Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель от вуза\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Ю. Андреева

(подпись) (и.о., фамилия)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

дата

Отчет о практике по получению первичных профессиональных умений и навыков, в том числе первичных умений научно-исследовательской деятельности (2-ая учебная практика)

«Имитационное моделирование игры «Мафия»»

УП 09.03.04. 13 О

Студент группы ПИ-81 Смирнов Роман Васильевич

и.о., фамилия

Руководитель практики доцент, к.ф.-м.н. А.Ю. Андреева

должность, ученое звание и.о., фамилия

Барнаул 2020

ФГБОУ ВО «Алтайский государственный технический университет   
им. И. И. Ползунова»

Кафедра «Прикладная математика»

**Индивидуальное задание**

**на практику по получению первичных профессиональных умений и навыков, в том числе первичных умений научно-исследовательской деятельности (2-ая учебная практика)**

студенту 2 курса группы ПИ-81 Смирнову Роману Васильевичу

*фамилия, имя, отчество*

Профильная организация Алтайский государственный технический университет

Сроки практики с 29.06.2020 г. по 13.07.2020 г.

Тема: \_\_\_\_\_\_\_Имитационное моделирование игры «Мафия» \_\_\_\_\_\_\_\_

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| **Наименование задач  (мероприятий),  составляющих задание** | **Сроки выполнения** | **Планируемые результаты** |
| 1 | 2 | 3 |
| Составить техническое задание | 1 неделя | Формирование части компетенций ОК-1, ОПК-1, ПК-1 |
| Разработать объектную модель | 1 неделя | Формирование части компетенций ОК-1, ОПК-1, ПК-1 |
| Реализовать модель в виде программы с графическим интерфейсом | 2 неделя | Формирование части компетенций ОК-1, ОПК-1, ПК-1 |
| Написать отчет | 2 неделя | Формирование части компетенции ПК-15 |

Руководитель практики от университета

Андреева А.Ю. доцент каф. ПМ

*подпись Ф. И. О., должность*

Задание принял к исполнению

Смирнов Р.В.

*подпись Ф. И. О.*

**Форма отзыва**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Код компетенции** | **Максимальное количество баллов** | **Итоговое количество баллов** |
| Качество объектной архитектуры ПО (включая диаграмму классов) | ОПК-1  ПК-1  ПК-15 | 15 |  |
| Качество кода программного обеспечения | ОПК-1  ПК-1 | 15 |  |
| Полнота выполнения задания (все ли требования задания выполнены) | ОК-1  ОПК-1  ПК-1  ПК-15 | 15 |  |
| Качество и наглядность графического интерфейса | ОК-1  ОПК-1  ПК-1 | 15 |  |
| Качество подготовки отчета:   * аннотация, введение, заключение, содержание; * техническое задание; * проектирование ПО; * описание ПО; * список использованных источников. | ОК-1  ОПК-1  ПК-15 | 40 |  |
| **Итого:** |  | **100** |  |

Дополнительные комментарии:

Андреева А.Ю., доцент каф. ПМ

*подпись Ф. И. О., должность*

**ОК-1:** способность к самоорганизации и самообразованию

**ОПК-1**: владение основными концепциями, принципами, теориями и фактами, связанными с информатикой

**ПК-1**: готовность применять основные методы и инструменты разработки программного обеспечения

**ПК-15**: способность готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях

В случае выполнения практики по нестандартному заданию или не на кафедре прикладной математики АлтГТУ требуется предоставить рецензию руководителя и отчет в установленной форме

**Аннотация**

Отчет об учебной практике содержит описание программы, реализующей имитационное моделирование трехуровневой пищевой цепочки: техническое задание, структуру данных, структуру файлов, систему конфигурирования, описание программного продукта. Код программы на языке или Qt C++ размещен в репозитории на Github. В приложении А приведены снимки экранных форм программы.

Отчет содержит 17 страниц, 4 рисунка, 5 источников литературы.

