

Техническое задание

Сетевой графический ASCII-арт редактор

Список ответственных исполнителей

Организация, должность	Фамилия, имя, отчество
УО «БГУИР», Студент	Баранович П. Л.
УО «БГУИР», Студент	Юревич А. А.

СОДЕРЖАНИЕ

Введение	5
1 Термины и определения.....	6
1.1 Общие термины и определения	6
1.2 Технические термины и определения	6
1.3 Другие термины и определения	11
2 Общие положения.....	12
2.1 Назначение документа	12
2.2 Цели создания системы.....	12
2.3 Краткое описание проекта	12
2.4 Основные функциональные возможности системы	13
2.5 Плановые сроки выполнения работ.....	13
2.6 Использование технического задания.....	13
3 Функциональные требования	14
3.1 Используемые технологии.....	14
3.2 Диаграммы вариантов использования.....	14
3.3 Описание вариантов использования.....	16
3.3.1 ВИ «Нарисовать графический объект»	16
3.3.2 ВИ «Выделить область холста»	17
3.3.3 ВИ «Масштабировать выделенную область».....	18
3.3.4 ВИ «Зеркально отразить выделенную область»	18
3.3.5 ВИ «Поворот на угол выделенной области»	19
3.3.6 ВИ «Копировать и переместить выделенную область»	20
3.3.7 ВИ «Сохранить изображение»	20
3.3.8 ВИ «Сохранить ASCII-арт»	21
3.3.9 ВИ «Очистить изображение»	21
3.3.10 ВИ «Очистить ASCII-арт»	21
3.3.11 ВИ «Заполнить замкнутую область цветом».....	22
3.3.12 ВИ «Загрузить изображение с компьютера»	22
3.3.13 ВИ «Посмотреть описание проекта».....	23
3.3.14 ВИ «Сменить IP-адрес и порт назначения»	23
3.4 Диаграмма состояний.....	24
3.5 Диаграмма компонентов	25

3.6	Диаграмма развертывания	26
3.7	Диаграмма последовательности.....	27
3.8	Диаграмма коммуникации	28
3.9	Диаграмма объектов	29
3.10	Внешний вид приложения	30
3.10.1	Внешний клиентской части	30
3.10.2	Внешний серверной части	34
4	Модель данных	36
4.1	Диаграмма классов	36
4.2	Описание диаграммы классов	37
5	Нефункциональные требования.....	38
5.1	Интерфейс пользователя.....	38
5.2	Производительность.....	38
5.3	Отображение на различных ОС	38
5.4	Документация.....	38
5.5	Порядок разработки.....	38
6	Перспективы развития	40

Введение

Работа с компьютерной графикой - одна из самых популярных направлений использования персонального компьютера. Для сферы обучения средства компьютерной графики открывают принципиально новые возможности: в процессе анализа изображений учащиеся могут динамически управлять их содержанием, формой, размерами и цветом, добиваясь наибольшей наглядности. Однако, данное техническое задание разработано не только для работы с компьютерной графикой, но и для отрисовки этого изображения в ASCII-арте на других компьютерах, связанных локальной сетью с главным, что, в целом, повысит интерес обучающихся.

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1.1. Общие термины и определения

ОС – операционная система.

ВИ – варианты использования.

Система – оконное клиент-серверное приложение «Сетевой графический ASCII-арт редактор», требования к которому указаны в данном документе.

Пользователь — лицо, участвующее в функционировании автоматизированной системы или использующее результаты её функционирования.

ASCII-арт – форма изобразительного искусства, использующая символы ASCII на моноширинном экране компьютерного терминала (терминальный сервер) или принтера для представления изображений. При создании такого изображения используется палитра, состоящая из буквенных, цифровых символов и символов знаков пунктуации из числа 95 символов таблицы ASCII. По причине высокой вероятности различий в представлении на системах с национальными вариантами таблицы остальные 160 символов, как правило, не используются.

«Клиент-сервер» – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиенты сервер – это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине.

1.2. Технические термины и определения

TCP/IP – сетевая модель передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю. В модели предполагается прохождение информации через четыре уровня, каждый из которых описывается правилом (протоколом передачи).

TCP, сокращенно от Transmission Control Protocol – это стандарт, определяющий как устанавливать и поддерживать связь, через которую две программы могут обмениваться данными. Механизм TCP предоставляет поток данных с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от UDP, целостность передаваемых данных и уведомление отправителя о результатах передачи.

UDP, сокращенно от User Datagram Protocol (протокол пользовательских датаграмм) – один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

C++ – компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником – языком C, – наибольшее

внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

Tkinter – кросс-платформенная графическая библиотека на основе средств Tk (широко распространённая в мире GNU/Linux и других UNIX-подобных систем, портирована также и на Microsoft Windows), написанная Стином Лумхольтом (Steen Lumholt) и Гвидо ван Россумом. Входит в стандартную библиотеку Python. Tkinter – это свободное программное обеспечение, распространяемое под Python-лицензией.

Python Imaging Library (Pillow) – библиотека языка Python (версии 2), предназначенная для работы с растровой графикой. Разработка библиотеки прекращена (последняя правка датируется 2011 годом). Однако проект под названием Pillow, являющийся форком PIL, развивается и включает, в том числе, поддержку Python 3.x. Этот форк был принят в качестве замены оригинальной библиотеки и включён в некоторые дистрибутивы Linux, включая Debian и Ubuntu.

Socket – модуль, который позволяет обмениваться данными с локальным или удалённым хостом по BSD-сокетах в режиме с установлением соединений (TCP) или на основе датаграмм (UDP). Семантика вызовов в API модуля socket точно соответствует семантике соответствующих вызовов в POSIX. Имена и сигнатуры функций по большей части совместимы с luasocket.

PEP8 можно определить, как документ, описывающий общепринятый стиль написания кода на языке Python. Python Enhanced Proposal (PEP) - переводится, как заявки по улучшению языка Python.

Git – это распределённая система контроля версий для отслеживания изменений в исходном коде во время разработки программного обеспечения. Он предназначен для координации работы программистов, но может использоваться для отслеживания изменений в любом наборе файлов. Его цели включают в себя скорость, целостность данных, и поддержку распределённых, нелинейных рабочих процессов. Как и в большинстве других распределённых систем контроля версий, и в отличие от большинства систем клиент-сервер, каждый каталог Git на каждом компьютере является полноценным хранилищем с полной историей и возможностями полного отслеживания версий, независимо от доступа к сети или центрального сервера. Git – это бесплатное программное обеспечение с открытым исходным кодом,

распространяемое в соответствии с условиями Стандартной общественной лицензии GNU версии 2.

В системах управления версиями **коммит** добавляет последние изменения в [часть] исходного кода в хранилище, делая эти изменения частью основной версии хранилища. В отличие от коммитов в управлении данными, коммиты в системах контроля версий хранятся в хранилище неограниченное время. Таким образом, когда другие пользователи делают update или checkout из репозитория, они получают последнюю зафиксированную версию, если только они не укажут, что хотят получить предыдущую версию исходного кода в репозитории. Системы контроля версий позволяют легко вернуться к предыдущим версиям. В этом контексте фиксация в системе управления версиями защищена, поскольку ее легко откатить даже после применения фиксации.

Ветка в Git'e — это просто легковесный подвижный указатель на один из этих коммитов. Ветка по умолчанию в Git'e называется master. Когда вы создаёте коммиты на начальном этапе, вам дана ветка master, указывающая на последний сделанный коммит. При каждом новом коммите она сдвигается вперёд автоматически.

Кросс-платформенность или межплатформенность – способность программного обеспечения работать с двумя и более аппаратными платформами и (или) операционными системами. Обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ. Типичным примером является программное обеспечение, предназначенное для работы в операционных системах Linux и Windows одновременно.

Диаграмма вариантов использования – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ

взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования

Диаграмма состояний – это, по существу, диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов.

Диаграмма компонентов – элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

Диаграмма развертывания – в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом.

Диаграмма последовательности – диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов). Используется в языке UML.

Диаграмма объектов – диаграмма, которая предназначена для демонстрации совокупности моделируемых объектов и связей между ними в фиксированный момент времени. Она описывает конкретные экземпляры объектов и напрямую соотносится с диаграммой классов,

которая даёт общее представление о конфигурации системы. Диаграмма объектов используется для документирования структур данных и создания статических снимков состояний объектов принимая во внимание реальные экземпляры или прототипы. Динамику поведения объектов обычно изображают в виде последовательности таких диаграмм.

