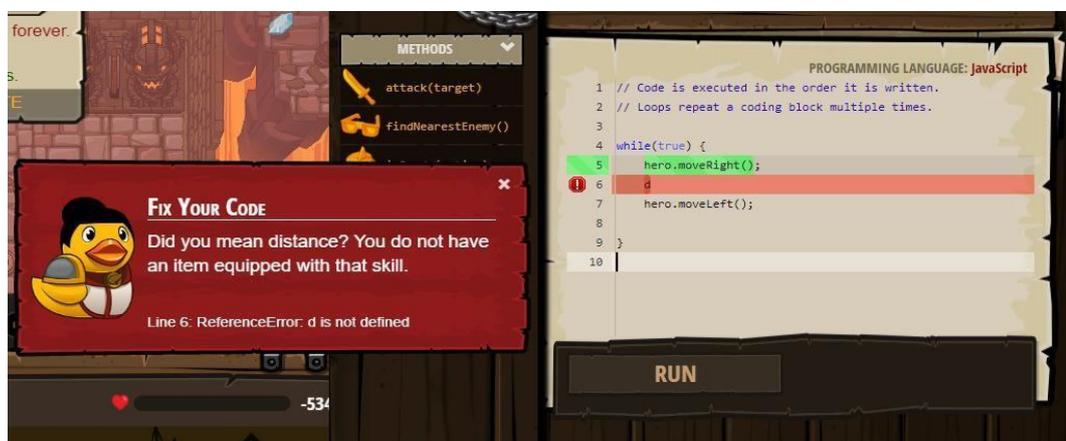


Рисунок 8. Автозамена и моментальная обратная связь в CodeCombat

Как только игрок заканчивает писать инструкции по кодированию, выполнение программы запускает анимацию, основанную на заданных инструкциях: персонаж игрока может, среди прочего, перемещаться, обнаруживать врагов и сражаться с ними. Эта игровая механика дает игроку чувство контроля, поскольку его или ее инструкции напрямую влияют на ход игры.

В отличие от обучения компьютерному программированию, другой подход к ознакомлению начинающих студентов с темами информатики заключается в обучении их алгоритмическому мышлению. Сначала необходимо развить у учащихся способность мыслить алгоритмически, прежде чем обучать программированию. Общее определение алгоритмического мышления - это «мыслительный процесс, связанный с формулированием проблем и их решений таким образом, чтобы решения были представлены в форме, которая может быть эффективно выполнена агентом обработки информации» [27].

Геймификация обучения с применением Scratch позволяет развивать творческое мышление, проектировать эффективное предметное взаимодействие; формировать навыки системного анализа. Участниками игрового сообщества могут быть не только игроки и педагоги, но и родители. Применять Scratch можно как на уроках математики, программирования, русского языка, истории, так и на уроках изобразительного искусства (азы дизайна). С позиции геймификации важна возможность конструировать и реализовывать интерактивные истории, организовывать игры в обучении с применением анимированных персонажей.

Scratch — это бесплатное веб-приложение и визуальный язык программирования, разработанный MIT Media Lab. Его цель состоит в том, чтобы обучить базовым концепциям программирования в наглядной форме, и в первую очередь он ориентирован на детей в возрасте от 8 до 16 лет. Язык

предоставляет блочный интерфейс, который напоминает типичные функции «реальных» языков программирования, такие как циклы, условные выражения и переменные. Программа Scratch написана путем склеивания различных блоков кода вместе и может быть расширена с помощью различных типов носителей, таких как изображения, звуки и даже реальные аппаратные датчики (например, Lego Mindstorms)[28]. Пользовательский интерфейс Scratch состоит из трех разделов: палитры блоков, области кодирования и области сцены, в которой визуализируется программа (рис. 9).

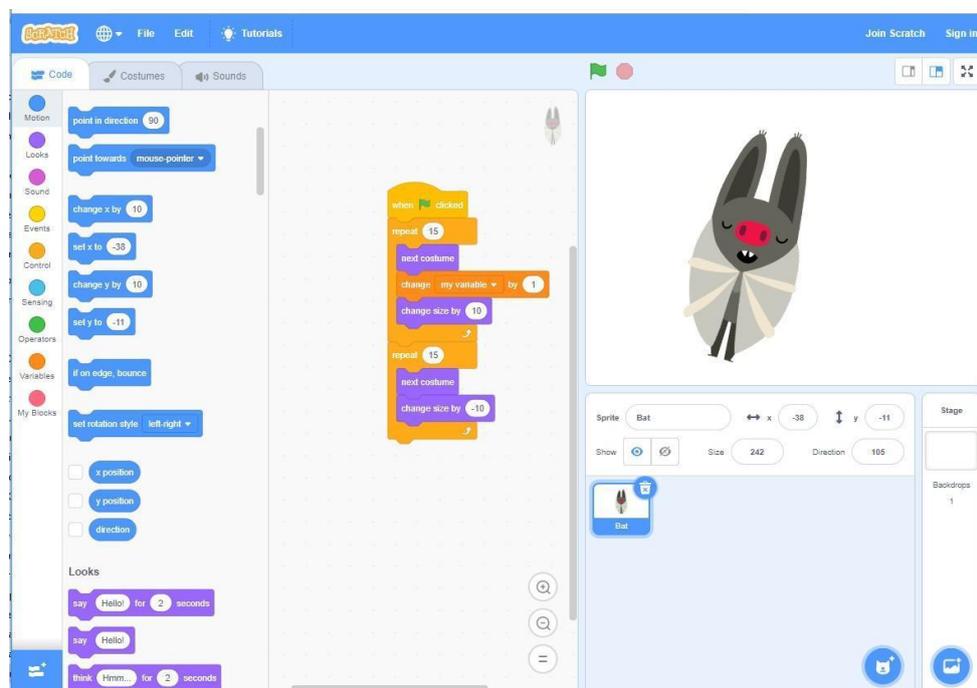


Рисунок 9. Пользовательский интерфейс Scratch

Coding Planets [29] — это мобильное приложение, доступное в Google Play Store. Она знакомит игроков с алгоритмическим мышлением и основными понятиями программирования, решая различные головоломки. Игровой мир основан на уровнях, каждый из которых посвящен различным темам алгоритмического мышления, например функциям и циклам. Цель каждой головоломки - переместить маленького робота по игровому полю, избегая препятствий и собирая все алмазы. Чтобы управлять роботом на игровом поле, игрок должен давать инструкции в виде блоков, которые служат различным

целям. Например, блоки предназначены для перемещения, сбора алмазов и поворота робота. Типичная доска-головоломка и пользовательский интерфейс Coding Planets показаны на рис. 10. Уровень сложности увеличивается с каждой головоломкой. На продвинутых уровнях игроки знакомятся с функциями, циклами и рекурсией, которые также могут быть использованы в качестве инструкций для робота. На протяжении всех уровней основная игровая механика (например, перемещение, сбор алмазов) никогда не меняется, что может привести к определенной степени разочарования и, следовательно, к потере вовлеченности.

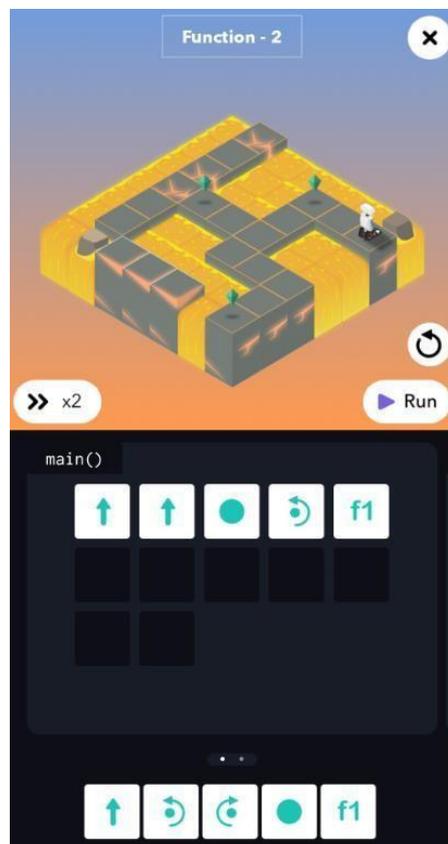


Рисунок 10. Головоломка с Coding Planets

Algorithm City обучает алгоритмическому мышлению, перемещая фигуру (в данном случае пингвина) по игровому полю, чтобы собирать виджеты (например, монеты). Единственное обнаруженное отличие носит визуальный характер (рис. 11): игрок перемещает пингвина вместо робота, основная задача

- собирать монеты вместо бриллиантов, кнопки для запуска программы находятся в разных местах экрана. В целом Coding Planets кажется более современным (цвета и стили), чем Algorithm City.



Рисунок 11. Algorithm City

Spritebox: Code Hour - это мобильное приложение, свободно доступное в Google Play Store. Жанр игры представляет собой смесь прыжка и бега, где препятствия преодолеваются путем написания программ. Такие компоненты, как блоки инструкций и область кодирования, спроектированы аналогично Algorithm City и Coding Planets, но они оснащены некоторыми дополнительными функциями. Игрок сначала настраивает игрока, выбирая причёску, цвет, пол и одежду. Как и в играх Jump and Run (например, Super Mario), игровой мир состоит из разных уровней, которые имеют различные темы (например, пустыня, луга, лед). Уровни снова начинают расти от простых к более сложным. Цель каждого уровня состоит в том, чтобы перемещать персонажа по уровню, преодолевая препятствия и собирая монеты, пока персонаж не достигнет финишной черты. В отличие от обычных игр «Прыгай и беги», препятствиями являются не NPC, которые пытаются лишить персонажей жизни, а, например, высокие края, которые слишком высоки, чтобы на них прыгать (рис. 12). Чтобы преодолеть такие препятствия, игрок должен написать небольшую программу, которая определяет, куда ставить новые блоки, чтобы персонаж мог двигаться вперед.

Spritebox, благодаря своему дизайну, игровому процессу и игровым персонажам, вероятно, ориентирован на более молодую аудиторию (6-12 лет), но в отличие от Algorithm City и Coding Planets, он содержит больше функций и, следовательно, обеспечивает больше развлечений.



Рисунок 12. Algorithm City

Озария разработана с нуля, чтобы эффективно обучать, увлечь и привить любовь к учебе (рис. 13).

