МБОУ «Средняя общеобразовательная школа №4» городского округа Судак

Рассмотрено и одобрено На заседании ШМО Протокол № 1 «___» _____ 2023г.	СОГЛАСОВАНО Заместитель директора по УВР _____ «___»_____2023г	УТВЕРЖДАЮ Директор МБОУ _____Ю.А.Собко Приказ № от .08.2023г

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Программирование на Python (второй год)» на 2023/2024 учебный год

Вид программы: **Авторская**
Составитель: учитель информатики
Козлов Сергей Владимирович

СУДАК – 2023

Оглавление

1.Комплекс основных характеристик программы.

1.1. Пояснительная записка.

1.1.1. Нормативно-правовая основа программы

Центры «Точка роста» — специальные образовательные центры, создаваемые на базе школ в селах и малых городах. Их работа направлена на подготовку детей по цифровому, естественно-научному, техническому и гуманитарному профилям. Их открытие предусмотрено федеральным проектом «Современная школа», входящим в национальный проект «Образование».
Направленность (профиль) программы – техническая

1.1.2. Направленность (профиль) программы – техническая

1.1.3. Актуальность программы

Динамичное развитие IT отрасли приводит к появлению и развитию новых языков программирования, таких как Python.
Важно быть на гребне информационно-технической волны и использовать лучшие практики современного рынка IT.

1.1.4. Новизна программы

Новизна программы заключается в использовании современных средств программирования, таких как Visual Studio Code, включающие в себя ассистентов (нейронная сеть, обученная на множестве примеров программного кода), таких как GitHub Copilot.
Кроме этого, программа построена на основе проектного подхода, где каждый блок занятий – это работа по созданию программного кода на заданную тему и с определённым функционалом.

1.1.5. Адресат программы

Данный курс предназначен для школьников 7 – 11 классов при условии успешного прохождения курса «Программирование на Python (первый год)»

1.1.6. Объем и срок освоения программы

Данный курс рассчитан на 34 занятия один раз в неделю.

1.2. Цель и задачи программы.

Основная цель курса – закрепление базовых понятий и принципов программирования на Python, полученных в предшествующем курсе и расширение кругозора в области использования дополнительных библиотек (PyQt5, json, PIL, Pygame, Pyinstaller и др.) и средств хранения и контроля версий написанного кода GitHub
Для достижения поставленной цели будет написано порядка 10 приложений разной направленности:

- Memory Card
- Умные заметки
- Easy Edior
- Игра "Лабиринт"
- И др.

1.3. Воспитательный потенциал программы.

Участие в данной образовательной программе открывает возможность участия в местных и республиканских образовательных конкурсах для проявления творческих и интеллектуальных способностей.

1.4. Содержание программы.

№	Тема	Количество часов
1	СТРУКТУРЫ ДАННЫХ	4
2	РАЗРАБОТКА ОКОННЫХ ПРИЛОЖЕНИЙ	6
3	РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ	4
4	АВТОМАТИЧЕСКАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ	4
5	ПРОДВИНУТАЯ РАЗРАБОТКА ИГР НА PYGAME	9
6	ПУБЛИКАЦИЯ И РАСПРОСТРАНЕНИЕ ПО	4
7	ИТОГОВОЕ ПОВТОРЕНИЕ	3

1. Структуры данных(4 часа)

1.1. Повторение. Обработка исключений

Цель урока — актуализировать базовые знания о Python и сформировать понимание обработки исключений как инструмента уведомления пользователя о некорректной работе программы. Ученики повторяют основные алгоритмические конструкции и применяют их на практике. Затем они изучают операторы обработки исключений и используют их в решении сюжетных задач.
Обсудите распространённую проблему: если пользователь вводит ответ в вольном стиле, то интерпретатор Python может выдать ошибку.
Хотя разработчики умеют читать сообщения об ошибках и понимать причину их возникновения, обычные пользователи не сумеют в них разобраться.
Возникает необходимость внедрения инструмента для выявления нежелательного поведения программы и уведомления пользователя о причине его возникновения — обработчика исключений.
Разберите конструкцию обработчика событий. Приведите примеры удачных и неудачных сообщений об ошибках («Ошибка! Введите данные одним числом без других символов» и «Неправильный ввод данных»).
Продемонстрируйте доработку программы с вводом данных обработкой исключений. Дополните исходный код операторами try и except.
Что выведет программа, если ввести: «10»? «20 см», а затем «20»? Подробно обсудите последовательный ввод данных «15 см»-«15см»-«15» (то есть некорректный ввод данных более одного раза).
Какая алгоритмическая конструкция помогла бы запрограммировать ввод данных до тех пор, пока они не будут получены в нужном виде? (Цикл).
Обсудите код программы, дополненный циклом while.