Диаграмма коммуникации – диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме коммуникации явно указываются отношения между объектами, а время как отдельное измерение не используется (применяются порядковые номера вызовов).

1.3. Другие термины и определения

2. ОБЩИЕ ПОЛОЖЕНИЯ

2.1. Назначение документа

В настоящем документе приводится полный набор требований системе, необходимых для реализации.

2.2. Цели создания системы

Создать развлекательное приложение для рисования. Практиковать использование сетевых протоколов TCP, UDP, в целом сетевой модели передачи данных TCP/IP и использование низкоуровневых сетевых интерфейсов.

2.3. Краткое описание проекта

Проект представляет собой клиент-серверное приложение.

На клиентской части пользователь может воспользоваться оконным интерфейсом для рисования и редактирования различных растровых изображений. Интерфейс поддерживает рисование прямых, изогнутых и ломаных линий (с регулировкой толщины и цвета линий), различных геометрических фигур (окружность, эллипс, квадрат, прямоугольник, пятиконечная звезда и некоторые другие виды многоугольников), выделения определённой области с последующими операциями над ней (увеличения в размерах, зеркального отражения и вращения вокруг центра), копирование и вставка фрагментов изображения, масштабирование, заливка замкнутых фигур цветом, сохранение и загрузка уже готовых изображений.

На серверной - пользователь видит в онлайн режиме изменения, производимые на клиентской части. Эти изменения отображаются в виде цветного ASCII-арта (отображаемого в консоли), в который переводится изображение клиента. Пользователь может сохранить данный ASCII-арт себе на компьютер в виде изображения в формате BMP.

Для ускорения передачи данных и обеспечения синхронизации обмен данными будет производиться через самописный простейший протокол TCP, который реализуется на основе UDP протокола,

устанавливает соединение с сервером, посредством локальной сети (IP адрес и порт заранее заданы), разбивает информацию на пакеты заданной длины, осуществляет повторный запрос данных в случае их потери, собирает пакеты в порядке их отправки и устраняет дублирование при получении двух копий одного и того же пакета.

2.4. Основные функциональные возможности системы

Визуальная часть:

- Клиентская часть
 - 1) Рисование стандартных фигур:
 - а) Цветные прямые, изогнутые и ломаные линии
 - б) Геометрических фигур (окружность, эллипс, квадрат, прямоугольник, пятиконечная звезда и некоторые другие виды многоугольников)
 - 2) Копирование и вставка фрагментов изображения
 - 3) Масштабирование
 - 4) Заливка замкнутых фигур цветом
 - 5) Сохранение и загрузка уже готовых изображений.
- Серверная часть
 - 1) Отображение ASCII-арт в терминале
 - 2) Сохранение ASCII-арта

Программно-аппаратная часть:

- Синхронный перехват изменений на форме
- Передача изображения (или его части) на сервер
- Преобразование изображения в ASCII-арт на стороне сервера
- Отображение ASCII-арта в терминале

2.5. Плановые сроки окончания работ

Проект планируется сдать до 19 мая 2019 года.

2.6. Использование технического задания

Для составления общей картины курсового проекта.

3. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

3.1. Используемые технологии

- а) Серверные технологии:
 - Язык программирования C++.
- б) Клиентские технологии:
 - Язык программирования Python;
 - Модуль Tkinter;
 - Модуль Pillow;
 - Модуль Socket;

3.2. Диаграммы вариантов использования

На диаграммах представлены основные варианты использования системы, детальное описание которых можно найти в п. 3.2 «Основные варианты использования».

Диаграмма 1. ВИ работы пользователя на стороне сервера

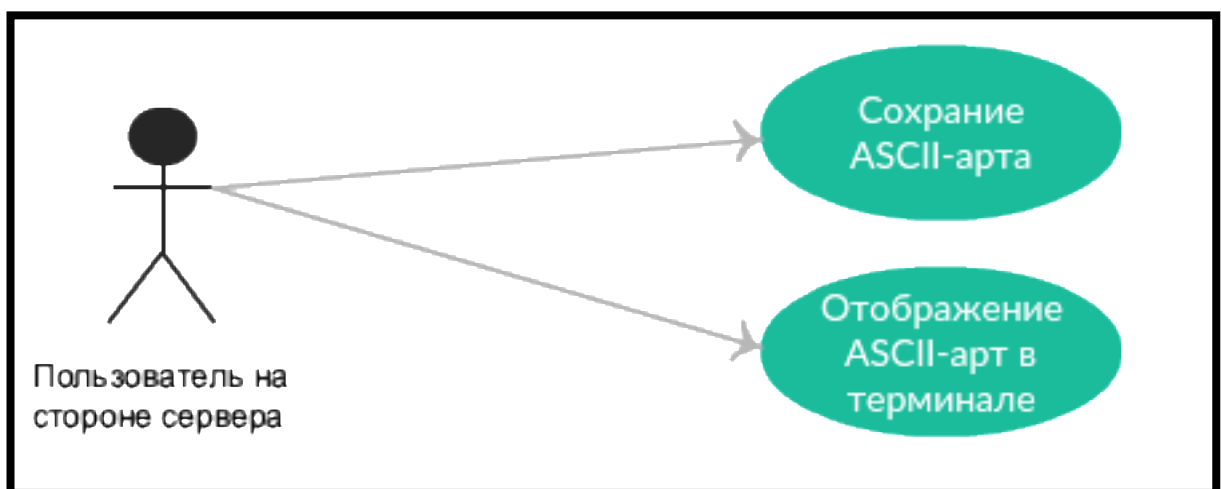


Диаграмма 2. ВИ работы пользователя на стороне клиента



Диаграмма 3. ВИ взаимодействие пользователей



3.3. Описание вариантов использования

3.3.1. ВИ «Нарисовать графический объект»

Описание ВИ:

Пользователь должен иметь возможность нарисовать различные графические объекты

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов;
- Пользователь выбрал толщину линии геометрического объекта.

Основной поток действий:

- Рисование линий:
 - Пользователь с помощью перемещений зажатой левой кнопки мыши рисует линию;
 - Система, во время перемещения мыши, отображает линию с заданной толщиной;
 - Система передаёт информацию о нарисованном объекте серверу.
- Рисование геометрических фигур:
 - Пользователь кликает левой кнопкой мыши в месте предполагаемого правого верхнего угла геометрической фигуры;

- Пользователь, не отпуская кнопку мыши, растягивает геометрическую фигуру до нужных размеров и отпускает кнопку мыши;
- Система, во время перемещения мыши, отображает геометрическую фигуру соответствующего размера;
- Пользователь отпускает правую кнопку мыши;
- Система передаёт информацию о нарисованном объекте серверу.

Альтернативный поток действий:

- Отмена рисования – нажатие клавиши Esc.

3.3.2. ВИ «Выделить область холста»

Описание ВИ:

Пользователь должен иметь возможность выделить прямоугольную область холста;

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов.

Основной поток действий:

- Пользователь кликает левой кнопкой мыши в месте предполагаемого правого верхнего угла выделенной области;
- Пользователь, не отпуская кнопку мыши, растягивает прямоугольник выделения до нужных размеров и отпускает кнопку мыши;
- Система, во время перемещения мыши, отображает прямоугольник выделения соответствующего размера.
- Пользователь отпускает правую кнопку мыши;
- Система подсвечивает выделенную область.

Альтернативный поток действий:

- Отмена выделения – нажатие клавиши Esc, либо нажатие в любое свободное место холста.

3.3.3. ВИ «Масштабировать выделенную область»

Описание ВИ:

Пользователь должен иметь возможность масштабировать выделенную прямоугольную область;

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов.
- Пользователь выполнил выделение прямоугольной области, см. п. 3.3.2. ВИ «Выделить область холста».

Основной поток действий:

- Пользователь, не отпуская кнопку мыши, растягивает прямоугольник масштабирования до нужных размеров и отпускает кнопку мыши;
- Система, во время перемещения мыши, отображает масштабированный объект.
- Пользователь отпускает правую кнопку мыши;
- Система передаёт информацию о масштабированной части объекта серверу.

Альтернативный поток действий:

- Отмена масштабирования – нажатие клавиши Esc.

3.3.4. ВИ «Зеркально отразить выделенную область»

Описание ВИ:

Пользователь должен иметь возможность зеркально отразить выделенную прямоугольную область;

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов.
- Пользователь выполнил выделение прямоугольной области, см. п. 3.3.2. ВИ «Выделить область холста».