**Оглавление**

[Введение 5](#_Toc45785519)

[1 Техническое задание 6](#_Toc45785520)

[1.1 Терминология 6](#_Toc45785521)

[1.2 Описание процесса функционирования модели 6](#_Toc45785522)

[1.3 Требования к функциональности программы 7](#_Toc45785523)

[2 Проект программного продукта 8](#_Toc45785524)

[2.1 Диаграмма классов 8](#_Toc45785525)

[2.2 Жизненный цикл объектов модели 9](#_Toc45785526)

[2.3 Математическая модель 9](#_Toc45785527)

[3 Описание программного продукта 10](#_Toc45785528)

[3.1 Выбор средств реализации 10](#_Toc45785529)

[3.2 Описание классов 11](#_Toc45785530)

[3.3 Конфигурационные файлы 12](#_Toc45785531)

[Заключение 13](#_Toc45785532)

[Список использованных источников 14](#_Toc45785533)

[Приложение А. Снимки экранных форм пользовательского интерфейса 15](#_Toc45785534)

[Приложение Б. Исходный код 16](#_Toc45785535)

Введение

Мафия — салонная командная психологическая ролевая игра с детективным сюжетом, моделирующая борьбу информированных друг о друге членов организованного меньшинства мафиози с неорганизованным большинством мирных жителей. В данной игре можно выделить два компонента. В первую очередь это, конечно же, психологический компонент. Психика человека невероятно сложный механизм, который не может следовать определённым алгоритмам, но эмоции и поведение человека могут раскрыть его планы. Поэтому в процессе игры игроки пытаются скрыть свои эмоции и прочитать эмоции противников. Второй компонент, который можно выделить, математический. Он позволяет оценить ситуацию в числах и определить некоторые вероятности. В отличии от числовой, объектная модель с примитивной психологией позволит определить условия игры, а также красиво визуализировать её процесс. В наше время имитационное моделирование является актуальной областью, развитие которой позволит упростить построение и изучение сложных структур.

1. Техническое задание
   1. Терминология

**Мафия** — преступная группировка, цель которых убить всех мирных жителей. Стараются избавиться от большинства мирных жителей (в т.ч. и от доктора). Не голосуют против своих на ночном голосовании. Побеждают в случае, если количество мирных жителей становится равным количеству мафии.

**Мирные жители** — обычные люди, цель которых вычислить и вывести дневными голосованиями всех игроков мафии. Ради выживания пытаются избавиться от всех членов мафии. Пребывают в полном неведении, никто из них не знает кому можно верить, а кому нет. Побеждают в случае, если все игроки мафии повержены.

**Игровой день** — один из двух временных отрезков игры, во время которого игроки по очереди проводят обсуждение, выставление и голосование друг против друга.

**Игровая ночь** — один из двух временных отрезков игры, во время которого мафия выбирает и стреляет в игрока, а доктор выбирает игрока для лечения.

**Доктор** — дополнительный Мирный персонаж, который обладает правом лечить ночью других игроков по своему усмотрению, но не более одного за Ночь. Это защищает игрока от ночного отстрела Мафии. Не может лечить одного и того же игрока (в том числе и себя) две ночи подряд.

* 1. Описание процесса функционирования модели

Игровой цикл разделён на несколько этапов – дневные и ночные. К ночным этапам относятся – голосование мафии, лечение игрока доктором. При голосовании мафии выбирается жертва, которая в последствии может быть убита. После голосования следует выбор игрока для лечения. Если доктор сделает правильный выбор, то состав игроков не изменяется к утру, иначе человек выбранный мафией выходит из игры и считается убитым. К дневным этапам относятся – обсуждение и голосование. Во время обсуждения игроки высказываются насчёт других игроков положительно или отрицательно. Игрок может высказаться только один раз и только об одном человеке, или воздержаться. Далее следует этап голосования, на котором игроки выбирают кого казнить. Игрок, которого казнили выходит из игры и считается убитым. Игровой цикл прерывается в случае, если выполняется условие победы для одной из сторон.

* 1. Требования к функциональности программы

В программе имеются две формы, одна из которых считается главной.

На главной форме два числовых поля: «Количество игроков» и «Количество преступников», - флажок «Скрыть роли» и кнопка «Сформировать игру». В поле «Количество игроков» содержится общее число игроков в игре. Данное поле может иметь значение в отрезке от 5 до 16. В поле «Количество преступников» содержится число членов мафии среди игроков. Минимально значение этого поля равно 1. Максимальное значение варьируется, в зависимости от значения поля «Количество игроков», и может достигать 6. Флажок «Скрыть роли» позволяет скрыть роли игроков от пользователя. На работу модели это не влияет. При нажатии на кнопку «Сформировать игру» формируется список игроков по данным из конфигурационного файла (см. п. 3.3), игрокам присваиваются роли. Список передаётся второй форме, главная форма закрывается.

