

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»
Направление «Информатика и вычислительная техника»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,

_____ К.А. Домбровский

«__»_____ 2017 г.

Платформа для обучения студентов созданию алгоритмов
принятия решений на примере игровых эмуляций

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ – 09.03.01.2017.382.ПЗ ВКР

Руководитель работы,
доцент каф. «Электронные
вычислительные машины»

_____ Е.С. Ярош

«__»_____ 2017 г.

Автор работы

студент группы КЭ-445

_____ Д.Е. Бухаров

«__»_____ 2017 г.

Нормоконтролер,
ассистент

_____ В. В. Лурье

«__»_____ 2017 г.

Челябинск 2017

АННОТАЦИЯ

Бухаров Д.Е. Платформа для обучения студентов созданию алгоритмов принятия решений на примере игровых эмуляций – Челябинск: ЮУрГУ, ВШЭЖН; 2017, 45 с. 21 ил., библиогр. список – 13 наим.

В выпускной квалификационной работе представлена разработка программного комплекса для обучения студентов созданию алгоритмов принятия решений на примере игровых эмуляций.

Был проведен анализ рынка, выделены достоинства и недостатки конкурирующих продуктов, на основе недостатка конкурентов были выделены требования к программному комплексу.

Проведен обзор программных средств для разработки программного обеспечения, и на основе требований к разрабатываемой системе были выбраны средства разработки и язык программирования.

Был спроектирован и реализован в виде кода программный продукт. Установлено соответствие предъявленным требованиям.

					<i>ЮУрГУ-09.03.01.2017.382 ПЗ ВКР</i>				
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп..</i>	<i>Дата</i>					
<i>Разраб.</i>		<i>Д.Е. Бухаров</i>			<i>Платформа для обучения студентов созданию алгоритмов принятия решений на примере игровых эмуляций</i>	<i>Лит.</i>		<i>Лист</i>	<i>Листов</i>
						<i>Д</i>		3	45
<i>Пров.</i>		<i>Е.С.Ярош</i>				<i>ФГБОУ ВПО «ЮУрГУ» (НИУ) Кафедра ЭВМ</i>			
<i>Н.контр.</i>		<i>В.В. Лурье</i>							
<i>Утв.</i>		<i>К.А.Домбровский</i>							

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1. ПОСТАНОВКА ЗАДАЧИ.....	7
1.1 Цель выпускной квалификационной работы	7
1.2 Задачи выпускной квалификационной работы	7
1.3 Конечный результат выпускной квалификационной работы	8
2. ОБЗОР ЛИТЕРАТУРЫ.....	9
2.1 Обзор литературы по теме выпускной квалификационной работы	9
2.2 Обзор литературы по обоснованию выбора языков программирования	9
2.3 Обзор литературы по обоснованию выбора программного обеспечения... ..	11
2.4 Обзор литературы по обоснованию выбора аппаратного обеспечения	11
3. АНАЛИЗ АНАЛОГОВ	13
3.1 BattleCode.....	13
3.2 StarCraft AI Tournament	15
3.3 Russian AI Cup	16
3.4 CodinGame	17
4. РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ КОМПЛЕКСУ.....	20
4.1. Требования к функциональным характеристикам	20
4.2 Требования к надежности	23
4.3. Требования к составу и параметрам технических средств.....	23
4.4. Требования к информационной и программной совместимости.....	23
4.5. Специальные требования	23
5. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ.....	24
5.1 Построение диаграммы использования	24
5.2 Построение объектной модели	25

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

5.3 Построение диаграммы требований.....	26
5.4 Выбор инструментов для разработки приложения	27
5.5 Создание программного кода	28
ЗАКЛЮЧЕНИЕ	43
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	44

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	<i>Лист</i>
						5
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

ВВЕДЕНИЕ

В настоящее время область разработки искусственного интеллекта для решения различного спектра задач является перспективным направлением для исследований, инвестиций и работы. Образуются новые соревнования по программированию искусственного интеллекта, которые проходят в форме игры, однако платформы имеют свои недостатки, например – языковой барьер при коммуникации с организаторами соревнований, высокий порог вхождения по знаниям программирования, закрытый исходный код платформы проведения соревнования, малое количество типов игр на определенной платформе. Планируется разработать систему, которая будет лишена вышеперечисленных недостатков, и в дополнение привлекать студентов в сферу разработки ИИ (искусственного интеллекта) посредством интерактивного представления результатов работы их кода.

Простейшей формой искусственного интеллекта является система на основе правил. Такая система дальше всего стоит от настоящего искусственного интеллекта, однако для изучения сферы ИИ является наиболее приемлемой. Набор заранее заданных алгоритмов определяет поведение игровых объектов. С учетом разнообразия действий конечный результат может быть неявной поведенческой системой, хотя такая система на самом деле вовсе не будет «интеллектуальной».

Разрабатываемый программный продукт впоследствии будет использоваться как платформа обучения студентов программированию алгоритмов поведения, позволит привлечь новую аудиторию в сферу программирования за счёт интерактивности представления работы исходного кода алгоритмов.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

1. ПОСТАНОВКА ЗАДАЧИ

От кафедры ЭВМ поступил заказ на разработку платформы для обучения созданию алгоритмов принятия решений на примере игровых эмуляций, а также был составлен перечень требований к программному продукту.

1.1 Цель выпускной квалификационной работы

Целью выпускной квалификационной работы является разработка программного комплекса по обучению созданию алгоритмов принятия решений на примере игровых эмуляций, ориентированного на использование студентами первых курсов, поддерживающего визуализацию результатов выполнения их кода.

1.2 Задачи выпускной квалификационной работы

Для достижения выделенной цели разработки были поставлены следующие задачи, требующие решения:

- ознакомиться со сферой программирования искусственного интеллекта и в частности с использованием ИИ в сфере игр;
- провести сравнительный анализ аналогов, выделить преимущества и недостатки;
- составить техническое задание на разрабатываемый продукт;
- на основе требований заказчика разработать архитектуру программного комплекса;
- обеспечить возможность переносимости проекта в дальнейшем на другую платформу;
- используя разработанную архитектуру, создать прототип программного обеспечения;

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

- в течение разработки и по её завершению оценить соответствие программного продукта техническим требованиям.

1.3 Конечный результат выпускной квалификационной работы

Конечным результатам выполнения выпускной квалификационной работы будет являться программный комплекс по визуализации работы алгоритмов принятия решений с возможностью дальнейшего расширения поддерживаемых платформ, языков программирования и типов игр.

Конечный результат разработки будет использован в образовательном процессе студентов при изучении программирования.

Исходный код программы предполагается сделать открытым, что позволит использовать результаты данной работы в других программных продуктах.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

2. ОБЗОР ЛИТЕРАТУРЫ

2.1 Обзор литературы по теме выпускной квалификационной работы

Целью обзора литературы по теме выпускной квалификационной работы является ознакомление с программированием искусственного интеллекта с упором на игровую сферу как наиболее подходящую для обучения студентов младших курсов созданию алгоритмов принятия решений.

Согласно [7], выделяются следующие основные направления использования искусственного интеллекта:

- представление знаний и разработка систем, основанных на знаниях;
- программное обеспечение систем ИИ;
- разработка естественно-языковых интерфейсов и машинный перевод;
- интеллектуальные роботы;
- обучение и самообучение;
- распознавание образов;
- новые архитектуры компьютеров;
- игры и машинное творчество.

Опираясь на [13], наиболее соответствующим форме обучения можно признать направление применения ИИ в играх и, в частности, программирование систем, основанных на правилах, так как преобразование алгоритма поведения в исходный код не является затруднительным.

