|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент группы ИУ5-13М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Бодунов А.Г.

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Виноградова М.В.

*2020 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы Бодунов А.Г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КП (учебный, исследовательский, практический, производственный, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра)

График выполнения проекта: 25% к 4 нед., 50% к 8 нед., 75% к 12 нед., 100% к 16 нед.

***Задание\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсового проекта:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «08» февраля 2020 г.

**Руководитель курсового проекта**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Виноградова М.В.

(Подпись, дата)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Бодунов А.Г.

(Подпись, дата)

**Задание на выполнение:**

1. **Этап анализа и планирования требований (Начало) – срок 4 неделя:**

* Определить область применения СОИУ (предназначение, границы, интерфейсы с внешней средой, критерии сдачи-приемки).
* Определить прецеденты, критические для системы (основные функции и главные решения).
* Определить основные элементы архитектуры (для выполнения основного сценария).
* Определить и оценить самые опасные риски (угрожающие успеху разработки), предложить способы управления ими.
* Оценить затраты, длительность и стоимость разработки по модели СОСОМО-2 этапа композиции приложения.

**В пояснительную записку (рабочие материалы на 4 неделе):**

* Постановка задачи (задание по варианту).
* Спецификация основных проектных требований, ключевых характеристик и главных ограничений (перечень требований-кандидатов, функциональных и нефункциональных требований).
* Модель предметной области (диаграмма классов предметной области, глоссарий понятий, бизнес-модели).
* Выявленные актеры.
* Выявленные прецеденты, их приоритеты и описание (кратко).
* Диаграмма основных прецедентов.
* Описание возможной архитектуры (перечень архитектурно-значимых прецедентов, обобщенные механизмы проектирования, системное ПО, используемые компоненты и/или каркасы). По каждой из выбранных технологий (обобщенные механизмы проектирования, системное ПО и т.д.) необходимо привести варианты, их достоинства и недостатки и обосновать выбор. Диаграммы уровней подсистем и развертывания.
* Перечень критических рисков (и рекомендации по управлению).
* Перечень экранных форм и их сложность; экспертные оценки скорости разработки и масштабных факторов; затраты, длительность и стоимость разработки (по модели СОСОМО-2 этапа композиции приложения).

**Содержание**

* Постановка задачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Спецификация основных проектных требований\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Модель предметной области\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Выявленные актеры\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Выявленные прецеденты, их приоритеты и описание\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Диаграмма основных прецедентов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Описание возможной архитектуры\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Диаграммы уровней подсистем и развертывания\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Перечень критических рисков и рекомендации по управлению\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Перечень экранных форм и их сложность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Экспертные оценки скорости разработки и масштабных факторов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Затраты, длительность и стоимость разработки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Основная часть**

* Постановка задачи (задание по варианту).

**Этапы разработки системы**

* Спроектировать архитектуру системы
* Написать программный код системы
* Написать подробную документацию с объяснениями работы каждого компонента системы
* Создать работоспособную систему для использования другими пользователями
* Поддерживать систему (исправлять ошибки, добавлять функции)
* Спецификация основных проектных требований, ключевых характеристик и главных ограничений (перечень требований-кандидатов, функциональных и нефункциональных требований).

**перечень требований-кандидатов**

Требования-кандидаты – требования, получившиеся в итоге процесса выяснения требований. К высокоуровневым требованиям-кандидатам можно отнести:

1. Система создает 3D модели через GUI на основе параметров, введенных пользователем
2. Система “на ходу” загружает произвольные нейронные сети, обучает их и тестирует
3. Система отслеживает внутренние параметры работающих приложений в режиме реального времени
4. Система позволяет создавать экспертные системы через GUI по шаблону
5. Система позволяет собирать приложения из готовых компонентов в режиме конвейера

Очевидно, что реализация всех вышеперечисленных требований невозможна за сколько-нибудь приемлемое время. Поэтому, после анализа и отсева требований-кандидатов остались следующие требования:

**перечень функциональных требований (по разделам интерфейса)**

1. Раздел “Вход”
   1. Регистрация пользователя
   2. Авторизация пользователя
   3. Копирование серверной БД в локальную БД
   4. Переключение системы на работу с локальной БД
   5. Переключение системы на работу с серверной БД
   6. Просмотр сведений о системе
2. Раздел “Пользователи”
   1. Создать гильдию (группу пользователей)
   2. Вступить в гильдию
   3. Пригласить человека в гильдию
   4. Найти гильдию
   5. Написать в чат гильдии

2.6 Загрузить обучающее видео

2.7 Отобразить список всех обучающих видео

2.8 Воспроизвести обучающее видео

1. Раздел “Оболочки Python”
   1. Создать виртуальное окружение Python
   2. Выбрать виртуальное окружение Python
   3. Протестировать jupyter notebook для выбранного виртуального окружения
   4. Ввести название пакета
   5. Установить пакет в виртуальное окружение (через pip)
   6. Установить пакет во все виртуальные окружения (через pip)
   7. Установить загруженный пакет в виртуальное окружение
2. Раздел “Алгоритмы” (скрипты)
   1. Загрузка входного скрипта из файла в БД
   2. Загрузка выходного скрипта из файла в БД
   3. Загрузка полного скрипта из файла в БД
   4. Выбор типа скрипта
   5. Сохранение скрипта из поля ввода в БД
   6. Загрузка скрипта из архива скриптов, загруженных другими пользователями
   7. Загрузка шаблона скрипта
   8. Выбор скрипта
   9. Подбор виртуальной среды для выбранного скрипта
   10. Функциональное тестирование для выбранного скрипта
   11. Тестирование производительности для выбранного скрипта
   12. Стресс-тестирование для выбранного скрипта
   13. Запуск выбранного скрипта
3. Раздел “Нейронные сети”
   1. Загрузка нейросети входного слоя из файла в БД
   2. Загрузка нейросети выходного слоя из файла в БД
   3. Загрузка полной нейронной сети из файла в БД
   4. Загрузка нейросети из архива нейросетей, загруженных другими пользователями
   5. Загрузка шаблона нейросети
   6. Выбор нейросети
   7. Подбор виртуального окружения для нейросети
   8. Тестирование нейросети в jupyter notebook
   9. Обучение нейросети
4. Раздел “Автоматы Маркова”
   1. Загрузка автомата Маркова из файла в БД
   2. Загрузка автомата Маркова из архива автоматов, загруженных другими пользователями
   3. Загрузка шаблона автомата Маркова
   4. Выбор автомата Маркова
   5. Редактирование выбранного автомата Маркова
   6. Создание пустого автомата Маркова
   7. Сохранение автомата Маркова без перезаписи
   8. Сохранение автомата Маркова под новым именем
5. Раздел “Сборка по шаблону”

7.1 Запуск нужного шаблона сборки (из подсистемы “Шаблоны сборки”)

1. Раздел “Сборка проекта”
   1. Привязка конкретного автомата Маркова к входному скрипту
   2. Задание коэффициента важности скрипта
   3. Привязка конкретного автомата Маркова к выходному скрипту
   4. Обнуление связей (всех привязок)
   5. Сборка проекта
2. Раздел “3D модели”
   1. Загрузить набор бесплатных 3D моделей (из Unity Assets Store)
   2. Загрузить набор бесплатных MoCap анимаций (из Unity Assets Store)
   3. Загрузить набор .dll для Unity 3D
   4. Загрузить пример контроллера для 3D персонажа
   5. Загрузить пример компонента “Animator” для 3D персонажа
3. Раздел “Мои проекты”
   1. Выбор проекта из списка проектов пользователя
   2. Запуск демонстрационного проекта
   3. Запуск выбранного проекта
   4. Загрузить видео с демонстрацией работы проектов (свои и других пользователей)
   5. Отобразить всех видео с демонстрацией работы проектов (свои и других пользователей)
   6. Воспроизвести демонстрационное видео

**перечень нефункциональных требований**

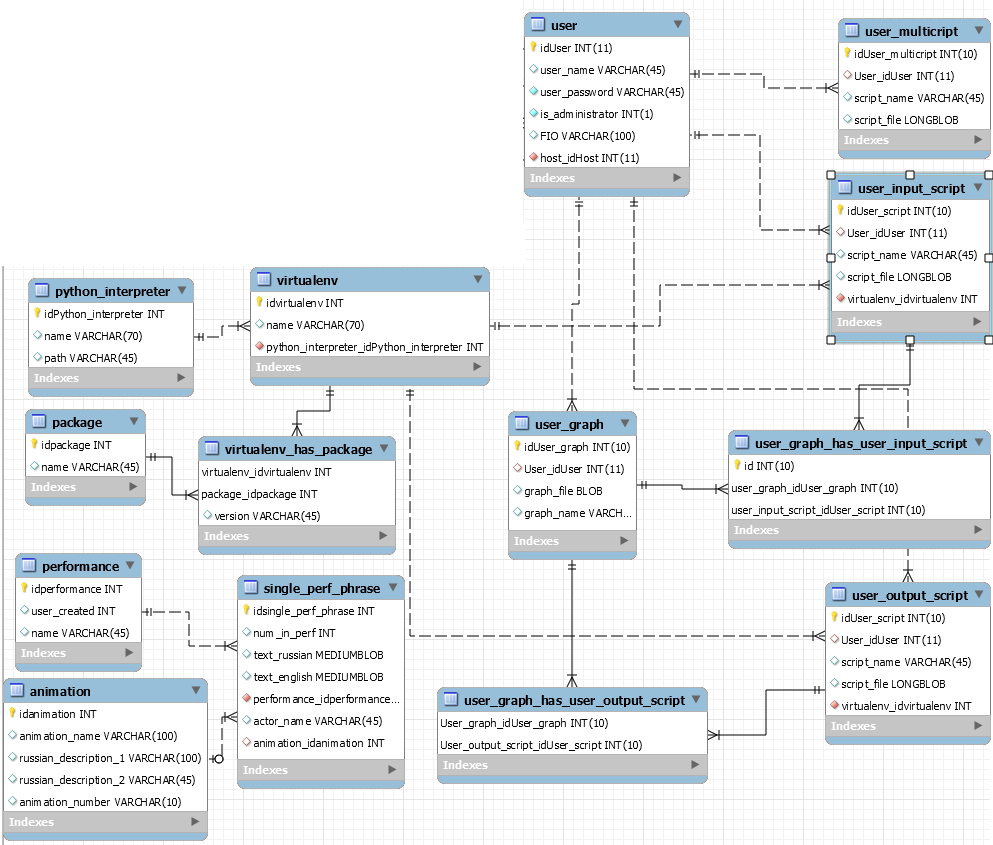
Критичные ограничения:

* На всех компьютерах пользователей должен быть установлен Unity3D
* На всех компьютерах пользователей должны быть установлена библиотека SignalR для Unity (можно скачать из программы)
* Пользователям нужно использовать Windows версии 10.
* Пользователю необходим выход в интернет (можно непостоянный)
* Пользователю необходимо иметь установленный интерпретатор python 3 версии и выше
* При запуске системы пользователю доступна только вкладка “Вход”, после входа в систему – все вкладки системы

Некритичные ограничения

* На всех компьютерах пользователей должно быть установлено расширение для Unity 3D – Vuforia Engine версии 8.6
* На всех компьютерах пользователей должно быть установлено расширение для Unity 3D – Windows 10 SDK (10.0.18362.0) версии не ниже 1903
* На всех компьютерах пользователей должен присутствовать набор MoCap анимаций версии не ниже 1.3 (бесплатный набор из 2500 анимаций в Unity Assets Store)
* На всех компьютерах пользователей должно присутствовать ПО для синтеза голоса с консольным вводом команд (в комплект поставки программы входит ESpeak версии 1.47).
* Пользователь должен отдельную систему сбора данных (например, на базе Raspberry Pi 3) с подключенном микрофоном (для записи звука без задержек и прерываний, поскольку приложение сильно нагружает пользовательский ПК)
* На всех компьютерах пользователей должно быть установлена программа GraphViz версии не ниже 2.38 с добавлением пути к исполняемому файлу в системную переменную PATH (есть в комплекте поставки программы)
* Модель предметной области (диаграмма классов предметной области, глоссарий понятий, бизнес-модели).