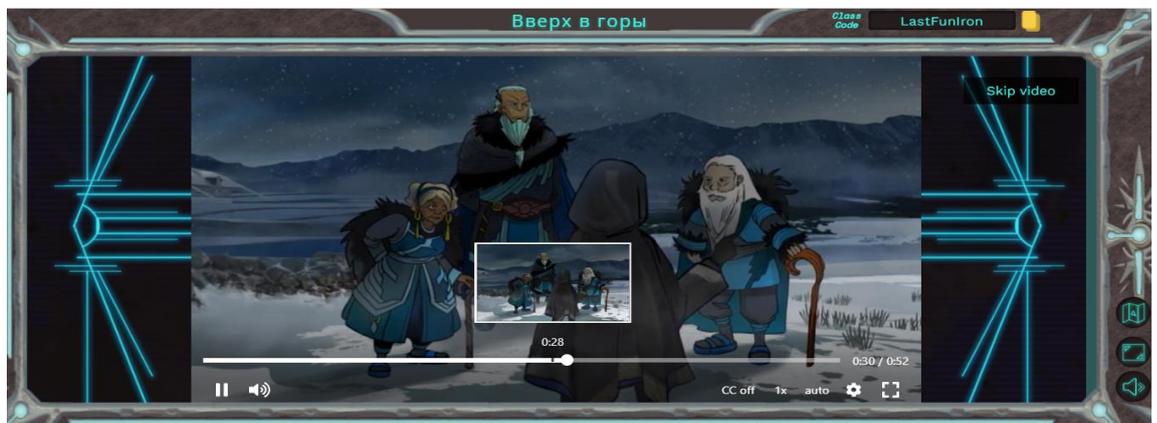


Рисунок 13. Видео контент Ozaria

Предметом является обучение компьютерному программированию на Python или JavaScript. Игра активно учит, так как игроки управляют своим персонажем с помощью кодирования и немедленно получают соответствующую обратную связь, используя доступное повествование, создавая состояние «потока» (рис. 14).

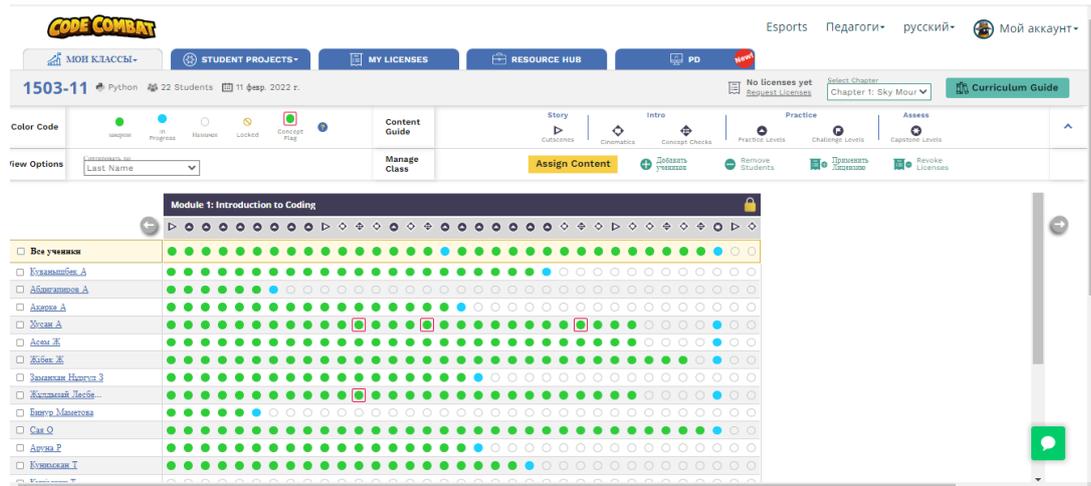


Рисунок 14. Страница учителя

Когда ученики повышают уровень своих героев в Озарии, они также повышают уровень своего обучения. Учащиеся набирают Python или JavaScript, решая проблемы с помощью вычислительного мышления, и развивают навыки 21-века, такие как критическое мышление, инициатива и уверенность (рис. 15).

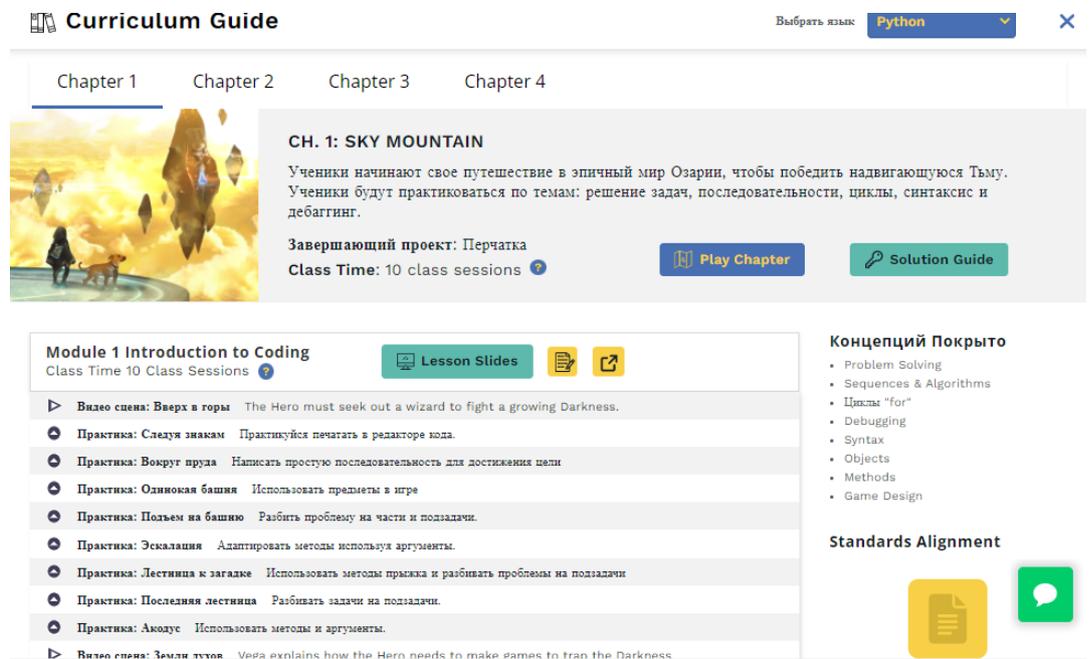


Рисунок 15. Учебная программа Ozaria

Несмотря на краткое представление теоретических терминов программирования для расширения декларативных знаний, основное

внимание в игре уделяется отработке и автоматизации навыков программирования с помощью конструктивистского подхода к конкретным уровням, ориентированным на цели обучения, которые требуют программного решения (рис. 16-17).

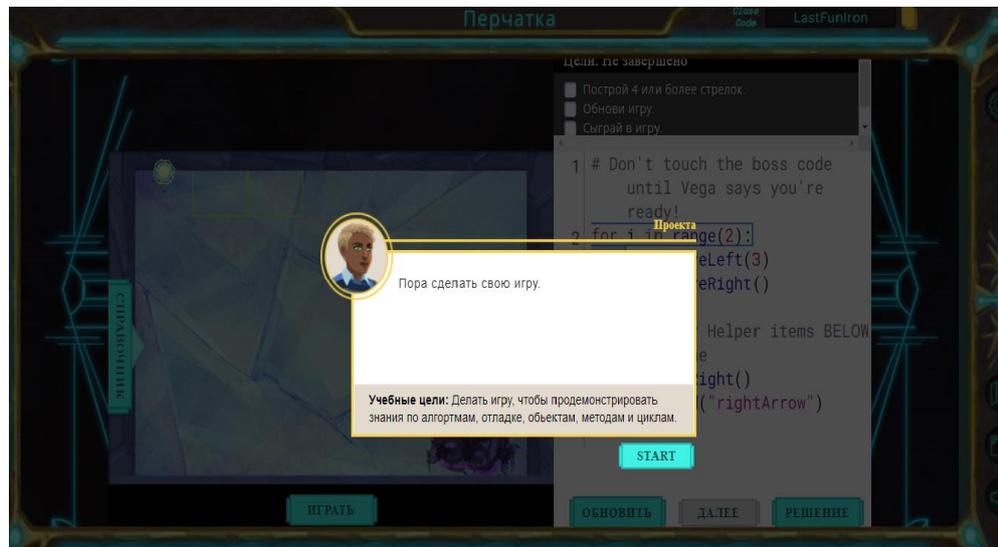


Рисунок 16. Задания Ozaria

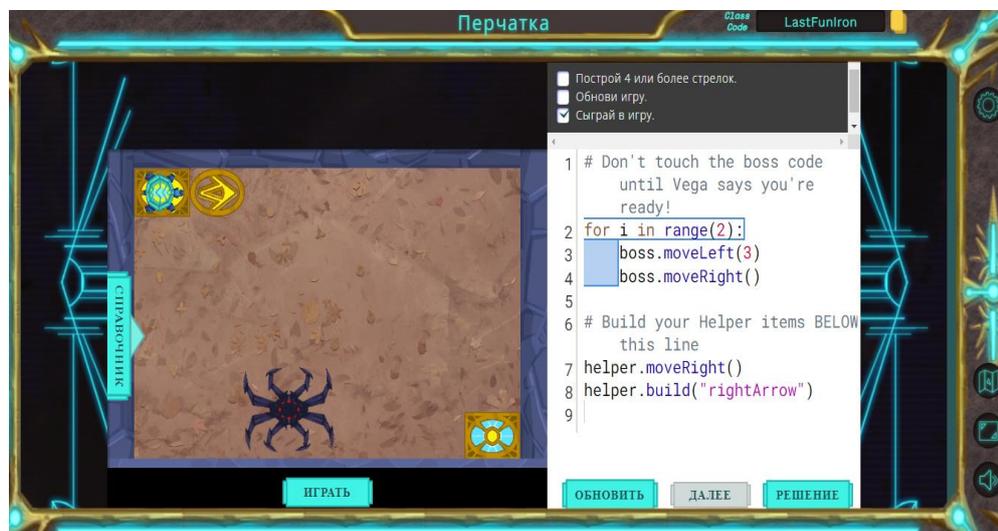


Рисунок 17. Программный код Ozaria

Учителя будут иметь доступ к данным, таким как руководства по внедрению, планы уроков и рубрики (рис. 18).

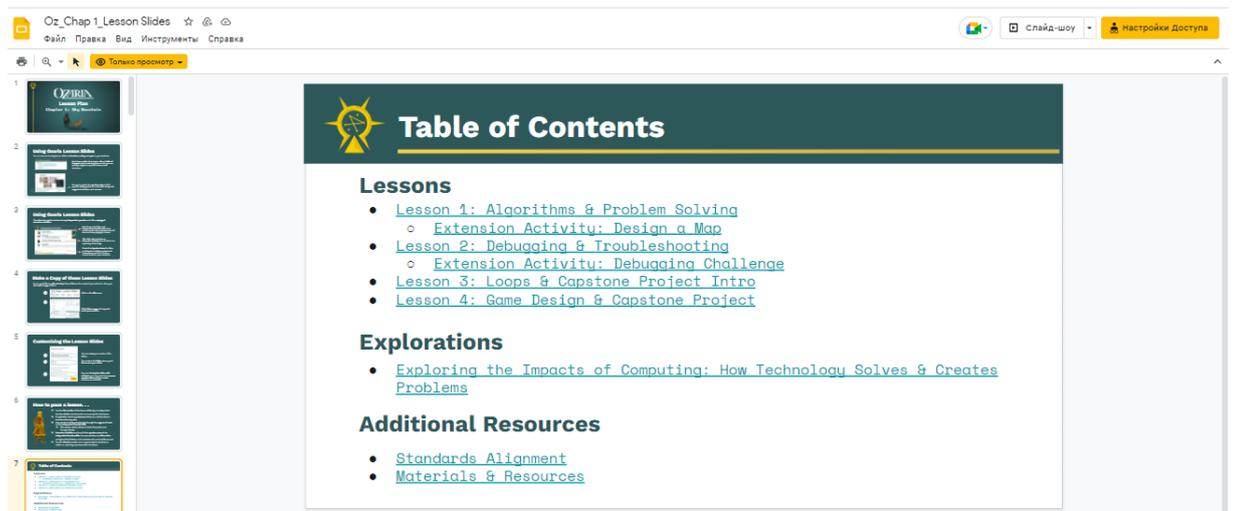


Рисунок 18. Методические учебные материалы Ozaria для учителя

GamiLab[33] - это еще одна платформа, где можно создавать свои собственные игры и делиться ими со своими учениками. Здесь можно создавать свои собственные игры, а также воссоздавать старые игры, созданные другими людьми, а затем делиться рип-кодом со своими учениками. Gamilab использует игровой элемент, чтобы мотивировать и привлекать игроков к обучению в интерактивном режиме.



