Создание компьютерных игр, как средство обучения программированию.

Исаева Оксана Сергеевна, преподаватель

I квалификационной категории,

ГАПОУ «Казанский колледж технологии и дизайна».

Аннотация

В статье анализируются существующие подходы к разработке игр как средству обучения программированию и методика обучения программированию на основе программирования динамических компьютерных игр.

Ключевые слова: методика обучения информатике, обучение программированию.

Попытки создания методик по включению разработки игр в процесс обучения детей программированию связаны, прежде всего, с философией конструктивизма, распространившейся в педагогике в конце XX в. Видные теоретики конструктивизма Л. С. Выготский, Дж. Дьюи, Ж.-Ж. Пиаже оценивали процесс движения к истине выше, чем саму истину. Ж.-Ж. Пиаже заявлял: «Научное знание — явление не статическое. Это есть процесс, более конкретно, процесс непрерывного конструирования и реорганизации» [1].

Согласно теории конструктивизма, обучение активный процесс, в ходе которого учащиеся активно конструируют знания на основе собственного опыта. Дети не получают идеи готовыми, а сами создают их. И этот процесс будет эффективнее, если они вовлечены в создание продуктов, наделенных для них особым смыслом. Одной из современных теорий в отечественной педагогике, воплощающей идеи конструктивизма, стала теория эвристического обучения.

Уникальность задачи по созданию компьютерной игры, помимо высокой внутренней мотивации на результат, заключается в том, что учащиеся детально знакомы с требованиями к конечному продукту, могут в каждый момент разработки проекта предположить результат следующего этапа. Это связано с их предыдущим опытом знакомства с компьютерными играми в качестве потребителей.

Налицо признаки эвристического задания, согласно А. В. Хуторскому [6]:

- цель - создание учеником личного образовательного продукта;

- открытость, т. е. отсутствие заранее известного результата (каждый учащийся делает собственную уникальную игру);

- опора на творческий потенциал ученика, обеспечение развития его творческих и познавательных способностей;

- наличие актуальной для решения проблемы (противоречия или потребности), касающейся ученика и принадлежащей заданной предметной (межпредметной) области;

- сочетание универсальной предметной основы и уникального ее рассмотрения учеником.

В обучении информатике и программированию конструктивизм как практическое направление в педагогике впервые применил С. Пейперт.

В своей работе «Переворот в сознании: дети, компьютеры и плодотворные идеи» С. Пейперт предложил концепцию школы будущего, в основе которой лежат естественное любопытство детей и средства для удовлетворения этого любопытства. Основная идея — это микромиры, представляющие собой некоторые модели реального мира, которые с той или иной степенью детализации творит сам ребенок [4].

Идеи и принципы таких миров С. Пейперта прослеживаются в большинстве визуальных конструкторов игр.

Использование разработки компьютерных игр в обучении программированию можно условно разделить на несколько подходов:

1) создание дополнительных уровней к играм, разработанным профессионалами, в том числе к играм с образовательным содержимым (обучающим играм);

2) использование визуальных конструкторов игр, в том числе со встроенным языком программирования;

3) введение небольших логических игр как элементов курса обучения какому-либо языку программирования.

Подавляющее большинство современных компьютерных игр имеет опцию «Конструктор уровней», то есть визуальный конструктор, который может использоваться для понимания детьми объектного мышления.

Интересен опыт сетевых игр с открытым кодом, построенных на принципе создания миров. В частности, чрезвычайно популярная в 2018-2020 гг. игра Minecraft [10] дает возможность разработки и последующего распространения классов и объектов игре. Несмотря на отсутствие системы обучения, популярность проекта привела к тому, что дети самостоятельно изучают принципы построения и механику игры.

Использование конструкторов уровней в готовых играх не относится непосредственно к программированию. Речь идет в основном о дополнительных объектах, внедряемых в игровой алгоритм с минимальными изменениями готового набора свойств и методов.

В настоящее время существует несколько сотен визуальных конструкторов компьютерных игр (со встроенным языком программирования или без), разработчики которых проводят эксперименты по обучению программированию на основе этих конструкторов. Эти программы очень похожи. Рассмотрим самые популярные из них.

Корпорация Microsoft выпустила финальную версию своей интегрированной среды разработки Kodu Game Lab для персональных компьютеров, которая ориентирована на детскую и подростковую аудитории [9]. Коду является интерактивной игрой, в которой можно создавать свои миры из предлагаемого множества блоков. Kodu активно продвигается корпорацией Microsoft через свои учебные центры и в ряде начальных школ США.

Принадлежащий компании Oracle проект Alice представляет собой что-то среднее между конструктором и языком программирования для создания 3D игр — программный код нужно писать, перетаскивая названия функций и переменных в специальные ячейки.

Scratch это разработка группы Lifelong Kindergarten MIT Media Lab [11], специально созданная для поддержки обучения школьников визуальному программированию. Позиционируемый как язык программирования, этот пакет позволяет формировать сценарий игры или анимации из блоков, но не из пиктограмм, как в большинстве конструкторов, а из стандартных текстовых фраз на формализованном естественном языке (английском). Эта интересная концепция имеет хорошую методическую поддержку разработчиков, широко применяется в школах США, Европы и в нашей стране.