2.2 Обзор литературы по обоснованию выбора языков программирования

В ходе обучения на технических специальностях студенты изучают различные языки программирования в том числе C, C++, C#, Assembler. C и

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

C++ опираясь на [11], [12], входят в топ-5 наиболее популярных языков программирования. После обсуждения требований с заказчиком, анализа задачи и вакансий в сфере программирования языком, на котором будет разработан графический интерфейс, был выбран C#, а для написания алгоритмов поведения C++. Рисунок 2.1 показывает рейтинг языков программирования на июнь 2017.

Jun 2017	Jun 2016	Change	Programming Language	Ratings
1	1		Java	14.493%
2	2		C	6.848%
3	3		C++	5.723%
4	4		Python	4.333%
5	5		C#	3.530%
6	9	▲	Visual Basic .NET	3.111%
7	7		JavaScript	3.025%
8	6	▼	PHP	2.774%
9	8	▼	Perl	2.309%
10	12	▲	Assembly language	2.252%

Рисунок 2.1 – Рейтинг языков программирования на июнь 2017

Опираясь на [6], также выделены основные достоинства выбора языка C++ для обучения программированию:

- компилируемый язык со статической типизацией;
- сочетание высокоуровневых и низкоуровневых средств;
- реализация ООП;
- STL.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

2.3 Обзор литературы по обоснованию выбора программного обеспечения

Следуя документу [5], – требованиям платформы .NET Framework 4.5, в рамках которой функционирует язык С# и происходит межъязыковое взаимодействие, разрабатываемый программный продукт будет функционировать исключительно на операционных системах семейства Windows, старше Windows XP.

При ознакомлении с [9] были выделены основные преимущества среды разработки Microsoft Visual Studio:

- использование вычислительных мощностей локального компьютера и облака;
- простая реализация общих задач и индивидуальный подход;
- быстрое создание высококачественного кода;
- широкий функционал тестирования приложений;
- возможность реализации идей и решений для широкого спектра платформ, включая Windows, Windows Server, веб-среду, облачную среду, Office и SharePoint.

2.4 Обзор литературы по обоснованию выбора аппаратного обеспечения

Разрабатываемый программный комплекс должен соответствовать требованию аппаратной платформонезависимости, следовательно, не требуется выбор конкретной платформы, на которой данное программное обеспечение будет эксплуатироваться. Однако, исходя из требований [5], аппаратное обеспечение пользователей должно подходить под данные системные требования:

					ЮУрГУ – 09.03.01.2017.382 ПЗ	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

- тактовая частота процессора – не менее 1 ГГц;
- объем оперативной памяти – не менее 512 мегабайт;
- минимальное свободное дисковое пространство – не менее 4.5 гигабайт.

Согласно [14], при последующем портировании приложения, поддерживаемыми архитектурами будут являться:

- s390, s390x;
- SPARC (32);
- PowerPC;
- x86;
- x86-64: AMD64 и EM64T;
- IA64 Itanium2;
- ARM;
- Alpha;
- MIPS;
- HPPA.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	<i>Лист</i>
						12
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

3. АНАЛИЗ АНАЛОГОВ

Аналогами для разрабатываемого программного продукта выступают платформенные приложения, веб-приложения и иные программы, которые включают в себя возможность визуализации кода алгоритма поведения.

Аналогами программного обеспечения «WarCode» на текущий момент выступают:

1. BattleCode [1];
2. StarCraft AI Tournament [2];
3. Russian AI Cup [3];
4. CodinGame [4];

3.1 BattleCode

Турнир, проходящий в Массачусетском институте технологий. Игра построена в жанре стратегии.

Интерфейс игры BattleCode представлен на рисунке 3.1.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

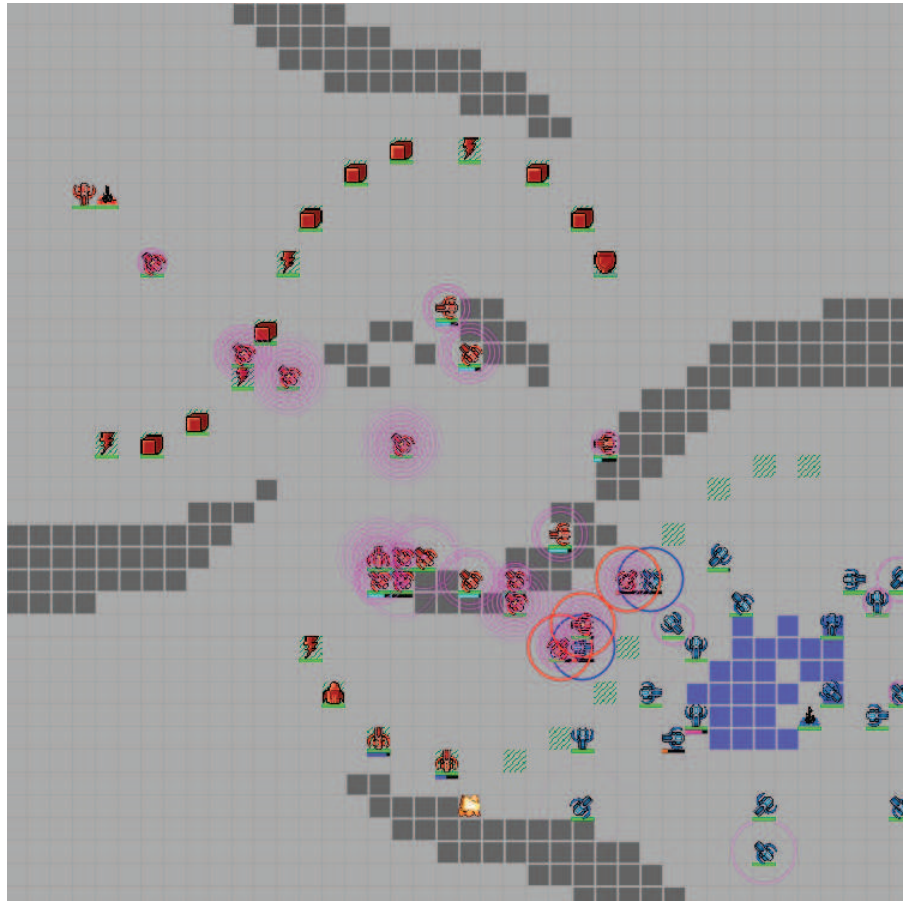


Рисунок 3.1 – интерфейс игры BattleCode

Поддерживаемые языки программирования:

- Java;
- Scala;
- Kotlin.

Достоинства:

- призовой фонд свыше 50 тыс. долл.;
- различные типы турниров (сложность, количество человек в команде).

Недостатки:

- один тип игры на все соревнования;
- языковой барьер для русскоговорящих игроков.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

3.2 StarCraft AI Tournament

Студенческий турнир StarCraft AI Tournament – это образовательное мероприятие, впервые состоявшееся в 2011 году. Он предоставляет конкурентную среду, в основном для студентов, интересующихся искусственным интеллектом и компьютерными науками. Матчи проходят в режиме 1 против 1 в стратегии StarCraft, проводится в Чешском техническом университете в Праге и университете Коменского в Братиславе.

Интерфейс игры StarCraft представлен на рисунке 3.2.



Рисунок 3.2 – интерфейс игры StarCraft AI Tournament

Поддерживаемые язык программирования:

- C++;
- Java.

Недостатки:

- один тип игры;
- языковой барьер для русскоговорящих игроков.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
					ВКР	15
Изм.	Лист	№ докум.	Подпись	Дата		

3.3 Russian AI Cup

Российское ежегодное соревнование по программированию искусственного интеллекта, организуемое Mail.Ru Group.

Интерфейс игры, проводимой в 2016 году на Russian AI Cup представлен на рисунке 3.3.

