

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО**

Кафедра информатики и программирования

**РАЗРАБОТКА ПРИЛОЖЕНИЯ «СБОРНИК  
ГОЛОВОЛОМОК» С ИСПОЛЬЗОВАНИЕМ ИГРОВОГО  
ДВИЖКА UNITY (C#)**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы  
направления 02.03.03— Математическое обеспечение и администрирование  
информационных систем  
факультета КНиИТ  
Ефимова Александра Сергеевича

Научный руководитель  
старший преподаватель \_\_\_\_\_

Е.Е. Лапшева

Заведующий кафедрой  
к. ф.-м. н., доцент \_\_\_\_\_

М.В. Огнева

Саратов 2023

## ВВЕДЕНИЕ

Ни для кого не секрет, что головоломки прочно заняли свое место в современной индустрии развлечений. Предпринимаются попытки выделить компьютерные игры в отдельное направление искусства, наряду с театром, кино и т. д. Разработка игр может быть как увлекательной, так и прибыльной, и история богата примерами. Первые примитивные компьютерные видеоигры были разработаны в 1950-х и 1960-х годах. Они работали на таких платформах, как осциллографы, университетские мейнфреймы и компьютеры EDSAC. Самой первой компьютерной игрой был симулятор ракеты, созданный в 1942 году Томасом Т. Голдсмитом-младшим и Эстле Рэем Манном. Позже, в 1952 году, появилась программа ОХО, имитирующая игру в крестики-нолики, созданную А.С. Дуглас в рамках своей докторской диссертации в Кембриджском университете. Игра работала на большом университетском компьютере, известном как EDSAC. Благодаря своей простоте на рынке мобильных игр появляется множество проектов, которые либо вводят новые жанры, либо меняют существующие. Одним из таких жанров является «головоломка», главное отличие мобильной версии этого жанра от ее версий на более старых платформах заключается в очень простой идее «головоломки» и игровых уровнях, не требующих много времени для решения. В настоящее время разработка игр — это многомиллионный процесс, в котором задействована целая команда разработчиков, сложные современные технологии и даже маркетинговые ходы. **Цель бакалаврской работы** — Разработка приложения «Сборник головоломок» на игровом движке Unity (C#).

Для достижения поставленной цели необходимо решить следующие задачи:

1. Обзор альтернативных игровых движков.
2. Изучение и освоение игрового движка Unity.
3. Описание алгоритмов для игр.
4. Реализация игрового приложения.

**Практическая значимость бакалаврской работы** заключается в том, что сборник головоломок имеет практическую значимость, способствуя развитию умственных способностей, улучшению памяти и когнитивных функций, обеспечивая развлечение и отдых, обучая и образовывая, а также способствуя развитию социальных навыков.

**Структура и объём работы.** Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 81 страниц, из них 55 страниц основное содержание, включая 20 рисунков, список использованных источников информации – 20 наименований.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Первый раздел «Обзор платформ разработки компьютерных игр»** посвящен обзору уже существующих платформ для разработки.

Игровой движок, или game engine, представляет собой совокупность прикладных программ, объединенных в комплекс, который позволяет создавать игры с визуальными эффектами, аудио поддержкой, регулированием передвижения персонажей и координированием их действий согласно сценариям. Он также предоставляет возможность многопользовательской игры в интернете, внедрения визуальных сцен, соблюдения законов физики и других функций.

Термин "игровой движок" стал популярным в середине 90-х годов двадцатого столетия, начиная с использования кода игры Doom для создания новых игр. Использование игрового движка позволяет разработчикам избавиться от необходимости писать основной программный код и сосредоточиться на реализации своих творческих идей, графики, игровой механики и сюжета.

Несколько компаний стали специализироваться на создании игровых движков, а разработчики игр приобретают лицензии на их использование. Некоторые игровые движки, такие как Unreal Engine и Unity, предоставляют бесплатный доступ для некоммерческого использования. Unreal Engine 4 и Unity отличаются своими особенностями и широкими возможностями для создания игр.

Unreal Engine является одним из самых популярных и широко используемых игровых движков в индустрии разработки игр. Он обладает удобством использования, глубокой проработкой и широким ассортиментом дополнительных библиотек. Unreal Engine поддерживает создание игр для различных платформ и операционных систем, включая консоли и мобильные платформы. Разработчики продолжают работу над его улучшением и

анонсировали выпуск Unreal Engine 5.

