

и функциональности способствует повышению удобства использования, увеличению вовлеченности, укреплению идентичности бренда и улучшению конверсии. Внесённые изменения, такие как сдержанные цветовые палитры, оптимизация визуальных элементов, улучшение читаемости текста и структурирование контента, формируют основу для разработки более интуитивных и комфортных интерфейсов, способных удовлетворять потребности пользователей и повышать эффективность веб-ресурсов.

1. Market Agency. – URL: <https://seizemarketingagency.com/color-psychology-in-web-design/> (дата обращения: 04.09.2025). – Текст: электронный.
 2. Divami. – URL: <https://divami.com/news/the-science-behind-color-psychology-in-ui-design/> (дата обращения: 05.09.2025). – Текст: электронный.
 3. International Journal of Advanced Human Computer Interaction. – URL: <https://www.ijahci.com/index.php/ijahci/article/view/21> (дата обращения: 05.09.2025). – Текст: электронный.
 4. Emerald. – URL: <https://www.emerald.com/jpbm/article-abstract/28/1/50/450728/The-influence-of-brand-color-identity-on-brand> (дата обращения: 06.09.2025). – Текст: электронный.
5. Якас, А. В. Влияние цвета и освещения на психологию восприятия в интерьере / А. В. Якас ; науч. рук. Е. О. Соколова // Молодость. Интеллект. Инициатива : материалы XI Междунар. науч.-практ. конф. студентов и магистрантов, Витебск, 21 апреля 2023 года : в 2 т. – Витебск : ВГУ имени П. М. Машерова, 2023. – Т. 2. – С. 440–442.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ РАБОТЫ С ГРАФИКОЙ В СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Шевцова А.Д.,

*учащаяся 3 курса Оршанского колледжа ВГУ имени П.М. Машерова,
г. Орша, Республика Беларусь*

Научный руководитель – Юржиц С.Л., магистр, преподаватель

Ключевые слова. Компьютерная графика, трехмерная графика, языки программирования, среды разработки.

Keywords. Computer graphics, three-dimensional computer graphics, programming languages, development environments.

В настоящее время трехмерная визуализация и работа с графикой являются важными направлениями в программировании, особенно, когда речь идет о создании компьютерных игр или интерактивных систем моделирования. Вследствие низкого, на первый взгляд, уровня вхождения в данную сферу, большое количество начинающих разработчиков стремятся опробовать себя в создании программ со сложной графикой. Существуют некоторые языки программирования и фреймворки со встроенными решениями для работы с графикой и моделями. Помимо конкретных языков, есть и сторонние библиотеки, которые позволяют создавать программы с трехмерными компонентами и сложным интерфейсом в тех средах программирования, в которых это изначально предусмотрено не было. Избыточность в выборе инструментов для разработки программ может затруднить начало проекта. Выбор среды разработки является проблемой, если рассматривать не только встроенные библиотеки, но еще и сторонние.

Целью является сравнительный анализ встроенных инструментов в языках программирования для работы со сложной графикой. Результат исследования направлен на помочь в выборе языка программирования начинающим в данной сфере разработчикам, который не требует сторонних библиотек или фреймворков для изучения компьютерной графики.

Материалы и методы. Для рассмотрения были выбраны следующие языки программирования: C#(.NET), C++, Java, Processing и Python. Для анализа выбранных языков программирования и их библиотек использовались статьи, книги, учебные материалы в открытом доступе, а также документация конкретных сред и библиотек.

Объектом данного исследования является разработка программ с использованием компьютерной графики. Выбор среды программирования – это предмет исследования.

Результаты и их обсуждение. В рамках проведения исследования были поставлены следующие задачи: анализ языков программирования на возможность работы с графикой; сравнение встроенных библиотек и инструментов для работы с графикой; подведение результатов анализа и сравнения.

Существует не так много языков программирования, которые были созданы исключительно для 3D-графики, чаще всего такими являются языки, направленные на написание программ, выполняемых графическим процессором во время обработки кадра (шейдеры) и языки, специализированные для конкретной области применения (предметно-ориентированные языки, DSL). Примерами языков программирования для написания шейдеров являются HLSL и GLSL. Из предметно-ориентированных языков можно выделить SQL для СУБД и HTML для разметки документов. Однако изучение данных языков программирования во время знакомства с разработкой графики может быть тяжелым и избыточным.

Ключевыми критериями для сравнения являются:

- наличие встроенных библиотек для работы с графикой;
- уровень абстракции;
- производительность;
- ключевые встроенные библиотеки;
- простота освоения.

Уровень абстракции означает степень детализации системы или процесса, при которой скрываются несущественные детали реализации и выделяются только главные признаки. Если у разработчика не стоит цели понять процесс работы графики на низком уровне, то начинать с изучения низкоуровневых API не имеет смысла.

На основе выделенных критериев был проведен сравнительный анализ выбранных языков программирования. Результаты представлены в сводной таблице и подробно описаны ниже.