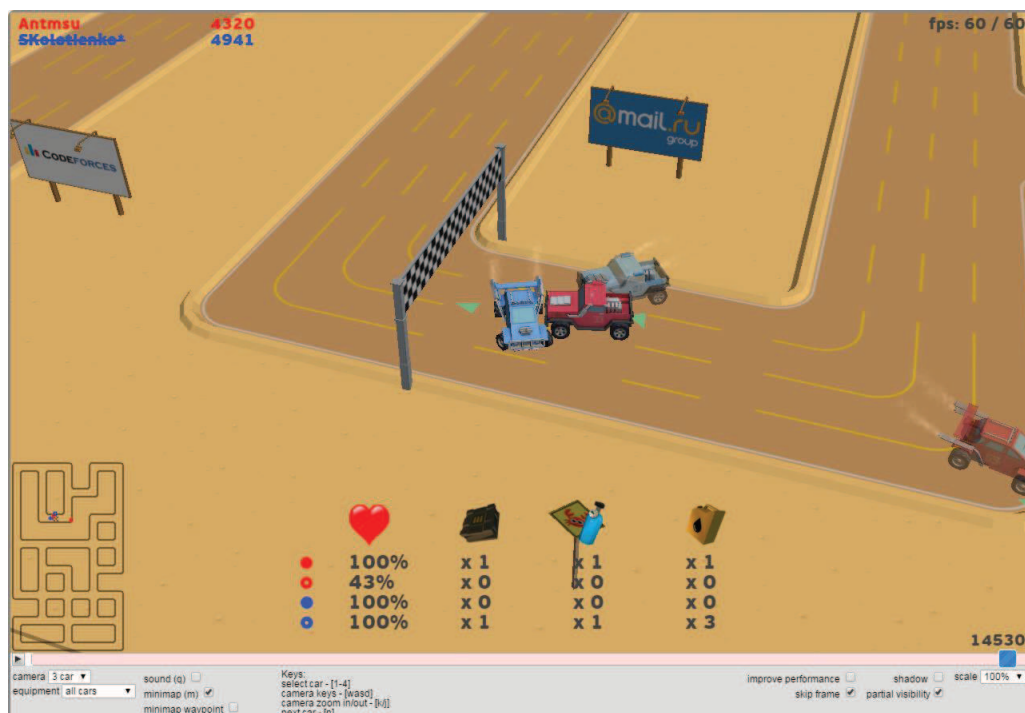


Рисунок 3.3 – интерфейс игры Russian AI Cup 2016

Поддерживаемые язык программирования:

- C++;
- Java;
- C#;
- Python;
- Pascal;
- Ruby.

Из достоинств можно выделить:

- русскоязычное сообщество;
- широкое финансирование, и, как следствие, ценные призы.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
					ВКР	16
Изм.	Лист	№ докум.	Подпись	Дата		

Недостатки:

- один тип игры на, время проведения одного соревнования.
- соревнования проходят раз в год.

3.4 CodinGame

Основанная на веб технологиях игра, позволяющая игроку написать алгоритм защиты или нападения для турели.

Интерфейс игры и среды для загрузки алгоритма представлена на рисунке 3.4.

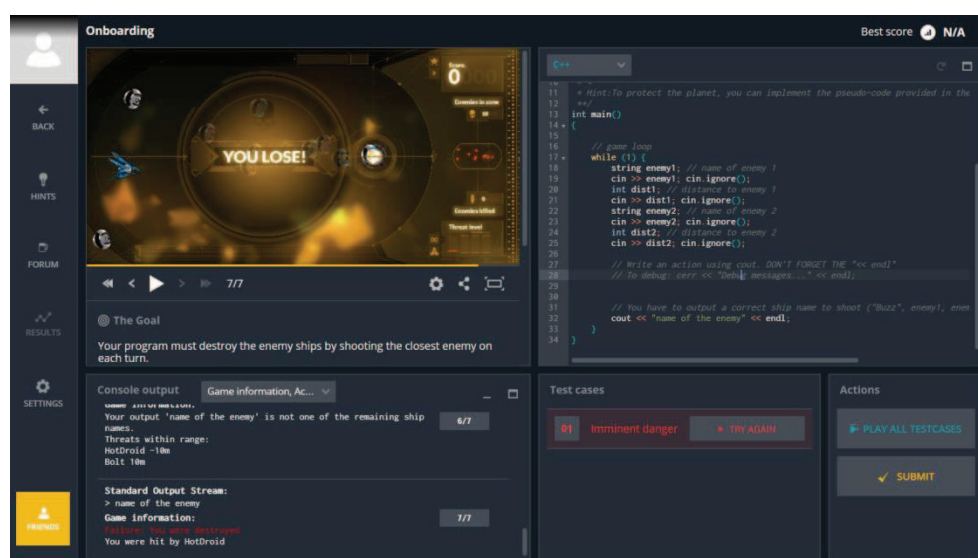


Рисунок 3.4 – интерфейс сайта CodinGame

Поддерживаемые язык программирования:

- C#;
- C++;
- Java;
- Python3;
- C;
- Clojure;
- Pascal;
- Perl;

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
					ВКР	17
Изм.	Лист	№ докум.	Подпись	Дата		

- Python;
- Ruby;
- Rust;
- Scala;
- Swift;
- Swift3.

Достоинства:

- возможность участия без регистрации;
- огромное количество поддерживаемых языков программирования.

Недостатки:

- один тип игры;
- невозможность выбора алгоритма противника.
- языковой барьер для русскоговорящих игроков.

Краткий перечень выявленных достоинств и недостатков аналогов представлен в таблице 3.1.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	<i>Лист</i>
						18
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Таблица 3.1 – сравнительный анализ аналогов

Критерий сравнения	BattleCode	StarCraft AI Tournament	Russian AI Cup	CodinGame
Мультиплатформенность	+	+	+	+
Поддерживание нескольких типов игр	-	-	-	-
Русская локализация	-	-	+	-
Использование вне соревнования	+	+	+	+
Выбор алгоритма противника	+	+	+	-

Изучив наиболее известные аналоги разрабатываемого ПО, можно выделить общие для продуктов недостатки:

- языковой барьер для русской аудитории;
- один поддерживаемый тип игры.

На основании анализа вышеуказанных недостатков были составлены требования, указанные в техническом задании.

Программное обеспечение «WarCode» должно быть лишено вышеуказанных недостатков.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

4. РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ КОМПЛЕКСУ

4.1. Требования к функциональным характеристикам

4.1.1. Составные части программного комплекса

Программный комплекс должен состоять из следующих частей:

- образ игры;
- библиотека с API игры;
- эмулятор.

Образ игры хранит правила механики, визуализации и параметры игрового мира.

API представляет собой либо подключаемую библиотеку, которая хранит доступное API для выбранного типа игры и подключается во время разработки алгоритма принятия решения, либо готовый проект на выбранном языке программирования.

Эмулятор отвечает за эмуляцию игрового мира в соответствии с подключенным образом игры и загруженными алгоритмами принятия решения игроков. Эмулятор отвечает за начало такта игрового мира, принятие сигналов от игровых алгоритмов и расчет состояния игрового мира в каждый такт игрового времени.

4.1.2. Требования к образу игры

Образ игры должен быть независим от языка программирования, на котором будут создаваться алгоритмы принятия решения. В образе хранятся правила взаимодействия объектов игрового мира, характеристики этих объектов, а также интерфейсы, которые характеризуют степени свободы игроков и доступные им действия.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

4.1.3. Требования к API образа игры

Т.к. образ игры независим от языка алгоритма принятия решений, то в нем не могут храниться реализации всех доступных действий объектов игрового мира образа игры. Для одного образа игры будет множество API библиотек для каждого языка программирования, который доступен для разработки алгоритма принятия решения.

API предоставляет полный функционал, доступный алгоритму принятия решения для влияния на игровой мир во время игрового процесса.

По усмотрению разработчика «WarCode» API может поставляться либо в виде подключаемой библиотеки, либо в виде готового проекта, в рамках которого будет производиться разработка алгоритма принятия решения.