Unity также пользуется популярностью среди разработчиков игр и имеет кроссплатформенную систему, позволяющую создавать игры в разных жанрах. Unity доступен для некоммерческого использования бесплатно, но при коммерческой разработке требуется выплата комиссионного сбора авторам движка. Unity прост в освоении, но может ограничивать графически.

**Второй раздел «Обзор платформы Unity»** посвящен детальному разбору платформы Unity, указаны его недостатки и преимущества, возможности движка и примеры игр.

Unity - универсальная среда разработки видеоигр, созданная компанией Unity Technologies. Она обладает простым интерфейсом Drag&Drop и поддерживает разработку игр для более чем 25 платформ, включая ПК, игровые приставки, мобильные устройства и веб-приложения. Unity была представлена в 2005 году и продолжает активно улучшаться.

Редактор Unity позволяет легко настраивать и использовать плагины KALI. Для написания скриптов в Unity используются C# и C++, а для расчетов физики - физические движки PhysX и Box2D. Unity поддерживает графический API DirectX.

Проект в Unity состоит из сцен (уровней), каждая из которых содержит игровые миры с объектами, сценариями и настройками. Объекты в Unity имеют компоненты, с которыми взаимодействуют скрипты. Редактор Unity также поддерживает систему наследования объектов и позволяет создавать анимации.

Unity поддерживает оптимизацию проекта с помощью системы Level Of Detail (LOD) и Occlusion Culling. После компиляции проекта создается исполняемый файл игры и папка с данными. Unity также поддерживает множество форматов для моделей, звуков, текстур и скриптов, которые могут быть упакованы в формат.unitypackage и

переданы другим разработчикам или опубликованы в магазине Unity Asset Store.

Unity Asset Server является инструментарием для совместной разработки на базе Unity, добавляющим контроль версий и другие решения.

Игровой движок Unity предоставляет множество функциональных возможностей, включая моделирование физических сред, создание карт нормалей, реализацию динамических теней и многое другое. Он отличается от других движков благодаря наличию визуальной среды разработки и поддержке множества платформ. Unity обладает гибким подходом визуального редактирования объектов, упрощает создание прототипов и поддерживает различные жанры игр, включая головоломки, аркады, платформеры и экшн-игры. Несмотря на некоторые недостатки, Unity остается популярным и мощным инструментом для разработки игр на различных платформах.

Unity имеет несколько недостатков. Во-первых, визуальный редактор может быть ограничен при работе с сложными сценами, содержащими множество компонентов. Во-вторых, отсутствует встроенная поддержка ссылок на внешние библиотеки, что требует дополнительной настройки программистов и может затруднить командную работу. Еще одним недостатком является сложность редактирования шаблонов экземпляров (prefabs). Кроме того, при использовании WebGL-версии движка могут возникать проблемы с производительностью, использованием памяти и работоспособностью на мобильных устройствах из-за особенностей архитектуры, которая требует трансляции кода из C# в C++, а затем в JavaScript.

Unity является платформой выбора для разработки множества игр, приложений и симуляций, и охватывает широкий спектр

платформ и жанров компьютерных игр. На Unity создаются популярные игры в жанре головоломок, такие как Candy Crush и Cut the Rope, а также аркадные игры, включая Flappy Bird и Doodle Jump. Платформеры, такие как Super Mario Run и Rayman Adventures, также успешно разрабатываются на Unity. Кроме того, Unity используется для создания экшн-игр, включающих динамичные бои и элементы приключений, например, Hollow Knight и Ori and the Blind Forest. Unity остается одним из самых популярных и мощных инструментов для разработки игр, обеспечивая разработчикам широкие возможности и гибкость при создании игровых проектов различных жанров и платформ.

