

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра информатики и программирования

РАЗРАБОТКА КОМПЬЮТЕРНОЙ ИГРЫ «РУССКИЕ ШАШКИ»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Алхазов Денис Германович

Научный руководитель:

доцент

Е.В. Кудрина

Зав. кафедрой ИиП:

к.ф.-м.н., доцент

М.В. Огнева

Саратов 2021

ВВЕДЕНИЕ

Актуальность темы. Разработка компьютерных логических игр является особенно актуальной в настоящее время, игровая индустрия растет в геометрической прогрессии, рынок развлечений непрерывно развивается, огромное число людей по всему миру желает провести время в компьютерных играх, объясняется это тем, что человек погружается в игровой процесс и получает удовольствие от игры благодаря своим решениям или действиям, способным привести к победе или поражению. Игровой мир развлечений является очень популярным среди людей любых возрастов. Доступность игр является одним из ключевых факторов такой популярности игровой индустрии. Сегодня компьютерные игры распространены по всему миру, эта тенденция продолжается уже много лет. Над созданием крупных игровых проектов работают целые компании и корпорации, для них создание игр является неотъемлемой частью работы, которую они выполняют. Так же, что немаловажно, игры приносят колоссальную прибыль, которая по своим масштабам приближается к киноиндустрии, а возможно и обойдет ее в будущем. В настоящее время трудно себе представить жизнь без развлечений. Это дает нам полное право говорить об актуальности разработки компьютерных игр.

Важно отметить, что игры не всегда бывают развлекательными, существуют и образовательные игры для развития логики, такие игры заставляют человека принимать решения и обдумывать свои действия для достижения максимального прогресса в игровом процессе, к такому виду игр можно отнести шахматы или шашки, данные игры являются образцовыми категории логических игр. Необходимо отметить, что такие игры имеют свое применение в учебных и образовательных целях, благодаря играм процесс обучения можно сделать не только разнообразным, но и более продуктивным.

Шашки и шахматы были придуманы в далекой древности, в наше время их адаптировали под современные реалии, выпустив их в цифровом виде.

Цель бакалаврской работы – разработать и реализовать полноценное игровое приложение «Русские шашки» с возможностью хранения статистики побед и поражений.

Поставленная цель определила **следующие задачи**:

1. познакомиться с историей развития компьютерных игр;
2. описать математический аппарат, используемый при проектировании приложений;
3. изучить алгоритмические основы проектирования логических игр;
4. рассмотреть инструментальные средства, применяемые в проектировании компьютерных игр;
5. разработать концепцию приложения «Русские шашки»;
6. разработать архитектуру приложения «Русские шашки» и реализовать её на языке программирования C#;
7. провести исследование влияния способа оценивания игровых позиций и глубины построения дерева игры на эффективность игровой стратегии компьютера.

Методологические основы программирования компьютерных логических игр представлены в работах Е.Р. Даниловцевой [1], М. Лопаткина [2], В.В. Потопахина [4], Скита Джона [5].

Практическая значимость бакалаврской работы состоит в получении опыта разработки игровых проектов, проведении исследовательской работы посредством экспериментов со временем, а также в получении полноценного приложения на платформе Windows.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 46 страниц, из них 42 страницы – основное содержание, включая 22 рисунка и 2 таблиц, цифровой носитель в качестве

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы программирования компьютерных логических игр» посвящен обзору истории развития компьютерных игр, рассмотрению математического аппарата и алгоритмических основ проектирования компьютерных игр, обзору инструментальных средств, применяемых при разработке компьютерных игры.

При рассмотрении истории развития компьютерных игр была приведена классификация игр по жанрам, подробно рассмотрен каждый жанр, приведены примеры игр для каждого жанра.

