

разовательная робототехника с использованием реального робота позволяет вовлечь учащихся «в процесс инженерного творчества»[2], что благотворно влияет на общее развитие, а также вводит в содержание обучения информатике и ИКТ содержательную линию «Основы конструирования и механики», освоение которой ознакомит школьников с принципами работы современных технических конструкций и устройств.

Библиографический список

1. Федеральный государственный образовательный стандарт основного общего образования [Текст]. Введ. 2010.12.17
2. Шимов И.В. Применение робототехнических устройств в обучении программированию школьников. Педагогическое образование в России.- 2013.-№1.-С. 185-188.
3. Официальный программного средства ROBOTC для LEGOMINDSTORMS [Электронный ресурс]/ RobotcInc. Режим доступа:<http://www.robotc.ru/download/robot-virtual-worlds.html>

ИСПОЛЬЗОВАНИЕ СРЕДЫ РАЗРАБОТКИ UNITY ДЛЯ ОБУЧЕНИЯ
БУДУЩИХ ИТ-СПЕЦИАЛИСТОВ ПРОГРАММИРОВАНИЮ

Е.А. Фролов

Научный руководитель: А.И. Газейкина, к.п.н., доцент

Уральский государственный педагогический университет

Аннотация.

Статья посвящена вопросу изучения программирования в вузе через разработку студентами игрового проекта с использованием готового игрового движка. В статье описываются преимущества использования среды разработки Unity и структурный план учебного пособия.

Ключевые слова: программирование, Unity, игровой проект, разработка видеоигры.

Регламентируемый государственным стандартом [1] перечень знаний и умений студента по программированию включает в себя множество компонентов. Однако те подходы, которые применяются в современном образовании в высших учебных заведениях в большинстве своем не гарантируют того, что студент сможет воспользоваться полученными знаниями комплексно, для выполнения крупного проекта. Зачастую студенты при постановке перед ними сложной задачи, для решения которой требуется применить полученные знания комплексно, испытывают затруднения.

При обучении студентов программированию используют различные виды учебных заданий, в том числе:

- решение олимпиадных задач;
- решение маленьких, локальных задач, затрагивающих какую-то конструкцию в программировании;
- создание собственного приложения;

- создание игрового проекта.

Остановимся подробнее на создании игрового проекта. Изучение программирования посредством создания игрового проекта основывается на методе проектов, который давно известен и набирает все большую популярность в мире. Прежде чем говорить о игровом проекте, следует пояснить, что из себя представляет метод проектов. Согласно Е. С. Полат: «В основу метода проектов положена идея, составляющая суть понятия "проект", его прагматическая направленность на результат, который можно получить при решении той или иной практически или теоретически значимой проблемы. Этот результат можно увидеть, осмыслить, применить в реальной практической деятельности.» [2]. Суть метода проектов заключается в том, что обучаемый, в группе или индивидуально, создает свой собственный проект, который он выполняет в течение определенного промежутка времени. При этом, при выполнении проекта подразумевается совокупность различных работ по исследованию темы, поиску нужной информации, и творческому подходу к решению поставленной задачи. Результатом выполнения проекта должна быть работа, готовая для использования. К проекту выдвигаются определенные требования, такие как:

- наличие проблемы или задачи, требующей исследования для ее решения;
- практическая значимость результатов выполнения проекта;
- самостоятельная или групповая деятельность учащегося;
- структурированность проекта и указание результатов достигнутых на определенных этапах выполнения проекта.

Под игровым проектом будем понимать некий программный продукт,, при создании которого перед учащимся ставится цель создать видеоигру.

Создать игровой проект можно различными способами:

- использовать готовые библиотеки для работы с графикой и звуками;
- написать проект «полностью с нуля», не прибегая к использованию сторонних разработок;
- использовать готовые игровые движки.

У каждого из этих способов есть свои преимущества и недостатки. При написании проекта с нуля затрагивается куда большее количество задач, нежели при использовании готовых библиотек или готового игрового движка, однако то количество времени и тот минимальный порог знаний, необходимый для успешного создания проекта при написании своего движка гораздо больше, нежели при использовании готовых продуктов. Несомненным плюсом такого подхода является полная независимость от той платформы, на которой мы собираемся разрабатывать проект. При использовании сторонних библиотек минусами является то, что, во-первых, возникает вопрос поиска и выбора более оптимальной библиотеки, а во-вторых требуется сначала изучить эти библиотеки, прежде чем начать с ними работать, что также влечет за

собой затраты временного ресурса. Наиболее оптимальным решением в этом случае является использование готовых игровых движков.

Существуют различные программные продукты в этой области, от узкоспециализированных, созданных для определенной платформы и ограниченных своим логическим языком программирования, до кроссплатформенных, с открытым исходным кодом и дающих выбор среди различных языков программирования. Минусами таких средств могут являться некоторая сложность в изучении, наличие платной лицензии, отсутствие локализованной документации, а также привязанность инструментария для разработки к какой-то одной определенной платформе.

Для создания игрового проекта могут быть использованы следующие программные продукты:

- Construct Classic (только 2d, собственный логический язык программирования, открытый исходный код, бесплатная);
- Construct 2 (только 2d, собственный логический язык программирования, основывается на технологии html5 и javascript, закрытый исходный код, бесплатная версия с ограничениями);
- Torque 3d (только 3d, собственный логический язык программирования, открытый исходный код);
- UDK (только 3d, язык программирования c++, платная);
- GameMaker (только 2d, собственный логический язык, бесплатная версия с ограничениями);
- Unity (2d и 3d, несколько языков программирования, бесплатная, закрытый исходный код).