Рисунок 19. Игра Велосипедная гонка

Велосипедная гонка заключается в том, чтобы как можно скорее добраться до финиша. Игрок должен пройти по трассе и ответить, как минимум на восемь вопросов. Если игрок отвечает правильно, он получает

больше энергии, может двигаться быстрее и быстрее достигает финишной черты. Игрок приближается к одной из восьми остановок, содержащих вопросы по дорожке. Надо держаться темп, точность и знания (рис. 19).

Велосипедная гонка идеально подходит для викторин. Его легко создать - это займет всего пять минут (рис. 20).

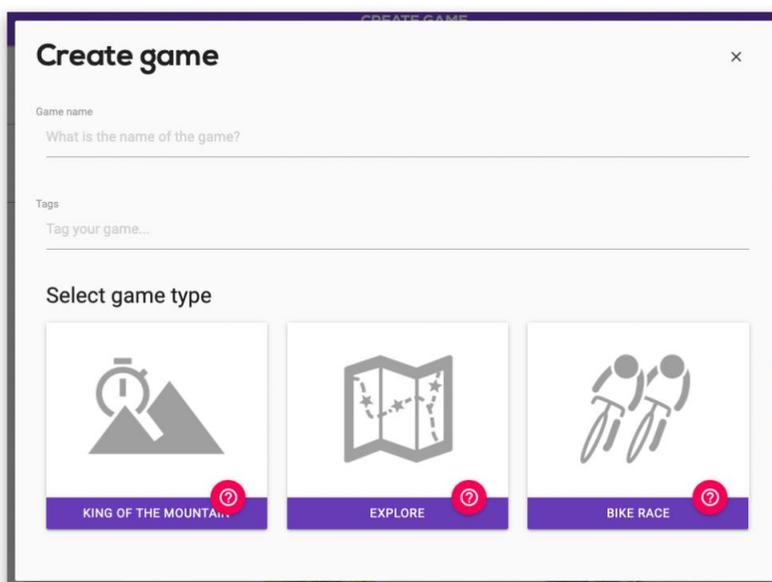


Рисунок 20. Типы игры Gamilab

Заполняется название и квест (цель) игры. Добавляется несколько тегов для упрощения поиска созданной игры; назначается целевая аудитория, чтобы определить, кто будет играть в игру. А также можно добавить любых соавторов, с которыми будете работать при создании игры. Для создания викторины игры выбирается «Игровая доска» и нажимается кнопка «Вопросы» (рис. 21).

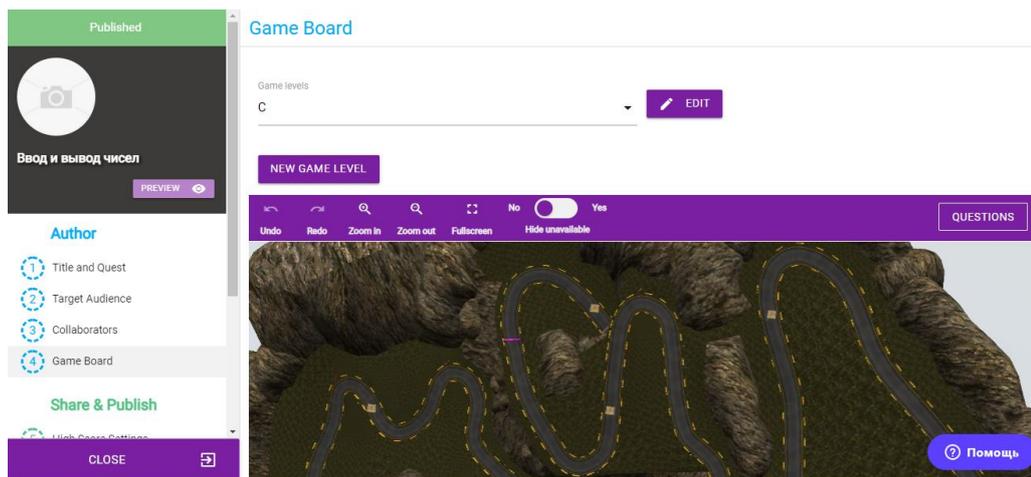


Рисунок 21. Игровая доска

Мы рекомендуем не добавлять более трех неправильных ответов, поскольку время, потраченное на игру в режиме «Велогонка», имеет решающее значение (рис. 22).

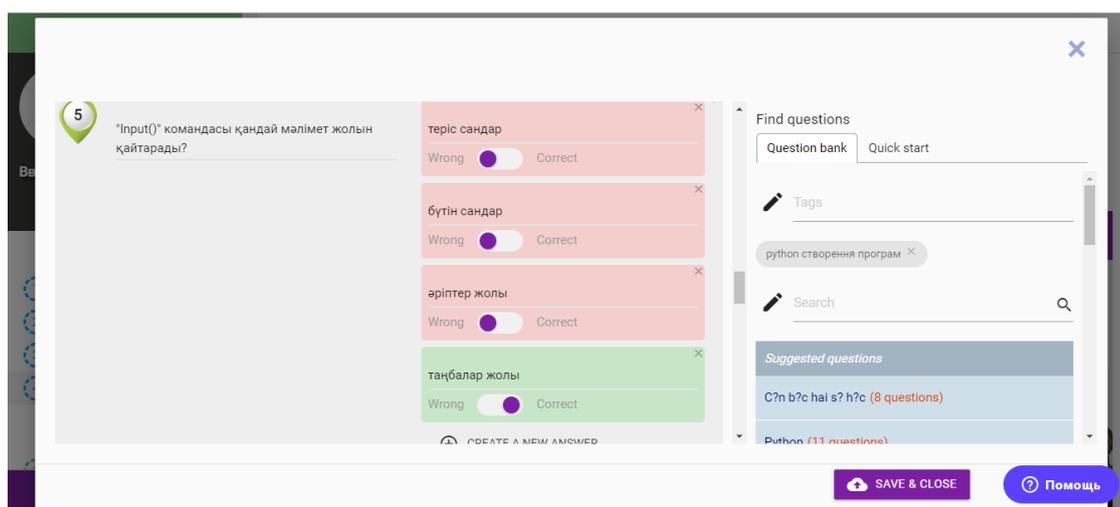


Рисунок 22. Окно для создания задания

А также можете создать игру «Велосипедная гонка», используя вопросы, созданные другими. Для этого нужно выбрать из предложенных вопросов в списке. Это вопросы и ответы, созданные другими пользователями, и предложения, основанные на названии игры и добавленных нами тегах.

Тип игры King of the Mountain — это достижение цели — вершины горы — как можно быстрее. Этот тип игры отлично подходит для быстрых и простых викторин. Игрок достиг вершины горы — и конец игры.

На пути к вершине игрок сталкивается как минимум с семью вопросами. Игрок должен правильно ответить на все вопросы, чтобы добраться до вершины.

Некоторые вопросы более опасны, чем другие, и игроку, возможно, придется вернуться к началу, если ответ будет неверным — так же, как Змеи и лестницы (рис. 23).

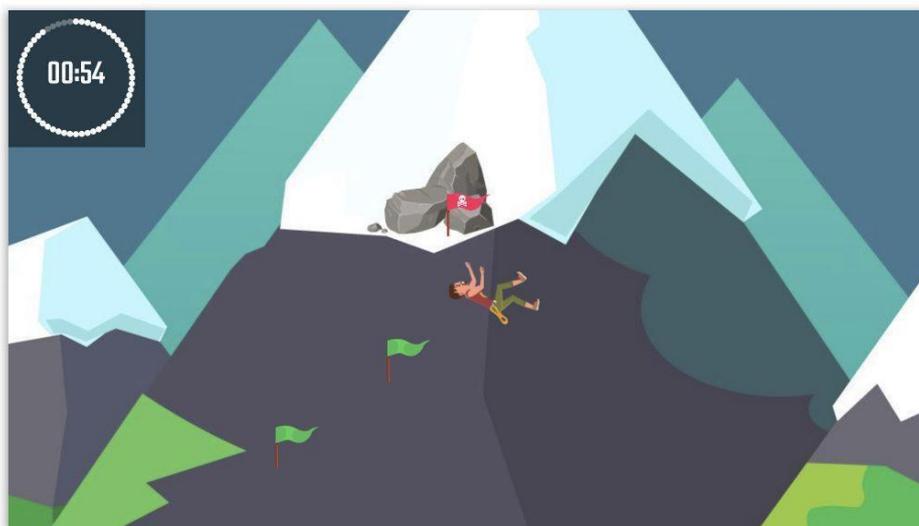


Рисунок 23. Игра Король горы

Правильный ответ на участке горы с красным флагом имеет решающее значение. Как только игрок прошел эту точку и достиг вершины, игрок получает обратную связь о времени, затраченном на подъем. Также отображается список самых высоких результатов, достигнутых в игре. В игровом режиме «Исследование» игрок движется от начала к цели в настольной игре, пытаясь набрать по пути как можно больше очков. Дело не в том, чтобы прийти к финишу первым. Речь идет о решении задач наилучшим образом (рис. 24).



Рисунок 24. Игра «Исследование»

Набирая очки, игрок будет следовать по пути от одной задачи или ресурса к другой. При создании игры «Исследуй» мы можем выбирать между несколькими типами задач или, даже упростить ее с помощью некоторых викторин. В игре «Исследуй» можно создать задачу в интерактивном видео.

Для проведения исследований в диссертационной работе на тему «Использование геймификации при обучении учащихся программированию на python» я выбрала платформу GamiLab норвежской It компании ([https://gamilab.com/ Gamilab](https://gamilab.com/)).

Сначала я зарегистрировалась на сайте GamiLab как учитель (рис. 25).

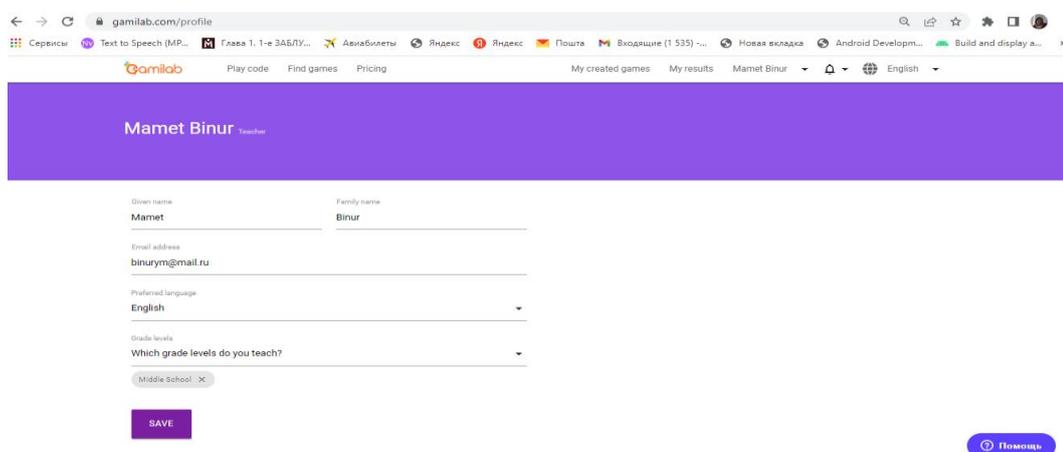


Рисунок 25. Моя страница аккаута в GamiLab

Было создано несколько игр в GamiLab в соответствии с тематикой и целями обучения приведены в таблице 4 (учебный план урока информатики)(рис. 26).