Основной поток действий:

- Пользователь кликает левой кнопкой мыши в сторону зеркального отображения;
- Система зеркально отрисовывает выделенную область в выбранную сторону.
- Система передаёт информацию о зеркально отраженной части холста серверу.

Альтернативный поток действий:

- Отмена зеркального отражения – нажатие клавиши Esc.

3.3.5. ВИ «Поворот на угол выделенной области»

Описание ВИ:

Пользователь должен иметь возможность повернуть на угол выделенную прямоугольную область;

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов.
- Пользователь выполнил выделение прямоугольной области, см. п. 3.3.2. ВИ «Выделить область холста».

Основной поток действий:

- Пользователь, не отпуская кнопку мыши, вращает прямоугольную область относительно центра и отпускает кнопку мыши;
- Система, во время перемещения мыши, отображает объект в каждом из положений.
- Пользователь отпускает правую кнопку мыши;
- Система передаёт информацию о повернутой на угол части холста серверу.

Альтернативный поток действий:

- Отмена поворота на угол – нажатие клавиши Esc.

3.3.6. ВИ «Копировать и переместить выделенную область»

Описание ВИ:

Пользователь должен иметь возможность копировать и вставить выделенную часть изображения в другое место на холсте;

Предусловия:

- Пользователь выполнил выделение прямоугольной области, см. п. 3.3.2. ВИ «Выделить область холста».

Основной поток действий:

- Пользователь нажимает сочетание клавиш Ctrl+C;
- Система запоминает скопированную часть.
- Пользователь нажимает сочетание клавиш Ctrl+V;
- Система вставляет ранее скопированную область на холст;
- Пользователь перемещает скопированную область в нужное место холста;
- Система сохраняет перемещённую часть изображения;
- Система передаёт информацию об измененной части холста серверу.

3.3.7. ВИ «Сохранить изображение»

Описание ВИ:

Пользователь должен иметь возможность сохранить изображение с холста на компьютер.

Основной поток действий:

- Пользователь выбирает пункт меню «Сохранить изображение»;
- Система открывает диалоговое окно выбора местоположения файла;
- Пользователь выбирает необходимое местоположение и формат изображения;
- Система сохраняет изображение в соответствии с выбранными критериями.

3.3.8. ВИ «Сохранить ASCII-арт»

Описание ВИ:

Пользователь должен иметь возможность сохранить ASCII-арт из терминала на компьютер.

Основной поток действий:

- Пользователь выбирает пункт меню «Сохранить ASCII-арт»;
- Система открывает диалоговое окно выбора местоположения файла;
- Пользователь выбирает необходимое местоположение и формат файла;
- Система сохраняет ASCII-арт в соответствии с выбранными критериями.

3.3.9. ВИ «Очистить изображение»

Описание ВИ:

Пользователь должен иметь возможность очистить холст от изображения.

Основной поток действий:

- Пользователь выбирает пункт меню «Очистить холст»;
- Система заполняет холст белым цветом.
- Система передает запрос об очищении ASCII-арта на стороне сервера.

3.3.10. ВИ «Очистить ASCII-арт»

Описание ВИ:

Пользователь должен иметь возможность очистить область отображения ASCII-арта.

Основной поток действий:

- Пользователь выбирает пункт меню «Очистить»;
- Система полностью очищает область отображения ASCII-арта.

3.3.11. ВИ «Заполнить замкнутую область цветом»

Описание ВИ:

Пользователь должен иметь возможность заполнить замкнутую область определённым цветом.

Предусловия:

- Пользователь выбрал соответствующий пункт на панели инструментов.
- Пользователь выбрал цвет-заполнитель.

Основной поток действий:

- Пользователь нажимает левой кнопкой мыши в пределах заполняемой области;
- Система заполняет замкнутую область заранее выбранным цветом.
- Система передаёт информацию о заполненной части изображения серверу.

3.3.12. ВИ «Загрузить изображение с компьютера»

Описание ВИ:

Пользователь должен иметь возможность загрузить готовое изображение различных форматов с компьютера.

Основной поток действий:

- Пользователь выбирает пункт меню «Загрузить изображение»;
- Система открывает диалоговое окно выбора изображения для последующей загрузки;
- Пользователь выбирает нужное изображение;
- Система запоминает выбор пользователя и закрывает диалоговое окно;
- Система загружает изображение и растягивает его по размерам холста;
- Система передаёт информацию о загруженном изображении серверу.

3.3.13. ВИ «Посмотреть описание проекта»

Описание ВИ:

Пользователь должен иметь возможность посмотреть информацию о приложении.

Основной поток действий:

- Пользователь выбирает пункт меню «About»;
- Система отображает диалоговое окно с информацией о созданном приложении.

Альтернативный поток действий:

- Заккрытие диалогового окна – нажатие клавиши Esc, либо соответствующей иконки окна.

3.3.14. ВИ «Сменить IP-адрес и порт назначения»

Описание ВИ:

Пользователь должен иметь возможность сменить IP-адрес и порт назначения для связи с сервером.

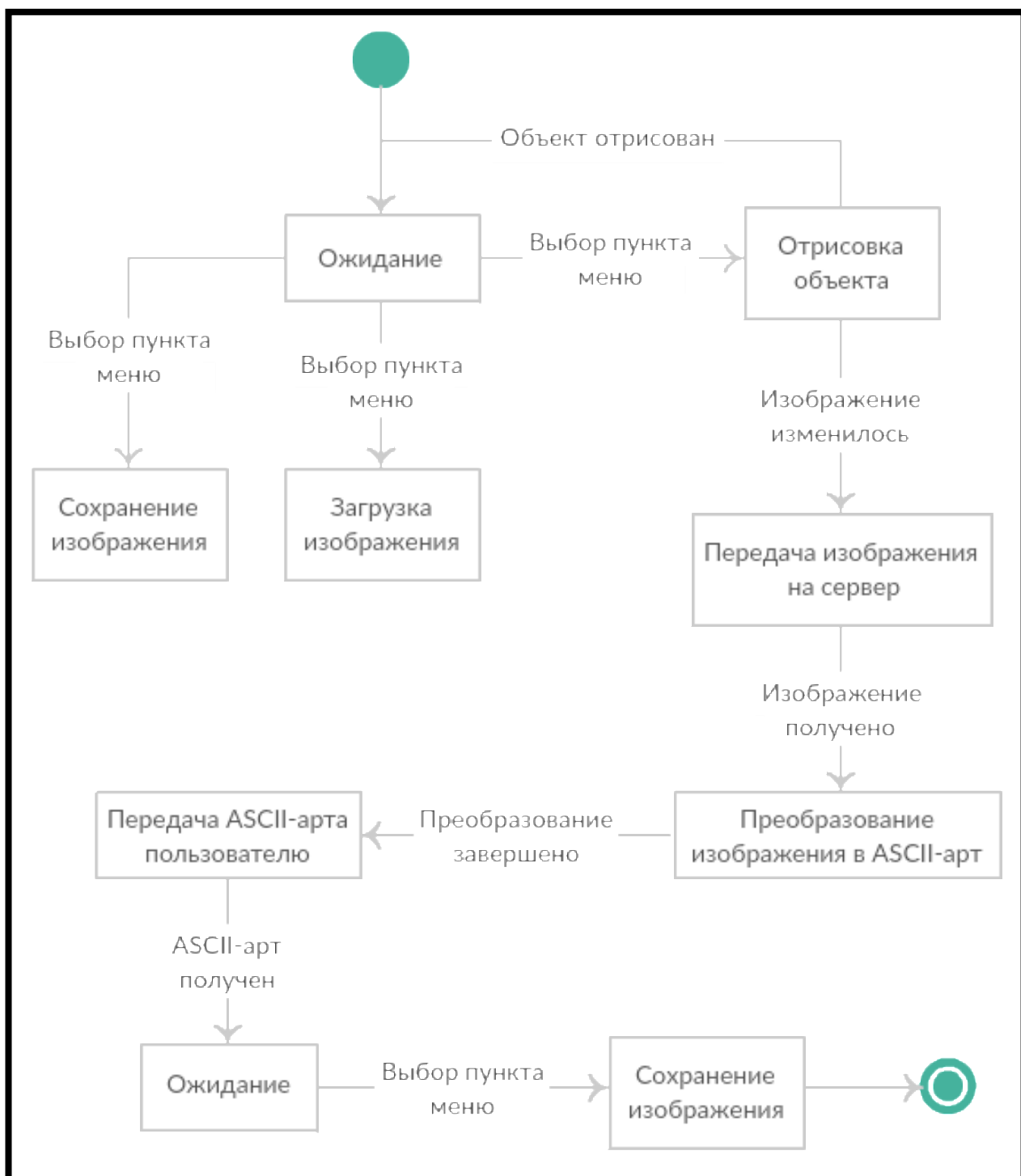
Основной поток действий:

- Пользователь выбирает пункт меню «Сменить IP-адрес и порт»;
- Система отображает диалоговое окно с полями для ввода значений;
- Пользователь вводит соответствующие значения;
- Система запоминает новый IP-адрес и порт назначения, использует их для дальнейшей связи с сервером.

