

УДК 001.92

ИСПОЛЬЗОВАНИЕ ИГР В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ



USING GAMES IN TEACHING PROGRAMMING

Тотухов Константин Евгеньевич

кандидат технических наук,
доцент кафедры «Информационные системы
и программирование»
Кубанский государственный технологический университет
ke.dnw@mail.ru

Харченко Анатолий Сергеевич

студент,
Институт компьютерных систем
и информационной безопасности,
Кубанский государственный технологический университет
anatoliy.h90@mail.ru

Новоженов Михаил Юрьевич

студент,
Институт компьютерных систем
и информационной безопасности,
Кубанский государственный технологический университет
mihail.novozhenov.01@mail.ru

Аннотация. Преподавание и изучение компьютерного программирования представляют учителя и студенты соответственно со многими проблемами, особенно при обучении с традиционным подходом. В предыдущие годы было предложено несколько альтернативных подходов к обучению, но они, похоже, не удовлетворяют потребностям 21 века. Современные студенты выросли в цифровом мире, они учатся и реагируют иначе. Учащиеся начинают пользоваться компьютерами еще до того, как они впервые посещают формальное образование, а компьютерные игры стали частью их повседневной жизни. Растущее количество учителей/исследователей предлагают включение обучающих игр (или серьезные игры) в обучении компьютерному программированию с целью усиления мотивации посредством сложных задач учащихся, пробуждая их любопытство и предоставление им чувства контроля и воображения. Эта статья направлена на рассмотрение возможностей, которые должны поддерживаться образовательными играми в целом и степень, в которой эти функции поддерживаются образовательными играми, в частности компьютерное программирование.

Сравнительный анализ нескольких обучающих игр для программирования: показывает, что большинство игр кажутся хорошо зарекомендовавшими себя, но они учат довольно простым концепциям, а их фактические испытания и проверки ограничены; предлагает будущее развитие образовательной игры, которое будет полностью поддерживать все указанные функции и обучать сложным элементам программирования.

Ключевые слова: ИИ, компьютерное программирование, обучение, развивающие игры.

Totukhov Konstantin Evgenyevich

Candidate of Technical Sciences,
Associate Professor of the department
«Information Systems and programming»,
Kuban State Technological University
ke.dnw@mail.ru

Kharchenko Anatoly Sergeevich

Student
Institute of Computer Systems
and information security,
Kuban State Technological University
anatoliy.h90@mail.ru

Novozhenov Mikhail Yurievich

Student,
Institute of Computer Systems
and information security,
Kuban State Technological University
mihail.novozhenov.01@mail.ru

Annotation. The teaching and learning of computer programming present teachers and students respectively with many challenges, especially when taught with a traditional approach. Several alternative approaches to learning have been proposed in previous years, but they do not seem to meet the needs of the 21st century. Modern students have grown up in a digital world, they learn and react differently. Students start using computers even before they first attend formal education, and computer games have become part of their daily lives. A growing number of teachers/researchers are proposing the inclusion of educational games (or serious games) in teaching computer programming in order to increase motivation through challenging tasks for students, arousing their curiosity and giving them a sense of control and imagination. This article aims to review the features that should be supported by educational games in general and the extent to which these features are supported by educational games, in particular computer programming.

Comparative analysis of several teaching games for programming: shows that most games seem to be well-established, but they teach fairly simple concepts, and their actual testing and verification is limited; offers a future development of the educational game that will fully support all of the above features and teach complex programming elements.

Keywords: AI, computer programming, learning, educational games.

Есть много трудностей, с которыми сталкиваются начинающие программисты на компьютерных курсах, в основном в процессе изучения программирования. Есть много причин для требований логического/математического мышления или скорость обучения учащихся. Изучение компьютерного программирования было выделено как трудная и сложная задача. Об этом свидетельствуют высокие показатели отсева выпускников школ на родственных курсах. Исследователи и педагоги приложили немало усилий, чтобы преодолеть трудности, возникающие при обучении программированию. Использование цифровых игр является методом, который изучается, предоставляя новичкам интерактивные и забавные моменты программирования [5]. Однако это использование по-прежнему ограничено отдельными инициативами и не известно, как и в какой степени игры способствуют процессу обучения программированию. Таким образом, целью данной работы было проведение систематического обзора литературы (SLR), чтобы исследовать использование цифровых игр в качестве инструмента, помогающего процессу обучения программированию. SLR – это исследование интересующего явления, которое производит подробные и конкретные результаты посредством анализа содержания и качества материала SELAW 474 исследовано. SLR использовался для удаления, каталогизации, анализа и синтеза данных из множество статей, опубликованных на конференциях и в международных журналах, посвященных теме этого обзора.