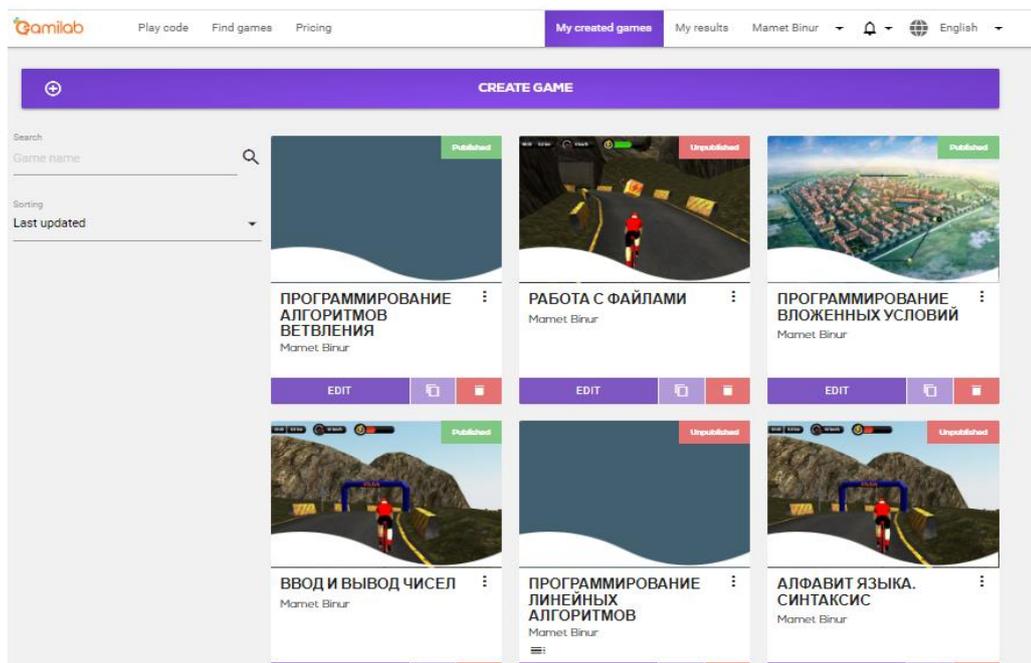


Рисунок 26. Игры GamiLab для обучения Python

Доступ к каждой созданной игре GamiLab предоставляется по специальным кодам (рис. 27).

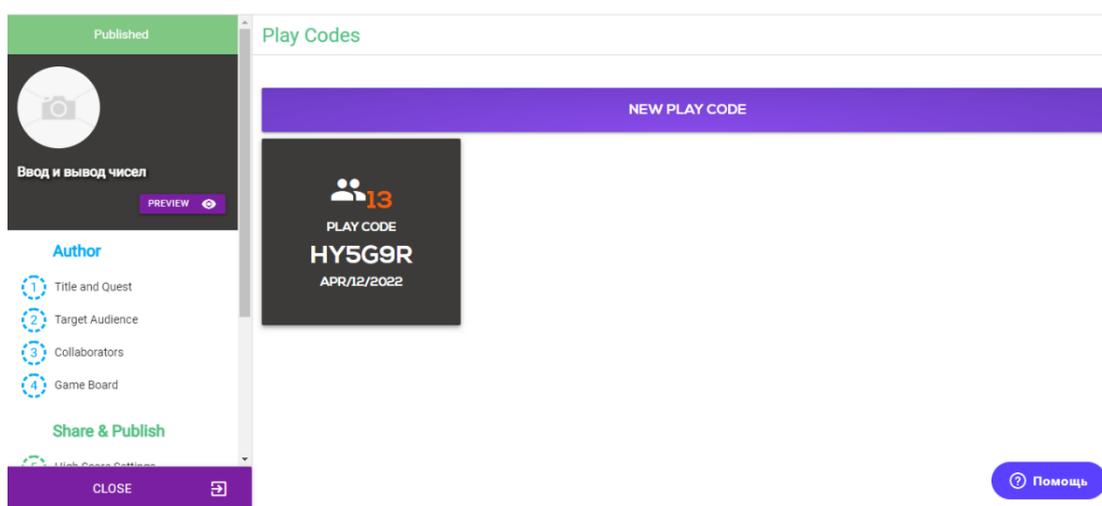


Рисунок 27. Код игры

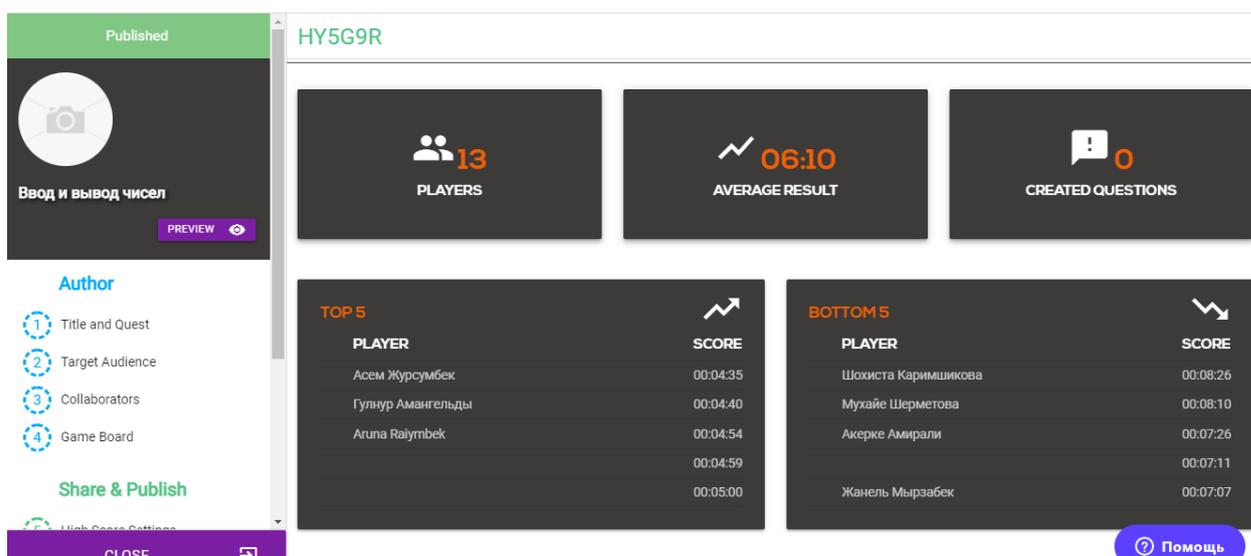


Рисунок 28. Результаты игры

В результате 13 учащихся группы 1503-11 зарегистрировались для участия в этих играх и играли онлайн по предоставленным ссылкам (рис. 28):

<https://gamilab.com/play/23580249-e833-4bb5-98b9-8b3db9eb2ba4>

<https://gamilab.com/play/d23f34a9-7402-4371-a274-cd6dfe765538>

<https://gamilab.com/play/36c86232-fc41-4d41-8d75-e12b1c4c8e37>

Особенность игр GamiLab в том, что они адаптированы для мобильных приложений и в них легко играть. Результаты игры показаны на рисунке 28.

Выводы по первой главе

В первом разделе диссертации раскрываются теоретические основы применения технологий геймификации на уроках информатики в старшей школе, а также возможности геймифицированных платформ для обучения программированию

Игровое обучение — это метод, с помощью которого учителя могут создавать учебную среду, стремящуюся мотивировать учащихся, удерживать их внимание во время разработки курса и позволять учащимся активно участвовать в достижении значимого обучения. Точно так же игра является инструментом взаимодействия с другими, способствуя самопознанию, ответственности, целеустремленности и самореализации учащегося. В процессе обучения компьютерному программированию, столкнувшись с проблемами в этой области, также были приняты стратегии, основанные на играх, чтобы способствовать более высокому уровню вовлеченности, участия, мотивации и поощрения, что позволяет учащимся достичь своих целей обучения и уменьшить отказ и уклонение от выполнения задач.

Результаты исследования показали, что геймификация способна оказывать положительное влияние на обучение учащихся с различных точек зрения, особенно на их достижения, вовлеченность и мотивацию в учебную деятельность. Лишь немногие исследователи сообщили о негативных последствиях геймификации в своих исследованиях. Элементами геймификации, которые часто применялись в преподавании и обучении в рассмотренных исследованиях, были баллы и награды, таблицы лидеров и цифровые значки. Основываясь на этом, можно сделать вывод, что при использовании определенных игровых элементов геймификация может принести положительный результат в обучении студентов, и она должна широко применяться учителями или преподавателями в процессе преподавания и обучения.

Во время исследования диссертации были проведены серия занятия в Южно-Казахстанском педагогическом университете. В ходе занятия в качестве метода геймификации данной компьютерной игры Озария студентам 1 курса, обучения (группа: 1503-11) по образовательной программе «6В01503 - Подготовка учителей информатики», были ознакомлены с данной компьютерной игрой Озария на занятиях «Языки программирования: Python». На программу «Озария» зарегистрировался 21 студент, которые успешно выполнили задания главы 1 (рисунок 19).

Учебный план Озарии, показанный в таблице 5, полностью охватывает темы и цели обучения школьной программы по информатике, указанные в таблице 4. Поэтому компьютерная игра «Озария» может широко использоваться школьниками для изучения Python.

Глава 2. Практические аспекты применения технологий геймификации при обучении программированию старшеклассников

2.1. Методические планирования занятий по программированию на Python

Курсы программирования — это один из курсов компьютерных языков, который существует уже более 100 лет. Существуют различные типы языков программирования, которые развились во времени, что позволяет программистам использовать их при разработке программ для приложений или программного обеспечения, а также любых других типов компьютерных инструкций, которые могут быть выполнены. Несмотря на то, что большинство языков программирования имеют сходство, каждый из них имеет свой собственный синтаксис программирования и среду. Когда программист понимает основные правила и условия языков, их структуру и синтаксис, он готов разработать исходный код и скомпилировать код в машинный язык, понятный компьютеру. Язык программирования является одним из основных предметов, изучаемых студентами компьютерных наук в высших учебных заведениях.

Алгоритмизация и программирование – одно из важнейших направлений школьного курса информатики и ИКТ. Раздел по направлению программирования в типовой учебной программе по предметам «Информатика» и «ИКТ» основного среднего образования по обновленному содержанию начинается с первого класса и язык программирования Python обучается с 6 по 9 класс в разделе «Алгоритмы и программирование» (табл. 4).