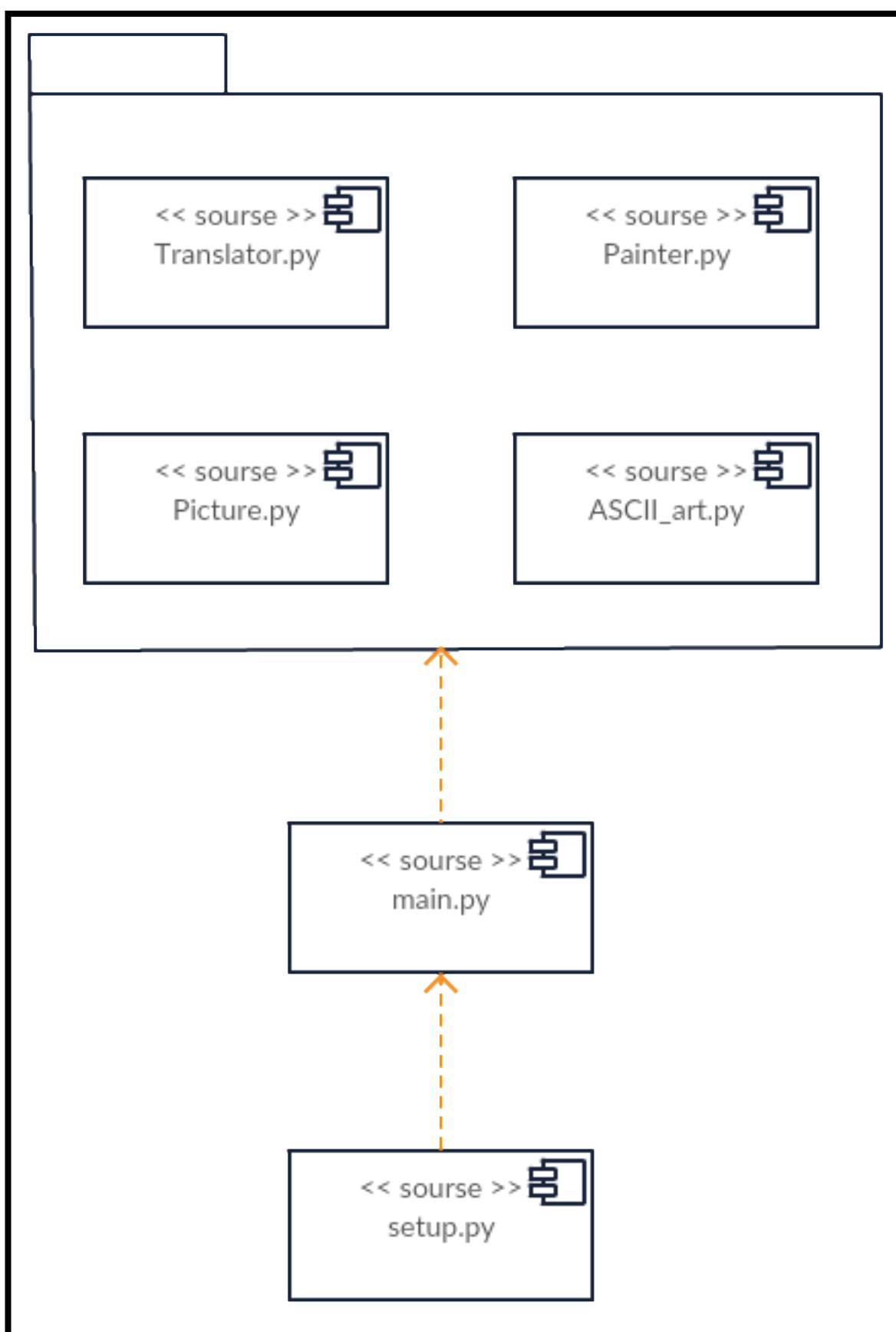
Альтернативный поток действий:

- Заккрытие диалогового окна – нажатие клавиши Esc, либо соответствующей иконки окна.