4.1.4. Требования к эмулятору

Эмулятор должен обладать двумя режимами работы:

- локальный;
- удаленный.

Локальный режим используется для отладки алгоритма принятия решения и запускается на рабочей машине разработчика алгоритма в режиме localhost, к которому затем подключается процесс самого алгоритма, после чего начинается эмуляция игры. Визуализация эмуляции игрового мира происходит в упрощенном режиме, схематично, отладочные данные представляются в расширенном виде: логирование всех событий игрового мира и состояния всех объектов, вывод координат объекта и его состояния во время эмуляции и т.п.

Удаленный режим рассчитан на развертывание на веб сервере для подключения множества игроков одновременно и проведения одновременно

					ЮУрГУ – 09.03.01.2017.382 ПЗ	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

множества игр. Количество одновременных игр должно быть настраиваемо, чтобы обеспечить устойчивость веб-сервера к нагрузке.

В удаленном режиме во время эмуляции также должно вестись логирование, но в упрощенном режиме, чтобы уменьшить нагрузку на сервер. Визуализация эмуляции, наоборот, должна идти в расширенном режиме.

Процесс подключения алгоритма к удаленному серверу должен происходить в несколько этапов:

- загрузка файла/файлов алгоритма на сервер;
- проверка алгоритма на компилируемость, корректность и соблюдение требований к отказоустойчивости и отсутствию запрещенных действий;
- проведение эмуляции с использованием данного алгоритма.

4.1.5 Требования к созданию алгоритма принятия решения

Алгоритм принятия решения должен представлять собой функцию или метод класса (в зависимости от языка программирования, на котором создается алгоритм принятия решения), запускаемый в каждый такт игрового времени. Сам алгоритм может храниться в нескольких файлах, вызывать другие размещенные в этом файле функции/методы, использовать собственные локальные переменные и т.п.

Алгоритму запрещено подключаться к сети интернет, использовать жесткий диск рабочей машины, пытаться вмешиваться в игровой мир любым другим способом кроме как посредством предоставляемого API, делать вызов системных функций, утилит и других средств ОС на рабочей машине, где он запущен, и т.п.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

4.1.6 Требования к начальному комплекту программного комплекса

Данный программный комплекс должен поставляться с по крайней мере с одним работающим образом игры, API для языка C++ и двумя примерами готовых алгоритмов принятия решений (новичок и продвинутый уровень) для этой игры.

4.2 Требования к надежности

При отказе оборудования программный комплекс должен обеспечить возможность продолжения работы с момента прерывания.

4.3. Требования к составу и параметрам технических средств

Вычислительные машины, на которых будет запущено программное изделие, должны обладать стандартным набором технических средств:

- устройства ввода\вывода;
- сетевая карта.

4.4. Требования к информационной и программной совместимости

Входными данными программного комплекса является исходный код алгоритма на C++. Должна быть заложена возможность в дальнейшем расширить область поддерживаемых языков.

Сам программный комплекс может быть исполнен на одном из следующих языков: C++, C#.

4.5. Специальные требования

Программный комплекс должен распространяться под лицензией GNU GPL.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

5. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРИЛОЖЕНИЯ

Для проектирования и разработки программного обеспечения были выделены следующие этапы:

- построение диаграммы использования;
- построение объектной модели;
- построение диаграммы требований;
- выбор инструментов для разработки приложения;
- создание программного кода;
- тестирование приложения.

Ранее были определены потребности пользователя и функционал, который должен предоставляться разрабатываемым продуктом. На основании этой информации были построены диаграмма использования, объектная модель, диаграмма требований разрабатываемого программного продукта.

5.1 Построение диаграммы использования

При проектировании программного обеспечения были составлены основные сценарии использования, на основании которых была построена диаграмма использования. Из актёров были выделены:

- пользователь – студент или организатор турнира;
- эмулятор;
- образ игры.

Отличием в использовании программного обеспечения студентом и организатором турнира является назначение использования – студент при написании алгоритма поведения проверяет его работоспособность, а организатор турнира из нескольких алгоритмов поведения выявляет тот, который одерживает победу над остальными.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Диаграмма использования представлена на рисунке 5.1.

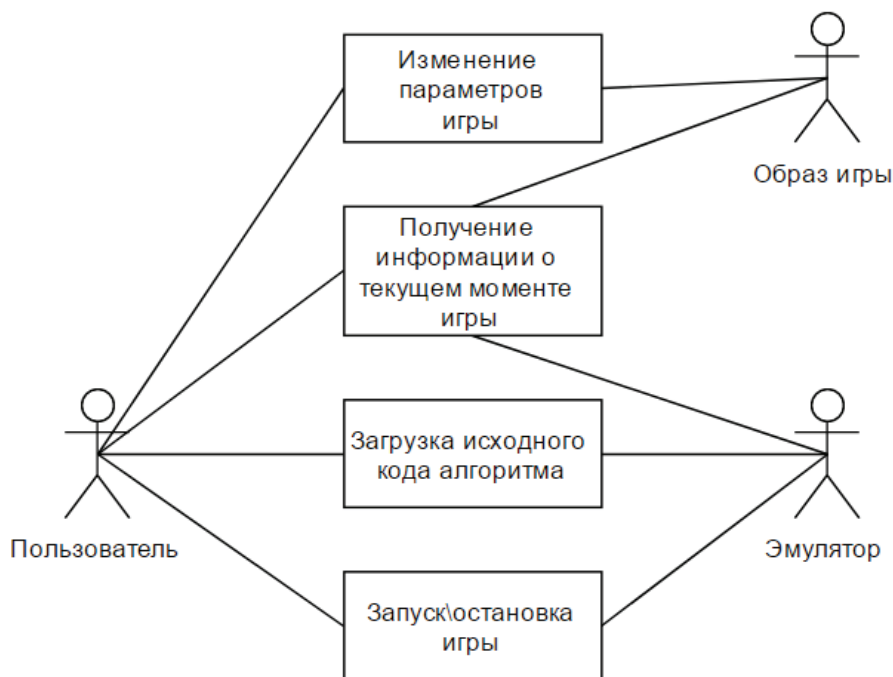


Рисунок 5.1 – диаграмма использования

5.2 Построение объектной модели

На основе диаграммы использования необходимо выделить бизнес-сущности и бизнес-логику.

Бизнес-сущности:

- человек – студент или организатор турнира (различия описаны в п. 5.1);
- эмулятор;
- образ игры;
- алгоритм поведения.

Бизнес-логика:

- получение статистики – текущий ход, номер игрока, которому принадлежит управление на текущем ходе, результат игры (победа, ничья, поражение);

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

- управление игрой;
- задание параметров игры;
- добавление/удаление алгоритмов поведения в игру.

Пользователь устанавливает эмулятор «WarCode», загружает в него предпочитаемый образ игры и написанный им (и/или его оппонентом) ранее один или более алгоритм поведения, использующий интерфейс взаимодействия (который описан в образе игры), настраивает параметры игры и запускает её. В процессе и по завершению игры пользователь должен получать информацию о текущем моменте игры.

Объектная модель представлена на рисунке 5.2.