Форма модели имеет набор иконок игроков (от 5 до 16) и текстовое поле. Иконки наглядно показывают состояние игроков. Иконка содержит имя игрока, одно из изображений (в зависимости от пола) и значок роли. Если игрок вышел из игры, то на иконка преобразуется (накладывается символ смерти и раскрывается роль, если она была скрыта). При наведении на иконку показывается информация о игроке. В текстовом поле выводятся сообщения о событиях в игре. События происходят в соответствии с порядком в игровом цикле (см. п. 1.2). Максимальное время одного цикла равно 90 секундам.

1. Проект программного продукта
   1. Диаграмма классов

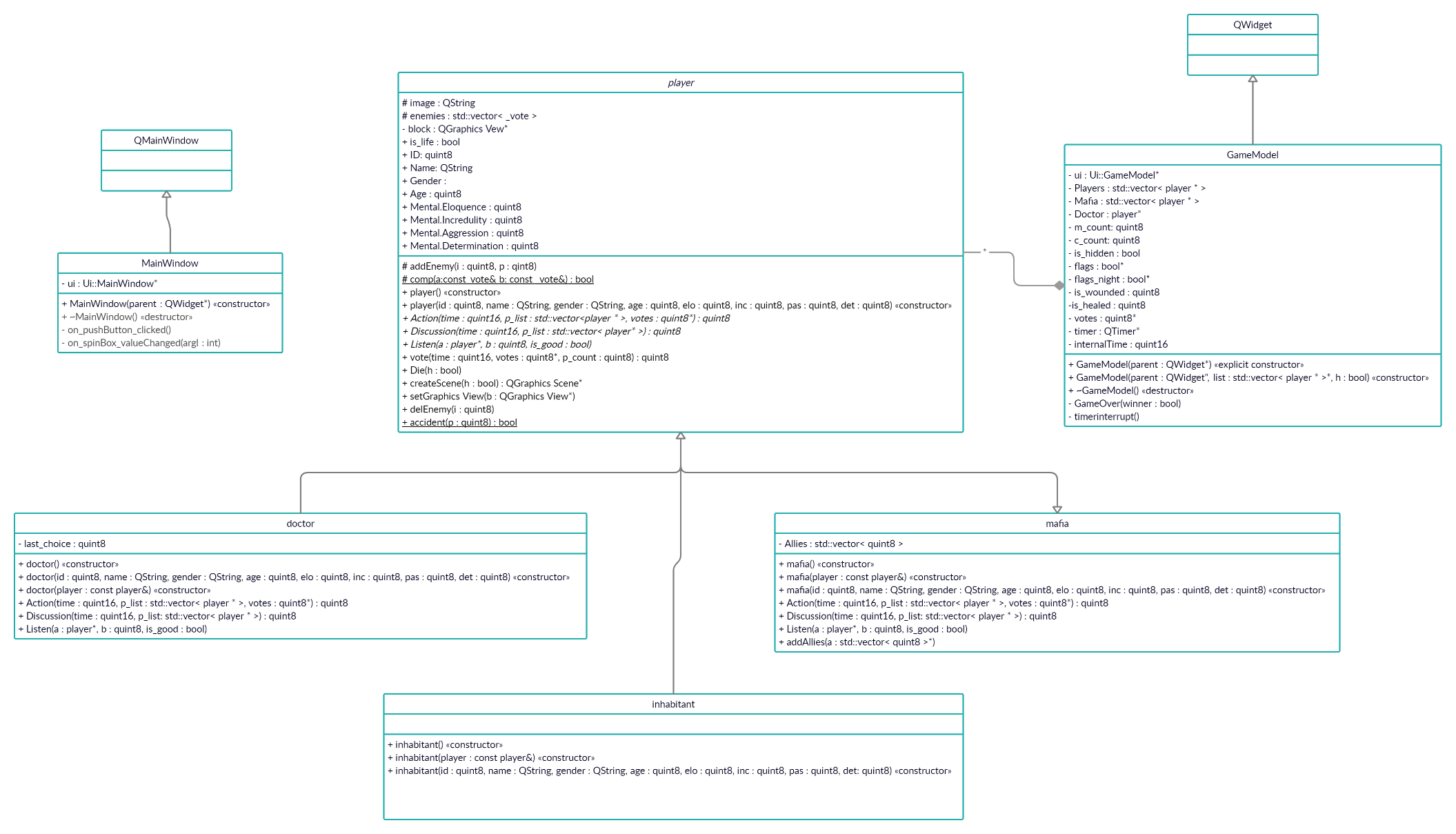


Рисунок 2.1 Диаграмма классов приложения

* 1. Жизненный цикл объектов модели

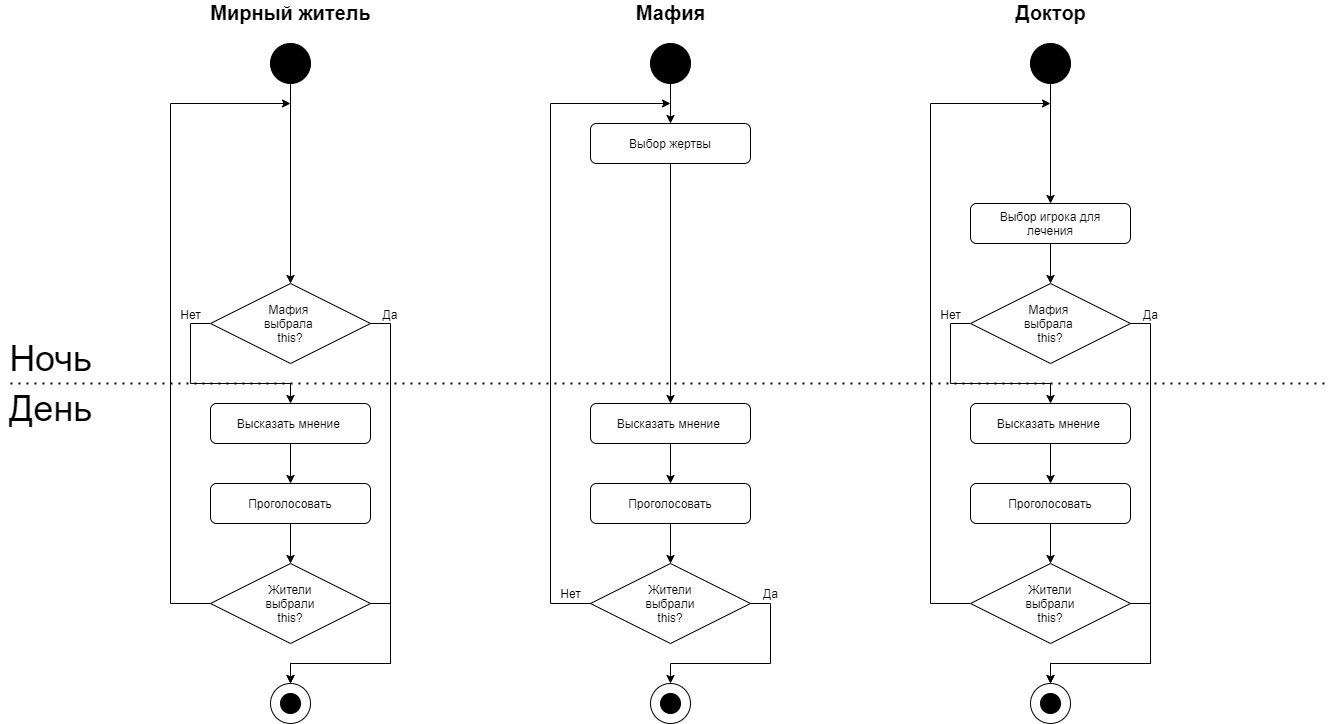


Рисунок 2.2 Диаграммы деятельности

* 1. Математическая модель

Психология личности - вещь совершенно непредсказуемая, и она не может подчиняться определённым алгоритмам. Поэтому действия объектов модели связанны с определёнными вероятностями.

**Основные обозначения:**

* E – уровень красноречия;
* I – уровень скептичности;
* A – уровень скрытой агрессии;
* D – уровень решимости.

**Формулы:**

*Вероятность хорошего отзыва о игроке:*

*Вероятность плохого отзыва о игроке:*

*Вероятность самостоятельного выбора:*

*Вероятность поверить человеку:*

*Вероятность занести человека в список подозрительных с повышенным коэффициентом:*

Из данных выше можно получить вероятности поступков для мирного населения. Н-p, вероятность того, что игрок выскажет плохой отзыв при пустом списке подозрительности равна . Вероятность при непустом списке равна, .