Спецификация требований к образовательным играм

Мы тщательно изучили фреймворки для обучающих игр (Becker, 2010; Freitas & Jarvis, 2006; Salen & Zimmerman, 2004; Yusoff et al., 2009; Zualkernan, 2006). Согласно этим работам, разработка обучающей игры должна осуществляться после рассмотрения ряда аспектов, где каждый аспект определяет особенности – это должно быть поддержано в образовательной игре [4]. Все фреймворки включают аналогичные требования как понятия, которые считаются важными. Однако мы выбираем исследования, предложенные Беккером (2010), поскольку его концепции охватывают все те, которые включены в другие рамки. Таким образом, мы считаем, что это более абстрактный расширенный набор функций, которые должны поддерживаться всеми образовательными играми. Первоначально предлагается исследовать образовательные цели по всем направлениям. Две оси используют разные точки зрения так, что их спецификация будет полной. Эти оси:

– *Когнитивная ось* в отношении умственных способностей (Знания). Образовательные цели должны следить за тем, чтобы информация, полученная учащимися, начиналась из первой категории в таксономии Блума (Знания) и заканчивалась в финальной и самой сложной категории (Оценка) успешно.

– *Эмоциональная ось* в отношении эмоций или эмоциональных областей (отношение). Образовательные цели должны позволить учащимся справляться с заданными ситуациями. Например, игры мотивируют учащихся быстрее и правильнее решать поставленную задачу.

Кроме того, важно выбрать правильную структуру, которая будет направлять процесс обучения с включением обучающих игр. Сборка среды разработки требует определения различных элементов, которые совместно структурируют компоненты структуры. Например, реальный мир, который нужно будет смоделировать в игре (например, какие движения будут разрешены, как будет устроен виртуальный мир, как будут представлены игроки и т.д.) должны быть четко идентифицированы. Это очень важный шаг, так как он определяет образовательные сценарии, которые могут поддерживаться игровой средой и, таким образом, влияют на весь процесс обучения. В соответствии со спецификацией фреймворка будет построена архитектура на основе идентифицированных компонентов, которые должны быть доступны в игре «Окружающая среда».

Эти компоненты включают в себя:

– *Сценарное пространство*. Студенты знакомятся с сюжетной линией игры один раз. Они входят в среду с кратким описанием сюжета, а также с кратким обзором основных действий, которые они должны будут выполнить.

– *Соответствующие дела.* Учащимся предоставляется набор, заранее решенных подобных кейсов, из которого они могут лучше понять требования к знаниям и навыкам игры и, таким образом, лучше подготовиться к тому, когда придет их очередь решать поставленные задачи.

– *Информационные ресурсы.* Учащиеся могут получить доступ к информации, относящейся к заданию, когда они нуждаются в помощи.

– *Инструменты облегчения.* Включен набор инструментов, которые учащиеся могут использовать, когда они пытаются выполнить задачу, и это помогает им накапливать новые знания. Также, игра предоставляет инструменты, которые лежат в основе общения учащихся, а также обсуждения с учителями любых вопросов, мыслей или размышлений о виртуальном мире.

Кроме того, важно различать информацию о студенте, такую как: цели обучения, стиль обучения (например, целостный, аналитический и т. д.), а также когнитивные ограничения (например, поведенческие компетенции, которые могут повлиять на их обучение). Учителя также должны иметь возможность ставить образовательные цели, которые должны быть достигнуты учащимися во время игры, назначая им конкретные виды деятельности. Таким образом, учащиеся достигают промежуточных целей, успешно выполняя задачи, которые приведут их к усвоению конечных результатов обучения, установленных учителями [3].

Вышеуказанные особенности будут лежать в основе выбора аутентичного сценария, который будет обеспечивать привлекательную историю, которая будет сопровождать виртуальный мир игры. К концу игры, учащимся должна быть предложена интересная и мотивирующая задача, которая нуждается в решении, и лучше всего, если сюжет будет похож на те сюжеты, которые доступны в существующих компьютерных играх. Таким образом, учащиеся уже будут знакомы с общей концепцией, а действия, в которых они будут участвовать, будут больше похожи на игры, чем на обучение.