1.2. Повторение. Списки

Цель урока: сформировать у учеников понимание списка как упорядоченной структуры данных, используемой для хранения объектов разных типов.
В течение урока ученики повторяют всё о списках: создание списка, нумерацию элементов, добавление, вставку и удаление элементов, сортировку элементов и подсчёт вхождения элементов в список.
Сначала обсудите задачу с записью школьников в кружок. Она состоит из двух частей: базовой и продвинутой.
Порассуждайте, как должна работать такая программа. Где должны храниться фамилии учеников?
Есть ли фиксированное количество учеников или пользователь может ввести всех желающих?
Как пользователь может остановить ввод?
Изучите оформление необходимых команд. Затем просмотрите код программы. Обратите внимание разработчиков, что ‘0’ не обязательно переводить в целочисленный тип — для распознавания сигнала окончания ввода это не имеет значения.
Обсудите усовершенствование программы. Как создать предзаполненный список учеников?
Какой оператор нужно использовать, чтобы проверить, содержится ли введённая фамилия в списке?
Какой метод сортирует элементы списка по алфавиту (в лексикографическом порядке)?

Затем обсудите задачу с перебором элементов списка и извлечением информации. По условию, нужно написать программу для определения среднего роста учеников в классе и роста самого высокого ученика: Опишите алгоритм работы программы.
Какие элементы будут храниться в списке? С помощью какой алгоритмической конструкции можно по одному перебрать все элементы списка?
Продемонстрируйте использование цикла for для перебора элементов списка.
Отметьте, что, например, указывать нумерацию элементов не нужно — цикл самостоятельно по одному перебирает элементы списка.
В конце обсуждения проверьте понимание разработчиками устройства программ, задав вопросы на понимание.

1.3. Словари

Цель урока: сформировать у учеников понимание словаря как структуры данных, хранящей пары «ключ-значение».
В течение урока ученики повторяют уже известную им структуру данных (списки), знакомятся с новой (словари), сравнивают и оценивают удобство их использования при решении задач.
Объясните суть технического задания на примерах возможных задач. Продемонстрируйте, что они требуют хранения и обработки пар значений, например, «книга-автор», «книга-расположение», «жанр-писатели» и др. Такие пары удобно хранить в словарях.
Продолжите «мозговой штурм» определением словаря, его оформлением и получением значений по ключам.
Для полного усвоения терминологии при обсуждении задавайте вопросы: Сколько пар значений находится в словаре?
Что хранится в одной паре? Сколько ключей в словаре? Перечислите их. Какую информацию хранит ключ?
Перечислите все значения словаря. Могут ли значения совпасть? А ключи?
Сколько экземпляров книги «Муму» есть в библиотеке? (И другие вопросы по примерам).
Продемонстрируйте методы для получения длины словаря, всех ключей и значений.
Сначала обсудите задачу с проверкой наличия книги в библиотеке. Она состоит из двух частей: базовой и продвинутой.
Порассуждайте, как должна работать такая программа. Какая информация должна храниться о книге?
Должна ли информация быть упорядочена? Что должно соответствовать каждому названию книги?
Изучите оформление необходимых команд. Затем просмотрите код программы.
Обратите внимание на добавление элемента в словарь (нет особого метода, как у списка).
Обсудите усовершенствование программы. Как уменьшить количество книг?
Затем обсудите задачу с рекомендацией разделения книг по жанрам:
Опишите алгоритм работы программы.
Какие элементы будут храниться в словаре? Что будет являться ключом, а что словарём? Как определить число имеющихся жанров? Как напечатать все имеющиеся жанры? Продемонстрируйте оптимальную форму записи длинного словаря и код программы.
Отметьте, что результатом работы метода books.keys() является набор значений.
Обсудите усовершенствование. Если нужный жанр не найден, как добавить рекомендацию «жанр-книга» в словарь?
В конце обсуждения проверьте понимание разработчиками устройства программ, задав вопросы на понимание.