Раздел *«Алгоритмы и программирование»* охватывает изучение практически всех разделов школьной информатики, связанных с изучением алгоритмизации и программирования в школе. Содержание раздела *«Алгоритмы и программирование»* представлено укрупненным модулем: *«Программирование алгоритмов на языке программирования Python»* (6-10 класс)

Таблица 4. Фрагмент типовой учебной программы по предмету «Информатика» для 6-10 классов уровня основного среднего образования по обновленному содержанию

Разделы долгосрочного плана	Темы, содержание	Цели обучения
Программирование на языке Python (пайтон)	Знакомство с IDE (Integrated Development Environment (интегрированный девелопмент энвайронмент))	6.1.2.2 использовать возможности интерфейса IDE (Integrated Development Environment (интегрированный девелопмент энвайронмент))
	Алфавит языка. Синтаксис	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)
	Типы данных	6.3.3.1 классифицировать типы данных
	Правила записи арифметических выражений	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)
	Ввод и вывод чисел	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)
	Программирование линейных алгоритмов	6.3.2.1 записывать линейные алгоритмы на языке Python (пайтон)
Программирование алгоритмов на языке Python (пайтон)	Работа с файлами	7.3.3.1 осуществлять чтение и запись файла на языке программирования Python (пайтон)
	Программирование алгоритмов ветвления	7.3.2.1 записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон)
	Программирование вложенных условий	7.3.3.2 использовать вложенные условия на языке программирования Python (пайтон)
	Программирование составных условий	7.3.3.3 использовать составные условия на языке программирования Python (пайтон)
	Организуем выбор	7.3.2.1 записывать разветвляющиеся алгоритмы на языке программирования Python (пайтон)
Программирование алгоритмов на языке	Цикл while (уайл)	8.3.3.1 использовать оператор цикла while (уайл)

программирования Python (пайтон)	Цикл for (фор)	8.3.3.2 использовать оператор цикла for (фор)
	Управление циклом: continue (континю),	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
	Управление циклом: break (брик),	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
	Управление циклом: else (элс)	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
	Трассировка алгоритма	8.3.2.1 осуществлять трассировку алгоритма
Программирование алгоритмов на языке программирования Python (пайтон)	Цикл while (уайл)	8.3.3.1 использовать оператор цикла while (уайл)
	Цикл for (фор)	8.3.3.2 использовать оператор цикла for (фор)
	Управление циклом: continue (континю),	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
	Управление циклом: break (брик),	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
	Управление циклом: else (элс)	8.3.3.3 использовать инструкции управления циклом (continue (континю), break (брик), else (элс))
Трассировка алгоритма	8.3.2.1 осуществлять трассировку алгоритма	
Программирование алгоритмов на языке программирования Python (пайтон)	Одномерный массив	9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов
	Поиск элемента заданными свойствами	9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов
	Перестановка элементов	9.3.3.1 создавать программы на языке программирования Python (пайтон) с

		использованием одномерных массивов
	Двумерный массив	9.3.3.2 создавать программы на языке программирования Python (пайтон) с использованием двумерных массивов
	Сортировка	9.3.2.1 применять алгоритмы сортировки; 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов; 9.3.3.2 создавать программы на языке программирования Python (пайтон) с использованием двумерных массивов
	Удаление и вставка элемента	9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов; 9.3.3.2 создавать программы на языке программирования Python (пайтон) с использованием двумерных массивов
10.2А Алгоритмизация и программирование	Пользовательские функции и процедуры	10.5.1.1 писать код на языке программирования, используя функции и процедуры
	Работа со строками	10.5.1.2 использовать процедуры и функции для обработки строк
	Работа с файлами	10.5.1.3 использовать файлы для чтения и записи информации
	Методы сортировки	10.5.1.4 реализовывать алгоритмы сортировки для решения практических задач
	Алгоритмы на графах	10.5.1.5 реализовывать алгоритмы поиска на графах для решения практических задач

Отличительной особенностью данного раздела является то, что в качестве учебного языка используется язык Python. Python - язык с динамической типизацией, в нём нет необходимости заранее декларировать функции и переменные для дальнейшего использования, что существенно укорачивает программу и делает её более наглядной. Многие преподаватели отрицательно относятся к такому старту в изучении основ программирования. Однако отсутствие строгой типизации позволяет уделить больше внимания логике программирования и управляющим структурам. Как известно, одним из важнейших дидактических принципов в методике обучения является принцип наглядности. За каждым изучаемым понятием в сознании ученика должен закрепиться какой-то визуальный образ, поэтому уже на первых занятиях учащиеся приступают к выполнению практических работ (использование Питон как калькулятора, работа в интерактивном режиме). При знакомстве с синтаксисом языка особое внимание уделяется «правилу отступов», поэтому учащиеся сразу привыкнут к хорошему стилю программирования.

Python – высокоуровневый интерпретируемый язык программирования общего назначения, ориентированный на повышение эффективности разработчика и улучшения читаемости кода. Поддерживает различные парадигмы программирования, в том числе структурное, объектно-ориентированное, функциональное программирование. Python широко применяется для решения разнообразных задач: разработка веб-приложений, игр, автоматизация различных процедур, работа с файлами, написание утилит. Для решения всех этих задач разработчики языка и другие специалисты написали множество библиотек. Кроме того, Python – это лаконичный и понятный язык, имеет синтаксис, легкий для чтения.

В Python имеются встроенные типы: булевы, строки, Unicode- строки, целые числа произвольной точности, числа с плавающей запятой, комплексные числа и некоторые другие. Из коллекций Python поддерживает

кортежи (tuples), списки, словари (ассоциативные массивы) и, начиная с версии 2.4, множества. Все значения в Питоне являются объектами, в том числе функции, методы, модули, классы. В теме «Типы данных» учащиеся знакомятся с тремя типами данных: целые числа (integer), числа с плавающей точкой (float), строки (string). Операторы while, if, for составляют операторы перемещения. В операторе for происходит сравнение переменной и списка. Чтобы получить список цифр до числа <number> - используйте функцию range (<number>). Вот пример использования операторов:

```
N= input ( 'Количество элементов : ' )
```

```
S=0
```

```
for i in range (N-1) :
```

```
    a=input ( 'Введите число : ' )
```

```
    S=S+a
```

```
C=S/N
```

```
print ('Результат : ', C)
```

После рассмотрения операторов управления и цикла учащиеся знакомятся со структурами данных: строки, списки, словари. Списки - похожи на одномерные массивы (но можно использовать Список, включающий списки - многомерный массив), кортежи - неизменяемые списки, словари - тоже списки, но индексы могут быть любого типа, а не только числовыми. «Массивы» в Python могут содержать данные любого типа, то есть в одном массиве могут находиться числовые, строковые и другие типы данных. Массивы начинаются с индекса 0, а последний элемент можно получить по индексу -1. Для объявления функции в Питоне служит ключевое слово «def». Аргументы функции задаются в скобках после названия функции. Можно задавать необязательные аргументы, присваивая им значение по умолчанию.

Python активно развивается благодаря усилиям разработчиков и прочих энтузиастов по сей день. Всё это позволило стать Python одним из самых

распространённых на сегодняшний день языков программирования. Так, этот язык программирования взяли на вооружение многие крупные компании, такие как Google, NASA, Facebook и многие другие. Python является подходящим языком для обучения программированию по ряду причин. Во-первых, он был создан на базе языка ABC, как раз предназначенного для обучения программированию. Вместе с тем Гвидо вон Россум, который считается создателем Python, «сделал Python неигрушечным языком для широкого круга пользователей, имеющим обширную коллекцию стандартных и сторонних прикладных модулей». Во-вторых, благодаря особенностям синтаксиса, обилию базовых структур, библиотек, упрощённой работе с переменными и функциями код, написанный на Python, получается весьма компактным, что приводит к улучшению его читаемости. Это продиктовано тем, что читать чей-то код приходится намного чаще, чем писать свой. Более того, это позволяет ученикам не отвлекаться на особенности написания кода и концентрироваться на изучение алгоритмов. Кроме того, Python поддерживает интерактивный режим (IPython), позволяющий сразу же выполнять записанные команды и операторы и получать результат (например, интерактивное выполнение команд на Python реализовано в средах IDLE, Jupyter Notebook), что бывает полезным для написания небольших учебных примеров и для тестирования кода. Тем не менее, существуют некоторые проблемы, с которыми можно столкнуться при изучении или преподавании Python, особенно при переходе с языков Pascal или C++. Первая из них – особенности синтаксиса. Например, тела циклов, условных операторов выделяются при помощи пробельных или табулирующих отступов, которые в других языках служат лишь для удобства читаемости кода, причём число отступов в пределах одного блока должно быть одинаковым. Из-за этого бывает порой нелегко найти и исправить ошибку (которая может оказаться как синтаксической, так и логической), когда, скажем, пропущен один пробел.

Кроме того, наличие различных функций и методов в Python позволяет решать множество задач буквально в одну строку. Это, с одной стороны, опять же значительно сокращает объём кода, улучшает его читаемость, а с другой – это может стать причиной того, что знание алгоритмов не будет закреплено на практике, так как работа этих алгоритмов скрыта средствами самого языка. «Научив школьников сортировать массивы вызовом метода `sort`, сложно потом объяснить, зачем написаны целые тома об алгоритмах сортировки». Ещё стоит отметить, что в настоящий момент используется версия языка Python 3.

Поскольку Python — интерпретируемый язык программирования, при его изучении можно использовать метод проб и ошибок, что превращает программирование в игру.

2.2. Особенности разработки и использования геймифицированных средств обучения

Основной задачей при создании образовательных игр является сохранение развлекательных и забавных аспектов игровой среды при одновременном включении учебного содержания. Поэтому для успешного внедрения обучающих игр необходимы соответствующие методы и рекомендации по проектированию. В этом разделе представлен обзор доступных фреймворков и методологий создания обучающих игр, а также их сравнение.

GOM основана на Парадигме объектно-ориентированного программирования и был впервые представлен А. Амори [12]. Объектно-ориентированный подход использовался для изображения взаимосвязей между объектами, позволяя поддерживать разработку и анализ сложных конструкций и облегчить понимание сложных ситуаций. Его можно упростить до трех основных компонентов: проблема, повествование и обсуждение.

Модель состоит из множества взаимосвязанных объектов, которые описываются через абстрактные и конкретные интерфейсы. Абстрактные интерфейсы относятся к педагогическим и теоретическим конструкциям, тогда как конкретные интерфейсы относятся к элементам дизайна игры. Представленный рисунок 29 основан на модели Амори. GOM, в которой прямоугольники со скругленными углами обозначают объекты. Интерфейсы представлены кругами, которые связаны с объектами. Объекты могут содержать дополнительные объекты; в этом случае эти объекты наследуют все родительские интерфейсы. Конкретные интерфейсы представлены открытыми кружками, а абстрактные интерфейсы представлены закрытыми кружками. Внешние объекты более абстрактны, поэтому содержат абстрактные интерфейсы, тогда как внутренние объекты содержат в основном конкретные интерфейсы. Основными концепциями GOM являются(рис.29).