Диаграмма классов предметной области:



**Глоссарий понятий**

**Автомат Маркова -** (НАМ, также марковский алгоритм) — один из стандартных способов формального определения понятия алгоритма

**SignalR Hub -** это логическая конструкция, своего рода точка, через которую приложение получает данные от пользователей и может обращаться к подключенным клиентам как скрипт узнает - по названию хаба - $.connection.chatHub и по названию функций, которые аналогичны методам хаба.

**TCP Socket -** название программного интерфейса для обеспечения обмена данными между процессами.

**Виртуальная среда python -** механизм, позволяющий разработчику содержать разные проекты в изолированных "песочницах".

**бизнес-модели**

Бизнес-модель является типичной для программных продуктов 2018-2020 гг.

Разрабатываемый программный продукт распространяется бесплатно. Получение прибыли планируется извлекать из вставляемой рекламы в обучающие ролики на Youtube.

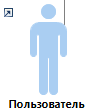
* Выявленные актеры.

Рассмотрим варианты подходов к разработке системы:



|  |  |  |  |
| --- | --- | --- | --- |
| Подсистема/актеры | Кто разрабатывает | Кто использует | Кто поддерживает |
| GUI пользователя(IDE) | Разработчик | Пользователи | Разработчик |
| Подсистема шаблоны | Разработчик и пользователи | Пользователи | Пользователи |
| Ядро собранных приложений | Разработчик | Пользователи | Разработчик |
| Собранные приложения | Пользователь | Пользователь | никто |
| 3D редактор | Unity Technologies | Пользователи | Unity Technologies |

Поскольку был выбран подход к разработке системы, предполагающий децентрализованное управление системой и децентрализованную поддержку (пользователи системы одновременно являются разработчиками шаблонов и приложений и диспетчерами системы), имеет смысл ввести только одного актера - пользователя



Выявленные прецеденты, их приоритеты и описание (кратко).

Диаграмма основных прецедентов.

Диаграмма работы со скриптами

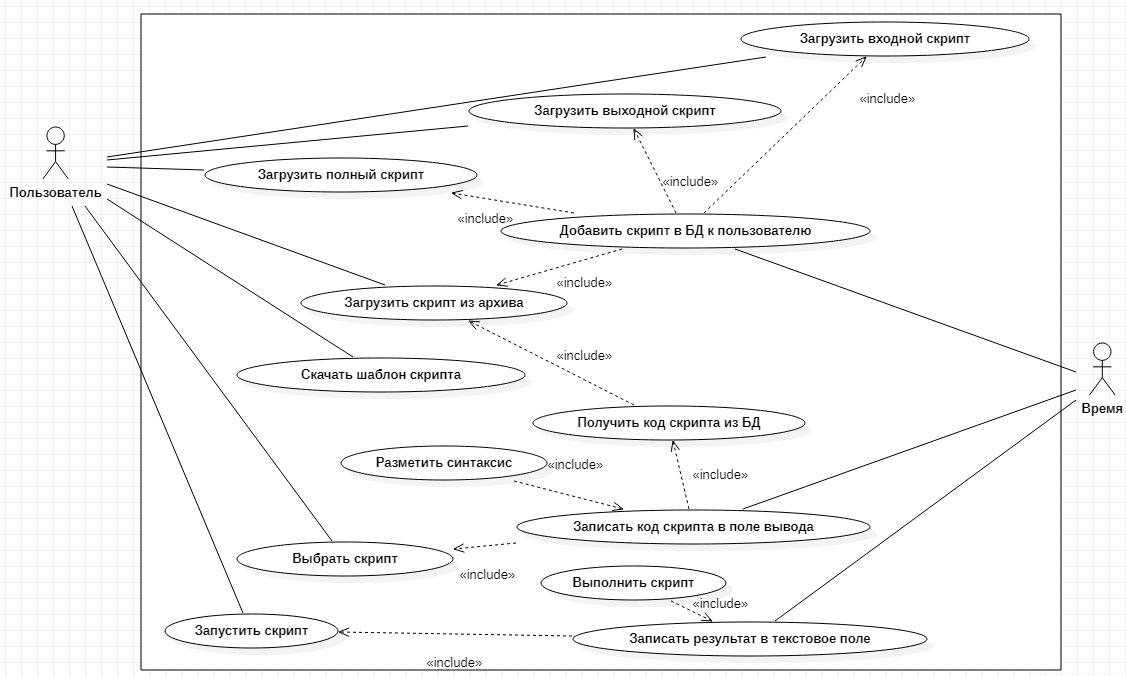


Диаграмма работы с нейронными сетями

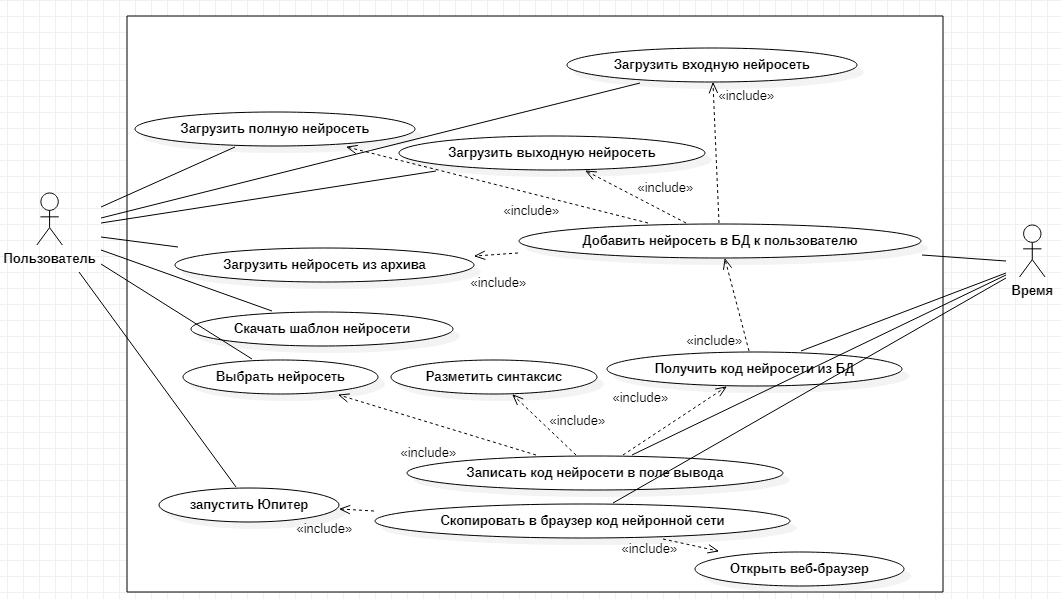


Диаграмма работы с конечными автоматами

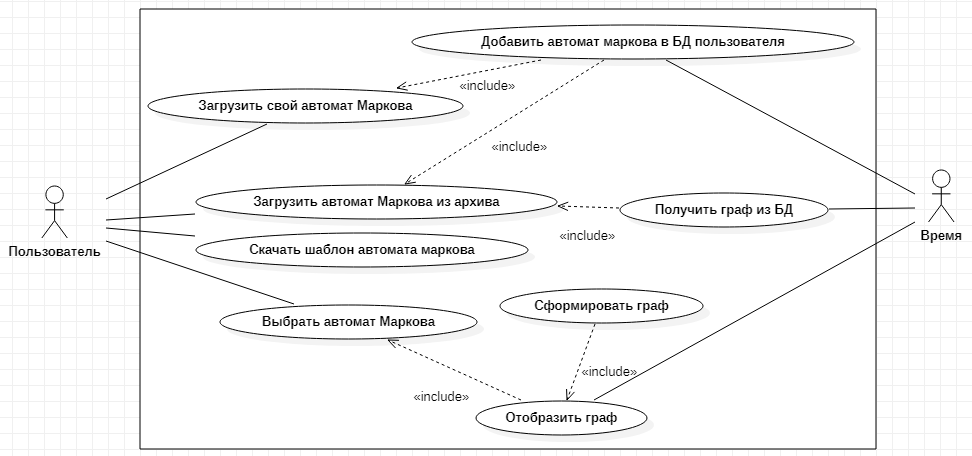
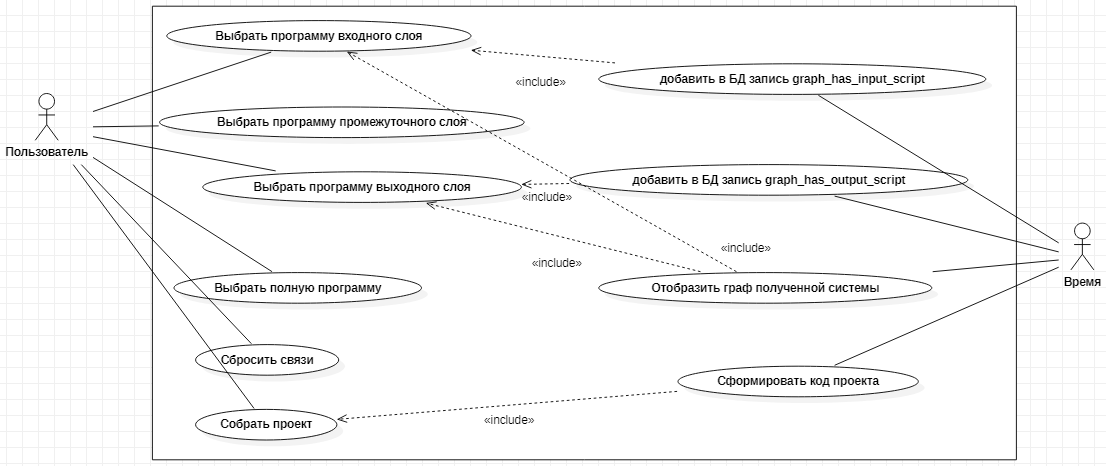


Диаграмма сборки пользовательских приложений



* Описание возможной архитектуры (перечень архитектурно-значимых прецедентов, обобщенные механизмы проектирования, системное ПО, используемые компоненты и/или каркасы). По каждой из выбранных технологий (обобщенные механизмы проектирования, системное ПО и т.д.) необходимо привести варианты, их достоинства и недостатки и обосновать выбор. Диаграммы уровней подсистем и развертывания.