Изучая математический аппарат, используемый при проектировании компьютерных игр, был рассмотрен алгоритм минимакса, в основе данного алгоритма лежат методы максимизации и минимизации. Узловое значение в дереве минимакса принимает следующее значение:

$$\text{Val}(p) = \begin{cases} \text{Utility}(p), & \text{если } p \text{ – терминальное состояние} \\ \max_{s \in \text{next}(p)} \text{Val}(s), & \text{если } p \text{ – узел MAX} \\ \min_{s \in \text{next}(p)} \text{Val}(s), & \text{если } p \text{ – узел MIN} \end{cases} \quad (1)$$

где:

$\text{next}(p)$ – узлы достижимые на следующем ходе из состояния p ,

$\text{Utility}(p)$ – функция оценки позиции p .

Выражение, приведенное в формуле 1 было использовано при реализации оценивания игрового дерева.

Для оценивания дерева игры были рассмотрены две оценки. Функция оценивания для «защитного» способа выглядит следующим образом:

$$\text{Utility}(p) = \begin{cases} N_w(p), & \text{если оцениваем позицию белого игрока} \\ N_b(p), & \text{если оцениваем позицию черного игрока} \end{cases} \quad (2)$$

где:

w – белый игрок;

b – черный игрок;

N_w , N_b – число живых фигур белого и черного игрока соответственно в позиции p.

Функция оценивания для «агрессивного» способа выглядит следующим образом:

$$Utility(p) = N_b(p) - N_w(p) \quad (3)$$

Вышеописанные формулы устанавливают поведение или игровую стратегию для компьютера, поскольку на основе оценок, полученных из выражений, приведенных выше, компьютер принимает свое игровое решение.

При рассмотрении алгоритмических основ проектирования компьютерных игр был представлен алгоритм построения дерева игры, а также метод альфа-бета отсечения – оптимизация алгоритма минимакса.

Изучая инструментарий, применяемый в проектировании компьютерных игр, был выбран язык программирования C# и игровой движок Unity, поскольку данные инструменты предоставляют широкий функционал для решения задач реализации приложения.

Второй раздел «Разработка компьютерной логической игры «Русские шашки»» посвящен описанию реализации приложения на языке программирования C# и игровом движке Unity.

При выборе концепции приложения за основу были взяты правила русских шашек:

- Первый ход всегда принадлежит игроку с белыми фигурами.
- Ходы делаются по диагонали, движение шашек осуществляется вперед.
- Пешки могут бить фигуры противника как вперед, так и назад по диагонали.
- Если пешка бьет фигуру противника, то она становится на клетку за ней. Сбитая фигура снимается с игрового поля и выходит из игры.
- Если фигура находится под боем, то бить ее нужно обязательно.

- Побеждает игрок, который сбил все фигуры противника или смог заблокировать все доступные ему ходы.
- Дамка может совершать ходы во всех 4 направлениях по диагонали. Количество таких фигур не ограничивается и может достигать до 12 штук в случае, если игроку удалось сохранить все фигуры и довести их до противоположной части игрового поля.

Архитектура приложения приведена на рисунке 1.

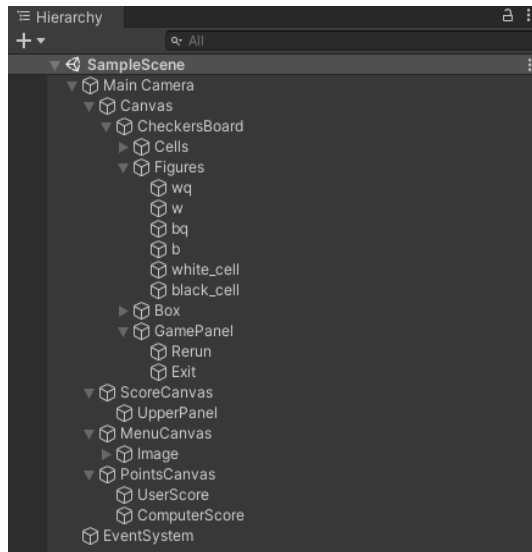


Рисунок 1 – Структура приложения.

Рассмотрим каждый игровой слой, а именно слой представления и слой логики или бизнес-слой. Слой представления состоял из 64 ячеек игрового поля, на которых были отрисованы фигуры. К слою представления был подключен бизнес-слой.