Точно так же индивидуальные проблемы, которые студенты будут решать во время игры должны соответствовать воспитательным целям, а также любые когнитивные ограничения, которые могут быть очевидны учителям. Это еще одна причина, по которой обе эти функции должны поддерживаться в образовательной игре.

Еще одной важной особенностью является постоянная и пояснительная обратная связь, предоставляемая ученикам во время навигации по игровым уровням. Этот отзыв должен быть представлен в виде сообщений, которые направляют учащихся к пониманию того, что они сделали правильно, что они сделали неправильно и как они могут достичь своих целей.

Развивающие игры для обучения компьютерному программированию

В этом разделе представлена серия игр, которые были разработаны специально для курсов компьютерного программирования. Обзор этих игр был проведен на основе спецификаций, определенных и описанных в предыдущем разделе, в случаях, где они были явно определены соответствующей литературой. Более того, все выбранные игры затрагивали как когнитивную, так и эмоциональную ось во время игры. Таким образом, в каждой игре учащиеся начинают с получения фишек и благодаря своему участию в игре они продвигаются вверх к этапу оценки, размышляя об их прогрессе и завершая задания. Также к эмоциональной оси обращаются через игровые сценарии. Все сценарии стимулируют эмоции, которые мотивируют учащихся на выполнение всех заданий для победы.

В ходе исследования можно было выделить две основные категории, которые сортируют образовательные игры в зависимости от образовательных целей. К первой категории относятся игры, направленные на обучение конкретным единицам компьютерного программирования. В то время как вторая категория представляет игры, которые охватывают несколько образовательных целей и материал для компьютерного программирования [2].

Представлен обзор для каждой категории в следующих двух подразделах с последующим сравнительным анализом, представленным в таблице 1.

Таблица 1 – Обзор обучающих игр для курсов компьютерного программирования

Игра	Элементы программирования	Язык программирования	Программирование	Особые характеристики
Catacombs	Переменные; Простые и вложенные операторы if; loop (бесконечный цикл)	Микроязык	Вопросы с несколькими вариантами ответов; Заполнение строк кода	Трёхмерный; Мультимедиа; Оценки успеха; Леса с пояснительными сообщениями от героя
Saving Sera	if операторы; Рекурсия	Микроязык	Заполнение строк кода; Отображение частей кода в соответствующих местах; Вопросы с несколькими вариантами ответов	Двумерные леса
EleMental	рекурсия; Алгоритм поиска в глубину (DFS)	C#	Алгоритм поиска в глубину; Перемещение героя по фантастическому бинарному дереву	Трёхмерный; строительные леса
Wu's Castle	loop (бесконечный цикл); Массивы	C++	Управление массивами; Движение героя; Выявление логических ошибок в коде	Взаимодействие; Ролевые игры
Robozzle	Функции	Код не используется	Создание функций через движение героя	Интерактивный; Доступный

Образовательные игры, ориентированные на обучение определенной единице обучения

Catacombs. Это трехмерная многопользовательская игра, целью которой является обучение студентов: как объявлять переменные и использовать простые и вложенные операторы if и циклы. По сценарию игры каждый игрок представляет собой волшебника, который должен спасти двух детей, застрявших в катакомбах. Для достижения этой цели волшебники должны отвечать на вопросы с несколькими вариантами ответов, пытаясь решить данный программный код, который поможет им выполнить их квесты. Ответы на заданные вопросы автоматически создаются исполняемыми строками кода на микроязыке. Если ответы верны, волшебники продвигаются по уровням игры; иначе им дают соответствующие отзывы о том, что они ответили неправильно, и им предлагается попробовать ещё раз. Игра записывает баллы опыта для каждого учащегося и предоставляет поясняющие сообщения в качестве вспомогательного механизма.

Saving Princess Sera. Это двумерная игра, которая позволяет учащимся создавать леса с помощью поясняющих сообщений, адресованных игроку. Каждый игрок должен попытаться спасти принцессу по имени Сера, которую похитил монстр по имени Гаргамель в свой шестнадцатый день рождения. Учащиеся должны заполнить число квестов для продвижения по сюжету игры. Для достижения этой цели они должны прописать полные строки кода, которые приведут к исполняемой программе, или они должны правильно сопоставлять существующие строки кода с их правильным положением или порядком в программе, используя функцию перетаскивания. Таким образом, студенты изучают алгоритм быстрой сортировки, а также простые и вложенные циклы с использованием микроязыка [1].