Для членов мафии есть некоторые нюансы:

* *Знание союзников позволяет не верить мирным жителям со 100% вероятностью;*
* *Во время дискуссии есть два варианта: плохо высказаться о мирном игроке с вероятностью или хорошо высказаться о члене мафии.*

1. Описание программного продукта
   1. Выбор средств реализации

Qt - кросс-платформенный инструментарий разработки ПО на языке программирования C++. Есть также <привязки> ко многим другим языкам программирования: Python - PyQt, Ruby - QtRuby, Java - Qt Jambi, PHP - PHP-Qt и другие.

Qt- это луч надежды для программистов, пишущих на языке С++, которые вынуждены сейчас выполнять тройную работу по реализации своих приложений для ОС Windows, Linux и Маc OS Х. Qt предоставляет поддержку большого числа операционных систем: Microsoft Windows, Маc OS Х, Linux, FreeBSD и других клонов UNIX с Х 11, а также и для мобильных операционных систем IOS, Android, Windows Phone, Windows RT и BlackBeгry. Qt использует интерфейс API низкого уровня, что позволяет кроссплатформенным приложениям работать столь же эффективно, как и приложениям, разработанным специально для конкретной платформы. [2]

Qt не является единым целым - она разбита на модули:

* **QtCore** – Основополагающий модуль, состоящий из классов, не связанных с графическим интерфейсом;
* **QtGui** – Модуль базовых классов для программирования графического интерфейса;
* **QtWidgets** – Модуль, дополняющий QtGui «строительным материалом» для графического интерфейса в виде виджетов на С++;
* **QtQuick** – Модуль, содержащий описательный фреймворк для быстрого создания графического интерфейса;
* **QtQМL** – Модуль, содержащий движок для языка QML и JavaScript;
* **QtNetwork** – Модуль для программирования сети;
* **QtSql** – Модуль для программирования баз данных;
* **QtSvg** – Модуль для работы с SVG (ScalaЫe Vector Graphics, масштабируемая векторная графика);
* **QtXml** – Модуль поддержки XML, классы, относящиеся к SAX и DOM;
* **QtXmlPatterns** – Модуль поддержки XPath, XQuery, XSL Т и XmlShemaValidator;
* **QtMultimedia** – Модуль мультимедиа. Собрание классов для работы со звуком, видео, камерой и радио;
* **QtMultimediaWidgets** – Модуль с виджетами для модуля *QtMultimedia;*
* **QPrintSupport** – Модуль для работы с принтером;
* **QtTest** – Модуль, содержащий классы для тестирования кода.

Для данного проекта выбрана библиотека Qt C++, с подключёнными модулями *QtCore, QtGui, QtWidgets* и *QtXml*.

* 1. Описание классов

**Player** –абстрактный класс, от которого наследуются классы игровых ролей.

* Action(quint16, vector<player\*>, quint8) – чистая виртуальная функция, отвечающая за действие игрока в ночное время.
* Discussion(quint16, vector<player\*>) – чистая виртуальная функция, отвечающая за действие игрока во время обсуждения.
* Listen(player\*, quint8, bool) – чистая виртуальная функция, отвечающая за действие игрока в случае когда ему высказывают своё мнение.
* vote(quint16, quint8\*, quint8) – возвращает голос игрока во время дневного голосования.
* Die(bool) – преобразует иконку игрока, показывая, что игрок вышел из игры.
* createScene(bool) – создаёт сцену для вывода иконки игрока
* setGraphicsView(QGraphicsView\*) – привязывает игрока к объекту QGraphicsView
* addEnemy(quint8, qint8) – Добавить номер игрока в список подозрений
* delEnemy(quint8) – Удалить номер игрока из списка подозрений
* comp(const\_vote&, const \_vote&) – Статическая функция, компаратор для некоторых функций библиотеки <algorithm>
* accident(quint8) – Статическая функция, возвращает true или false, с определённой вероятностью

**Inhabitant** – класс мирного жителя, наследник класса player.

* Action(quint16, vector<player \*>, quint8\*) – Действие игрока ночью
* Discussion(quint16, vector< player \* >) – Действие игрока во время обсуждения
* Listen(player\*, quint8, bool) – Действие игрока в случае когда ему высказывают своё мнение

**Doctor** – класс доктора, наследник класса player.

* Action(quint16, vector<player \*>, quint8\*) – Действие игрока ночью
* Discussion(quint16, vector< player \* >) – Действие игрока во время обсуждения
* Listen(player\*, quint8, bool) – Действие игрока в случае когда ему высказывают своё мнение