Четырехмерная структура. Четырехмерная структура является результатом исследования преподавателей и учащихся по правильному выбору обучающих игр и тому, как их использовать

Фреймворк также можно использовать в качестве инструмента проектирования для разработки серьезных игр. Четыре измерения структуры: моделирование и профилирование учащегося, представление самой игры, роль педагогических подходов для поддержки обучения и контекста в процессе обучения (табл. 5). В нем рассматриваются элементы образовательного и игрового дизайна, в том числе необходимость участия заинтересованных сторон с использованием подходов к дизайну с совместным участием [14].

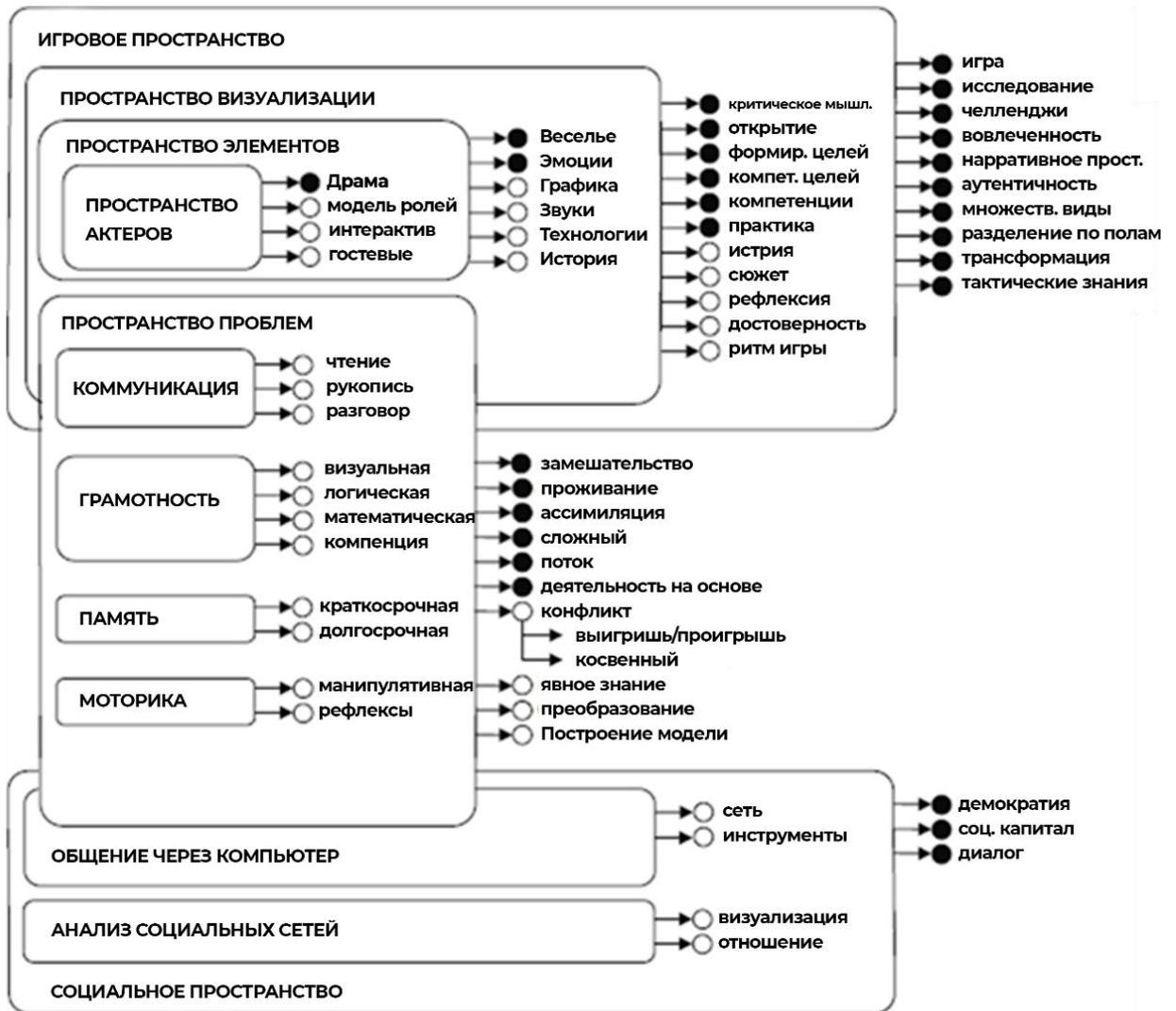


Рисунок 29. GOM: основные концепции

Четырехмерная структура была усовершенствована за счет включения в игровой процесс познавательного и обучающего подходов. Кроме того, структура была расширена за счет взаимосвязей подходов с четырьмя измерениями (рис. 30). Авторы утверждают, что его можно использовать для разработки серьезных игр, а также в качестве инструмента оценки существующих игр. Кроме того, он служит контрольным списком и напоминанием для разработчиков серьезных игр.

Таблица 5. Четырехмерная структура

<p>Особенности учащегося</p> <ul style="list-style-type: none"> - проблемы - конфликт - прогресс 	<p>Педагогика</p> <ul style="list-style-type: none"> - приспособление - Оценка/ Обратная связь - Подведение итогов/оценка - Инструкции/Помощь/Подсказки - безопасность
<p>Представление</p> <ul style="list-style-type: none"> - Действие-Управление ссылками на Домен - Взаимодействие (Оборудование) - Взаимодействие (Межличностное) - Взаимодействие (социальное) - Местоположение - Связь между Проблемой и учащимся - Представительство Сенсорные раздражители 	<p>Контекст</p> <ul style="list-style-type: none"> - фантазии - Цели задачи - Язык/Общение - тайны - Фигуры или игроки - состав игроков - Правила - тема



Рисунок 30. Четырехмерная структура

Образовательная модель игрового дизайна. Дизайн образовательных игр состоит из трех основных факторов: игрового дизайна, педагогики и моделирования содержания обучения. Основными факторами в игровом дизайне являются элементы тестирования удобства использования и мультимодальность. Развлечение является основным отличием образовательных игр от других подходов к обучению, что приводит к мотивации и вовлеченности учащихся. Таким образом, веселье — еще одна ключевая характеристика категории игрового дизайна(рис.31).



Рисунок 31. Модель дизайна образовательных игр

Чтобы оценить, насколько игра будет соответствовать результатам обучения используются первые три уровня таксономии Блума, а именно знание, понимание и применение, для определения и классификации образовательных целей обучения. Игры так же должны поддерживать механизмы самообучения и, следовательно, рефлексии. Это позволяет учащиеся учатся во время игры и оценивают свою работу, одновременно

повышая свои способности решать проблемы. Чтобы обеспечить способность к самообучению и решению проблем, должны быть предоставлены соответствующие учебные материалы. Фактор моделирования содержания обучения связан с включением содержания в игру, чтобы установить правильное решение задач и, следовательно, привести к намеченным результатам обучения [15].

Основная цель Трехуровневой модели мышления (ТММ) состоит в том, чтобы сделать разработку и оценку сетевых образовательных игр менее сложными и более эффективными. ТММ состоит из трехслойной матрицы для анализа этапов процесса разработки образовательной игры (рис. 32). Чтобы связать образовательные цели с задачами в игровом дизайне, модель делает упор на преобразование целей учебной программы в игровые цели.

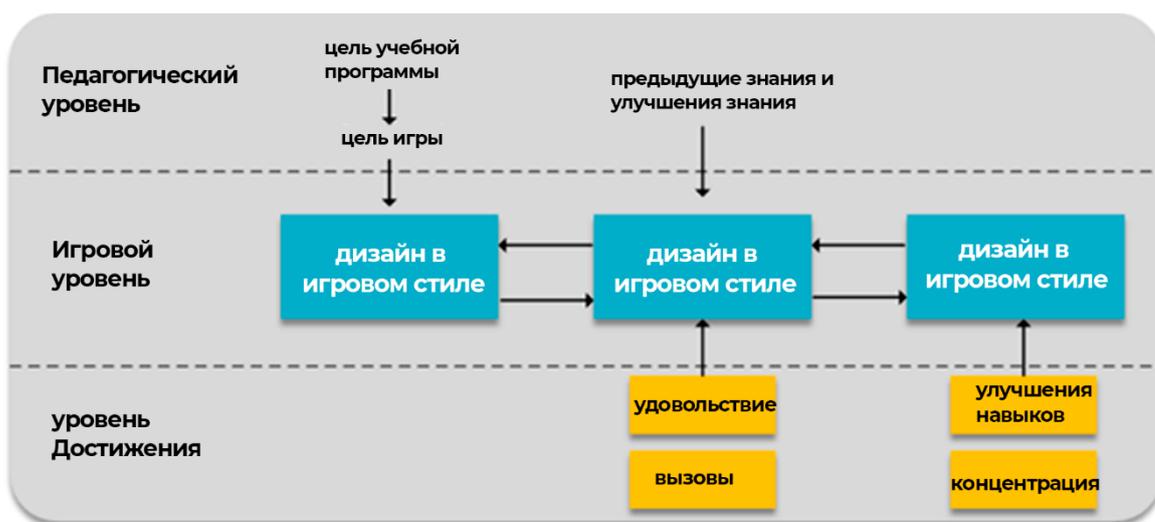


Рисунок 32. Трехслойная модель мышления

Экспериментальная игровая модель была разработана для преодоления разрыва между игровым дизайном и педагогическими элементами. В ней подчеркивается важность рассмотрения теории потока Чиксентмихайи для обеспечения оптимального опыта обучения. Таким образом, модель позволяет связать игровой процесс с эмпирическим обучением. В центре внимания методологии находится опыт потока, который показан на рисунке 5 (рис. 5).

Модель основана на циклическом процессе обучения. В результате цикл потока адаптируется к каждому результату предыдущих задач. Задачи, вытекающие из целей обучения, составляют ядро модели и отвечают за поддержание мотивации и вовлеченности игрока. Игрок генерирует идеи для решения задач. Эти решения могут быть либо прединвазивными, то есть приниматься без учета ограничений системы, либо решениями, учитывающими доступные ресурсы в игре [16].

