

ТЕХНОЛОГИИ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ НА ПРИМЕРЕ SCRATCH 2

Рассматривается как Scratch 2 и иные среды визуального программирования позволяют изучить азы программирования и алгоритмизации.

ВВЕДЕНИЕ

Визуальное программирование — это способ образного, графического представления разрабатываемого алгоритма, который наиболее естественен для восприятия человека. При визуальной разработке существенно уменьшается количество вынужденных ошибок управления в программе, а следовательно, повышается качество программного продукта.

I. ПОПУЛЯРНОСТЬ SCRATCH 2

Scratch 2.0 — это бесплатная визуальная среда программирования, созданная для обучения детей и начинающих программистов. Она была разработана Массачусетским технологическим институтом в 2013 году. Данная среда использует блоки кода для создания программ и приложений.

Одна из главных особенностей Scratch 2.0 — это то, что он позволяет создавать проекты без необходимости писать код с нуля. Это делает его идеальным инструментом для обучения детей и новичков, которые хотят изучить основы программирования.

Вот несколько причин, почему Scratch стал такой популярной:

1. Простота использования.
2. Доступность.
3. Сообщество.
4. Визуализация.
5. Обучение.

II. SCRATCH 2: ДОСТОИНСТВА И НЕДОСТАТКИ В СРАВНЕНИИ С ДРУГИМИ СРЕДАМИ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ

Некоторые преимущества и недостатки Scratch 2 по сравнению с другими средами:

Преимущества Scratch 2:

1. Простота использования.
2. Большое сообщество.
3. Низкий порог вхождения.
4. Многообразие проектов.

Недостатки Scratch 2:

1. Ограниченные возможности.
2. Низкая производительность.
3. Ограниченный доступ к исходному коду.

В целом, Scratch 2 — отличный выбор для начинающих программистов, которые хотят изучить основы программирования и создавать простые проекты. Однако, для создания более слож-

ных приложений, лучше подходят другие системы визуального программирования, такие как App Inventor или Blueprint.

App Inventor — это интегрированная среда разработки веб-приложений, изначально предоставленная Google, а теперь поддерживаемая Массачусетским технологическим институтом. Он позволяет новичкам в компьютерном программировании создавать приложения для двух операционных систем: Android и iOS, которое по состоянию на 20 января 2023 года находится в стадии финального бета-тестирования. Он использует графический пользовательский интерфейс, очень похожий на Scratch, который позволяет пользователям перетаскивать визуальные объекты для создания приложения, которое можно протестировать на устройствах Android и iOS и построить для работы как Android-приложение. Он использует сопутствующее мобильное приложение, которое позволяет мгновенно проводить тестирование и отладку в реальном времени. При создании App Inventor Google опирался на значительные предшествующие исследования в области образовательных вычислений и работу, проделанную в Google над онлайн-средами разработки.

Преимущества App Inventor:

1. Группировка всей требуемой функциональности в одном блоке.
2. Возможность динамического создания объектов и свойств.

Недостатки App Inventor:

1. Сложность.
2. Необходимость перегруппировки блоков при изменении функциональности.
3. Некорректная работа редактора с высокими блоками.
4. Увеличение вероятности совершения ошибки при указании в строке имени команды, объекта или свойства по сравнению с выбора в блоках имён переменных из списка существующих.

Blueprint — это система визуального программирования, которая позволяет создавать игровые механики и функции без необходимости писать код на языке программирования. Blueprint был создан Epic Games, компанией, разработавшей Unreal Engine, и был впервые представлен в версии Unreal Engine 4.

С помощью Blueprint можно создавать игровые объекты, настраивать поведение персона-

жей, создавать сценарии, включать звук и музыку, настраивать эффекты и освещение, а также многое другое. Эта система программирования предназначена для всех уровней пользователей, от новичков до профессионалов в области разработки игр.

Система Blueprint позволяет создавать сложные системы и механики, используя простые блоки, называемые узлами. Узлы можно соединять между собой, чтобы создавать логические цепочки, которые определяют поведение игровых объектов. При этом не требуется знание языка программирования, хотя это может быть полезно для создания более сложных систем.

Преимущества Blueprint:

1. Быстрое прототипирование.
2. Упрощённое написание скриптов для непрограммистов.

Недостатки Blueprint:

1. Производительность.
2. Трудность с объединением проектов.

III. ПОЧЕМУ ДАННЫЕ СРЕДЫ ПРОГРАММИРОВАНИЯ ВОСТРЕБОВАНЫ

В современном мире визуальное программирование используется в разных сферах деятельности и является востребованным. В отличие от текстового программирования, визуальное программирование может иметь ряд преимуществ, такие как:

1. Возможность вносить изменения в алгоритм прямо во время его выполнения.
2. Легкий способ отлаживать алгоритм благодаря визуальному отображению работы программы.

При исследовании различных сфер применения визуального программирования, таких как: образование (Scratch), робототехника, разработка видеоигр (Unreal Engine Blueprints, Unity Visual Scripting, Unity Shader Graph), 3D-моделировании (Blender) – выясняется, что в разных сферах визуальное программирование показывает свою эффективность в отличие от текстовых в определенных задачах. Если речь идёт об универсальных языках, то лучше себя проявляют классические языки, однако если рассматривать более узкие сферы, то более эффек-

тивным себя может показать визуальное программирование. Так как работа с данными является задачей узконаправленной, то предполагается, что визуальный язык в этой сфере окажется более эффективным средством разработки.

IV. ВЫВОДЫ

Визуальное программирование предусматривает создание приложений с помощью наглядных средств. Средства визуального программирования обычно решают задачи построения пользовательского интерфейса и упрощения разработки приложения путем замены метода "написания программы" на метод конструирования. Визуальное программирование обладает достоинством наглядного представления информации и гораздо лучше соответствует природе человеческого восприятия, чем методы традиционного, текстового программирования. Однако практически все визуальные средства нуждаются в дополнении функциями, которые не могут быть представлены в виде графических конструкций и требуют текстового выражения. Концепция визуального программирования реализована во многих современных средах разработки программных систем. Все ведущие фирмы, создающие средства для программирования и конструирования имеют системы, поддерживающие технологию визуального программирования.

Список литературы

1. New frameworks for studying and assessing the development of computational thinking: [Электронный ресурс] // URL: <https://www.media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/> (Дата обращения: 12.04.2023).
2. Association for Computing Machinery: [Электронный ресурс] // URL: <https://dl.acm.org/doi/10.1145/1592761.1592779> (Дата обращения: 12.04.2023).
3. Сервис Хабр: [Электронный ресурс] // URL: <https://habr.com/ru/articles/344446/> (Дата обращения: 11.04.2023).
4. Сервис DTF: [Электронный ресурс] // URL: <https://dtf.ru/gamedev/192498-blueprints-i-c-v-unreal-engine-plyusy-i-minusy> (Дата обращения: 10.04.2023).

Пименов Илья Александрович, студент 1 курса Факультета информационных технологий и управления БГУИР, mituk996@gmail.com.

Клявда Тимур Олегович, студент 1 курса Факультета информационных технологий и управления БГУИР, urTEN@tut.by.

Шкадинский Артём Дмитриевич, студент 1 курса Факультета информационных технологий и управления БГУИР, artemtiger_2005@mail.ru.

Научный руководитель: Шатилова Ольга Олеговна, старший преподаватель кафедры вычислительных методов и программирования БГУИР, o.shatilova@bsuir.by.