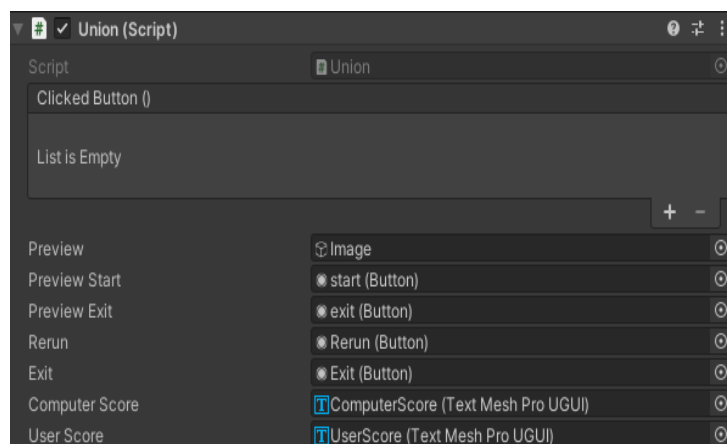


Рисунок 2 – Подключение бизнес-слоя.

Бизнес-слой содержит в себе всю логику развития игры, именно в нем содержится вся реализация алгоритма минимакса построения игрового дерева и нахождения лучшего хода, хранение и обновление статистики побед и поражений.

Немалую часть работы заняла демонстрация самого приложения. Игровое меню выглядит следующим образом:



Рисунок 3 – Игровое меню.

В игровом меню, представленном на рисунке 3 всего 2 кнопки: старт и выход. После нажатия кнопки старт игровое меню исчезает, и пользователь переходит в игровой режим.

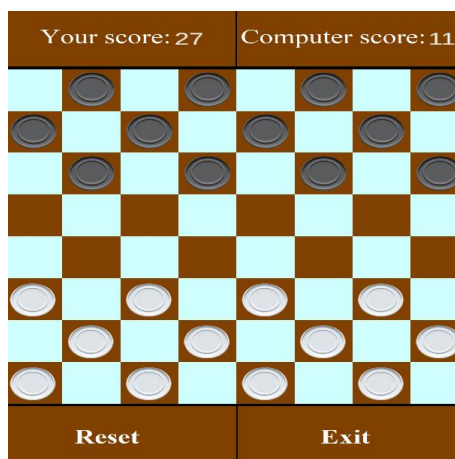


Рисунок 4 – Игровое поле.

На рисунке 4 представлено игровое поле, именно сюда пользователь попадает при нажатии на кнопку старта игры. Здесь пользователь играет за игрока с белыми фигурами, поэтому он ходит первым.

Помимо демонстрации самого игрового поля, были продемонстрированы такие моменты, как выбор фигуры для хода. Фигура подсвечивалась синим цветом при ее выборе.

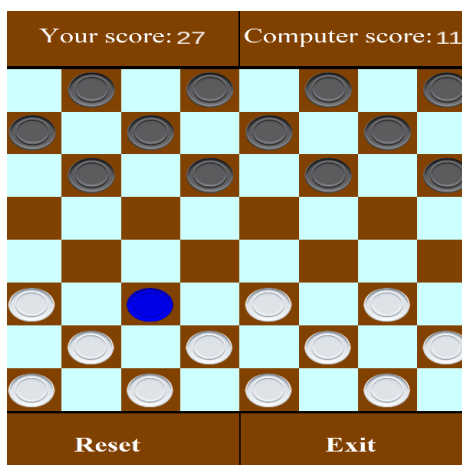


Рисунок 5 – Подсветка фигуры.

На рисунке 5 продемонстрирован выбор фигуры для хода. Таким образом, пользователь понимает, какую фигуру он выбрал в данный момент. Повторное нажатие на данную фигуру отменяет ее выбор.

На рисунках 4 и 5 в верхней части представлена игровая статистика побед игроков. Данная статистика сохраняется посредством сериализации в формате json. При победе игрока с белыми или черными фигурами вызывается метод, который обновляет статистику, вследствие этого json файл статистики обновляется.