Жанр "Головоломка" выбирается по нескольким причинам. Во-первых, он пользуется популярностью и востребованностью среди игроков, так как предлагает решение разнообразных задач и головоломок, что обеспечивает успех проекта. Во-вторых, этот жанр обладает интуитивным геймплеем, который привлекает широкую аудиторию, включая новичков и casual-геймеров. В-третьих, разработка игр в жанре "Головоломка" способствует развитию логического и аналитического мышления у игроков, а также улучшению их когнитивных способностей. Кроме того, этот жанр предоставляет возможность для творчества и экспериментов, позволяя разработчикам создавать уникальные головоломки. Жанр "Головоломка" также подходит для различных платформ, и разработка на Unity упрощает портирование игры на разные устройства. Наконец, создание игр в жанре "Головоломка" требует меньше ресурсов и времени, чем некоторые другие жанры, а Unity предоставляет мощные инструменты и редактор для удобной и быстрой разработки головоломок.

Игры, которые будут реализованы в Сборнике приложений.

- Игра «Три в ряд»
- Игра «Крестики-нолики»
- Игра «Змейка»

- Игра «Морской бой»

**В третьем разделе «Разработка мобильного приложения»** рассмотрена пошаговая реализация проекта.

В процессе разработки игры "Три в ряд" на Unity были осуществлены следующие шаги:

Изменение вида плиток (тайлов) на игровом поле. В скрипте BoardManager была добавлена строка кода в метод CreateBoard, которая задает новый вид плиток  $[x, y] = \text{newTile}$ .

Избавление от повторения плиток. Для этого были созданы новые переменные, включая Sprite, которые использовались в цикле для выбора спрайта, находящегося слева и под текущей плиткой. Это гарантировало отсутствие строк из трех одинаковых плиток при начале игры.

Добавление новых сценариев для основной механики игры "Три в ряд". Скрипт Titles.cs содержал методы Select и CancelSelection, которые меняли цвет плитки и производили звуковые эффекты при выборе и отмене выбора плитки соответственно. Для управления плитками использовалась левая кнопка мыши.

Добавление метода SwapSpriteUnderneath в скрипт Tile.cs, который позволял менять спрайты соседних плиток местами при клике на плитку. В методе использовался параметр render2, который вместе с render использовался для обмена спрайтами.

Создание условия игры, чтобы плитки можно было менять только с соседними. Для этого был добавлен метод в скрипт Tile.cs, который отправлял лучевую трансляцию к соседней плитке и возвращал объект, если он был найден.

Реализация функционала перемещения двух соседних плиток. Теперь игрок может менять местами только две плитки, находящиеся рядом.

Разработка алгоритма для поиска и удаления трех и более

одинаковых плиток в ряду. Был добавлен метод FindMatch, который находил все совпадающие плитки по заданным путям и удалял их спрайты.

Реализация принципа "Домино" для удаления совпадающих плиток. Метод ClearMatch вызывал метод FindMatch для каждого направления (горизонтального и вертикального) и удалял найденные совпадения.

Добавление метода ClearAllMatches, который вызывался после каждого перемещения плиток, чтобы обеспечить корректное функционирование игры.

Добавление сопрограммы в метод CreateBoard в скрипте BoardManager.cs. Сопрограмма проходит по всей доске и находит пустые места для смещения плиток.

Реализация функционала смещения плиток вниз. Сопрограмма перебирает плитки, находит пробелы для смещения и заменяет каждый спрайт с тем, который находится над ним.

Введение системы отслеживания результата игры. Были добавлены переменные Score и MoveCounter, которые обновляются при изменении и отображаются в текстовых компонентах.

Начисление игроку очков за комбо из трех плиток и окончание игры, когда заканчиваются ходы. Для этого была создана сопрограмма, которая ожидает завершения всех изменений в BoardManager.cs и затем вызывает событие GameOver.

В процессе разработки игры "Змейка" на Unity были осуществлены следующие шаги:

Создан новый скрипт GridManager, отвечающий за отрисовку и ограничение движения змейки в пределах сетки. Был создан объект Grid, которому был присвоен пустой GameObject.

Добавлен метод OnDrawGizmos() в GridManager для отрисовки сетки в редакторе Unity и линий спрайтов для отображения сетки во время выполнения игры.

Реализован контроль позиции змейки в скрипте Snake. Змейка

представляет собой последовательность сегментов, где первый сегмент является головой. На каждом шаге змейка обновляет свою позицию в указанном направлении, и при столкновении со стеной игра завершается.