EleMental: The Recurrence. Это трехмерная игра, цель которой научить студентов выполнять рекурсию и поиск в глубину, используя C # язык программирования. Игрок должен перемещаться по виртуальному бинарному дереву, используя трансверсал в глубину, и выполнять три квеста, применяя рекурсию. Два аватара по имени Эль и Сера помогают ученикам во время игры различными способами. Например, как только код написан, Ele пересекает бинарное дерево в соответствии с тем, как разворачивается написанный код, в то время как Sera объясняет, что именно представляет собой код производства в определенный момент [1].

Wu's Castle. Это двухмерная ролевая игра, целью которой является научить студентов использовать циклы и массивы с помощью интерактивных действий. Каждый игрок – волшебник, который может управлять армией снеговиков. Игроки распознают

логические ошибки в строках кода, написанного на языке программирования C++. Игра позволяет управлять массивами за счет изменения параметров внутри циклов и движения персонажей посредством выполнения вложенных циклов.

Robozzle. Это онлайн игра – головоломка, которая предоставляет ряд predetermined команд, готовых к использованию и не показывает никакого фактического кода. Судя по игре, по сценарию пользователи должны создавать функции, которые помогут им выполнить каждую поставленную задачу. Пользователи могут запускать свои функции и видеть, как их герой будет перемещаться по миру и поэтому могут легко определить какие ошибки они сделали и перепрограммировать.

Обсуждение

В этом разделе мы даем общий обзор того, насколько хорошо развита существующие развивающие игры по программированию, и каковы их ограничения. Результаты представлены ниже.

Образовательные цели. Образовательные цели охватывают как познавательные, так и эмоциональные оси. В играх эти цели четко сфокусированы на компьютере. Особенно это касается образовательные игры, которые охватывают определенные единицы обучения, и, таким образом, желаемые результаты обучения более четко определены. Эмоциональные цели, похоже, достигаются благодаря многочисленным привлекательным сценариям, доступным в каждой игре.

Задачи, которые необходимо решить учащимся, соответствуют поставленной учебной цели и их когнитивные ограничения. В обучающих играх, ориентированных на конкретные единицы обучения, студенты выполняют и завершают квесты, которые учат их знаниям. Это имеет отношение к концепциям программирования, установленным в целях. Например, простые и вложенные циклы в Catacombs обучаются через завершение строк кода, а их правильный синтаксис позволяет учащимся перейти на следующий уровень, в то время как те же концепции преподаются в Wu's Castle, когда ученики перемещают своих персонажей по всему миру и распознавать логические ошибки.

Рамки. Образовательные игры, которые фокусируются на конкретных единицах обучения, должным образом определили рамки для своей работы в образовательных контекстах. Однако игры, которые учат множеству и более сложным единицам обучения и, таким образом, охватывают несколько образовательных целей, обычно устанавливают несколько рамок. Следует отметить, что игра Robozzle не определяет никаких рамок.

Пространство сценария. Все обучающие игры присутствуют и работают по сценарию для привлечения и мотивации студентов. В некоторых случаях вводная информация предоставляется игрокам в отношении виртуального мира (например, Wu's Castle).

Информационные ресурсы. Большинство обучающих игр содержат поясняющие сообщения.

Игры, в которых эта функция поддерживается более полно, – Catacombs, Saving Sera, EleMental: The Recurrence and Wu's Castle. Кроме того, техника подмостей предоставляется через эти пояснительные сообщения, которые появляются, когда учащиеся пытаются решить свои квесты (например, Catacombs, Saving Sera, EleMental).

Общие условия. Учтены общие условия. Это было выполнено более эффективно в обучающих играх, которые охватывают определенные единицы обучения, а не в множественных и сложных концепциях компьютерного программирования. С другой стороны, они вообще не учитывались при разработке игры Robozzle. Следует отметить, что ни одна из исследованных игр не дает релевантных кейсов, которыми можно подготовить учащихся к действиям, которые они должны будут выполнять. Существование этой функции значительно повысит качество игр, поскольку предоставит полезные учебные пособия и рекомендации для учащихся.