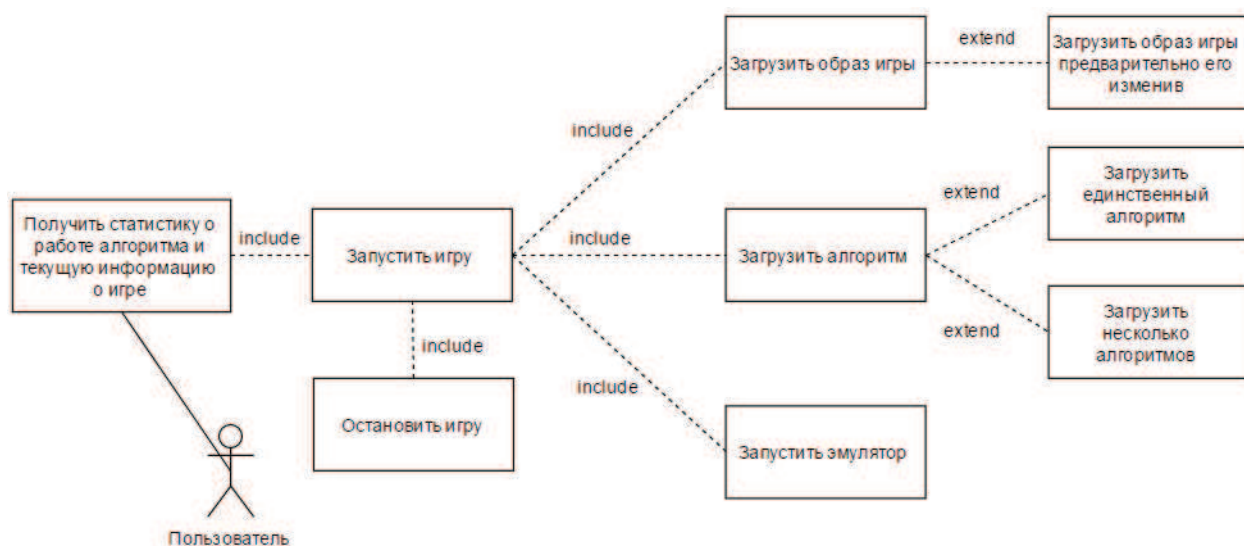


Рисунок 5.2 – объектная модель

5.3 Построение диаграммы требований

На основе обобщения требований технического задания, требований среды, в которой будет использоваться программный продукт, и возможностей самой разработки была составлена диаграмма требований.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

Описание диаграммы:

- для работы программного комплекса «WarCode» требуется два объекта – образ игры и эмулятор (графическая оболочка и объект для обеспечения взаимодействия образа игры и алгоритма поведения);
- эмулятор, являясь приложением, выдвигает требования к аппаратному обеспечению персонального компьютера, а также подключает библиотеку с алгоритмом поведения;
- образ игры основывается на правилах игры и интерфейсе, предоставляющим доступ к управлению субъектами игры (игровое поле, юниты, подконтрольные игроку и т. д.);
- для создания алгоритма поведения необходимо знать, какой функционал (интерфейс взаимодействия с субъектами игры) предоставляет образ игры.

Диаграмма требований представлена на рисунке 5.3.

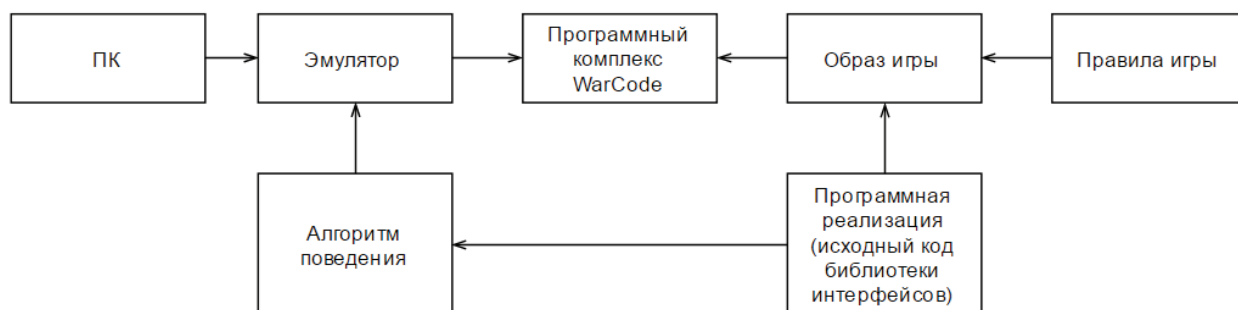


Рисунок 5.3 – диаграмма требований

5.4 Выбор инструментов для разработки приложения

Под платформу .NET на языке C# на данный момент существует несколько сред разработки:

- Microsoft Visual Studio;
- MonoDevelop;

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

- SharpDevelop;
- Rider.

Предпочтительным вариант является Microsoft Visual Studio 2017, так как данная среда разработки предоставляет следующий функционал:

- широкие возможности по отладке и поиску ошибок в коде, так, например, замер времени исполнения каждой функции, что позволяет выявить плохо оптимизированные участки кода;
- встроенные механизмы работы с репозиториями (облачные, в том числе с поддержкой системы версий)
- продукт распространяется бесплатно при условии некоммерческого использования;
- детализированная документация с примерами исходных кодов.

Для переноса проекта на другие операционные системы будет использовано платформенное ПО MonoDevelop, так как весь реализуемый функционал будет поддерживаться в операционных системах, на которых существует реализация Mono:

- Windows;
- Linux;
- Mac OS X;
- Apple iOS.

5.5 Создание программного кода

В ходе разработки программы были написаны исходные коды для следующих компонентов программного комплекса:

- Эмулятор – программа, написанная на языке C# с использованием интерфейса программирования приложение Windows Forms. Обеспечивает связь двух типов DLL библиотек, описанных ниже.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

- Образ игры – DLL библиотека, написанная на языке C++, включает в себя интерфейс и правила, относящиеся к определенному типу игры.
- Алгоритм поведения – DLL библиотека, написанная на языке C++, включает в себя набор правил поведения игрока и интерфейс для взаимодействия с образом игры.

Алгоритм взаимодействия между составляющими программы следующий:

- в основном окне программы пользователь выбирает образ игры, два или более алгоритмов поведения и задает настройки игры;
- эмулятор назначает каждому игроку (алгоритму поведения) номер – 0, 1, 2 и так далее;
- в зависимости от выбранного режима проведения игры (в реальном времени, либо пошаговый по нажатию клавиши) эмулятор вызывает основную функцию (подробнее в пункте 5.5.3) алгоритма поведения по очереди у всех игроков, получает от неё информацию о сделанном ходе и передает её в образ игры;
- после каждого хода игрока эмулятор проверяет текущее состояние игры (необходимо продолжить игру/победа одного из игроков/ничья) при помощи вызова соответствующей функции из образа игры (подробнее в пункте 5.5.2), при необходимости продолжить игру управление передаётся следующему игроку, а в противном случае – выводится информация о победе/ничьей.

Написание кода производилось параллельно во всех трех компонентах продукта.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

5.5.1 Разработка эмулятора

В ходе разработки программного продукта с учетом работы других компонентов системы был спроектирован интерфейс основного окна программы.

Основное окно программы представлено на рисунке 5.4.

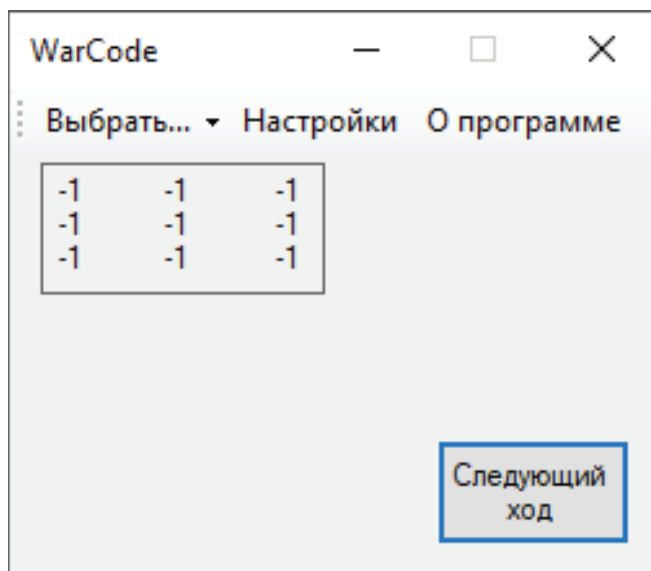


Рисунок 5.4 – интерфейс эмулятора

В верхней части окна находится элемент ToolStrip, включающий в себя выпадающее меню и 2 кнопки.