Создан скрипт `AppleSpawner` для создания и управления яблоками на сцене. Скрипт создает новый объект яблока и обновляет его позицию каждый кадр, если яблоко отсутствует на сетке.

Реализована обработка столкновения змейки с яблоками в методе `OnTriggerEnter2D`. При столкновении с яблоком, в змейку добавляется новый сегмент, яблоко удаляется, и игра продолжается. В случае столкновения с самой змейкой, вызывается метод `ResetState` для перезапуска игры.

В процессе разработки игры "Крестики нолики" на Unity были осуществлены следующие шаги:

Создан `UI Canvas`, который представляет собой контейнер для UI элементов.

Созданы UI элементы, такие как кнопки, для отображения ячеек на игровом поле.

С использованием компонента `GridLayoutGroup` на объекте `Panel` была создана сетка 3x3, автоматически размещающая кнопки.

Каждой кнопке было присвоено уникальное имя или тег для их идентификации в скриптах.

Написаны скрипты `GameManager` и `Cell` для управления игрой и отдельными клетками соответственно.

Скрипт `GameManager` отвечает за общее управление игровым процессом, включая определение текущего игрока, проверку победителя и смену игрока после каждого хода.

Скрипт `Cell` отвечает за каждую отдельную клетку на поле, включая обновление внешнего вида клетки и вызов функций проверки победителя и смены игрока.

Созданы два игровых поля размером 10x10, представляющие собой 2D сетку для каждого из игроков.

Созданы списки или массивы для представления игровых полей, где каждый элемент представляет отдельную ячейку на поле.

Создан префаб ячейки и клонирован для каждой ячейки на игровом поле, чтобы визуализировать сетку на экране.

Используя систему ввода Unity, отслеживаются клики мыши, чтобы определить выбранную игроком ячейку.

Реализована функциональность размещения кораблей на игровом поле посредством кнопки "Разместить флот". Корабли могут иметь различные размеры и могут размещаться только горизонтально или вертикально.

Игрок выбирает ячейку на игровом поле противника для атаки, а затем происходит атака. Если в выбранной ячейке находится корабль противника, он получает урон и ячейка отмечается как попадание.

После каждого хода проверяется, остались ли у противника непотопленные корабли. Если все корабли противника потоплены, игра заканчивается, и объявляется победитель.

Ход переходит к другому игроку, и процесс повторяется с начала, если игра не закончена.

Интерфейс пользователя включает следующие элементы:

Игровые поля: два игровых поля, по одному на каждого игрока, отображающие размещение кораблей и результаты выстрелов.

Панель информации: отображает информацию об игре, очереди хода и количестве оставшихся кораблей.

Кнопки действий: позволяют игроку совершать определенные действия в игре.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы был обоснован выбор программной платформы для создания игр (игрового движка) Unity. Были выбраны примеры игр-головоломок, которые решено было объединить в едином приложении. Было создано приложение, включающее в себя три популярные игры: «Три в ряд», «Змейка» и «Морской бой». Было подробно описан процесс создания данных игр с использованием игрового движка Unity.

### Основные источники информации:

- 1 Зарождение компьютерных игр [Электронный ресурс]  
URL: <http://www.gamer.ru/everything/zarozhdenie-kompyuternyh-igr>
- 2 Calabrese D. Unity 2D Game Development. – USA: Packt Publishing Ltd, 2014. – 126 p
- 3 Menard M., Wagstaff B. Game development with Unity. – Nelson Education, 2015. – 433 p
- 4 Что такое Unity [Электронный ресурс] URL:  
<http://web.spt42.ru/index.php/chto-takoe-unity-3d>
- 5 Pereira V. Learning Unity 2D Game Development by Example. – US: Packt Publishing Ltd, 2014. – 266 p
- 6 Игровой движок – написать самому или взять готовый?  
[Электронный ресурс] URL: <http://3dg.me/ru/gamedev/basics/igrovoy-dvizhok-napisat-samomu-ili-vzyat-gotovyuy>
- 7 Unity Real-Time Development [Электронный ресурс]  
URL: <https://unity.com/>
- 8 Пасько Д.Н. Современные игровые движки. Инновационная наука, 2016 – № 2-3 (14). – С. 127–130.