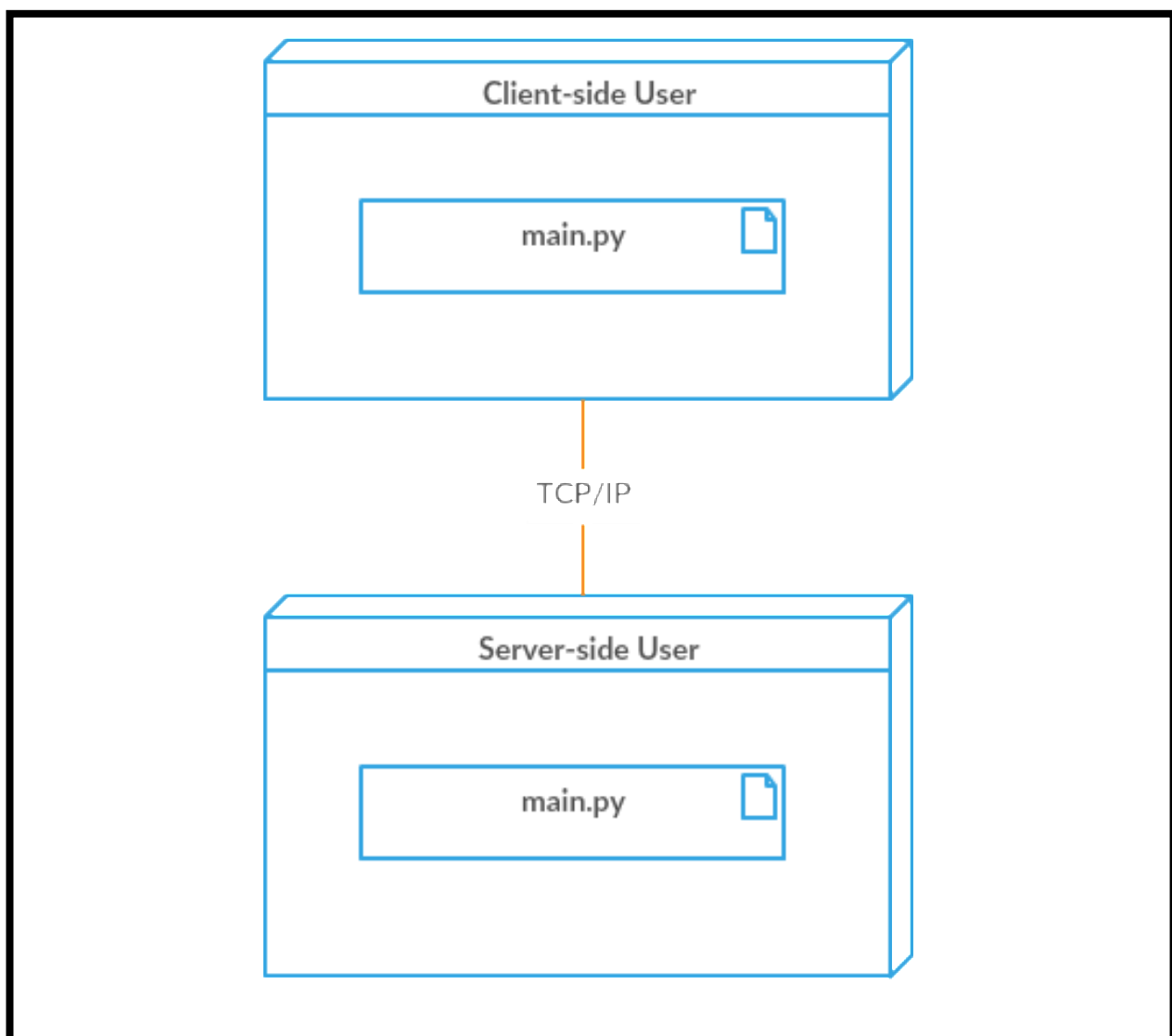
3.4. Диаграмма состояний



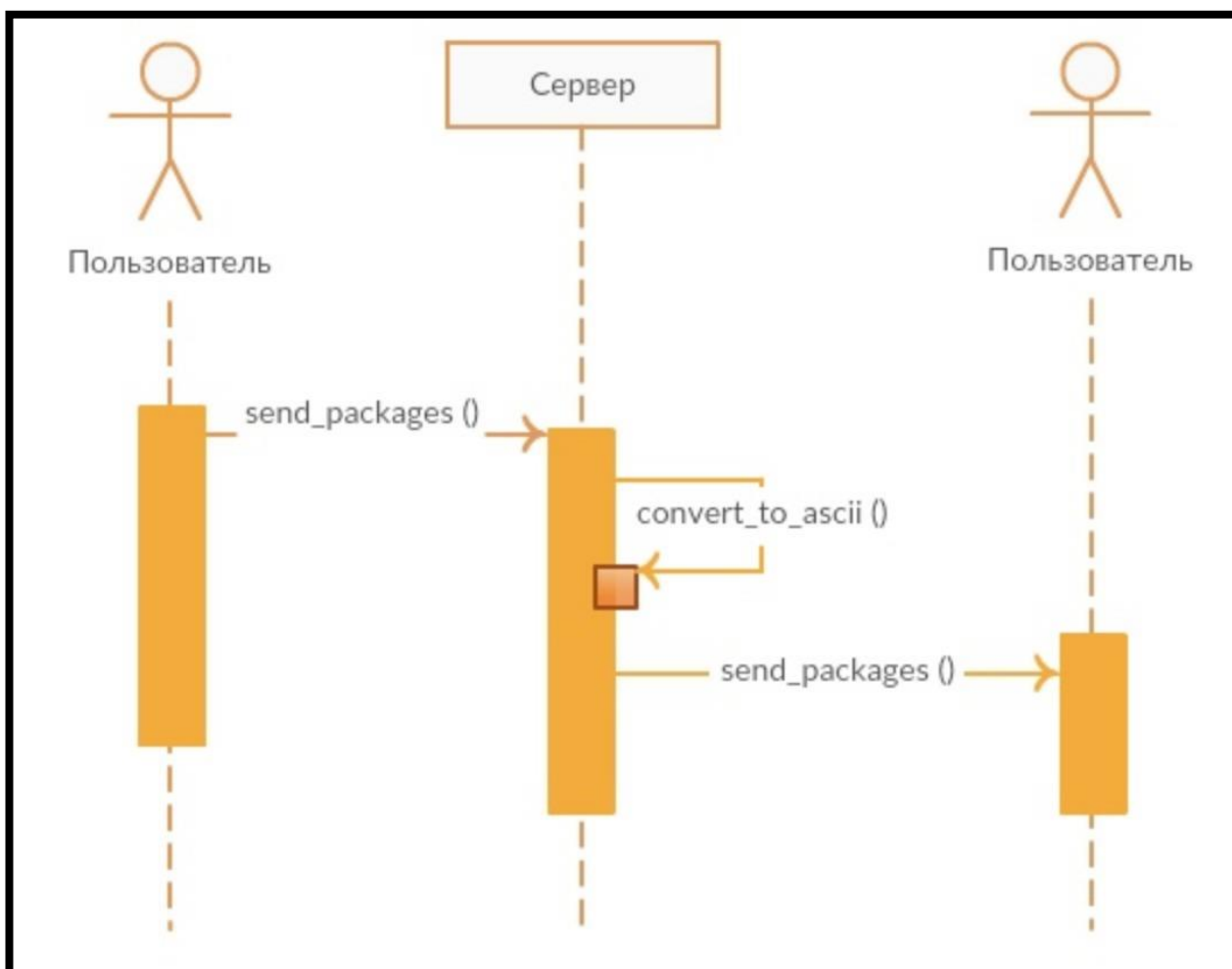
3.5. Диаграмма компонентов



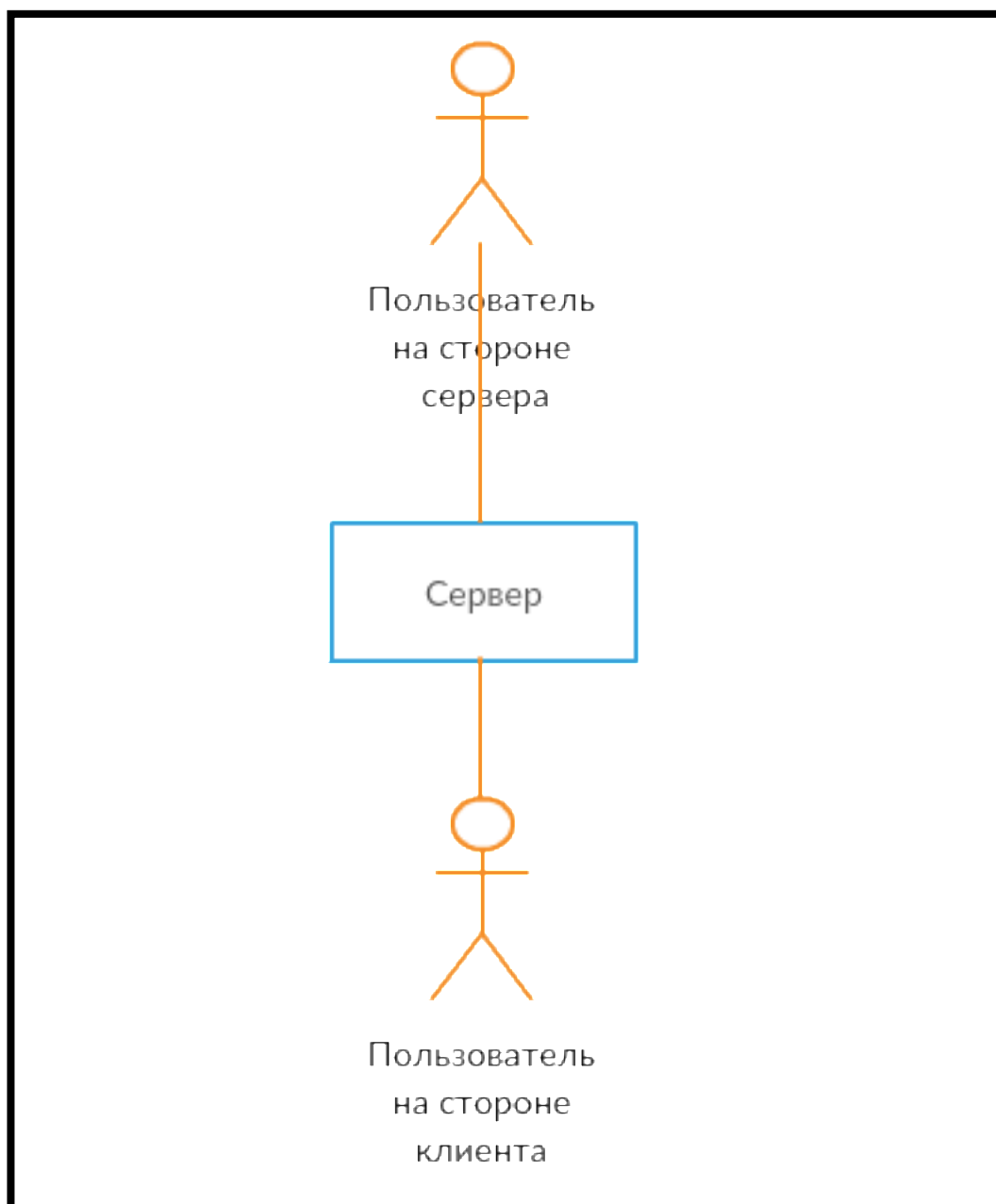
3.6. Диаграмма развертывания



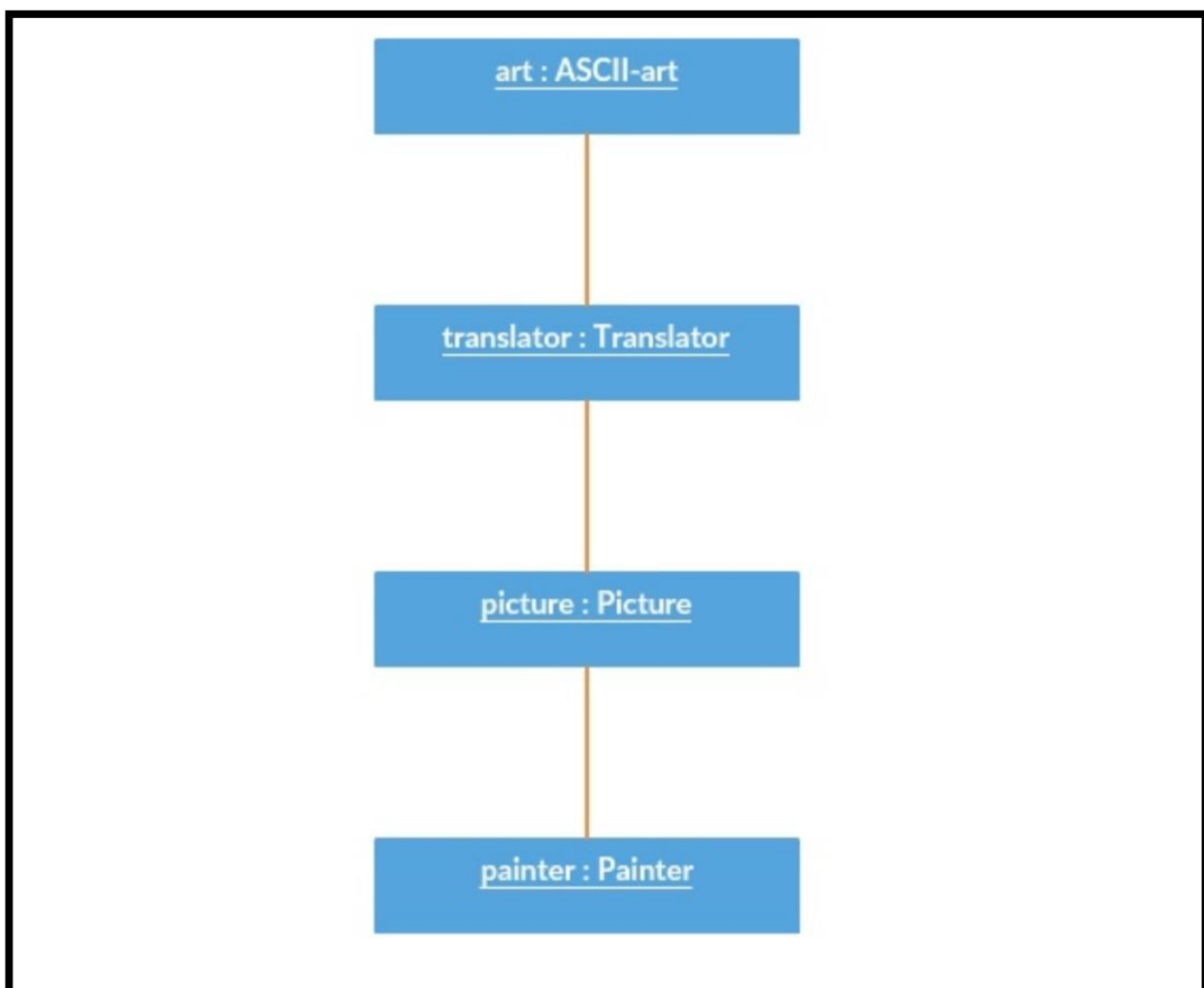
3.7. Диаграмма последовательности



3.8. Диаграмма коммуникации