**Перечень архитектурно-значимых прецедентов**

**Набор основных прецедентов, реализующих действия пользователя**

1. Раздел “Вход”
   1. Регистрация пользователя
   2. Авторизация пользователя
2. Раздел “Пользователи”
3. Раздел “Оболочки Python”
   1. Создать виртуальное окружение Python
   2. Установить пакет в виртуальное окружение (через pip)
   3. Установить пакет во все виртуальные окружения (через pip)
4. Раздел “Алгоритмы” (скрипты)
   1. Выбор типа скрипта
   2. Сохранение скрипта из поля ввода в БД
   3. Загрузка скрипта из архива скриптов, загруженных другими пользователями
   4. Выбор скрипта
   5. Подбор виртуальной среды для выбранного скрипта
5. Раздел “Нейронные сети”
   1. Загрузка нейросети входного слоя из файла в БД
   2. Загрузка нейросети выходного слоя из файла в БД
   3. Загрузка полной нейронной сети из файла в БД
   4. Загрузка нейросети из архива нейросетей, загруженных другими пользователями
   5. Подбор виртуального окружения для нейросети
   6. Тестирование нейросети в jupyter notebook
6. Раздел “Автоматы Маркова”
   1. Загрузка автомата Маркова из файла в БД
   2. Загрузка автомата Маркова из архива автоматов, загруженных другими пользователями
   3. Сохранение автомата Маркова под новым именем
7. Раздел “Сборка по шаблону”

7.1 Запуск нужного шаблона сборки (из подсистемы “Шаблоны сборки”)

1. Раздел “Сборка проекта”
   1. Привязка конкретного автомата Маркова к входному скрипту
   2. Привязка конкретного автомата Маркова к выходному скрипту
   3. Сборка проекта
2. Раздел “3D модели”
   1. Загрузить набор .dll для Unity 3D
3. Раздел “Мои проекты”
   1. Запуск демонстрационного проекта
   2. Запуск выбранного проекта

**Набор основных прецедентов, реализующих действия системы**

1. Для языка программирования
   1. Одновременное использование одной и той же библиотеки разных версий
2. Для системы
   1. Запуск приложения пользователя должен происходить отдельно от запуска Unity 3D
   2. Логика приложений пользователя должна быть сосредоточена в одном месте
   3. Система должна использовать БД как при наличии подключения к интернету, так и при его отсутствии
   4. В системе должна быть предусмотрена поддержка многопользовательского режима работы (для многопользовательских приложений)
3. Для Unity 3D

3.1 Должна быть предусмотрена возможность приема пользовательских команд для каждого персонажа на сцене

**Необходимые компетенции пользователя для взаимодействия с системой**

Для эксплуатации системы пользователю необходимо

* Владеть общими навыками программирования
* Владеть языком программирования Python
* Иметь опыт в практическом применении алгоритмов и структур данных
* Понимать устройство конечных автоматов
* Уметь загружать ассеты – готовые 3D модели с анимацией

При этом пользователь системы может не понимать:

* Принципы сборки веб-сервисов
* Принципы работы АСОИУ
* Принципы отладки сложных систем
* Принципы работы клиент-серверной архитектуры
* Тонкости работы с Unity 3D на уровне профессионального разработчика

**Выбранная архитектура системы**

С развитием технологий дополненной реальности и нейронных сетей встает вопрос о создании систем иного порядка – взаимодействующих с реальным миром, а не только с устройствами ввода пользователя. При этом, система Unity 3D используется уже не как ядро системы, а отходит на периферию, или на уровень представления, теряя функцию контроллера.

Логичным решением была бы система с архитектурой, показанной на рисунке ниже.



рис. Целевая схема взаимодействия контроллера и Unity

Систему с такой архитектурой разработать невозможно, исходя из ограничений Unity API. Нельзя обратиться к объектам 3D сцены извне Unity3D.

Единственным “мостом” в Unity 3D является поддержка TCP сокетов, которые предназначены для обмена сообщениями между клиентами в многопользовательских играх, как показано на рисунке.

 Рис архитектура разрабатываемой системы



Разрабатываемая система состоит из 2 контуров:

1. IDE для сборки проекта из python скриптов (C# Windows Forms Application)
2. Созданного приложения, которое взаимодействует с приложением Unity через Web сокеты (посредством стандартной библиотеки SignalR)

В качестве языка создания пользовательского интерфейса был выбран язык C#, поскольку:

* Ориентация на классы позволяет реализовать трехзвенную архитектуру MVC (Model-View-Controller)
* C# позволяет разрабатывать Desktop – приложения с помощью простого и понятного интерфейса в IDE – Visual Studio с поддержкой визуального конструктора форм.
* В отличие от Qt и PyQt, интерфейсы на C# Windows Forms обладают большей стабильностью

В качестве языка задания игровой логики приложения был выбран Python, поскольку:

* Программы на языке Python компактны
* Python де-факто является стандартом в системах машинного обучения
* Python поддерживается большим числом специализированных библиотек
* Python имеет динамическую типизацию
* Python является скриптовым языком, не требующим предварительной компиляции
* Python имеет развитое сообщество

В качестве программы визуализации 3D сцены была выбрана Unity3D по следующим причинам:

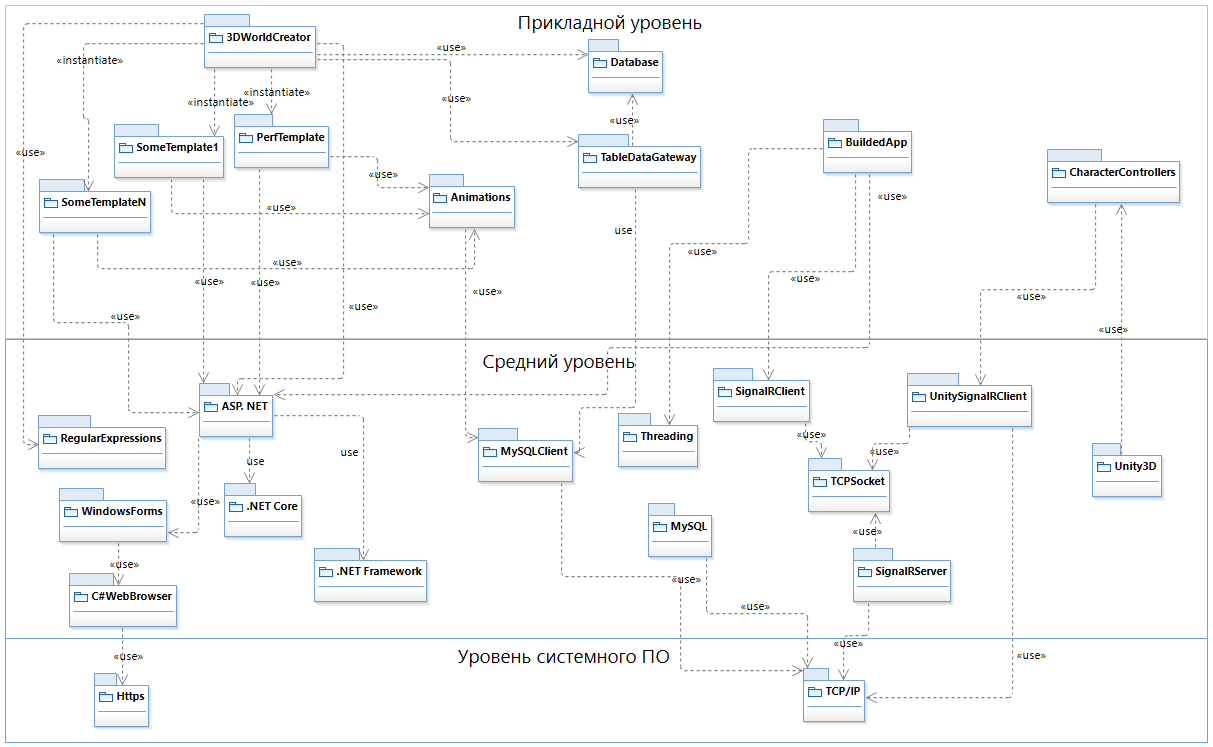
* Unity 3D узко направлена на визуализацию 3D сцены (в отличие от таких программ, как 3DsMax, предназначенной для создания 3D моделей)
* Unity 3D имеет магазин ассетов, в котором можно загрузить уже готовые и анимированные 3D модели
* Unity 3D имеет расширение Vuforia, предназначенное для работы с очками дополненной реальности (Microsoft HoloLens)

В качестве библиотеки взаимодействия контроллера и Unity 3D была выбрана библиотека SignalR, поскольку:

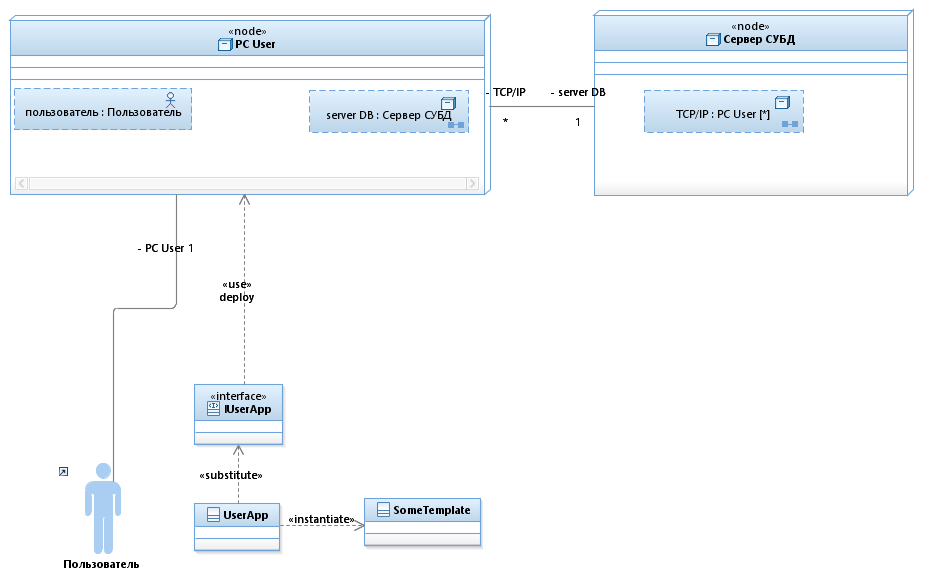
* SignalR – бесплатная библиотека
* SignalR использует TCP сокеты, но позволяет абстрагироваться от них на уровне API
* SignalR позволяет распространять сообщения от одного клиента всем остальным клиентам (благодаря наличию класса Hub – концентратора)
* SignalR позволяет организовывать как локальные, так и распределенные сетевые приложения (приложение для одного пользователя можно легко сделать многопользоватеским, изменив только часть программы с SignalR)
* SignalR поддерживает вызов удаленный процедур – RPC.
* SignalR поддерживается компанией Microsoft и имеет поддержку в Unity 3D

Итак, имеется система, включающая в себя несколько технологий: нейронные сети, системы принятия решений, системы передачи информации по сети и 3D графику.