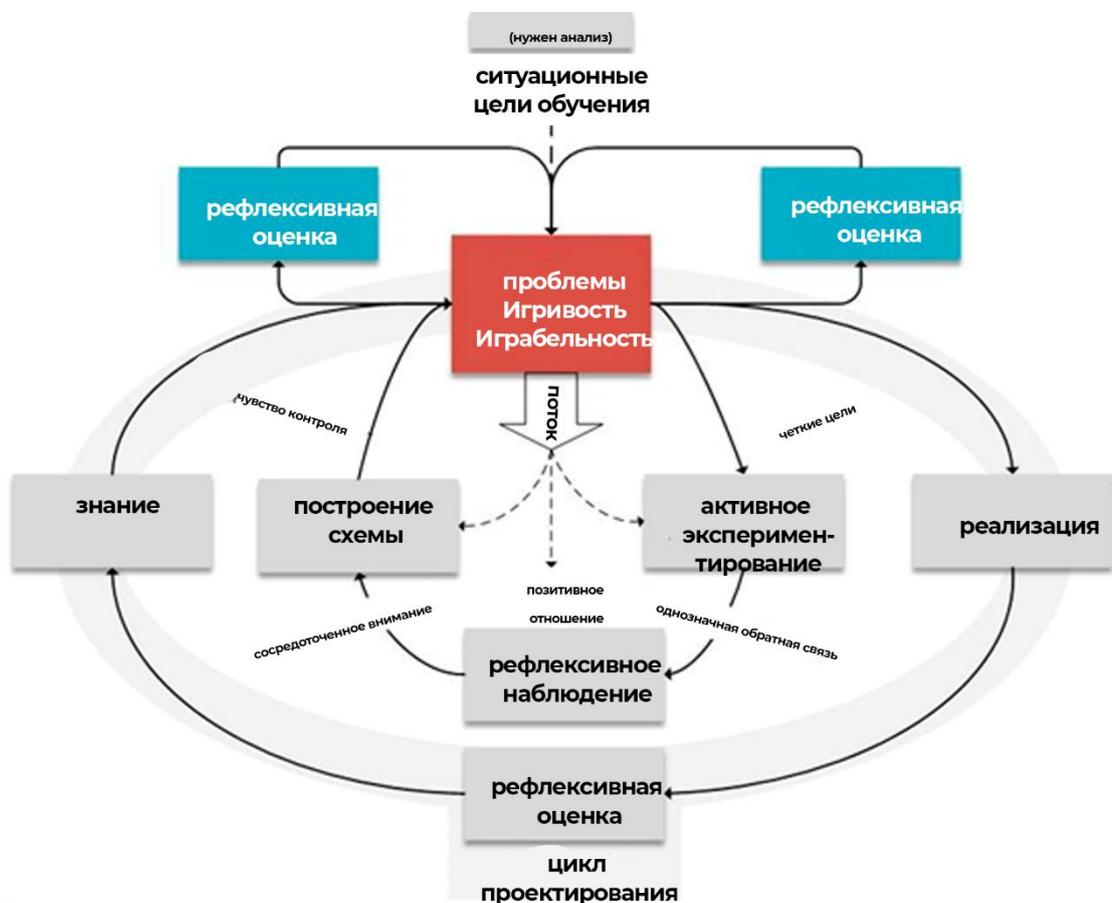


Рисунок 34. Экспериментальная игровая модель

6.Методология ELEKTRA (расширенный опыт обучения и передача знаний) определяет 4М как психолого-педагогическое ядро структуры, которые независимы от конкретного учебного содержания игры: Макроадаптивность, Микроадаптивность, Метапознание и Мотивация. Кроме того, структура состоит из 8 этапов управления рабочим процессом:

Фаза 1: определение целей обучения;

Фаза 2: Учебный анализ;

Фаза 3: Анализ учащихся и контекст обучения;

Этап 4. Напишите цели производительности и общую структуру игры;

Фаза 5: Изучение игрового дизайна;

Этап 6: Производство и разработка;

Фаза 7: Оценка обучения;

Этап 8: Пересмотреть инструкцию.

Эти фазы не следуют линейному порядку, но имеют различные взаимосвязи и циклы обратной связи (рис. 35).



Рисунок 35. Обзор 8 этапов модели ELEKTRA

Важным выводом анализа является то, что только две методологии предоставляют инструмент оценки для измерения эффективности их структур. В Трехслойной модели мышления, Эффективность измеряется на основе расчета общей мотивации. Kiili[17] предлагает так называемую шкалу потока для игр (FSG), чтобы оценить эффективность модели в отношении опыта

потока. в ФСТ учитывает девять параметров потока, каждый из которых оценивается участниками с помощью формата ответов типа Лайкерта из 5 баллов (5 = согласен до 1 = не согласен), что приводит к переменной суммы. Определенными параметрами потока являются: вызов, цель, обратная связь, контроль, играбельность, фреймовая история, концентрация, искажение времени и аутоотелический опыт [17].

2.3. Результаты оценки разработанных средств

В исследовании было использовано упомянутая платформу Ozaria и GamiLab. Проведены занятия для студентов двух групп 1 курса обучения в ЮКГПУ по образовательной программе «6В01503- Подготовка учителей информатики». Обе группы (мы обозначим их буквами А и В) не имеют опыта программирования. А (n = 21), и В (n = 17) были разделены на две группы (A_G , A_N и B_G , B_N), где группы с индексом G использовали геймифицированную платформу для лекции, а группы с индексом N - нет. Группа A_G состояла из 11 студентов, группа A_N из 10 студентов, группа B_G из 9 студентов и группа B_N из 8 студентов. Не было никаких существенных различий между группами, связанных с возрастом, полом, этнической принадлежностью или успеваемостью.

Концепция системы оценивания определился на основе идеи ученого-педагога В. П. Беспалько. С точки зрения В. П. Беспалько, качество подготовки обучающихся характеризуется различными показателями, такими как широта опыта, уровень и прочность усвоения определенных способов деятельности, автоматизация умений, осознанное применение знаний. Автор выделяет четыре уровня усвоения способов деятельности как способности решать определенные задачи [34].

Первый уровень предполагает узнавание ранее усвоенной информации и воспроизведение действий по алгоритму «с подсказкой». Второй уровень —

самостоятельное воспроизведение ранее усвоенных знаний, репродуктивное алгоритмическое действие по решению типовой задачи. Третий уровень заключается в применении усвоенных знаний в новой ситуации без предъявления алгоритма решения. Четвертый уровень — это продуктивная деятельность творческого типа, в процессе выполнения которой рождается объективно новая информация.

В качестве объективного метода диагностики качества знаний обучающихся В. П. Беспалько предлагает специально разработанные тестовые задания, соответствующие перечисленным выше уровням усвоения деятельности. Коэффициент усвоения задание вычисляется по формуле $K = \frac{\alpha}{p}$ ($0 \leq K \leq 1$), где α — количество правильно выполненных операций, p — число существенных операций. При $K \geq 0,5$ можно считать, что программный материал на данном уровне освоен и обучающийся в дальнейшем способен самостоятельно развивать свои знания и умения. При $K < 0,5$ обучающийся будет совершать систематические ошибки, которые он не в состоянии сам исправить, соответственно, нужно вносить изменения в образовательный процесс.

Для диагностики усвоения учебного материала на первом и на втором уровне предлагается тестовые задание, то есть на опознание, различение или классификацию изученных объектов и решения типовых задач. На третьем и студенты должны выполнять тестовые задания открытой формы, которые демонстрирует правильное решение задачи по программированию. На четвертом уровне студент участвует в взаимооценивание, которые при проверке логики рассуждения других работ у студентов формируется критические мышления. Диагностика всех перечисленных уровней предложенной В. П. Беспалько соответствует результатом обучения по таксаномии Блума.

Несмотря на то, что в GamiLab существует множество функций, связанных с играми, мы решили оценить использование квестов в качестве средства самообучения учащихся и проверки их достижений. Квесты - это серия заданий, которые необходимо решить, чтобы получить дополнительные очки, которые впоследствии можно использовать либо для получения оценки, либо, например, для использования некоторой силы. GamiLab предоставляет нам рынок уже готовых квестов, связанных с игрой, которые можно использовать в качестве шаблонов для разработки собственных уникальных квестов для студентов.

Как указывалось ранее, всем группам студентов была дана одинаковая серия заданий: группы с индексом G использовали квест GamiLab для решения своих задач, в то время как группы с индексом N использовали неигровой подход.

Далее, демонстрируется анализ результатов задания, выполненных студентами (табл. 8).

Таблица 8. Результаты выполнения задания

Группы	Число студентов, получивших соответствующую балл(чел.)				Доля выполненных тестов (%)	Средняя оценка по группе M_x	Среднее квадратическое отклонение σ
	До 50 баллов	50-70	70-90	Выше 90			
A_G, B_G (геймификационное обучения)	1	5	8	6	95	3,95	0,75
A_N, B_N (традиционное обучения)	5	5	4	4	73	3,44	1

Сравнение данных дало следующие результаты. Средняя оценка экспериментальной группы достаточно высокая, обычная группа — существенно ниже. Доля выполненных заданий в первом случае составляет 95 %, а во втором — 75 %. Эти показатели говорят о том, что студенты

достаточно хорошо усвоили материалы при занятиях с использованием геймификации чем студенты при традиционном обучении.

Среднее квадратическое отклонение σ характеризует степень индивидуализации успеваемости в группе. Чем он меньше, то есть чем ближе оценки студентов к среднему баллу, тем лучше идет обучение и плодотворнее работает преподаватель. В. П. Беспалько считает, что идеальное значение $\sigma = 0,25$. Однако, как указывает в своем исследовании Л. П. Квашко, после трехлет систематических замеров этого показателя у школьных учителей был сделан вывод, что показатель колеблется от $\sigma = 0,6$ до $\sigma = 1,5$, а его среднее значение равно $\sigma = 1$ [35].

В ходе проведенного исследования было получено 15 релевантных ответов от студентов группы 1503-11 по образовательной программе «6В01503 - Подготовка учителей информатики». В опросе были следующие вопросы. Повлияло ли на ваш интерес к занятиям задания на этой платформе в виде геймификации(рис.35)? Как вы считаете, использование геймификации на уроке улучшает качество занятия(рис.36)? Вы получили дополнительную мотивацию после пройденных занятия(рис.37)?

Повлияло ли на ваш интерес к занятиям задания на этой платформе в виде геймификации ?
15 ответов

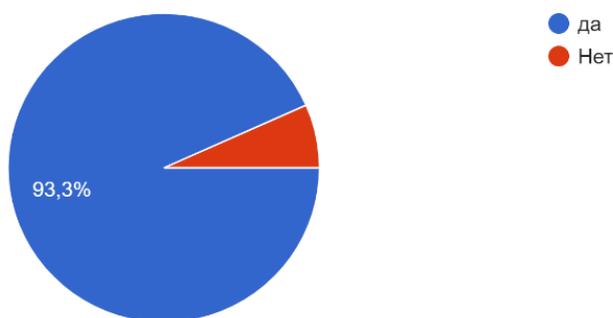


Рисунок 35. Влияние геймификации на мотивацию

Как вы считаете, использование геймификации на уроке улучшает качество занятия ?

15 ответов

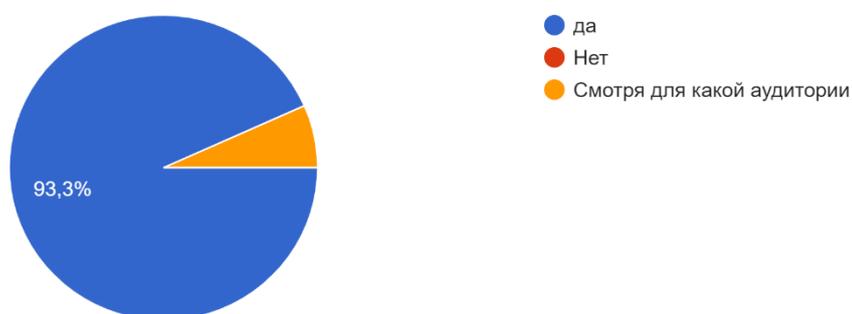


Рисунок 36. Влияние геймификации на качество знания

Вы получили дополнительную мотивацию после пройденных занятия ?