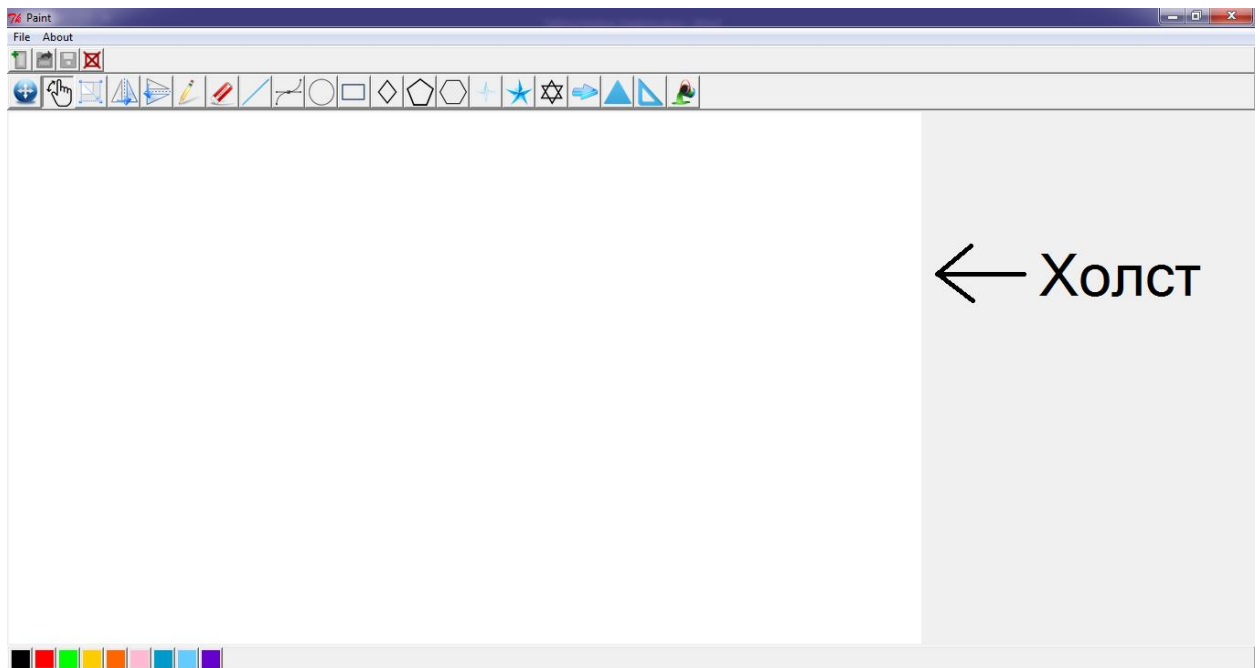
3.9. Диаграмма объектов



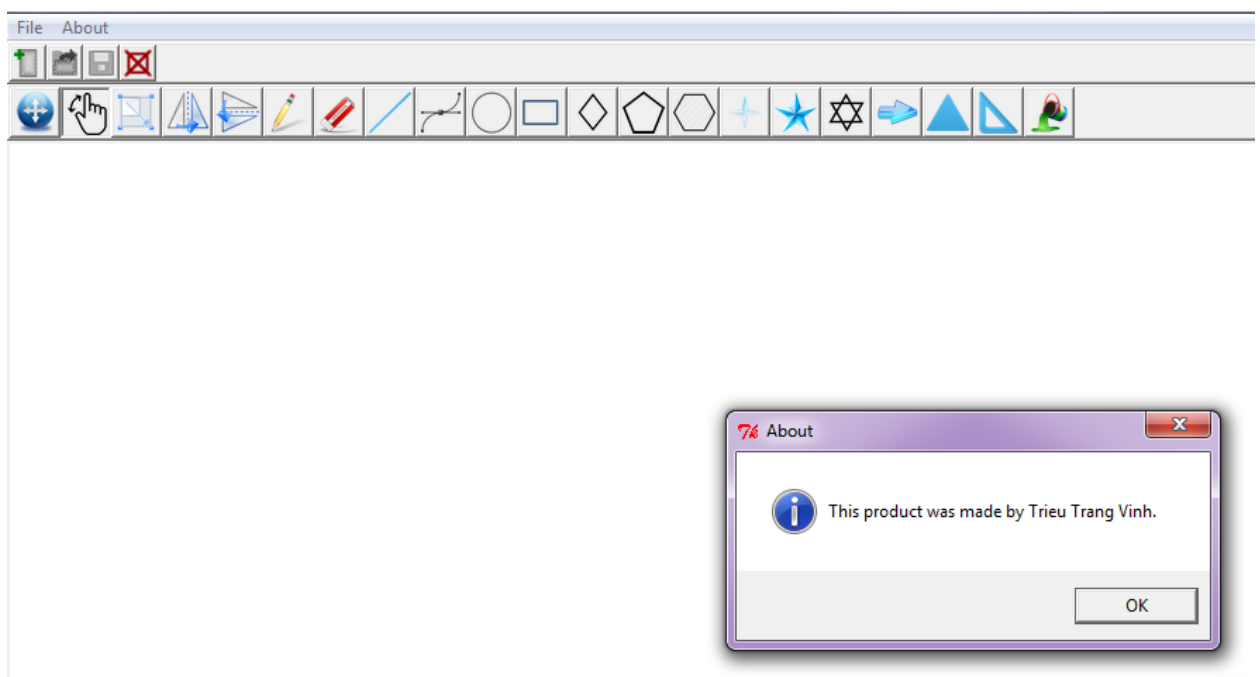
3.10. Внешний вид приложения

3.10.1. Внешний вид клиентской части

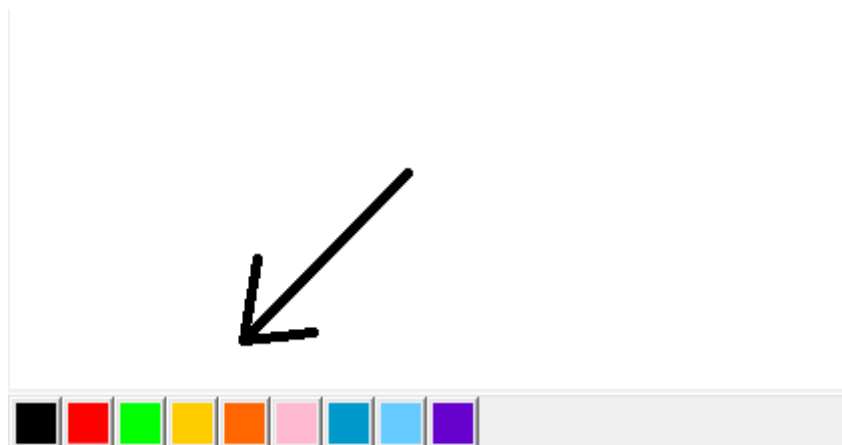
Внешний вид приложения на клиентской части



Диалоговое окно описания приложения («About»)



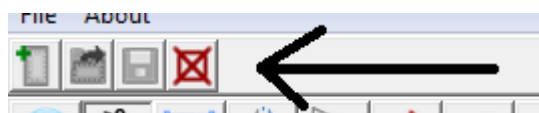