**Диаграмма уровня подсистем**

****

**Диаграмма развертывания**



* Перечень критических рисков (и рекомендации по управлению).

###### Возьмем типичные риски из [1], общие для всех проектов разработки программного обеспечения

1. Внутренние изъяны календарного планирования – вследствие малого опыта, исполнитель устанавливает чересчур оптимистичные сроки.

Рекомендация по управлению – постоянно отслеживать сроки завершения этапов работ и корректировать, в связи с этим, сроки завершения следующих этапов

1. Раздувание требований (изменение требований) – появление в процессе разработки новых требований, время на реализацию которых не было учтено в первоначальном графике.

Рекомендация по управлению – поскольку мы не можем увеличивать сроки разработки(срок сдачи данной работы определен преподавателем), нам придется периодически просматривать список требований и выполнять их приоретизацию (выкидывать уже имеющиеся низкоприоритетные требования при необходимости добавления новых, высокоприоритетных)

1. Текучесть кадров – в нашем случае не является риском, поскольку исполнителем является единственный человек, и выдача работы на out-source не планируется.
2. Нарушение спецификаций – отклонение свойств программного продукта от указанных в спецификациях. Требует значительного времени на “доводку” продукта. Рекомендация по управлению – раннее тестирование, позволяющее вовремя выявить появляющиеся отклонения от спецификаций.
3. Низкая производительность – в нашем случае означает необходимость периодически отвлекаться на выполнение учебного плана по другим предметам. Рекомендация по управлению – мы принимаем этот риск, поскольку никак не можем управлять объемом учебной программы курса.

* Перечень экранных форм и их сложность; экспертные оценки скорости разработки и масштабных факторов; затраты, длительность и стоимость разработки (по модели СОСОМО-2 этапа композиции приложения).

**Перечень экранных форм и их сложность**

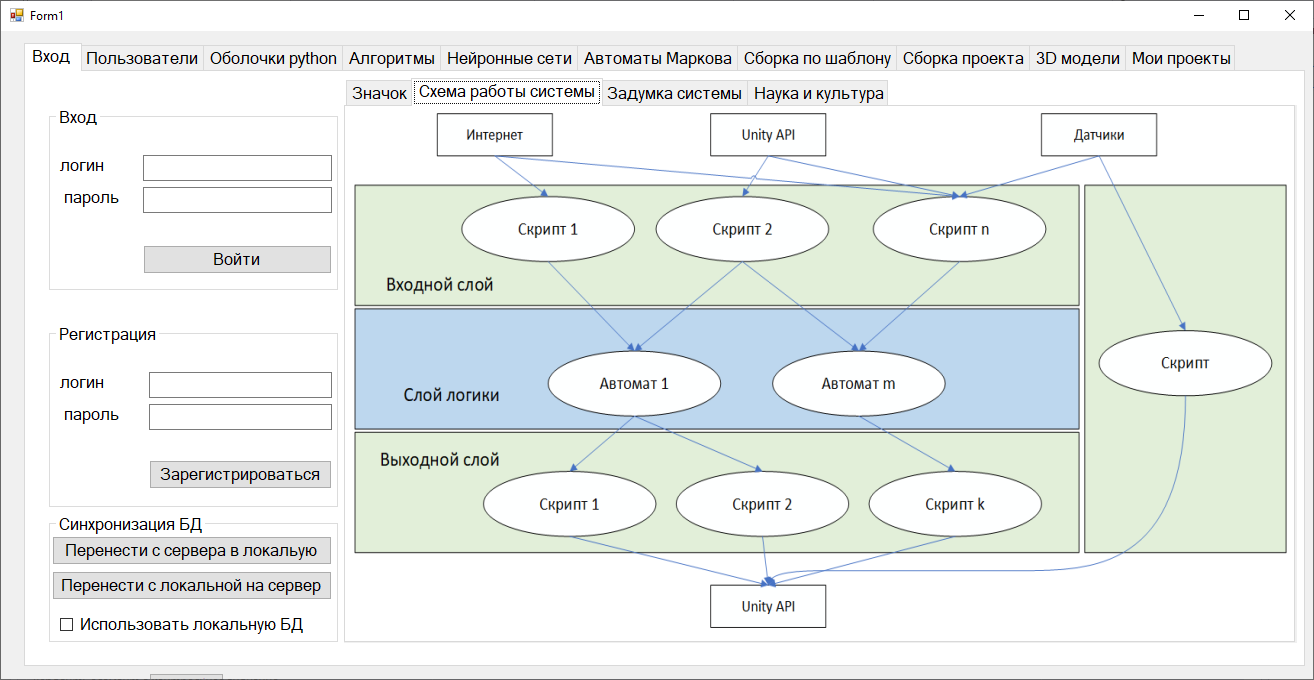
**Подсистема “GUI пользователя”**

Не существует единых и общепризнанных оценок сложности пользовательского интерфейса. Поэтому можно провести оценку, исходя из степени компетентности конечного пользователя системы в вопросах программирования на языке python (понимания алгоритмов, паттернов, степени знакомства с технологиями машинного обучения и понимания работы Unity 3D).

При этом можно выделить 3 основных категории пользователей:

1. Новичок – человек, имеющий базовые знания на уровне компьютерных курсов
2. Любитель – человек с неплохой компетенцией, работающий в данной предметной области
3. Профессионал – эксперт в программировании на Python, имеющий большой опыт разработки

Далее приведем скриншоты спроектированных экранных форм. (подписывать названия рисунков не будем, они очевидны из названия активной вкладки)

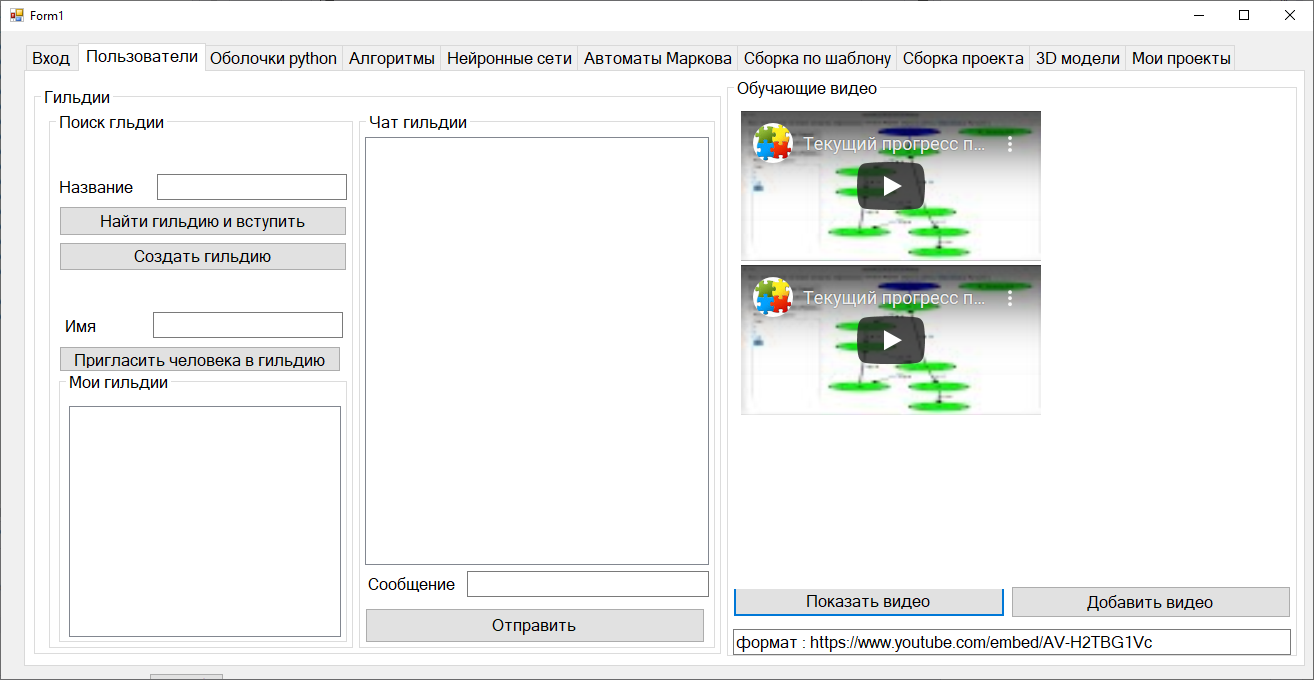


Сложность интерфейса:

Для новичков: 1

Для любителей: 1

Для профессионалов: 1

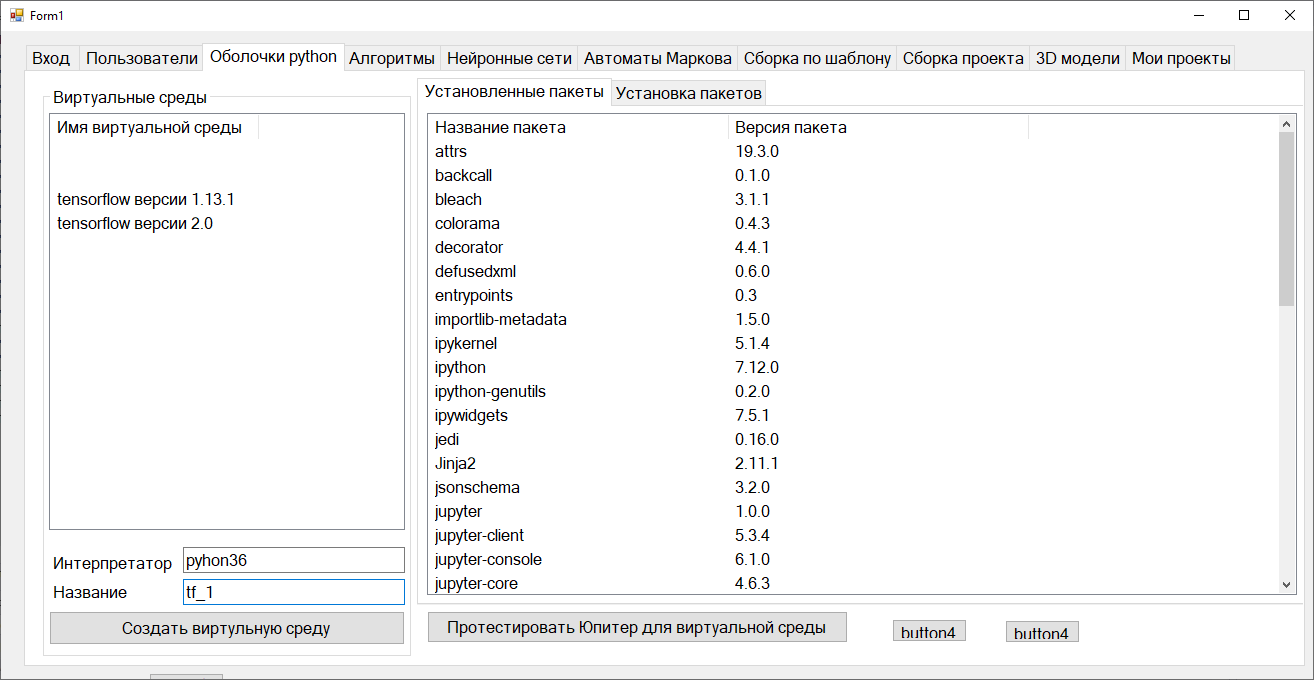


Сложность интерфейса:

Для новичков: 1

Для любителей: 1

Для профессионалов: 1

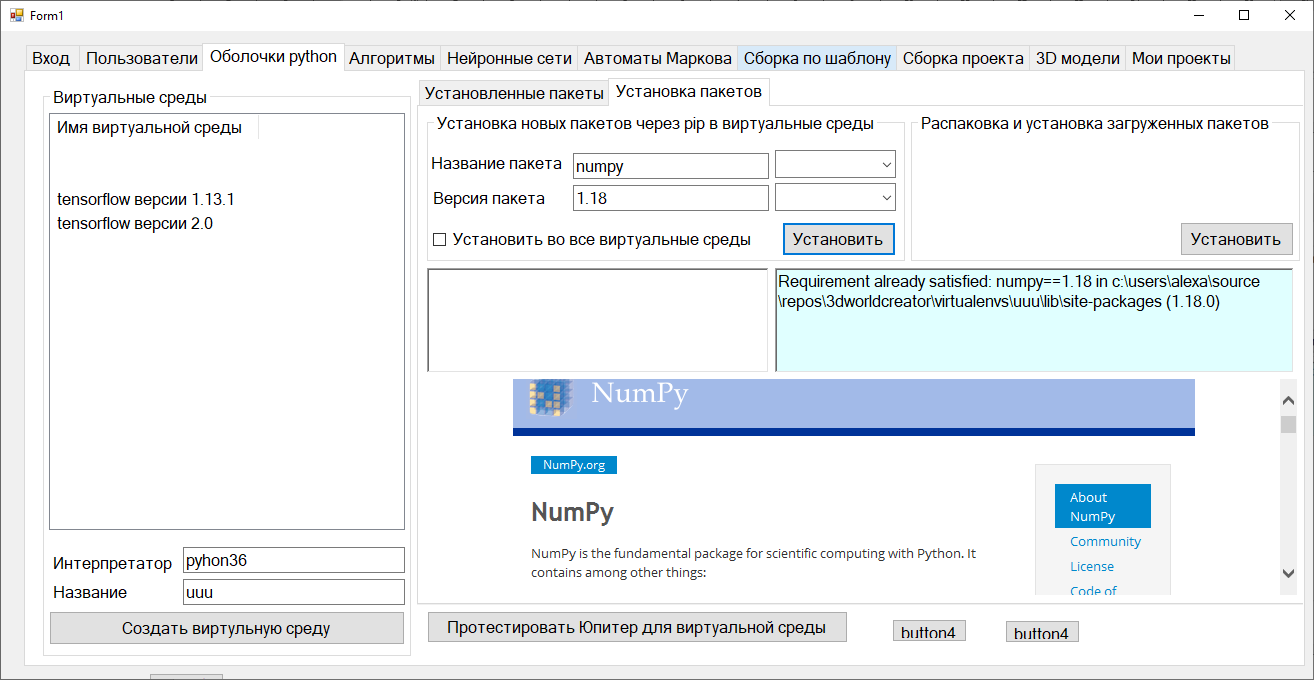


Сложность интерфейса:

Для новичков: 4

Для любителей: 2

Для профессионалов: 1

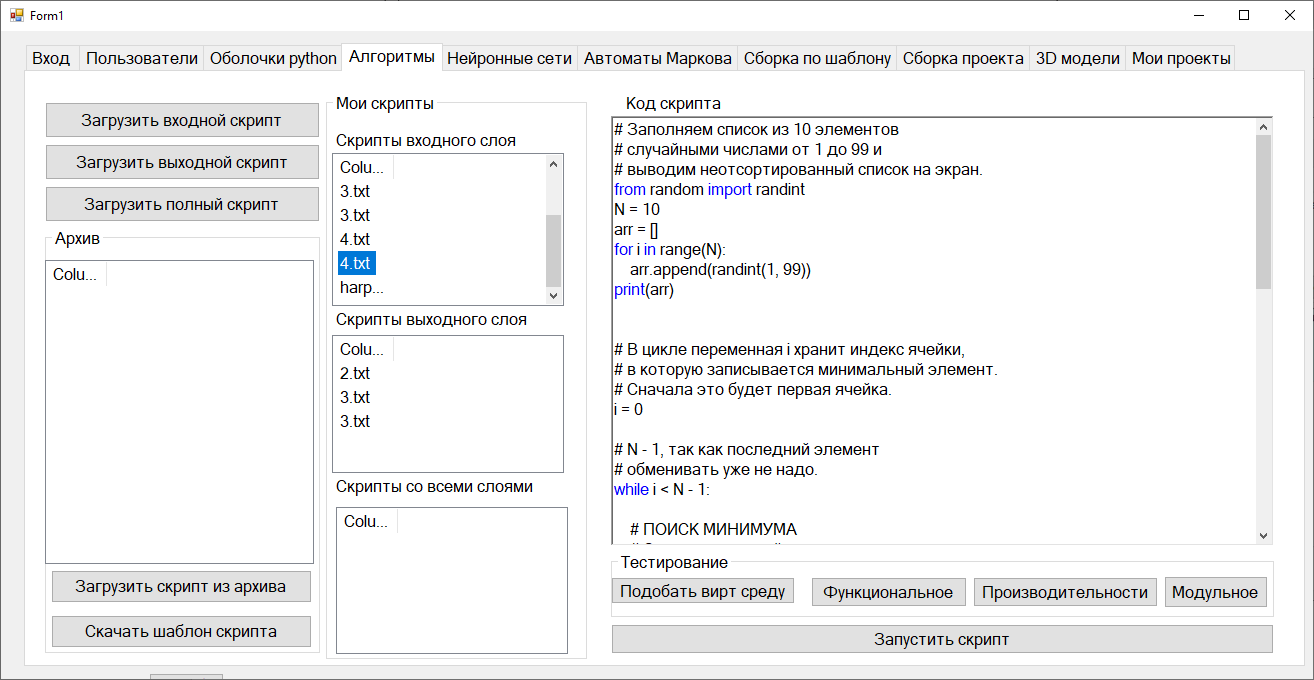


Сложность интерфейса:

Для новичков: 4

Для любителей: 2

Для профессионалов: 1

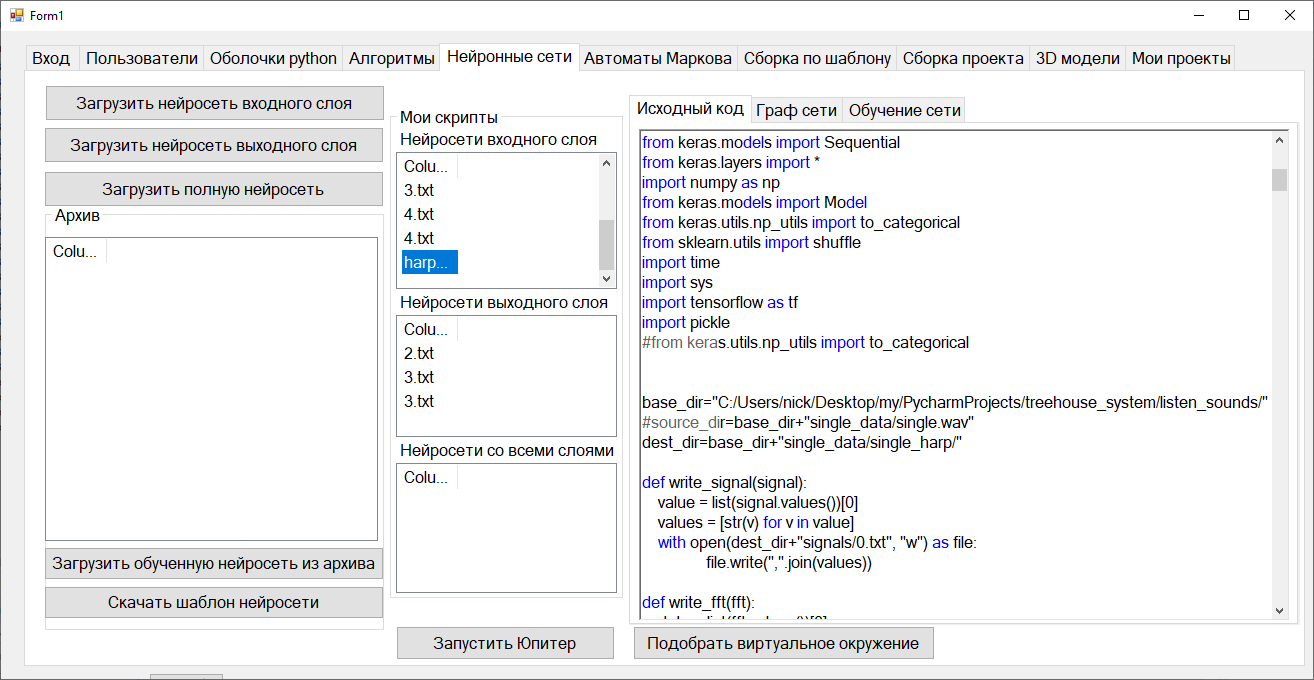


Сложность интерфейса:

Для новичков: 5

Для любителей: 3

Для профессионалов: 1

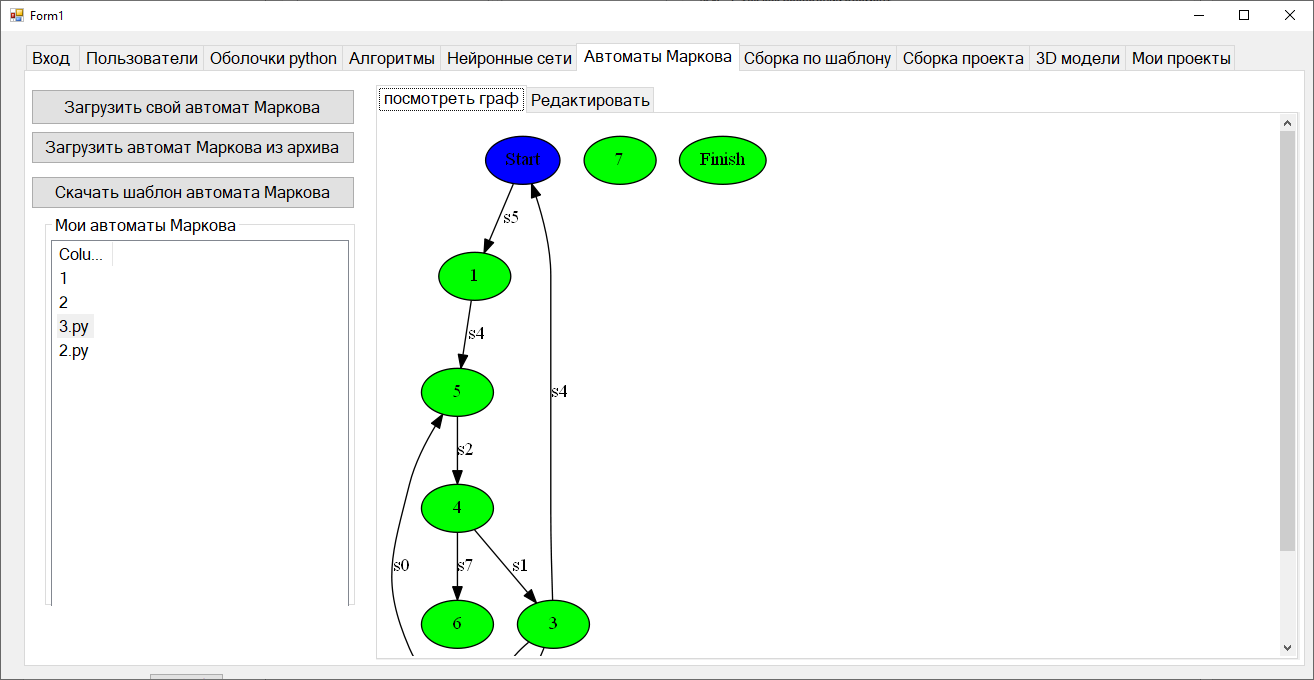


Сложность интерфейса:

Для новичков: 7

Для любителей: 5

Для профессионалов: 3

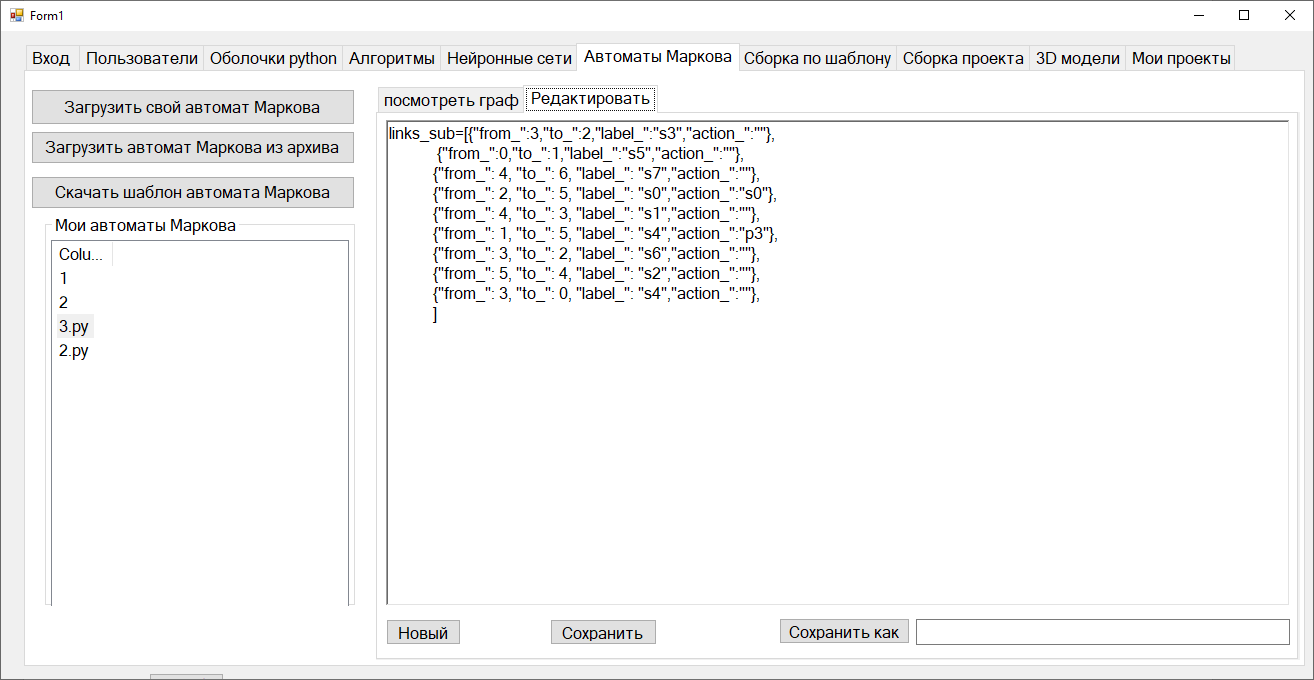


Сложность интерфейса:

Для новичков: 2

Для любителей: 1

Для профессионалов: 1

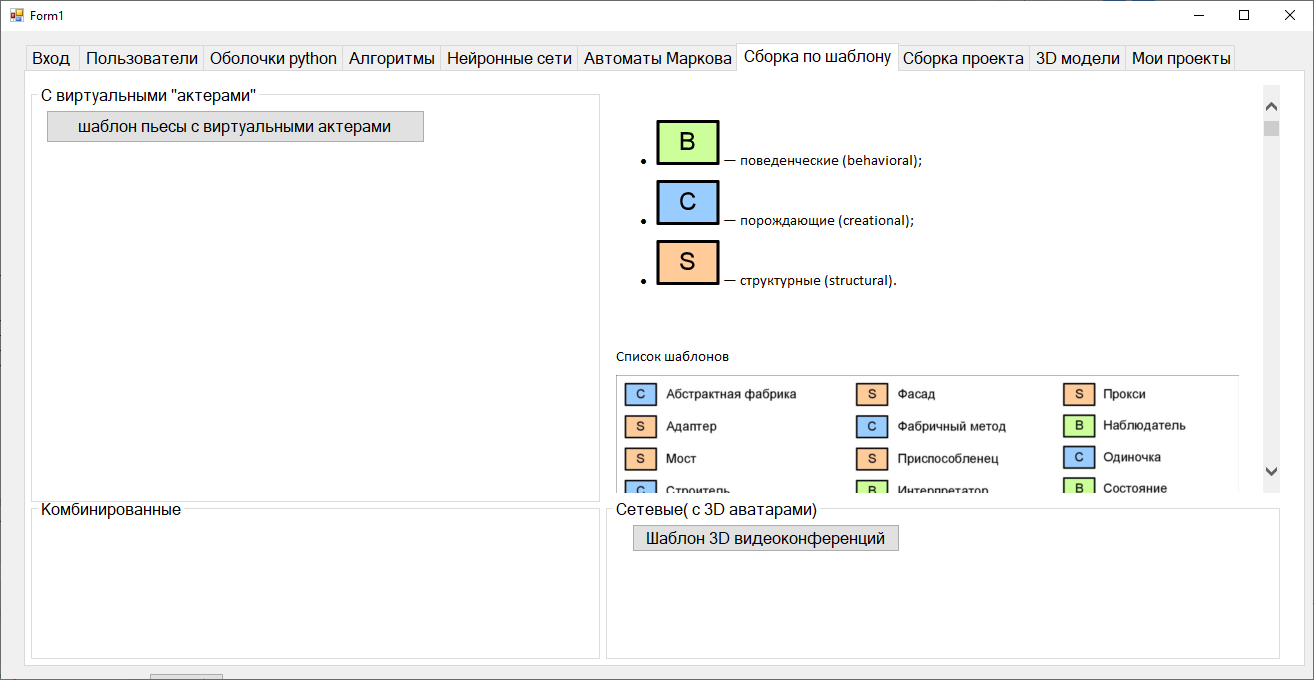


Сложность интерфейса:

Для новичков: 2

Для любителей: 1

Для профессионалов: 1

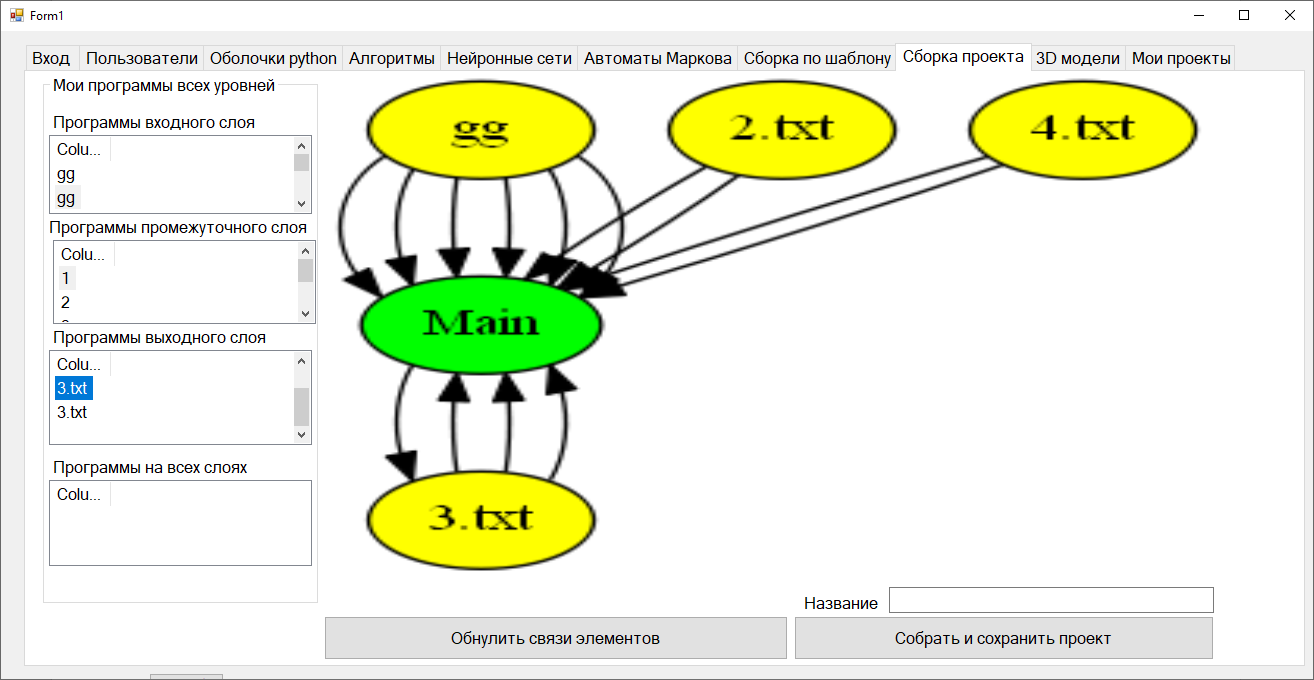


Сложность интерфейса:

Для новичков: 1

Для любителей: 1

Для профессионалов: 1

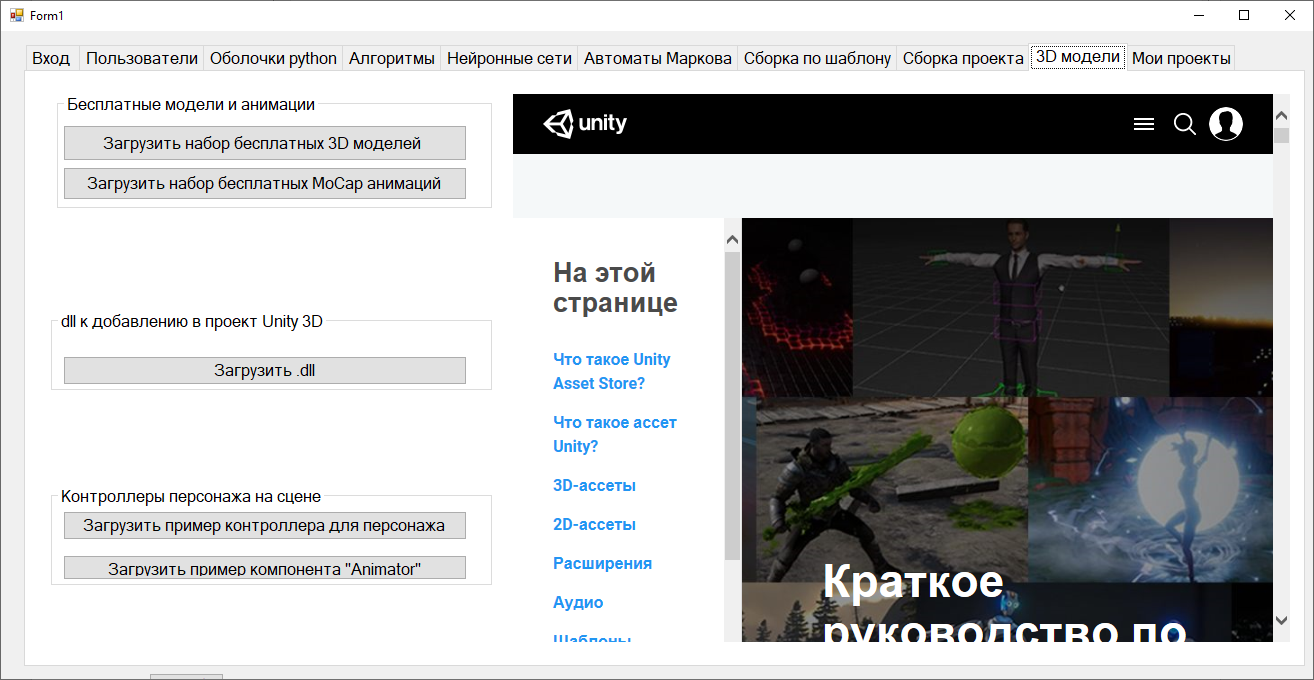


Сложность интерфейса:

Для новичков: 6 (неочевидный способ соединения элементов)

Для любителей: 1

Для профессионалов: 1

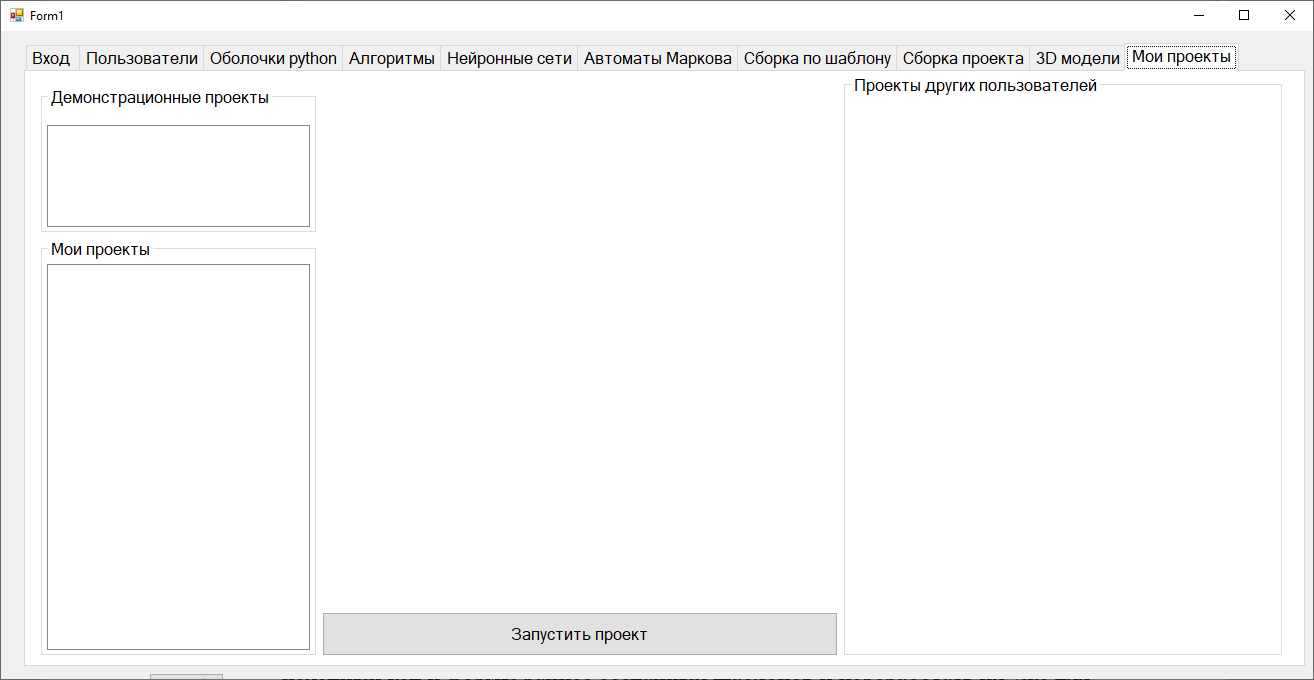


Сложность интерфейса:

Для новичков: 3

Для любителей: 1

Для профессионалов: 1



Сложность интерфейса:

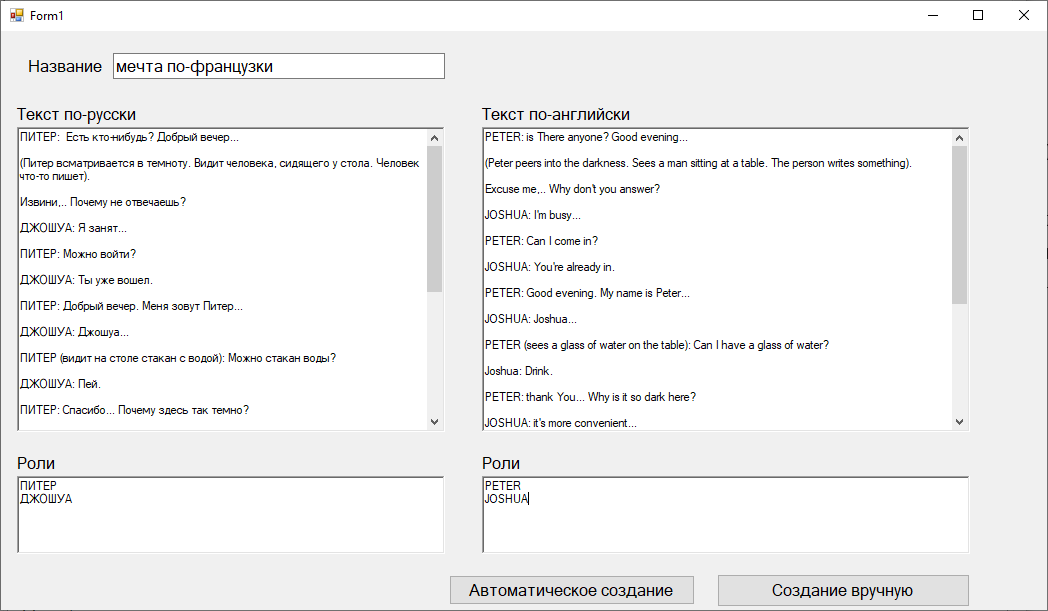
Для новичков: 1

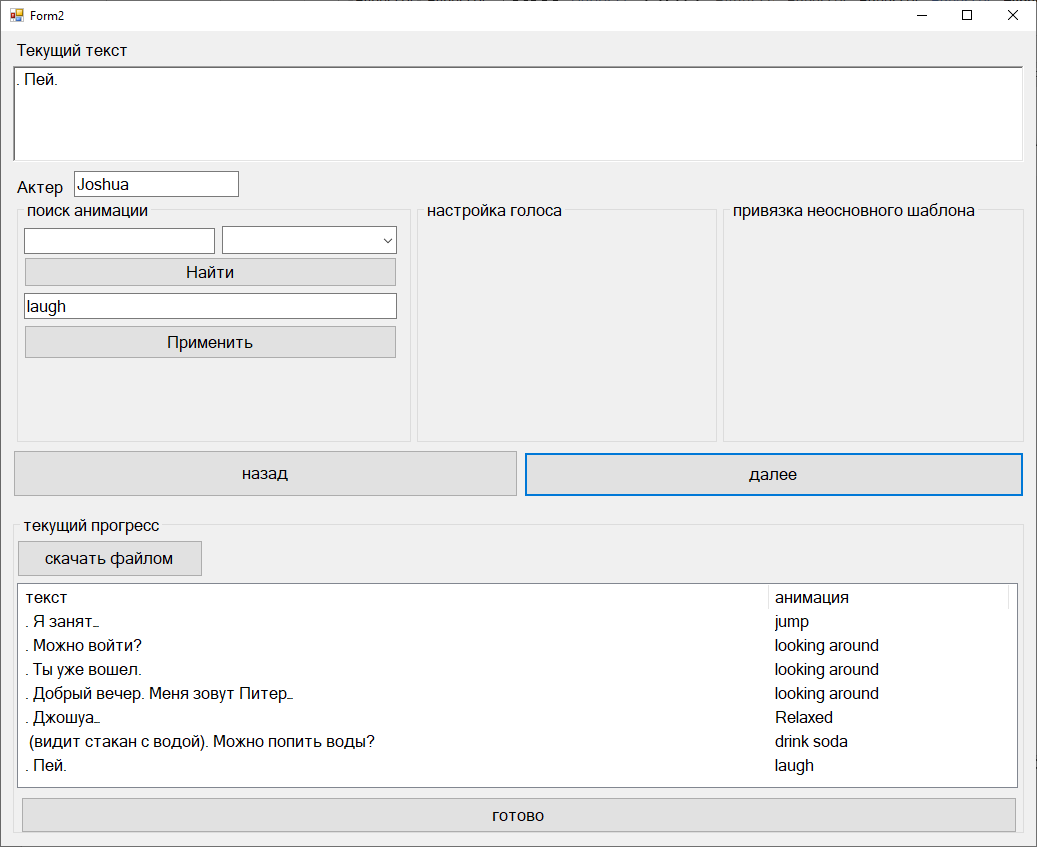
Для любителей: 1

Для профессионалов: 1

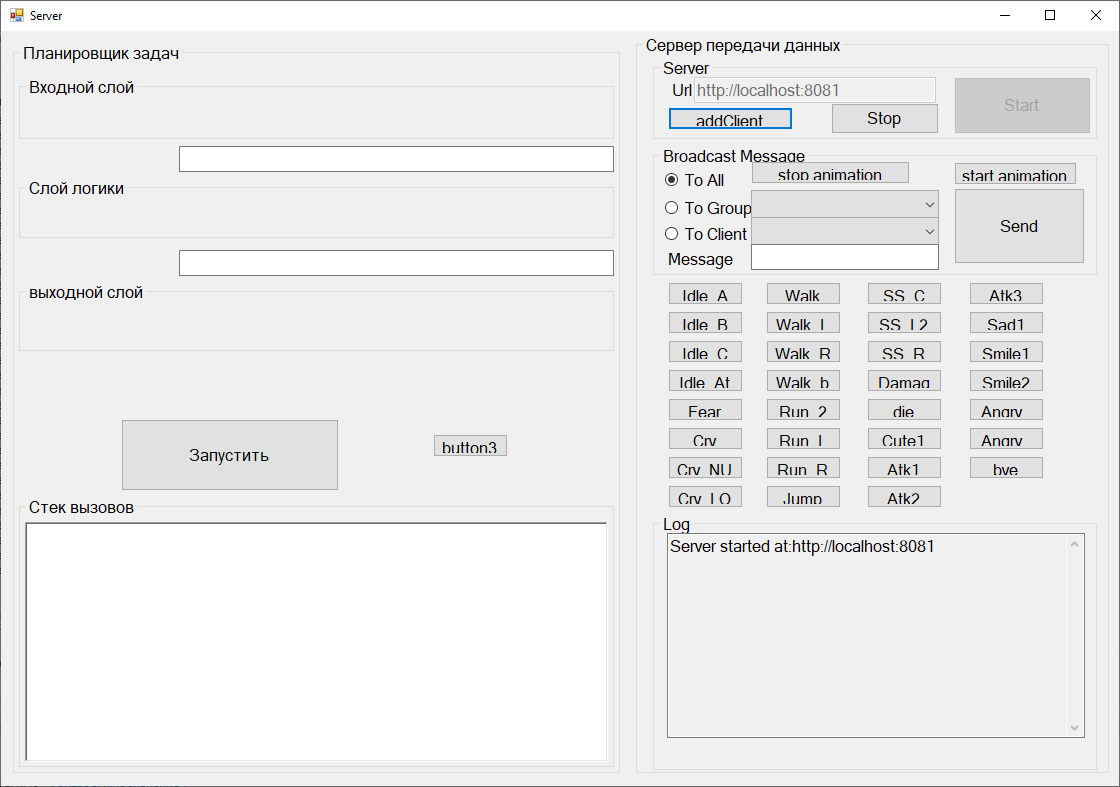
**Подсистема “Шаблоны сборки”**

Данная подсистема может включать в себя неограниченное число шаблонов. В соответствии со спирально-инкрементной моделью, предполагается, что при разработке разработчиками или пользователями новых шаблонов, можно взять исходный код и формы раннее созданных шаблонов и переработать их уже для решения новой задачи. В данную подсистему был добавлен первый шаблон – шаблон пьесы с виртуальными актерами.





**Подсистема “Пользовательское приложение”**



**Экспертные оценки скорости разработки и масштабных факторов; затраты, длительность и стоимость разработки (по модели СОСОМО-2 этапа композиции приложения).**

**2) Расчет функционального указателя FP**

External inputs, EI- внешние входные транзакции, элементарная операция по обработке данных или управляющей информации, поступающих в систему извне;

External outputs, EO (внешние выходные транзакции)- элементарная операция по генерации данных или управляющей информации, которые выходят за пределы системы. Предполагает определенную логику обработки или вычислений информации из одного или более ILF; EQ External inquiries, EQ ( внешние запросы) - элементарная операция, которая в ответ на внешний запрос извлекает данные или управляющую информацию из ILF или EIF

Internal logical files, ILF (внутренние логические файлы) -логические группы данных, которые могут быть частью базы данных или отдельным файлом.

External interface files, EIF (внешние интерфейсные файлы) - все логические файлы из других приложений, на которые ссылается данное приложение.

Форма окна входа пользователя (рис.1)

1. Количество внешних вводов = 3
2. Количество внешних выводов =1
3. Количество внешних запросов = 0
4. Количество внутренних логических файлов = 1
5. Количество внешних интерфейсных файлов = 0

Форма окна управления заданиями (рис.2)

1. Количество внешних вводов = 7
2. Количество внешних выводов = 1
3. Количество внешних запросов = 0
4. Количество внутренних логических файлов = 1
5. Количество внешних интерфейсных файлов = 0

Форма окна администрирования проектов (рис.3)

1. Количество внешних вводов = 10
2. Количество внешних выводов = 0
3. Количество внешних запросов = 0
4. Количество внутренних логических файлов = 1
5. Количество внешних интерфейсных файлов = 0

Форма окна администрирования пользователей (рис.4)

1. Количество внешних вводов = 6
2. Количество внешних выводов = 0
3. Количество внешних запросов = 0
4. Количество внутренних логических файлов = 1
5. Количество внешних интерфейсных файлов = 0

**2.Указываем количество элементов данных и рассчитываем коэффициенты сложности**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Элементы данных** | EI | EO | EQ | ILF | EIF |
| Форма окна входа пользователя | 3 | 1 | 1 | 1 | 0 |
| Форма окна управления заданиями | 7 | 1 | 1 | 1 | 0 |
| Форма окна администрирования проектов | 10 | 0 | 2 | 1 | 0 |
| Форма окна администрирования пользователей | 6 | 0 | 1 | 1 | 0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Оценка сложности** | EI | EO | EQ | ILF | EIF |
| Форма окна входа пользователя | 3 | 4 | 3 | 7 | 0 |
| Форма окна управления заданиями | 3 | 4 | 3 | 7 | 0 |
| Форма окна администрирования проектов | 3 | 0 | 4 | 3 | 0 |
| Форма окна администрирования пользователей | 3 | 0 | 3 | 7 | 0 |

Подсчитываем функциональный указатель (FP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Исходные данные для расчета FP - метрик** | Ранг, сложность, количество | | | |
| Имя характеристики | Низкий | Средний | Высокий | Итого |
| внешние вводы | 36 | 0 | 0 | 36 |
| внешние выводы | 32 | 0 | 0 | 32 |
| внешние запросы | 39 | 0 | 0 | 39 |
| внутренние логические файлы | 168 | 0 | 0 | 168 |
| внешние интерфейсные файлы | 0 | 0 | 0 | 0 |
| Общее количество |  |  |  | 275 |

Подсчитываем количество строк кода, исходя из предположения, что код пишется на Java.

LOC=FP\*53 = 13555

**4 Расчет затрат**

Подсчет затрат (в чел. - мес.)

ЗАТРАТЫ = А х РАЗМЕРB х Ме + ЗАТРАТЫаuto[чел.-мес],

Где А – масштабный коэффициент = 2,5

B=1,01+0,01 (зависимость затрат от размера проекта )

|  |  |
| --- | --- |
| **Масштабный фактор (Wi)** | 2 |
| Предсказуемость PREC | 3 |
| Гибкость разработки FLEX | 4 |
| Разрешение архитектуры /риска RESL | 2 |
| Связность группы TEAM | 4 |
| Зрелость процесса РМАТ | 2 |
| **Сумма** | 17 |
| **B** | 1,18 |

Me – множитель поправки, учитывающий влияние масштабных факторов, связанных с компанией - разработчиком



|  |  |  |
| --- | --- | --- |
| **Формирователи затрат для раннего этапа проектирования** |  |  |
| Возможности персонала (Personnel Capability) | PERS | 1 |
| Надежность и сложность продукта (Product Reliability and Complexity) | RCPX | 1 |
| Требуемое повторное использование (Required Reuse) | RUSE | 1 |
| Трудность платформы (Platform Difficulty) | PDIF | 1 |
| Опытность персонала (Personnel Experience) | PREX | 2 |
| Средства поддержки (Facilities) | FСIL | 2 |
| График (Schedule) | SCED | 1 |
|  | Me | 4 |

ЗАТРАТЫauto= (КALOC x (AT /100)) / ATPROD,

где

KALOC — количество строк автоматически генерируемого кода (в тысячах строк);

AT — процент автоматически генерируемого кода (от всего кода системы);

ATPROD — производительность автоматической генерации кода.

ЗАТРАТЫauto= (5000\*(30/100))/1000=1,5

ЗАТРАТЫ =2,5\*13,51,18\*4+1,5= 217,172[чел./мес.]

1. **Расчет длительности и стоимости разработки**

Предположим, что в проекте задействовано 10 разработчиков, что означает продолжительность проекта в ~22 месяца. При средних затратах на разработчика в 100 т.р./мес. Стоимость проекта составит 21 700 000 р.

1. **Зависимости между параметрами модели**

Зависимости между параметрами модели рассмотрим на примере взаимосвязи масштабных факторов. Так, гибкость разработки(FLEX) отрицательно влияет на зрелость процесса PMAT, так что увеличение одного параметра приводит к уменьшению другого. Наоборот, увеличение предсказуемости (PREC) уменьшает разрешение риска (RESL).

Варьируя масштабные факторы, можно получить оптимальное значение (локальный минимум) для конкретной организации , в которой ведется разработка. Но для этого необходимо в течение длительного времени собирать метрики масштабных факторов в данной организации, что является непростой задачей.