Важную часть работы занимают проведенные эксперименты над построением игрового дерева игры в зависимости от глубины, на которую данное дерево строится.

Важно уточнить, что замеры производились на следующей конфигурации ПК:

- Процессор: Intel Core i5-8250U CPU @ 1.60GHz 1.80GHz (4 ядра, 8 потоков)
- Оперативная память: 8 GB (DDR4, 1 канал)
- Видеокарта: NVIDIA GeForce MX250 (GDDR5, 2 GB видеопамяти)
- Внутренний накопитель: Samsung PM961 256 GB NVMe SSD

Как упоминалось ранее, были рассмотрены две оценки игровых позиций, представленные в формулах 2 и 3. В формуле 2 представлена «защитная» оценка, в формуле 3 – «агрессивная» оценка.

Таблица 1 – Зависимость времени построения дерева игры от его глубины при «защитной» стратегии.

Глубина поиска	Измерение времени (с)			Среднее время (с)
	Измерение № 1	Измерение № 2	Измерение № 3	
4	0,047	0,024	0,037	0,036
5	0,173	0,111	0,166	0,150
6	0,896	0,533	0,791	0,740
7	4,418	2,483	3,820	3,574
8	22,825	11,742	18,954	17,847
9	117,606	69,617	107,302	98,175

При анализе данных, представленных в таблице 1 можно заметить разницу между измерениями, во втором измерении достигается наиболее быстрое оценивание игрового дерева, относительно остальных измерений.

Таблица 2 – Зависимость времени построения дерева игры от его глубины при «агрессивной» стратегии.

Глубина поиска	Измерение времени (с)			Среднее время (с)
	Измерение № 1	Измерение № 2	Измерение № 3	
4	0,037	0,027	0,038	0,034
5	0,184	0,114	0,204	0,167
6	0,903	0,547	0,826	0,759
7	4,387	2,706	3,890	3,661
8	21,226	12,478	29,347	21,017
9	149,658	97,305	161,779	136,247

Из данных, представленных в таблице 2 можно заметить разницу между измерениями двух оценок, вторая оценка показала себя чуть более медленной, поскольку в учет берутся фигуры обоих игроков, однако при игре с компьютером вторая оценка показала себя гораздо лучше, чем первая, поэтому в основе приложения лежит «агрессивный» способ оценивания игровых позиций. Что касается глубины поиска, то в приложении за основу взята глубина поиска равная 6, при данной глубине ходы компьютера просчитываются около секунды, благодаря чему пользователю не приходится долго ждать ответа компьютера.

ЗАКЛЮЧЕНИЕ

В заключение хочется отметить, что все поставленные задачи были решены: знакомство с историей развития компьютерных игр, описан математический аппарат, используемый при проектировании компьютерных игр, изучены алгоритмические основы проектирования логических игр, изучены инструментальные средства, применяемые в проектировании игр, разработана концепция и архитектура приложения, проведены эксперименты, позволяющие выбрать и обосновать способ оценивания игровых позиций и глубины дерева игры. В результате была разработана игра «Русские шашки» на языке программирования C# и платформе Unity.

Мы добились хорошего уровня игры от компьютера благодаря построению дерева игры, основанному на использовании метода минимакса и альфа-бета-отсечения. А проведенные эксперименты позволили нам определить, что лучшим способом оценивания игровых позиций для компьютера является «агрессивная» стратегия и дерево игры глубиной 6.

Основные источники информации:

1. Даниловцева Е.Р. Теория игр. Основные понятия. Санкт-Петербург: СПбГУАП, 2003. – 36 с.
2. Лопаткин М. Минимаксный перебор в игровых деревьях. – Нижний Новгород: НГТУ, 2010. – 44 с.
3. Выбор движка для первой игры. [Электронный ресурс]. URL: <https://vc.ru/pixononic/46004-development-engine> (Дата обращения 02.05.2021).
4. Потопахин В.В. Искусство алгоритмизации. Москва: ДМК Пресс, 2011. – 320 с.
5. Скит Джон. C# для профессионалов: тонкости программирования. Москва: Вильямс, 2019. 608 с.