1.4. Вложенные структуры данных

Цель урока — узнать и применить на практике способы использования вложенных структур данных.
На занятии ученики узнают, как ограничения использования вложенности, так и эффективные способы использования вложенных структур данных. Вместе с этим ученики повторяют типы данных и функции.
Обсудите задачу с базой веб-разработчиков. Покажите, какая информация должна быть доступна по фамилии сотрудника: должность, стаж, портфолио.
В данном случае требуется только описать структуру хранения данных, проговаривать код не нужно.
Обсудите вопросы: с помощью чего можно организовать хранение и обработку набора данных?
Какая структура данных удобна для получения данных сотрудников по их фамилиям?
Сколько уровней будет в базе данных? Почему?
Придите к необходимости использовать вложенные структуры данных (в данном случае — словарь словарей) и повышению мастерства работы с ними перед хакатоном на специальном тренинге.
Начните тренинг с общих ограничений использования типов данных в списках и словарях. Затем обсудите задачу со структурой для хранения данных отдела дизайна.
Покажите, что информацию о сотруднике легко получить по его фамилии, а сами данные удобно хранить в парах «параметр-значение».
Особое внимание уделите командам получения частей данных по ключам и индексам.
Для полного усвоения задайте вопросы: Обобщите содержание теоретической части тренинга и перейдите к практике.

2. Разработка оконных приложений (6 часов)

2.1. Классы. Введение в PyQt

Цель урока — изучить процесс создания оконных приложений и применить его на практике. Ученики узнают, что элементы оконных приложений являются экземплярами классов, описанных в библиотеке PyQt. Для осмысленной работы с виджетами занятие начинается со знакомства с классами, их свойствами и методами. Затем ученики переходят к реализации сюжетной задачи с помощью PyQt.
Расскажите, что приложение должно генерировать и показывать случайный номер участника розыгрыша. При этом это должно быть именно компьютерное оконное приложение (не консольная программа). Перечислите инструменты для реализации заказа. Покажите, что знаний для программирования функционала хватает, а для создания графического интерфейса пользователя требуется изучить новый инструмент — библиотеку PyQt.
Раскройте необходимость повторения основ объектно-ориентированного подхода и изучения классов.
Каждый элемент интерфейса (объект) наделён свойствами и методами, а виды элементов могут быть взаимосвязаны. Чтобы осознанно применять инструменты PyQt, нужно не просто уметь работать с готовыми объектами, но и понимать, как они устроены внутри, и уметь создавать собственные типы объектов (классы). Не проводя привычную квалификацию рабочих навыков, вспомните основополагающие понятия: объект, свойство и метод. На примере объектов типа «черепашка»

напомните, что в программе может быть много объектов одного типа и каждый их них может обладать некоторыми свойствами.

Расскажите, что подобных «типов объектов» в программировании бесконечно много, и в объектно-ориентированном подходе они называются классами. Сформулируйте определение класса. Отметьте, что многие классы уже описаны в библиотеках (например, Turtle), но не менее часто, программисты часто создают и свои собственные описания объектов. Обсудите алгоритм создания собственного класса. Придумать название класса. Оно должно отражать устройство совокупности объектов. Запрограммировать конструктор — специальную функцию, создающую экземпляр класса и наделяющую его свойствами. Свойства нового объекта можно будет изменить отдельно. Примечание. Обратите внимание разработчиков, что при создании конструктора всегда используется одно и то же служебное слово __init__, выделяющееся двумя нижними подчёркиваниями с каждой стороны. При использовании конструктора в программе он вызывается по названию соответствующего класса. Запрограммировать методы управления экземплярами класса. Назначения методов могут быть разными: от печати информации об объекте в консоль (ведь для интерпретатора этот тип новый, и он не знает, как выводить его на экран) до анализа свойств и внесения в них изменений. Примечание. Обратите внимание разработчиков, что при описании конструктора и методов класса используется служебное слово self. Это параметр, указывающий на объект, к которому применяется метод. Именно благодаря нему впоследствие метод «применяется» к экземпляру класса с помощью точки. Обсудите пример класса — Application. Заметьте, что это учебный класс, и реальный Application устроен гораздо сложнее. Затем задайте вопросы на понимание темы «Мозгового штурма» и переходите к заданию в VSC.