Выпадающее меню представлено на рисунке 5.5.

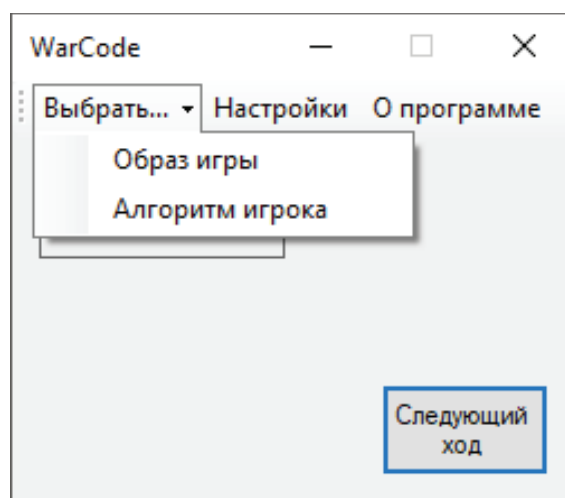


Рисунок 5.5 – выпадающее меню

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

В выпадающем меню пользователь должен по очереди, в любой последовательности при помощи программы проводник выбрать один образ игры и 2 или более алгоритмов поведения игрока.

После выбора образа игры появляется возможность активировать кнопку настройки. При нажатии открывается новая форма, с соответствующими типу игры параметрами. Фрагмент кода, отвечающий за открытие формы настроек представлен в листинге 5.1.

Листинг 5.1 – функция нажатия кнопки настроек

```
private void settingsButton_Click(object sender, EventArgs e)
{
    SettingsForm settings = new SettingsForm(typeOfGame);
    settings.ShowDialog();
}
```

Форма с заданием параметров игры представлена на рисунке 5.6.

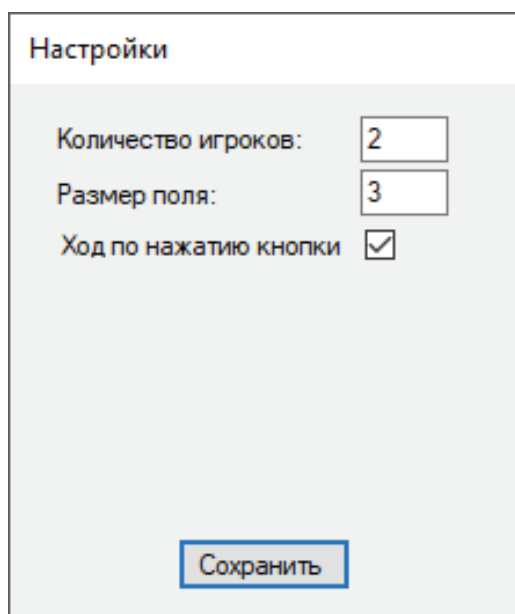


Рисунок 5.6 – интерфейс эмулятора

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
					ВКР	31
Изм.	Лист	№ докум.	Подпись	Дата		

При нажатии кнопки сохранить форма настроек передает введенные параметры основной форме программы.

Для игры типа крестики-нолики возможно пошаговое проведение игры, при установлении соответствующей галки в форме настроек становится доступна кнопка «Следующий ход» в основной форме программы.

При нажатии указанной клавиши происходит выполнение функции tick, в которой производится передача хода следующему игроку и обновление поля игры в основном окне программы, после его хода.

Фрагмент кода функции tick представлен в листинге 5.2.

Листинг 5.2 – функция tick.

```
unsafe private void tick(object sender, EventArgs e)
```

```
{  
    if (typeofGame == 0)  
    {  
        int** area = getMap();  
  
        int* move;  
  
        if (currentPlayer != players)  
            currentPlayer++;  
        else  
            currentPlayer = 0;  
    }  
}
```

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

```

if (currentPlayer == 0)

    move = movePlayer1(area, players, currentPlayer, size);

else

    move = movePlayer2(area, players, currentPlayer, size);

moveImage(move[0], move[1], currentPlayer);

area = getMap();

int i = 0, j = 0;

foreach (Label lbl in Controls[0].Controls.OfType<Label>())

{

    lbl.Text = Convert.ToString(area[i][j]);

    if (j != size - 1)

        j++;

    else

    {

        j = 0;

        i++;

    }

}

}

```

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Если же пользователем в окне настроек не была выбрана галка, отвечающая за ход по нажатию кнопки, то в таком случае функция tick будет исполняться раз в 50 мс.

5.5.2 Разработка библиотеки образа игры

Образ игры включает в себя переменные, отвечающие за текущее состояние игры и интерфейс – перечень функций для корректного (соответствующего правилам игры) изменения обозначенных переменных.

Интерфейс состоит из четырех функций:

- void Init(int, int);
- int** getMap();
- int moveImage(int, int, int);
- int checkGame().

Функция Init – принимает два целочисленных параметра – размер поля и количество игроков. Она нужна для выделения памяти для двумерного массива переменных типа int(поле игры) и сохранения количества игроков для дальнейшего использования в других функциях. Значение в ячейках двумерного массива могут иметь значения:

- -1, если в ячейке нет отметки игрока;
- 0 и более – номер игрока, который поставил отметку в данной клетке игры.

Функция getMap требуется для передачи двумерного массива в эмулятор.

Функция moveImage принимает три параметра – две координаты и номер игрока, который хочет поставить отметку в данные координаты и затем устанавливает значение в двумерном массиве.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Функция `checkGame` – проверяет текущее состояние игры. Возвращает следующие значения:

- -2, если игра не окончена;
- -1, если текущий момент игры обозначает ничью.
- 0 и более – победа соответствующего игрока.

Возможность экспорта этих функций описана в заголовочном файле `export.h`

5.5.3 Разработка библиотеки алгоритма поведения

При пользовании данным программным комплексом исходный код алгоритмов поведения впоследствии будет разрабатываться пользователем. Однако для этого в комплекте с программой должен поставляться тестовый алгоритм и шаблон для написания алгоритма.

Шаблоном для написания алгоритма является DLL библиотека, написанная на языке C++, которая включает в себя одну функцию

```
int* movePlayer(int** area, int players, int player, int size)
```

Данная функция вызывается эмулятором при передаче управления игроку (у каждого игрока индивидуальная реализация этой функции). В параметрах функции передается двумерный массив с текущим состоянием поля игры, количество игроков, номер игрока с доступным в данный момент ходом и размер поля.

Шаблон функции `movePlayer` представлен в листинге 5.3.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

Листинг 5.3 – функция movePlayer

```
int* movePlayer(int** area, int players, int player, int size)
```

```
{
```

```
    int x, y;
```

```
    /*area[i][j] имеет значения:
```

```
    -1 - клетка свободна
```

```
    0..1..2.. - установлен символ данного игрока(игрок №0, игрок №1 и т.д.)
```

```
    players - количество игроков в данной сессии
```

```
    player - номер вашего игрока
```

```
    size - выбранный размер поля
```

```
    Ваш алгоритм далее.
```

```
    */
```

```
    //-----
```

```
    /*Установите координаты для данного шага игры
```

```
    Поле игры нумеруется так:
```

```
    X - сверху вниз, начинается с 0 включительно
```

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

Y - слева направо, начинается с 0 включительно

```
*/
```

```
x = 0;
```

```
y = 0;
```

```
//-----
```

```
int* move = new int[2]{x, y};
```

```
return move;
```

```
}
```

5.5.4 Тестирование приложения

Тестирование приложения проводилось на примере игры 2-х оппонентов. На начальном этапе кнопка настройки не активна. Необходимо выбрать образ игры.