15 ответов

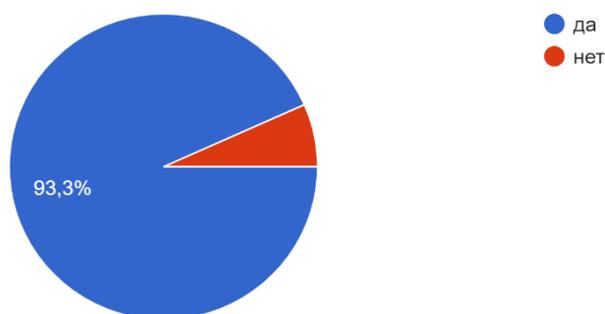


Рисунок 37. Влияние геймификации на мотивацию

Подавляющее большинство опрошенных (93,3%) считают, что пройденные занятия на платформе Ozaria и GamiLab повлияли на их интерес к занятиям; улучшает качество занятия; получили дополнительную мотивацию после пройденных занятия.

Выводы по второй главе

Подведем итоги второй главы диссертационного исследования. Здесь были показаны методические планирования занятий по программированию на Python, особенности разработки и использования геймифицированных средств обучения, а также результаты оценки разработанных средств

Сравнивались ранее упомянутые методологии разработки, проектирования и оценки обучающих игр. Сравнение преследовало две цели: выделение пересекающихся аспектов и различий между методологиями и указание на нерешенные проблемы, а также возможности для дальнейших исследований и оценок. По этой причине рамки сравниваются с разных точек зрения.

Сравнение показало, что четкое определение и отражение целей обучения имеет основополагающее значение при разработке образовательных игр — каждая структура включает по крайней мере одну фазу (обычно начальную фазу), на которой формулируются общие цели обучения. Кроме того, все фреймворки подчеркивают важность разработки и реализации задач, поскольку это улучшает процесс обучения. Тем не менее, структуры различаются с точки зрения детальной реализации и их направленности.

Важным выводом анализа является то, что только две методологии предоставляют инструмент оценки для измерения эффективности их структур. В *Трехслойной модели мышления*, Эффективность измеряется на основе расчета общей мотивации. Kiili[17] предлагает так называемую шкалу потока для игр (FSG), чтобы оценить эффективность модели в отношении опыта потока. в *ФСГ* учитывает девять параметров потока, каждый из которых оценивается участниками с помощью формата ответов типа Лайкерта из 5 баллов (5 = согласен до 1 = не согласен), что приводит к переменной суммы. Определенными параметрами потока являются: вызов, цель, обратная связь,

контроль, играбельность, фреймовая история, концентрация, искажение времени и аутоотелический опыт[17].

Так же был сделан анализ результатов по проведенным занятиям и опрос среди для студентов двух групп 1 курса обучения в ЮКГПУ по образовательной программе «6В01503- Подготовка учителей информатики»

Сравнение данных дало следующие результаты. Средняя оценка экспериментальной группы достаточно высокая, обычная группа — существенно ниже. Доля выполненных заданий в первом случае составляет 95 %, а во втором — 75 %. Эти показатели говорят о том, что студенты достаточно хорошо усвоили материалы при занятиях с использованием геймификации чем студенты при традиционном обучения.

Результатом опроса стал вывод о том, что подавляющее большинство опрошенных (93,3%) считают, что пройденные занятия на платформе Ozaria и GamiLab повлияли на их интерес к занятиям; улучшает качество занятия; получили дополнительную мотивацию после пройденных занятия.

Заключение

В данной диссертации представлены различные подходы к внедрению игр и игровых элементов в образовательном контексте (обучение на основе игр и геймификация), касающиеся обучения основным понятиям программирования на языке Python. На первом этапе обобщаются основы игрового обучения. Одним из выводов этого анализа является то, что только две из шести методологий предоставляют инструмент оценки для измерения их эффективности.

На основе результатов проанализированных игровых подходов к обучению, рассмотренных методологий, цифровых игр и платформ было разработаны серия задач и занятия для обучения школьников вводным концепциям программирования на языке Python. интервью.

В ходе диссертационной работе были выполнены все поставленные задачи, а именно:

1. На основе анализа научно-методических источников были описаны теоретические основы применения технологий геймификации на уроках информатики в старшей школе.

2. Выделены виды технологий и признаки дидактической геймификации.

3. Спроектированы серия занятий по программированию в старшей школе с применением технологий геймификации.

4. Разработаны учебные материалы и задания для применения технологий геймификации.

5. Проведена оценка разработанных средств и проанализированы ее результаты.

Поэтому можно сделать вывод о том, что все задачи исследования решены, заявленная цель достигнута, поскольку гипотеза нашла свое подтверждение.

Список использованных источников

1. Бартл Р. Виртуальные миры: зачем люди играют // Разработка многопользовательских игр. - 2005. - Т. 2. - № 1. - С. 3-18.
2. Аль-Азави, Азми, С., Яхад, Н.А., и Ахмад, Н. (2016). Привлечение студентов к программированию Курсы с геймификацией. Конференция IEEE по электронному обучению, электронному управлению и электронным услугам. IEEE.
3. Аттали И. Миллениум: победители и проигравшие в грядущем мировом порядке. Нью-Йорк: Рэндом Хаус, 2005. 132 с.- ISBN 0-812-91913-0
4. Варенина Л. П. Геймификация в образовании. Историко-социально-просветительская идея, 2014, т. 1, с. 6, нет. 6-2 (28), стр. 314-317. (на русском языке) URL: <https://elibrary.ru/item.asp?id=22981456>
5. Олейник Ю. П. Геймификация в образовании: к определению. Современные проблемы науки и образования, 2015, № 3, с. 476. URL: <https://elibrary.ru/item.asp?id=23703904>.
6. Орлова О. В., Титова В. Н. Геймификация как способ обучения организации. Вестник Томского государственного педагогического университета. 9, стр. 60-64. (на русском языке) URL: <https://elibrary.ru/item.asp?id=24309337>
7. Детердинг С., Сикарт М., Наке Л., О'Хара К., Диксон Д. Геймификация. Использование элементов игрового дизайна в неигровом контексте. CHI '11 Расширенные рефераты о человеческом факторе в вычислительных системах. Нью-Йорк, США, издательство ACM Press, 2011, стр. 2425-2428. DOI: <http://doi.org/10.1145/1979742.1979575>
8. Капп Н.Д., Игра и умственное развитие. Альманах Института специального образования, 2005. – 233с.
9. Систематический литературный обзор эмпирических данных о компьютерных играх и серьезных играх / Т. Коннолли, Э. А. Бойл, Э.

МакАртур, Т. Хейни, Дж. М. Бойл // Компьютеры и образование. - 2012. - Т. 59. - № 2. - С. 661-686. DOI: 10.1016 / j.compedu.2012.03.004.

10. Салин А. С. Критический проект геймификации // Логос. - 2015. - Т. 25 - Страница 1 (103). - С. 100-129.

11. Э. Симпсон Образовательные цели в психомоторной области // Поведенческие цели в разработке учебных программ: Избранные материалы и библиография / Под ред. Мириам Б. Капфер. 1-е изд. - Иглвудские скалы: Публикации по образовательным технологиям, 1971. - С. 60-68.

12. Амори, А. (2, 2007 г.). Модель игровых объектов версии II: теоретическая основа для образовательных игр и их разработка. Исследования и разработки в области образовательных технологий, 55(1), 51-77.

13. Амори, А. (2, 2007 г.). Модель игровых объектов версии II: теоретическая основа для образовательных игр и их разработка. Исследования и разработки в области образовательных технологий, 55(1), 51-77.

14. Куни Дж., Снайдер Л. и Винг Дж. (2010). Демистификация вычислительного мышления для не-компьютеров, стр. 265-274.

15. Маллиаракис, К., Сатрацеми, М., и Ксиногалос, С. (2014). Разработка обучающих игр для компьютеров программирование: целостная структура. Электронный журнал электронного обучения, 12(3), 281-298.

16. Киили, К. (2005). Об образовательном игровом дизайне: строительные блоки потокового опыта.

17. Киили, К. (31.10.2006). Оценки эмпирической игровой модели. Человеческие технологии: Ан Междисциплинарный журнал о людях в среде ИКТ, 2(2), 187-201.

18. Скиннер Б. Ф. Понятие о рефлексе в описании поведения // Журнал общей психологии. 1931. № 5. Стр. 427–458;

19. Аль-Азави, Р., Аль-Блуши, М., и Аль-Фалити, Ф. (2016). Образовательная геймификация против игрового обучения: Сравнительное

исследование.Международный журнал инноваций, менеджмента и технологий Том 7,№ 4, (стр. 131 - 136).

20. Альберт, Д., и Лукас, Дж. (1999).Пространства знаний:

теории, эмпирические исследования и приложения.

21. Лоуренс Эрлбаум Ассошиэйтс, Амбросио Ана, (2011). Выявление когнитивных способностей для улучшения результатов CS1. (IEEE, изд.)Границы в образовании.

22. Амори, А (2001). Создание образовательной приключенческой игры: теория, дизайн и уроки.Журнал Исследование интерактивного обучения, том. 12, стр. 249-263.

23. Андерсон П., Тернер С., Диркшайде Дж. и Макколи Р. (2014). Расширяемая онлайн-среда для обучения концепциям науки о данных с помощью геймификации.Конференция IEEE Frontiers in Education, стр. 1 - 8.

24. Структура LM-GM для анализа серьезных игр / Т. Лим, М.Б. Карвальо, Ф. Беллотти, С. Арнаб, С. де Фрейтас, С. Лушар, Н. Сатти, Р.Б., А. Де Глория. URL:

<https://pdfs.semanticscholar.org/7df0/20237a6d3995860e7345c77dab28e4d0a001.pdf> (дата подачи: 12.08.2017).

25. Оценка игрового опыта: подход, основанный на теории личностных конструктов / Ф. Беллотти, Р. Берта, А. де Глория, Л. Примавера // Конспект лекций по информатике. - 2009. - С. 120-131. URL: https://link.springer.com/content/pdf/10.1007%2F978-3-642-04052-8_11.pdf (дата 20.10.2018).

26. AMRO, JS; ROMLI, R. Исследование методов обучения программированию с помощью мобильного обучающего приложения. В: 4-я Международная конференция и семинары по последним достижениям и инновациям в технике (ICRAIE). IEEE, с. 1-7, 2019.

27. Азми, С., Нурминшах, И., и Нораснита, А. (12, 2015 г.) Геймификация в совместном онлайн-обучении для курса программирования: Обзор литературы. Журнал инженерных и прикладных наук ARPN, том 10, № 23.
28. Валк, Лоренс Большая книга LEGO MINDSTORMS EV3 / Лоренс Валк. - М.: Эксмо, 2014. - 687 с.
29. Сайт [https://play.google.com/store/apps/details?id=com.material.design.codingplanet &hl=ru&gl=US](https://play.google.com/store/apps/details?id=com.material.design.codingplanet&hl=ru&gl=US)
30. Сайт <https://play.google.com/store/apps/details?id=air.MusterenGames.EIHarezmiCoding&hl=ru&gl=US>
31. Сайт <https://play.google.com/store/apps/details?id=com.lightbot.SpriteBoxCoding&hl=ru&gl=US>
32. Сайт <https://www.ozaria.com/>
33. Сайт <https://gamilab.com/>
34. Беспалько В. П. Слагаемые педагогической технологии. М., 1989. 192 с.
35. Квашко Л. П. Тесты – в практику преподавания математики // Математика в школе. 1996. № 6. С. 50–55.