Используя презентацию расскажите ученикам о новом разделе программирования — разработке оконных приложений. Объясните, что называется оконными приложением, в чём отличие приложения от консольной программы; назовите базовые компоненты любого оконного приложения и покажите виджеты, которые в них встречаются (надпись, кнопка, переключатель и другие). Каждый новый термин иллюстрируйте примером и обращайтесь к опыту работы за компьютером у учеников. Говоря о виджетах, спросите, каким образом их можно расположить в окне. Разработчикам уже известно размещение спрайтов по координатам. От лица ведущего разработчика Кости заметьте, что такой способ очень неудобен, особенно если виджетов много. Чтобы избежать неудобства, многие программисты используют лэйауты — вертикальные и горизонтальные направляющие линии, по которым можно размещать содержимое. Завершая введение в оконные приложения, расскажите о двух фазах работы приложения. Используйте опыт разработчиков и проведите аналогию в фазами работы игры. Перейдите к пошаговому созданию первого приложения с помощью PyQt. Сначала опишите процесс создания интерфейса. Назовите элементы, необходимые для реализации заказа. Затем покажите, что иерархическая структура библиотеки PyQt требует аккуратного подключения соответствующих модулей. По очереди демонстрируйте названия виджетов, их конструкторы и базовые методы. Затем покажите процесс обработки события мыши. Покажите, что все действия, которые должны произойти при нажатии на кнопку, нужно поместить в функцию-обработчик. Затем при запуске приложения нужно подключить эту функцию к нужному виджету.

2.2. Проектирование интерфейса

Цель урока — изучить процесс проектирования интерфейса приложения с помощью лэйаутов и применить его в решении сюжетных задач. Ученики узнают, что библиотека PyQt имеет иерархическую многомодульную структуру. Для осмысленной

работы с её элементами занятие начинается со знакомства с наследованием и программированием класса-родителя и класса-наследника.
Расскажите, что нужно написать приложение для конкурса, которое будет задавать вопрос об истории канала с выбором ответа. Необходимо, чтобы в зависимости от ответа появлялось окно уведомления о выигранном призе. Перечислите инструменты для реализации заказа. Для этого нужно освоить два новых виджета (переключатель и окно уведомления) и научиться располагать виджеты в нескольких лэйаутах (а не просто в строку или в столбец).
Раскройте необходимость изучения наследования в рамках объектно-ориентированного подхода. Библиотека PyQt имеет иерархическую структуру, и многие элементы являются «предками» и «потомками» по отношению друг к другу. Чтобы осознанно применять инструменты PyQt и понимать, почему у разных классов виджетов могут встречаться одинаковые поля свойств, нужно освоить наследование. Разговор о наследовании начните с нескольких примеров из жизни. Вместе с разработчиками приведите примеры классов объектов, в которых свойства и действия одного класса включаются в другой. На одном из примеров обратите внимание на то, что мы редко называем «общий» тип объекта, если знаем его уточнение (говорим не «усатое пушистое домашнее животное», «кот»). Отметьте, что поскольку профессиональные программисты постоянно работают с объектами разных видов, возникла идея переносить все умения одного класса в другой. Введите понятия «суперкласс» и «класс-наследник». Продемонстрируйте создание класса-наследника: сначала упрощённого, только с дополнительным методом, затем — с новыми свойствами, методами. Обратите внимание разработчиков на метод super, который позволяет обратиться к полям суперкласса. Обсудите класс Application, который уже рассматривался на прошлом занятии. Предложите разработчикам рассмотреть Application как суперкласс и создать класс-наследник MobileApplication. Какими свойствами и методами он может быть дополнен? Пусть MobileApplication расширяется свойством system_type (вид операционной системы телефона) и методом setup_application (печатает «Установка <вид системы> приложения завершена»). Что должно быть в конструкторе класса MobileApplication? Как присвоить значения свойствам, наследуемым от суперкласса Application? Что напечатает программа, если программист решит вывести информацию об экземпляре класса MobileApplication методом, выводящим поля класса Application? Почему не напечаталось поле system_type класса-наследника? Отметьте, что это учебные примеры, но они позволяют понять возможные взаимосвязи между готовыми классами. Затем задайте вопросы на понимание темы и переходите к заданию в VSC.
Напомните задачу рабочего дня и перечислите средства разработки оконных приложений, необходимые для изучения. Затем по шагам расскажите, как создать оконное приложения для конкурса Crazy People. Начните с перечисления всех необходимых виджетов и подключения библиотек. Напомните разработчикам важные разделы PyQt: QtCore и QtWidgets. Затем напомните, как создать пустое окно, сделать его видимым и убрать автоматическое завершение работы приложения. Продемонстрируйте новый виджет QRadioButton, напомните процесс создания лэйаута с одной направляющей линией и продемонстрируйте макет интерфейса с вопросом и вариантами ответа в столбик. Как нужно изменить расположение виджетов, чтобы приложение соответствовало пожеланиям заказчика? Располагать виджеты по лэйаутам можно по-разному. Предложите один из вариантов — привязка виджетов к горизонтальным направляющим и добавление горизонтальных направляющих к главной вертикальной. Продемонстрируйте ещё один новый виджет QMessageBox для отображения окна уведомления. Отметьте, что для этого окна тоже нужно предотвращать автоматическое закрытие. Обсудите обработку событий приложения. Реализовать вывод разных