**Mafia** – класс мафии, наследник класса player.

* Action(quint16, vector<player \*>, quint8\*) – Действие игрока ночью
* Discussion(quint16, vector< player \* >) – Действие игрока во время обсуждения
* Listen(player\*, quint8, bool) – Действие игрока в случае когда ему высказывают своё мнение
* addAlies() – Добавить список союзников

**MainWindow** – класс главной формы.

* on\_pushButton\_clicked() – реакция на нажатие кнопки
* on\_spinBox\_valueChanged(int) – реакция на изменение значения в первом поле

**GameModel** – класс формы модели.

* GameOver(bool) – останавливает таймер и выводит сообщение о победителе;
* timerinterrupt() – прерывание по таймеру, данная функция содержит игровой цикл.
  1. Конфигурационные файлы

Программа работает с одним конфигурационным файлом “players.xml”. В данном файле содержится список игроков с параметрами. Файл представлен в синтаксисе XML.

Корневым является элемент <list> … </list>. В нём содержится список игроков, представленный в виде пустых тегов <player/> со следующими атрибутами:

* **Id** – порядковый номер в списке;
* **name** – имя игрока
* **gender** – пол игрока (“male”, “female”)
* **age** – возраст игрока
* **eloquence** – красноречие (0 - 9)
* **incredulity** – скептичность (0 - 9)
* **aggression** – латентная (скрытая) агрессия (0 - 9)
* **determination** – решимость (0 – 9)

Заключение

Разработанные программы обладают следующими функциональными возможностями:

* имитационное моделирование поведения игроков в процессе игры «Мафия»;
* визуализация процесса моделирования.

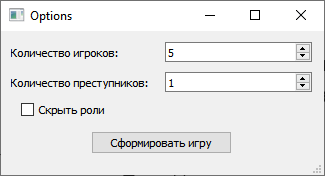
В дальнейшем возможно расширение возможностей программы:

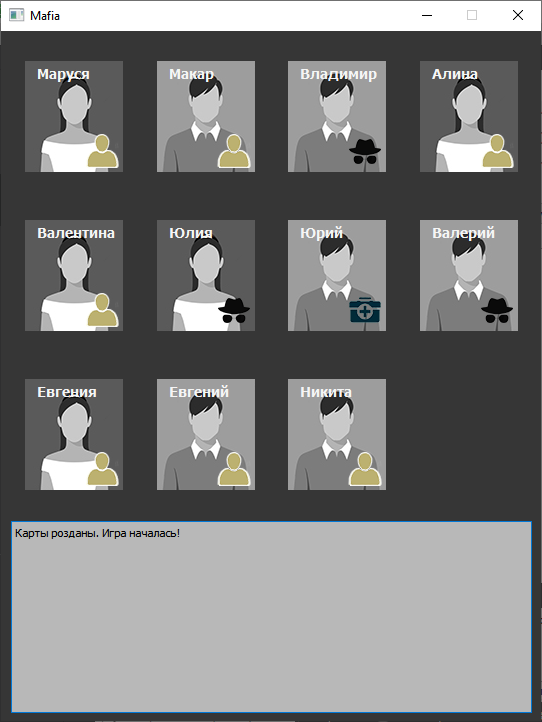
* улучшение искусственного интеллекта игроков;
* реализация возможности работы с базой данных игроков;
* добавление новых игровых ролей (см. правила игры «Мафия. Большой город»);
* добавление игрового режима.

Список использованных источников

1. Роберт А. Максимчук, Эрик Дж. Нейбург UML для простых смертных. Лори, 2016. 304 с.
2. Шлее М. Qt 5.10. Профессиональное программирование на С++. СПб.: БХВ-Петербург, 2018. 1072 с.
3. Справочная документация по Qt 4.8 // СrossPlatform.ru URL: <http://doc.crossplatform.ru/qt/4.8.x/html-qt/>
4. Qt 5.15 Documentation // Qt | Cross-platform software development for embedded & desktop URL: <https://doc.qt.io/qt-5/index.html>
5. Филипп Кан, Всеволод Чернов Мафия: Большой город. Правила. GaGa Games, 2016.

Приложение А. Снимки экранных форм   
пользовательского интерфейса

Главная форма:  


Форма модели:  


Приложение Б. Исходный код

Исходный код программы на язык Qt C++ находится в прилагаемом к отчету архиве в каталоге qt.