Палитра выбора цветов



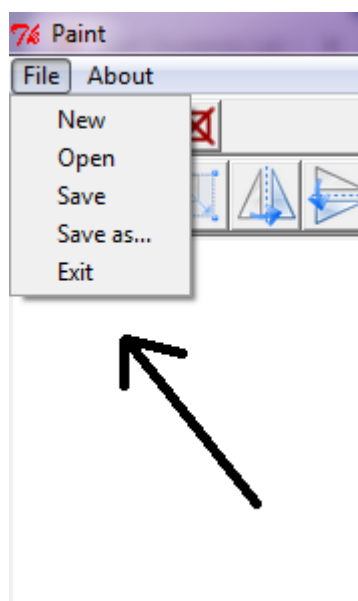
Панель инструментов



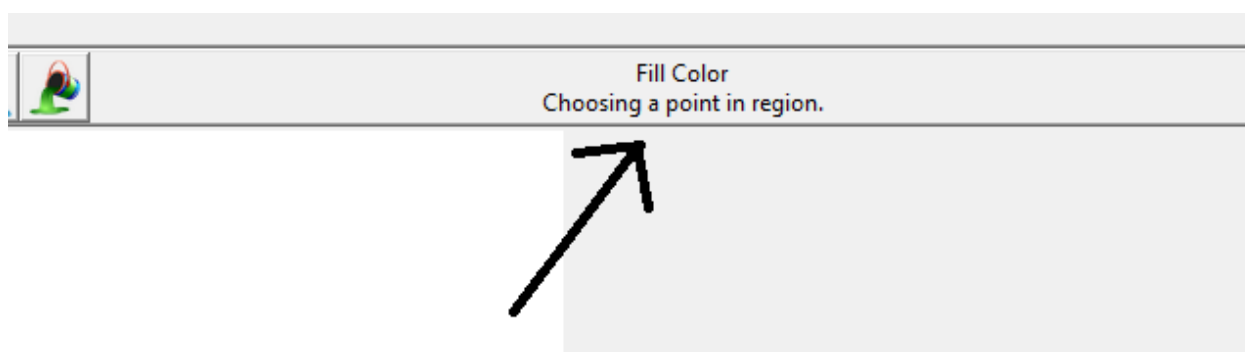
Панель работы с файлами



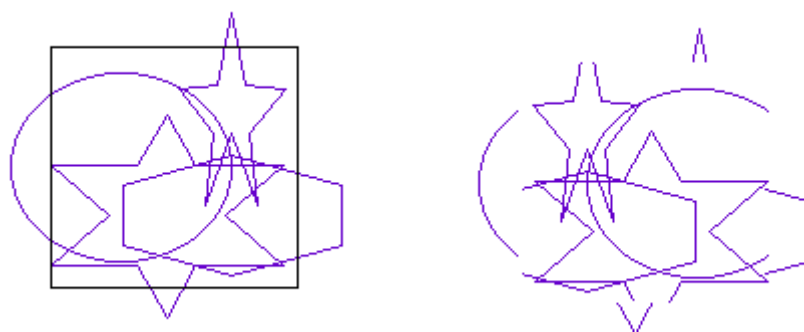
Меню работы с файлами



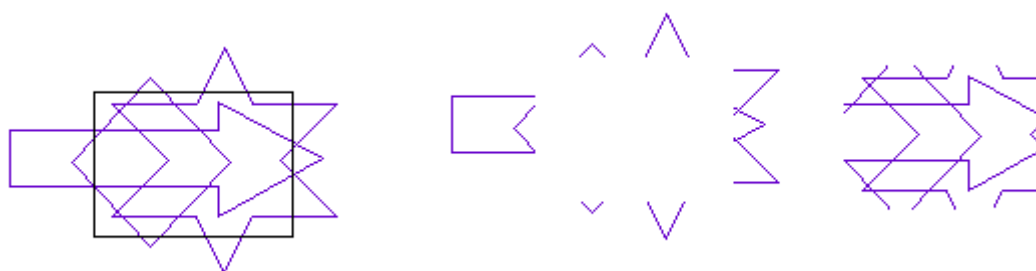
Подсказка об инструменте при наведении на него