Мы также должны упомянуть, что многие игры взяты исключительно из соответствующей литературы, так как они недоступны. Этот факт также приводит к нашей неспособности внедрять их в процесс обучения и фактически проверять их на соответствие поставленным образовательным целям на компьютере.

Подводя итоги, представляется, что все изученные игры включают в себя сценарии, которые мотивируют обучающихся, четко обозначают образовательные цели, которые необходимо достигнуть и включают проблемы, установленные, как указано выше. Другие особенности, такие как вспомогательные инструменты, информационные ресурсы, одна или несколько рамок поддерживаются большинством обучающих игр.

Вывод

Основные выводы, сделанные в результате анализа, заключаются в том, что большинство игр разработаны для охвата таких концепций программирования, как переменные, простые и вложенные операторы if, циклы, массивы, функции. Мы не учитываем тот факт, что игры не решают все концепции программирования как недостаток, поскольку они успешно выполняют образовательные цели, к которым они стремятся.

Кроме того, в исследовании подробно рассматриваются дополнительные преимущества использования обучающих игр.

В компьютерном программировании присутствует также ряд интересных принципов, которые помогают нам понять, почему образовательные игры могут улучшить преподавание и изучение компьютерного программирования. Например, игры могут играть маленькую или большую роль в реализации всего процесса в зависимости от общих условий. Точнее, в зависимости от характера курса (онлайн, оффлайн, смешанный), материалов, коммуникаций, экзаменов и т.д. могут поддерживаться на разных уровнях игровой средой. В конце концов, образовательные игры могут обладать рядом характеристик, таких как рассказывание историй, построение лесов и интерактивность, которые повышают мотивацию к участию у класса, а также возможность привлечь учащихся к выполнению своих заданий с помощью интересных сценариев.

Более конкретно, мы рассмотрели изученный кейс, в котором тестировались игры, основанные на какую образовательную ценность привносят, и мы пришли к выводу, что они дают учащимся:

Четкие образовательные цели и результаты обучения, гарантирующие получить необходимые знания и навыки.

Знакомая и иммерсивная среда, которая привлекает внимание учащихся, способствует их активному участию и повышает их мотивацию.

Интересные сценарии с комплексными проблемами, которые им предстоит решить, что позволяет им учиться в контекстуальной манере (изучение конкретных единиц обучения периодически).

Такие игры будут полностью поддерживать все указанные спецификации и функции и будут выполнять цель – углубленное изучение более сложных концепций, таких как объектно-ориентированное программирование.

Литература

1. Чаффин А., Доран К., Хикс Д. и Барнс Т. Экспериментальная оценка обучения рекурсии в видеоигре // В проц. Симпозиум ACM SIGGRAPH по видеоиграм. – 2009. – С. 79–86.
2. Серия Инина И. Игры, которые учат программированию [Электронный ресурс]. – URL : <https://habr.com/post/273003/>
3. Жемчужников Д.Г. Методика обучения программированию, основанная на создании школьниками динамических компьютерных игр : автореф. дис. ... канд. пед. наук. – М., 2013. – С. 25.
4. Тихонова Т.И. В какую информатику будем играть? // Вестник НГУ. Серия: Информационные технологии. – 2012. – Т.10. – № 2. – С. 100–105.
5. Данова Н.С., Пономарев О.П. Роль программирования в школьном курсе информатики // Гаудеамус. – 2005. – № 1(7) – С. 191–196.

References

1. Chaffin A., Doran K., Hicks D., and Barnes T. Experimental evaluation of recursion learning in a video game // In Proc. ACM SIGGRAPH symposium on video games. – 2009. – P. 79–86.
2. Inina I. Games that teach programming [Electronic resource]. – URL : <https://habr.com/post/273003/>
3. Zhemchuzhnikov D.G. Methodology of teaching programming, based on the creation of dynamic computer games by schoolchildren : abstract of the ... Candidate of pedagogical sciences. – M., 2013. – P. 25.
4. Tikhonova T.I. What kind of informatics will we play? // Bulletin of the NSU. Series: Information Technology. – 2012. – V.10. – № 2. – P. 100–105.
5. Danova N.S., Ponomarev O.P. The role of programming in school computer science course // Gaudeamus. – 2005. – № 1(7) – P. 191–196.