Однако данный конструктор в настоящее время не поддерживает ряд важнейших программных конструкций (процедуры и функции, передачу параметров, рекурсию, наследование и т. д.), что делает его использование и обучении школьников программированию весьма ограниченным.

Самый современный конструктор двухмерных игр Game Maker имеет встроенный скриптовый Си-подобный язык GML (Game Maker Language) и используется на всех уровнях обучения. Некоторые образовательные организации США применяют программу в летних лагерях с техническим уклоном, в частности, Children's Technology Workshop [8] на условиях лицензирования предлагает законченный курс обучения для таких лагерей на базе Game Maker. В программы информатики для средней школы педагоги Германии все чаще включают курсы игрового дизайна. Обычно учащиеся с энтузиазмом воспринимают эти курсы, потому что обычным занятиям они предпочитают программирование в Game Maker.

Создание игр с помощью конструкторов и, соответственно, методики обучения программированию на их основе имеют следующие существенные недостатки:

- ограниченность создаваемых игровых сюжетов возможностями конструктора, что может привести к снижению мотивации;

- из-за того, что игру формируют простым перемещением элементов, учащиеся не используют встроенный язык программирования даже при его наличии;

- невозможность изучения на основе этих игр ряда необходимых по стандарту программных конструкций.

Следует заметить, что языки программирования, встроенные в конструкторы игр, всегда объектно -ориентированные и чаще всего скриптовые (диалекты языка Си и ЈavaScript, реже Visual Basic).

Достаточно часто в учебных пособиях по объектно-ориентированным языкам программирования в качестве задачи повышенной сложности предъявляется разработка небольших игр. Особенно часто такой подход используется для скриптовых языков, например, в работах по Javascript [5] и ActionScript [2].

Следует отметить, что такие пособия пишутся профессиональными программистами для программистов, самостоятельно осваивающих новый язык. Педагогические методики в них разработаны слабо, для обучения школьников они не подходят. Для создания игр авторы используют сложные профессиональные программные конструкции, часто не разобранные ранее в предыдущих задачах.

Ряд преподавателей информатики, особенно профессиональных программистов, а также интересующихся информатикой учащихся эпизодически используют разработку игр для обучения программированию. Но цельной методики создания игр школьниками на языке программирования до сих пор не существовало.

Концепция методики обучения на основе со здания динамических игр [3] состоит в том, чтобы связать все этапы обучения программированию в школьном курсе одной сквозной практической за дачей, в ходе решения которой каждый учащийся поэтапно будет создавать свою собственную динамическую компьютерную игру.

Средой разработки выбран пакет Adobe Flash с встроенным Си-подобным языком ActionScript 2.0. Этот пакет активно используется в школах, в основном как графический редактор и редактор анимации. На объектно-ориентированном языке Action Script полноценно реализуются все элементы программирования, требуемые на профильном курсе информатики.

Каждый этап разработки игры призван иллюстрировать соответствующую программную конструкцию. Например, простое поступательное движение персонажей игры линейный алгоритм и программное обращение к свойствам экземпляров объектов на экране; замкнутое движение в границах экрана - множество вариантов конструкции условия и выбора; сценарии поведения группы одинаковых персонажей — массивы, иерархию объектов и т. д.

Помимо традиционных конструкций структурного программирования в процессе разработки игры существует возможность с высокой степенью наглядности освоить и воплотить принципы объектно-ориентированного программирования. На примере игры, тем более создаваемой самим учащимся, сложные понятия наследования, инкапсуляции, полиморфизма воспринимаются достаточно легко.

Литературные и интернет - источники

1. Жан Пиаже: теория, эксперименты, дискуссии: [Сб.ст.): учеб. пособие для вузов / под ред. Л. Ф. Обуховой, г. В. Бурменской. М.: Гардарики, 2001,

2. Мельников С. В. Создание игр во Flash MX. СПб.: вхв Петербург, 2005.

3. Обучение программированию на основе создания динамических игр. Учебно-методическое пособие. М.: Мэйлер, 2019.

4. Пейперт С. Переворот в сознании: дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.

5. Соколов С. А. JavaScript в примерах, типовых решениях и задачах. Профессиональная работа. М.: Вильямс, 2017.

6. Хуторской А. В. Эвристическое обучение: Теория, методология, практика. М.: Международная педагогическая академия, 2000.

7. Чошанов М. А. Инженерия обучающих технологий. М.: Бином. Лаборатория знаний, 2011.

8. Children's Technology Workshop. http://ctwork- shop.com

9. Коду — Актуальные — Конструкторы игр — Файлы для игроделов — Создание игр, Игровые движки, Конструкторы игр. http://fuse.microsoft.com/page/ kodu.aspx

10. Minecraft.su <http://minecraft.su> русское сообщество Minecraft.

11. Scratch Home | imagine, program, share. http://scratch.mit.edu