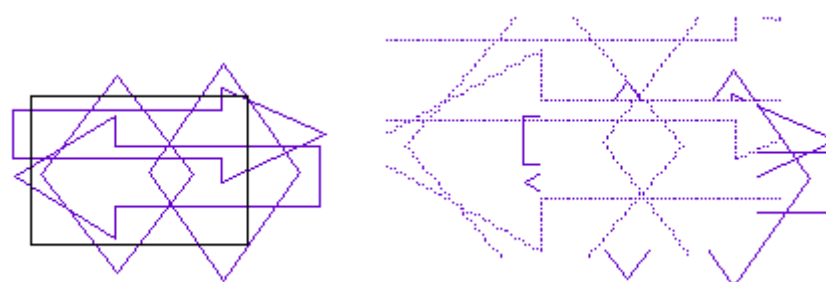
Зеркальное отражение



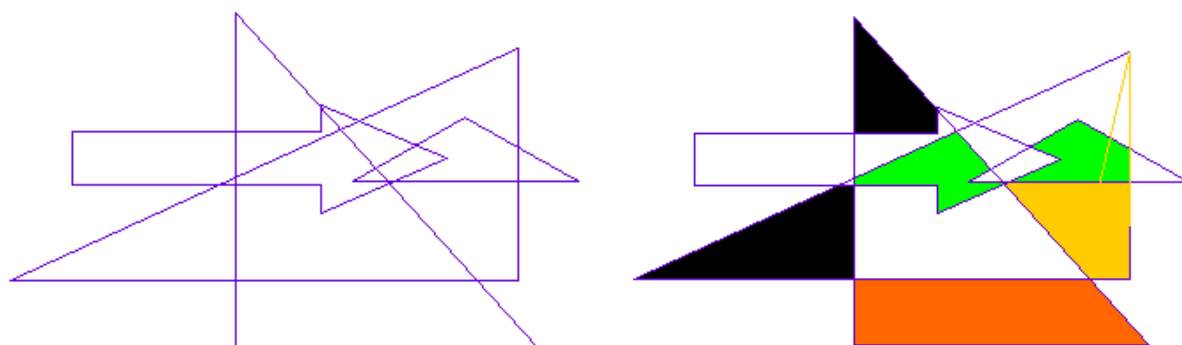
Перемещение части объекта



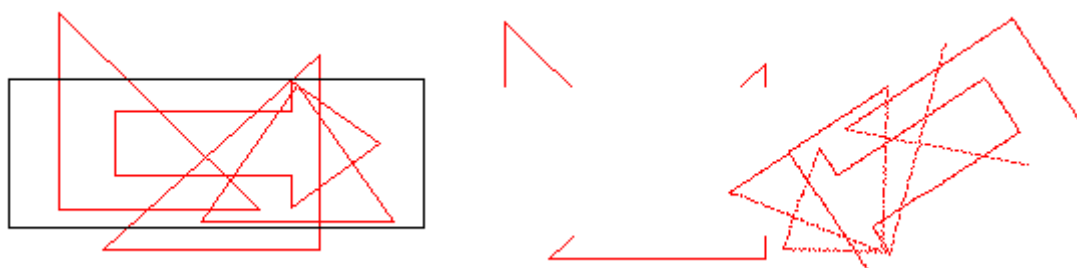
Масштабирование



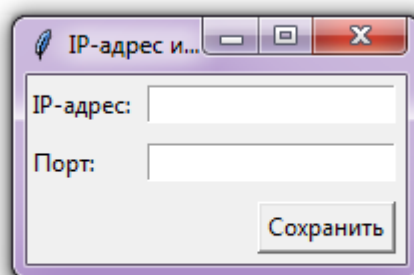
Заливка цветом



Вращение относительно центра выделенной области

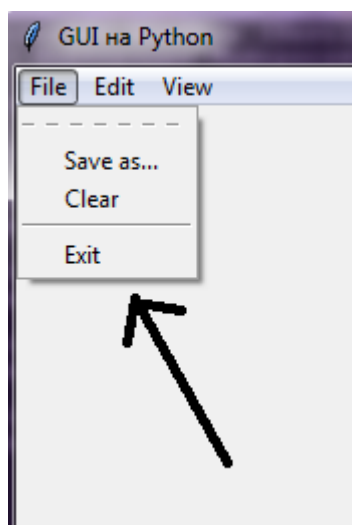


Диалоговое окно ввода, нового IP-адреса и порта назначения



3.10.2. Внешний серверной части

Меню приложения



Отрисовка черно-белого изображения



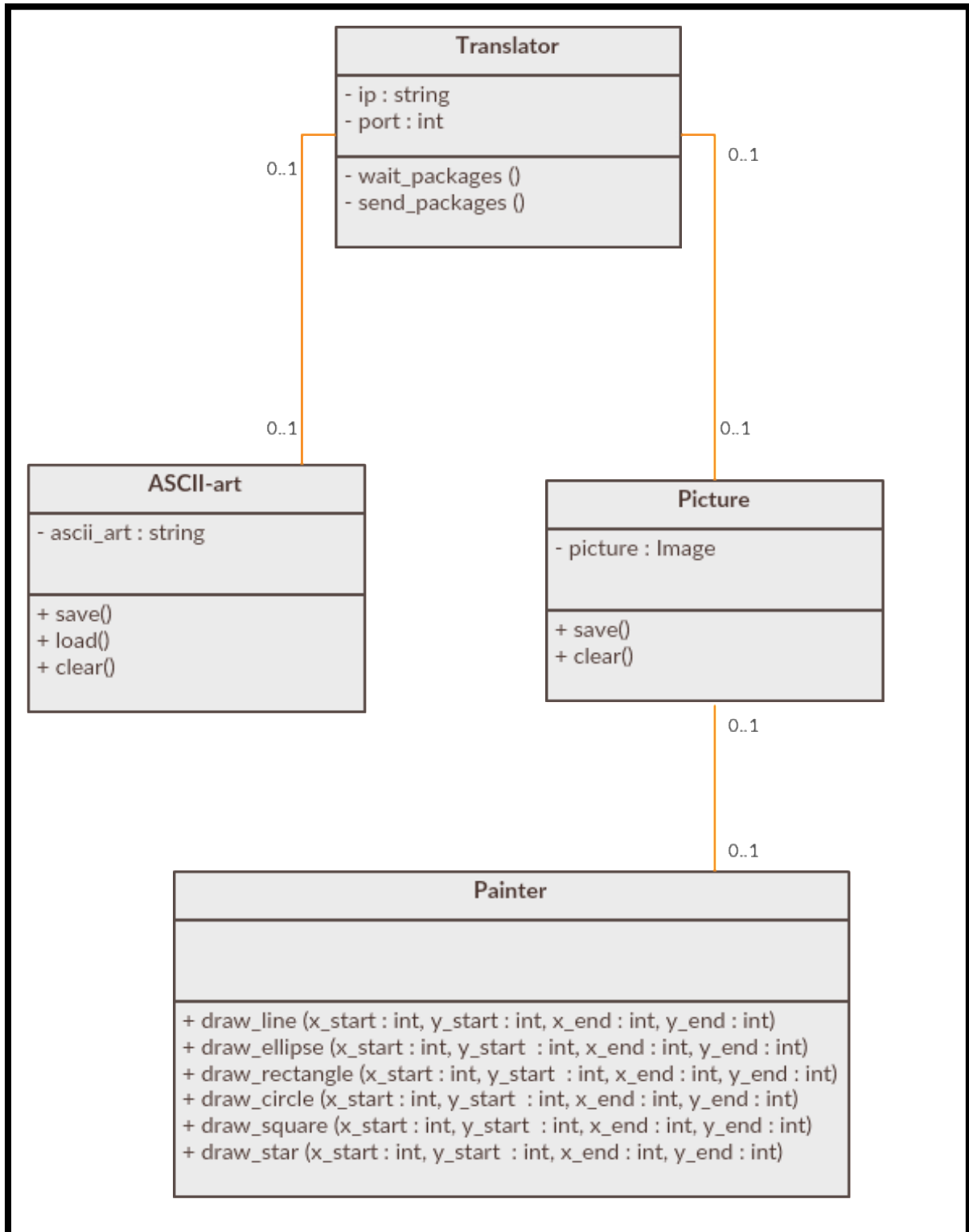
Отрисовка цветного изображения



4. МОДЕЛЬ ДАННЫХ

4.1. Диаграмма классов

На диаграмме показаны основные сущности системы и взаимосвязь между ними.



4.2. Описание диаграммы классов

Класс Painter – представляет собой класс для непосредственного рисования геометрических объектов, таких как линия, эллипс, круг, прямоугольник, квадрат, звезда.

Класс ASCII-art – класс для сохранения, загрузки и очистки отображаемого ASCII-арта.

Класс Picture – класс для загрузки, сохранения и очищения изображения рабочей области.

Класс Translator – служит для передачи информации по локальной сети от клиента к серверу.

5. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

5.1. Интерфейс пользователя

Требования к пользовательскому интерфейсу:

- Интерфейс пользователя должен быть мультиязычным, в начале два языка: русский и английский.
- Система должна отображать корректно интерфейс пользователя с разрешением от 1024x600 пикселей.

5.2. Производительность

Требование к производительности:

- Система должна стабильно отображать ASCII-арт с задержкой не более 5 секунд

5.3. Отображение на различных ОС

Приложение должно быть кроссплатформенным, в том числе, с возможностью включения клиентской части на одной ОС, а серверной – на другой.

5.4. Документация

Документация должна соответствовать следующим требованиям:

- Разрабатываемые программные модули должны быть самодокументированы, т. е. тексты программ должны содержать все необходимые комментарии;
- В состав сопровождающей документации должна входить пояснительная записка на 60 листах (30 из них - это описание реализации приложения со вставками кода) + 10 страниц кода.

5.5. Порядок разработки

Требования к процессу разработки программного продукта:

- Вся разработка должна сопровождаться гранулярными коммитами (на английском языке) в системе GIT. Коммиты

должны быть понятны и содержать лишь основную суть изменений кода программы. При разработке должны быть активно использованы соответствующие ветки (весь процесс разработки не может производиться только в ветке master);

- Разрабатываемая программа должна соответствовать руководству по стилю кода (PEP-8 для Python).

6. ПЕРСПЕКТИВЫ РАЗВИТИЯ

В дальнейшем планируется улучшение интерфейса графического редактора, увеличение скорости передачи данных на серверную часть за счёт обработки только редактируемой части изображений, повышение удобства использования, возможность сжатия изображений и выделения свободной области (в том числе, поддержкой всех функций, доступных при выделении прямоугольной области).