Таблица – Сравнительный анализ языков программирования для работы с графикой

Критерий	C#(.NET)	Java	Processing	Python
Наличие встроенных библиотек	Частично	Есть	Есть	Частично
Уровень абстракции	Высокий	Высокая	Очень высокий	Высокий
Производительность	Высокая	Высокая	Средняя	Низкая
Ключевые библиотеки	WinForms	JavaFX	Вся среда	Tkinter
Простота освоения	Высокая	Высокая	Очень высокая	Очень высокая

Данные таблицы требуют более глубокого рассмотрения. Результат «частично» по критерию сравнения «наличие встроенных библиотек» для языков программирования C#(.NET) и Python обусловлен тем, что во встроенных библиотеках функций для работы с графикой недостаточно. В основном, возможности, которые предоставляют вышеупомянутые библиотеки, – это создание интерфейса или рисование простых двухмерных объектов, что недостаточно для разработки программ со сложной графикой.

Уровень абстракции определялся на основе некоторого количества критериев. Самыми значимыми были критерии: близость к аппаратному обеспечению и низкоуровневым API, объем и сложность кода для выполнения базовой задачи, степень контроля над процессом рендеринга и производительностью. Во всех выбранных библиотеках уровень абстракции достаточно высокий, во-первых, потому что разработчику не обязательны знания работы с низкоуровневыми API, во-вторых, многие процессы скрыты, и, наконец, для написания базовой программы не требуется большого количества сложного кода.

Производительность зависит от метода работы языка с кодом. C#(.NET) и Java используют JIT-компиляцию (Just-In-Time). Код сначала преобразуется в промежуточный байткод, который затем компилируется в машинный код во время выполнения. Python является интерпретируемым языком – код выполняется построчно без предварительной компиляции, что значительно медленнее. Производительность Processing сложно срав-

нить с остальными языками программирования, так как он решает принципиально другие задачи и построен поверх Java. Работает он методом трансляции сначала в Java-код, который затем компилируется и выполняется на виртуальной машине Java. Его производительность напрямую наследует все преимущества и недостатки Java.

Простота освоения у всех выбранных языков достаточно высокая за счет большого количества учебных материалов, а также понятный синтаксис самого языка.

Сравнивая библиотеки по функциональности, заметно лучше библиотека JavaFX [2] и язык программирования Processing, в котором нельзя выделить определенную библиотеку, потому что сам язык программирования создан специально для графики.

Библиотеки WinForm [3] и Tkinter [1] не совсем подходят для установленных задач. Они предназначены скорее для простого и быстрого создания интерфейса, а не двухмерной или трехмерной компьютерной графики. Эти библиотеки представляют лишь возможность рисования примитивных 2D-фигур и линий.

Язык программирования Processing подходит для легких проектов, визуализации, рисования и т.п. Однако для создания игр с фотoreалистичной и сложной графикой он не подойдет. Этот язык подойдет тем, кому необходимо относительно быстро и просто создать программу без использования сложной графики.

Библиотека JavaFX лидирует среди вышеописанных библиотек в создании более профессиональной графики за счет своей кроссплатформенности, возможности работы со сложной трехмерной и двухмерной графикой, сценой и освещением, а также высоким уровнем абстракции и умеренной сложностью языка Java.

Итак, по итогам проведенного анализа были сделаны следующие выводы:

Язык программирования C#.NET является хорошим решением для разработки интерфейсов и кроссплатформенных приложений. Встроенных библиотек для работы с графиком помимо примитивных линий и фигур в нем нет. Однако при помощи сторонних библиотек можно писать программы с использованием сложной и даже фотoreалистичной графикой. Примерами таких библиотек являются Veldrid и SharpDX. Наиболее простое решение для разработчиков, которые хотят заниматься графикой на этом языке программирования – это использование игрового движка Unity.

Python редко используется для разработки программ с 3D-графикой. Если с помощью встроенных библиотек является возможным сделать только графический интерфейс, то при добавления сторонних решений спектр возможностей расширяется. Python подходит для работы с 2D-графикой с помощью библиотек Pygame и Arcade.

Язык программирования Processing позволяет работать с графикой и создавать сложные фигуры и освещения. Однако, он не позиционируется как язык, на котором можно создавать приложения со сложной физикой и интерактивной, фотoreалистичной графикой. Для этого язык Java, на котором построен Processing, подходит больше.

Язык программирования Java имеет больше возможностей работы с графикой без дополнительных инструментов. Для начинающего в данной сфере разработчика, который не желает устанавливать сложные и «тяжелые» игровые движки, Java с использованием библиотеки JavaFX будет лучшим вариантом.

Заключение. Таким образом, в ходе проведения исследования был произведен анализ и сравнение языков программирования, и выбранных библиотек в рамках создания компьютерной графики. По итогу исследования был получен следующий результат: для разработки программы со сложной графикой и сценой наиболее подходящим является язык программирования Java со встроенной библиотекой JavaFX, которая позволяет работать со сценой, освещением и графическими объектами.

1. Tkinter – Python interface to Tcl/Tk [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/library/tkinter.html>. – Дата доступа: 12.09.2025.

2. Руководство по JavaFX [Электронный ресурс]. – Режим доступа: <https://metanit.com/java/javafx/>. – Дата доступа: 12.09.2025.

3. Руководство по программированию в Windows Forms [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/windowsforms/>. – Дата доступа: 12.09.2025.