Начальное состояние окна программы представлено на рисунке 5.7.

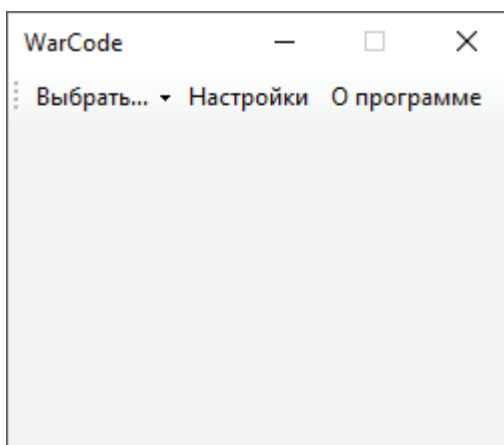


Рисунок 5.7 – начальное состояние окна программы

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

Для продолжения необходимо выбрать образ игры в выпадающем меню в панели инструментов.

Выпадающее меню представлено на рисунке 5.8.

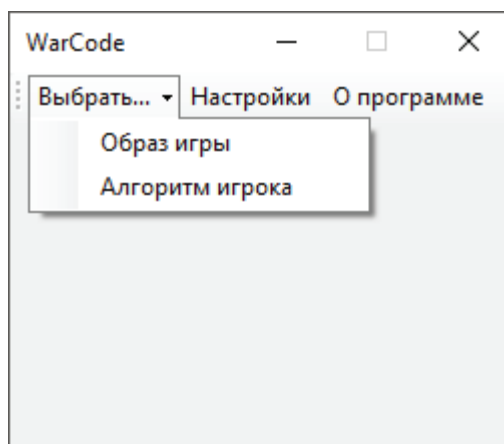


Рисунок 5.8 – выпадающее меню

После нажатия на кнопку образ игры откроется диалоговое окно с возможностью выбора образа игры.

Окно с выбором образа игры представлено на рисунке 5.9

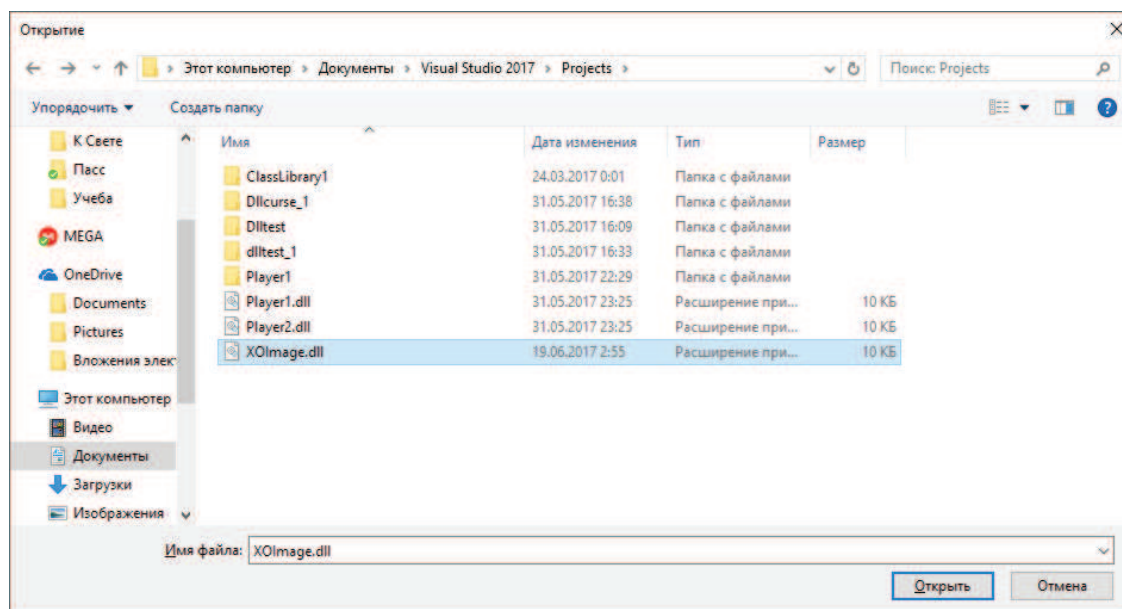


Рисунок 5.9 – выбор образа игры

После того, как пользователь выберет образ игры, станет доступна кнопка настройки в главном окне программы, при нажатии на которую откроется

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

окно с настройками, присущими данному типу игры. После задания необходимых параметров нужно нажать кнопку сохранения для применения изменений. Окно настроек представлено на рисунке 5.10.

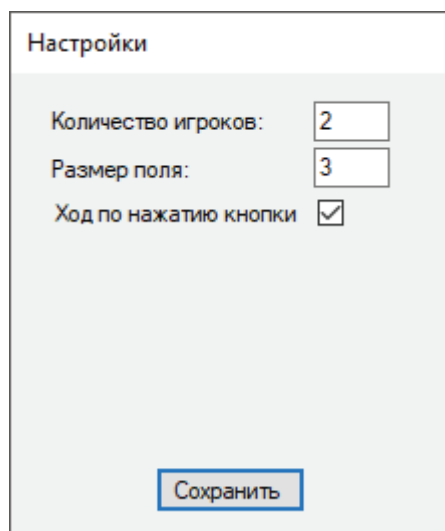


Рисунок 5.10 – задание настроек игры

Также после выбора образа игры станет доступен выбор алгоритмов поведения в выпадающем меню главного окна программы. В зависимости от количества игроков, указанного в настройках, изменяется количество доступных позиций для выбора. Выбор алгоритмов поведения представлен на рисунке 5.11.

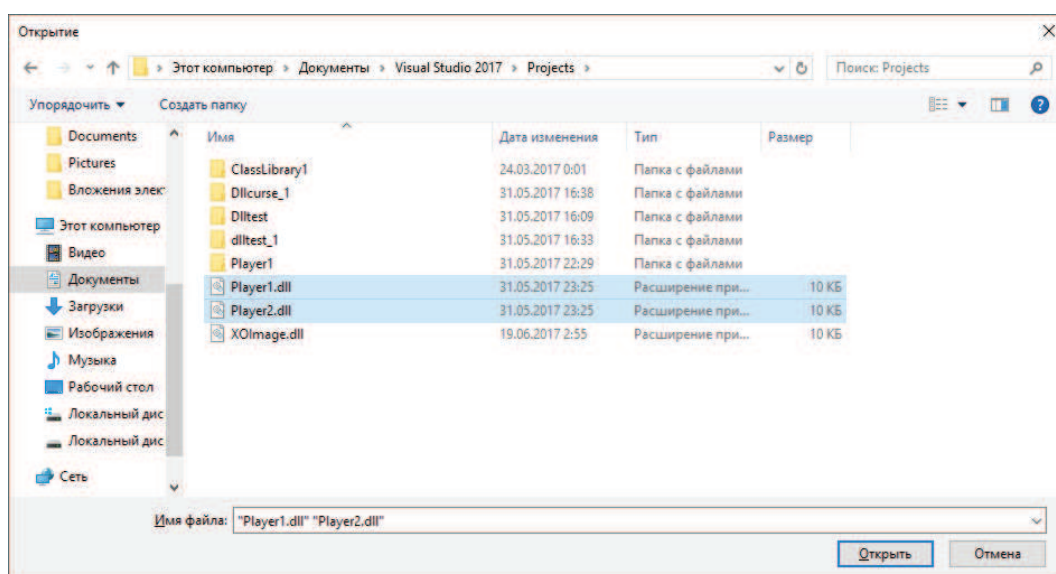


Рисунок 5.11 – выбор алгоритмов поведения

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

После выбора алгоритмов поведения в главном окне программы отобразится статистика игры, присущая данному типу игры, а также поле игры.

Основное окно программы после выбора алгоритмов поведения представлено на рисунке 5.12.