На основе анализа основных характеристик и функциональных особенностей перечисленных выше программных продуктов нами была выбрана среда Unity [3]. Преимуществами Unity по сравнению с аналогичными программными средами являются:

- бесплатность (в том числе и для создания коммерческих проектов);
- возможность писать на разных языках программирования (на выбор: javascript, boo, c#);
- интуитивно-понятный графический интерфейс;
- возможность компиляции проекта под различные платформы (Windows, Mac, Linux, iOS, Android, web-плагин для браузера);
- большое сообщество пользователей в интернете, обширная официальная документация и как следствие большое количество обучающего материала;
- наличие примеров и различных дополнительных плагинов и материалов, расширяющих функционал редактора.

В состав Unity входят как сам визуальный редактор проекта (Unity Editor), так и среда для программирования Mono. С помощью Unity возможно создать практически любое приложение, не только игровое. Библиотека соз-

данных проектов насчитывает огромное количество, от обучающих и демонстрационных приложений, до любительских и профессионально разработанных игр и сложных систем симуляции различных явлений [4, 5]. Непосредственно в сам Unity входит мощный, при этом простой в понимании графический движок, который может работать как с 3d графикой, так и с 2d. Для упрощения работы, процесс разработки 2d и 3d приложения унифицирован. Стандартные графические возможности Unity могут быть расширены с помощью шейдеров, написанных на HLSL (High-level shader language, язык для написания графических шейдеров, созданный Microsoft). Программирование в Unity осуществляется в отдельных файлах. Сам код можно писать как в сторонних программах (например, в любом текстовом редакторе), так и в среде, которая поставляется вместе с Unity, Mono. Существенных отличий, кроме как в синтаксисе, при выборе языка программирования нет, так как весь написанный пользователем код транслируется в специальный язык, понятный игровому движку. Работа осуществляется в интуитивно понятном визуальном редакторе по принципу WYSIWYG (What you see is what you get — что видишь, то и получаешь). Редактор понимает большинство 3D форматов и позволяет импортировать 3d-модели, кости и анимации из большинства популярных 3d приложений (3ds max, Maya, Blender, SketchUp...). При изменении файлов редактор Unity автоматически обновляет измененные файлы, которые были добавлены в проект. Так же поддерживается большинство 2d форматов — jpeg, bmp, png, gif, tga; различные аудио и видео форматы. Unity допускает работу с текстовыми файлами — xml и txt, и позволяет обращаться к их данным во время исполнения приложения. Встроенный сетевой движок позволяет создавать онлайн приложения прибегая к минимуму усилий. Все свойства объектов в игре могут быть синхронизированы между игроками используя алгоритм сжатия или же могут быть переданы несжатые данные для особо важных ситуаций.

В ходе исследования разрабатывается учебно-методическое пособие «Создание игрового проекта с использованием Unity», имеющее следующую структуру:

- Знакомство с интерфейсом Unity.

Создание нового проекта. Основные принципы работы в Unity.

Иерархическая структура проекта. Добавление стандартных объектов. Основные параметры объектов.

Использование сторонних ресурсов для объектов. Настройки текстур и материалов для объектов.

Использование скриптов в проекте. Глобальные и локальные скрипты. Триггеры. Динамическое создание объектов.

- Основные приемы при разработке игрового проекта.

Создание игрока. Добавление перемещения игроку.

Камера и освещение.

Создания окружения и добавление декораций.

Взаимодействие игрока с окружением.

Дополнительные возможности для игрока. Базовый инвентарь.

Создание врагов. Добавление логики врагов.

Добавление эффектов.

- Игровой проект «Arkanoid».
- Игровой проект «2D side-scroller».
- Игровой проект «3D platformer».
- Создание собственного игрового проекта.

Такая структура учебного материала является наиболее рациональной и позволяет студентам сначала ознакомиться с работой в среде Unity, узнать основные принципы и подходы, применяемые при разработке различных видеоигр, а затем использовать полученные знания, умения и навыки для создания собственного игрового проекта.

Библиографический список

1. Российская федерация. Министерство образования и науки. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 010300 Фундаментальная информатика и информационные технологии (квалификация «бакалавр») [Текст:] : приказ : [утвер. Министерством образования и науки от 8 декабря 2009 г. №712].
2. Полат, Е.С. Метод проектов. // <https://docs.google.com/document/d/13xOCJ50yaEkIzYq2kuRf3nbzVDewud6fcIkMzFqyrq4/>
3. Официальный сайт Unity // unity3d.com URL: <http://unity3d.com/unity>
4. Made with Unity // unity3d.com URL: <http://unity3d.com/showcase/gallery>
5. Unity SIM // unity3d.com URL: <http://unity3d.com/company/sim/>

ОСОБЕННОСТИ ОБУЧЕНИЯ СЛУШАТЕЛЕЙ НА КУРСАХ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ С ПРИМЕНЕНИЕМ СМЕШАННОЙ ФОРМЫ ОБУЧЕНИЯ

Фролова Т.М.

Научный руководитель: кандидат педагогических наук Рожина И.В.

Уральский государственный педагогический университет

Аннотация: статья посвящена проблемам организации смешанного обучения на курсах повышения квалификации. Сочетание очной и заочной форм обучения позволяет решить проблему различного уровня подготовки, мотивации, потребности в объеме знаний слушателей в учебной группе.

Ключевые слова: ИКТ- компетентность, рабочая тетрадь, Google, дистанционное обучение, очное обучение.

Tatiana M. Frolova

FEATURES OF TEACHING STUDENTS REFRESHER COURSES
WITH APPLICATION BLENDED LEARNING