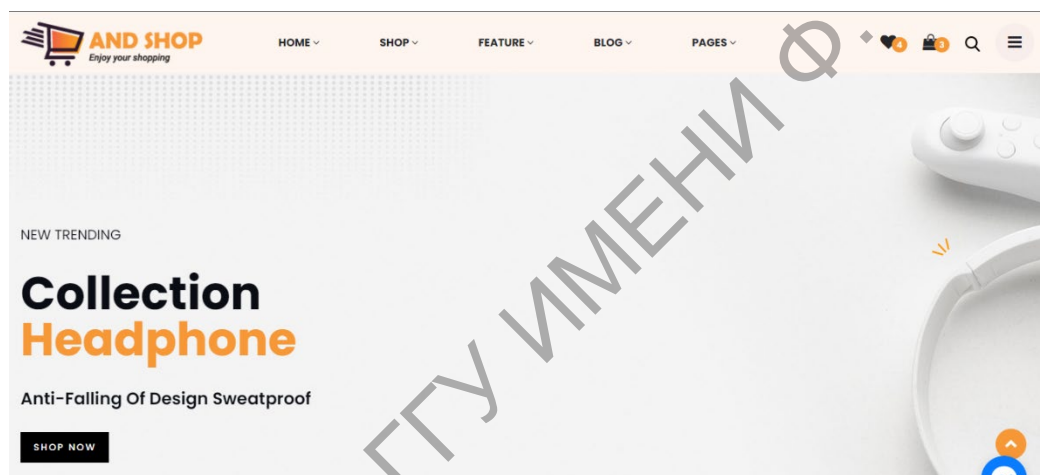


Достаточно иметь доступ к глобальной сети интернет, чтобы в браузере каждый покупатель мог посмотреть на фото товара, а иногда и в 3d-изображении, что даст возможность иметь лучшее представление о продукте, потом прочитав описание и сравнив отзывы можно приобрести этот товар при помощи заказа и доставки по почте. С оплатой тоже все просто, вы можете рассчитаться как безналичным платежом и электронными деньгами, так и наличными получения товара в руки.

На главной странице web-сайта, изображенной на рисунке ниже, представлена стартовая страница интернет-магазина, на которой содержатся различные информационные ссылки в основном меню сверху. Также имеется корзина заказов клиента и поиск.



В результате работы создан web-сайт, в котором размещена информация клиенту о разных товарах с электронной начинкой. При разработке сайта были использованы Spring Boot, Reactjs.

**Н. А. Доломакин**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **С. А. Лукашевич**, ст. преподаватель

## **ОСНОВНЫЕ МЕТОДЫ РАЗРАБОТКИ ИГР НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C#**

В данной работе рассмотрим основные методы разработки компьютерных игр на языке программирования C#. Для создания развлекательных приложений и игр используется игровой движок, с помощью которого мы способны обеспечить музыкальное и звуковое со-

провожение, сетевую игру, графическую визуализацию, встроенные графические сцены и многое другое.

C# («си шарп») – современный объектно-ориентированный и типобезопасный язык программирования. C# предоставляет разработчикам возможность создавать различные типы безопасных и надежных приложений, осуществляющихся в .NET. C# относится к довольно известному семейству языков C, и будет довольно знаком людям, кто работал с такими языками, как Java, JavaScript, C или C++.

Для работы с языком C# многие программисты отдают предпочтение программе Visual Studio. Для работы с C# мы должны установить компонент Microsoft Visual C#, так же мы можем установить различные Framework'и для работы с языком [1–3].

C# популярен за счет своего удобства и простоты. Эти факторы способствуют увеличению количества программистов на данном языке, так как они могут в небольшие сроки создавать функциональные и производительные приложения. Этому способствуют нетипичные конструкции языка и специфичный синтаксис, помогающий максимально органично реализовать намеченные функции.

Популярность языка – это еще один плюс в копилку C#. Большое количество почитателей C# способствуют его развитию. Также это положительно влияет на увеличение вакансий, связанных с разработкой на языке от Microsoft. Программисты, хорошо знакомые с C#, довольно востребованы, несмотря на их большое и постоянно увеличивающееся количество.

C# крайне популярен среди разработчиков видеоигр. Язык используется для разработки игр под Windows, macOS, Android и iOS. Секрет популярности кроется в Unity – платформе для работы с 3D-графикой. C# лучше остальных языков подходит для работы с этим движком. Это послужило причиной выбора многих игровых разработчиков сразу использовать связку Unity+C# для воплощения своих идей в игру.

Оставив Unity, мы можем найти альтернативные способы написания игр на языке C#. Как изложено ранее, для создания полноценной качественной игры мы можем использовать Unity, но если мы хотим написать что-то простое, допустим рассмотрим разработку игры тетрис, тогда мы можем использовать 2D-графику.

Какими же способами мы можем написать тетрис, не используя Unity:

– первый, и самый простой способ написать тетрис – это сделать его консольным приложением. В данном способе мы используем среду разработки Visual Studio. В данном методе нам понадобится напи-

сать консольное приложение, реализующее падение фигур, их движение в стороны и поворот, а также удаление заполненных строк фигур и окончание игры;

– второй способ – графический. В данном способе нам понадобится Windows Forms. Это платформа пользовательского интерфейса для создания классических приложений Windows. Для рисовки нашей игры мы можем использовать пространство имен System.Drawing, которое обеспечивает доступ к функциональным возможностям графического интерфейса GDI+, используя около 50 классов, в том числе класс **Graphics**. Класс **Graphics** предоставляет методы рисования на *устройстве отображения* (другие термины – *графический контекст*, «холст»). С помощью этого класса мы можем нарисовать все контуры, фигуры, но остальной код, связанный с движением фигур остается похож на предыдущий способ;

– третий способ, вызывающий множество неудобств – это использование класса PictureBox в разработке нашей игры. В этом способе мы рисуем основу тетриса, используя данный класс. К примеру, в классическом тетрисе, размерами 10x20, мы будем использовать 200 компонентов PictureBox. Данный способ неудобен тем, что программисту легко запутаться в большом количестве этих компонентов, а также при работе с ними, мы будем писать огромное количество однотипного кода. Код этого способа будет самым длинным из представленных выше.

Конечно, используя платформу Unity, мы сможем сделать хорошую графическую составляющую нашей игры, кроссплатформенность, но сама концепция игры не изменится.

## Литература

1. Краткий обзор языка C# [Электронный ресурс] / Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>. – Дата доступа: 21.03.2022.

2. Двумерная графика на C#, классы Graphics, Pen и Brush [Электронный ресурс] / c-sharp.pro. – URL: <https://sharp.pro/общие-замечания-классы-graphics-pen-и-brush/>. – Дата доступа: 21.03.2022.

3. Фленов, М. Е. Библия C#. 2-е издание / М. Е. Фленов. – БХВ-Петербург, 2011. – 560 с.