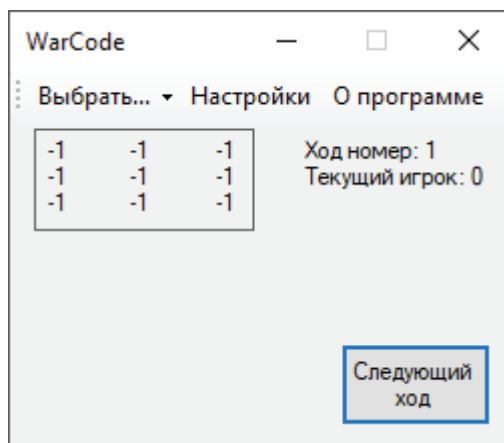


Рисунок 5.12 – начало игры

Так как пользователем в настройках был выбран ход по нажатию клавиши, для хода игрока и передачи управления следующему игроку необходимо нажать клавишу «Следующий ход». После первого нажатия кнопки следующего хода будет произведен ход игрока под номером 0, передано управление игроку номер 1 и обновлена информация в главном окне программы.

Состояние главного окна программы после первого хода представлено на рисунке 5.13.

					ЮУрГУ – 09.03.01.2017.382 ПЗ ВКР	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

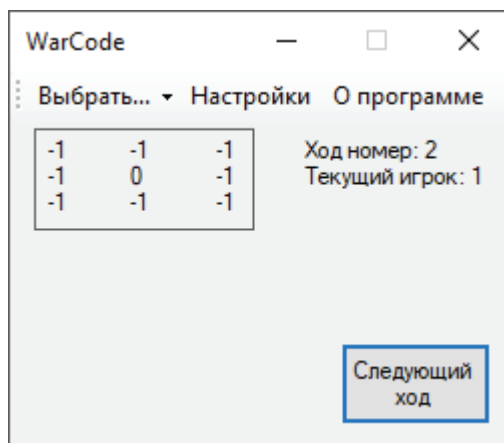


Рисунок 5.13 – состояние после первого хода игрока 0

После повторного нажатия кнопки следующий ход будет произведен ход игроком под номером 1, также обновится информация в главном окне программы.

Состояние главного окна программы после первого хода игрока под номером 1 представлено на рисунке 5.14.

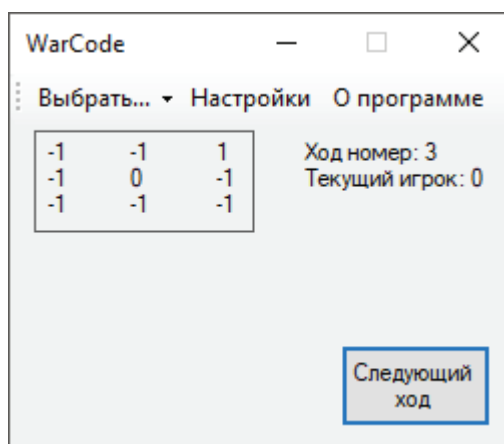


Рисунок 5.14 – состояние после первого хода игрока 1

После заполнения всех доступных клеток, либо до этого момента, но при победе определенного игрока будет выведено сообщение с результатами игры. В данном примере закончились доступные ячейки для ходов.

Состояние главного окна программы представлено на рисунке 5.15.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	<i>Лист</i>
						ВКР
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

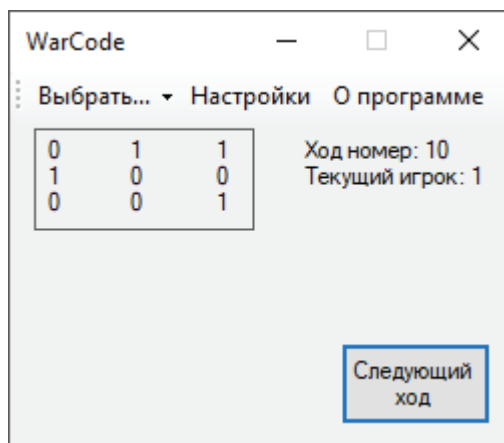


Рисунок 5.15 – состояние после последнего доступного хода

Одновременно с завершением последнего доступного хода появится диалоговое окно с сообщением о результатах игры. В данном примере результатом игры является ничья.

Окно с результатами игры представлено на рисунке 5.16.

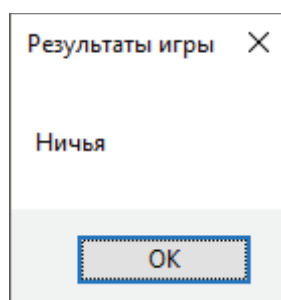


Рисунок 5.16 – сообщение с результатами игры

Было проведено тестирование работы интерфейса приложения, двух алгоритмов поведения и поведение приложения в ходе игры. Программа действует согласно правилам игры, ошибок в ходе работы программы не обнаружено.

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

В результате проведенной работы был рассмотрен перечень аналогов, выявлены их общие недостатки. Изучены технологии программирования интерфейсов пользователя и межъязыковое взаимодействие программного кода. Был проведен обзор и анализ литературы по теме выпускной квалификационной работы, получена информация по современным технологиям программирования из различных источников информации.

При взаимодействии с заказчиком было сформировано техническое задание на разрабатываемое программное обеспечение.

На основе общих недостатков аналогов разработан программный комплекс, лишенный этих недостатков.

Полученное ПО в дальнейшем будет использовано в образовательном процессе студентов при изучении программирования.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Battlecode - AI Programming Competition [Электронный ресурс]. – Режим доступа: <https://www.battlecode.org>. – Заглавие с экрана. – (Дата обращения: 29.05.2017).

2. [SSCAIT] Student StarCraft AI Tournament & Ladder [Электронный ресурс]. – Режим доступа: <http://sscaitournament.com>. – Заглавие с экрана. – (Дата обращения: 28.05.2017).

3. Главная - CodeWizards 2016 [Электронный ресурс]. – Режим доступа: <http://russianaicup.ru>. – Заглавие с экрана. – (Дата обращения: 28.05.2017).

4. Coding Games and Programming Challenges to Code Better [Электронный ресурс]. – Режим доступа: <https://www.codingame.com>. – Заглавие с экрана. – (Дата обращения: 29.05.2017).

5. .NET Framework system requirements [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/framework/get-started/system-requirements>. – Заглавие с экрана. – (Дата обращения: 28.05.2017).

6. Преимущества C++ как первого языка для обучения программированию / Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/202466/>. – Заглавие с экрана. – (Дата обращения: 30.05.2017).

7. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский — СПб: Питер, 2000. — 384 с.

8. Understanding Mono C# Compiler / Mohammad A Rahman, - CreateSpace Independent Publishing Platform, 2012. – 288 с.

9. C#. Разработка компонентов в MS Visual Studio 2005/2008 / Павел Агуров — СПб: БХВ-Петербург, 2008. — 466 с

10. Visual C++ 2010. Полный курс / Айвор Хортон — М.: Вильямс, 2011. — 1207 с

					ЮУрГУ – 09.03.01.2017.382 ПЗ	Лист
						ВКР
Изм.	Лист	№ докум.	Подпись	Дата		

11. The 2016 Top Programming Languages [Электронный ресурс]. – Режим доступа: <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>. – Заглавие с экрана. – (Дата обращения: 29.05.2017).

12. TIOBE Index for June 2017 [Электронный ресурс]. – Режим доступа: <https://www.tiobe.com/tiobe-index>. – Заглавие с экрана. – (Дата обращения: 31.05.2017).

13. Программирование искусственного интеллекта в приложениях / М. Тим Джонс — М.: ДМК Пресс, 2012. — 312 с.

					<i>ЮУрГУ – 09.03.01.2017.382 ПЗ</i>	<i>Лист</i>
						<i>ВКР</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		