сообщений, в зависимости от выбранного переключателя, можно по-разному. Предложите вариант с двумя функциями-обработчиками. Переключатель с правильным ответом должен обрабатываться функцией show_win(), а остальные — функцией show_lose().

2.3. Приложение Memory Card. Часть 1

Цель урока — применить знания по основным виджетам и проектированию интерфейсов для создания приложения Memory Card. В первой половине занятия ученики проектируют и программируют форму вопроса с вариантами ответов. Во второй половине занятия дописывается форма отображения результата.

Расскажите, что приложение должно: хранить набор вопросов и вариантов ответа; в случайном порядке задавать вопросы и предлагать варианты ответов; после того, как пользователь выберет ответ и нажмёт на кнопку «Ответить», отображать результат («Правильно» или «Неправильно» и правильный ответ); при нажатии на кнопку «Следующий вопрос» переходить к следующему вопросу. Перечислите инструменты для реализации заказа. Покажите, что такая программа требует как работу с базовыми инструментами Python (случайные числа и структуры данных), так и уверенного знания PyQt. Отметьте, что такой большой заказ требует комплексного использования разных разделов программирования и планирования работы, например, разбиения задачи на подзадачи и составления mind map проекта. Сформулируйте цель рабочего дня и анонсируйте его содержание.

Начните «мозговой штурм» с деления проекта на задачи. Объясните разработчикам, что это не просто приложение-тест с одним вопросом, а серьёзная программа с набором данных, в которой переключение между формами вопроса и правильного ответа должно происходить в одном окне. Обсудите работу над проектом Memory Card: Предложите разработчикам выделить важные задачи проекта. Это могут быть: интерфейс формы вопроса, интерфейс формы правильного ответа, переключение между формами, хранение данных и прочее. Выделите в каждой задаче подзадачи. Например, для формы вопроса это могут быть: создание необходимых виджетов, расположение виджетов по лэйаутам, отображение верхнеуровневого лэйаута. Предложите разработчикам упорядочить рабочие задачи и выделить несколько для текущего рабочего дня. Итогом обсуждения должна стать mind map с выделенной задачей на первую половину рабочего дня. Затем перейдите к обсуждению интерфейса формы приложения. ..... Попросите разработчиков перечислить необходимые виджеты. Напомните, что все они импортируются из модуля QtWidgets библиотеки PyQt. Продемонстрируйте объединение нескольких виджетов в группу. Для этого нужно создать группу вида QGroupBox, расположить нужные виджеты по лэйаутам (как — спросите у разработчиков) и установить в группе главный лэйаут. Обратите внимание на параметры, влияющие на расположение содержимого: выравнивание по центру или по краю, сжатие или растяжение виджетов, добавление пробелов между виджетами или лэйаутами. Такие параметры импортируются из модуля QtCore библиотеки PyQt. Продемонстрируйте ожидаемый результат работы, сформулируйте текущие задачи и переходите к заданию в VSC.

«Отметьте, что и форма вопроса, и форма ответа находятся в одном окне и сменяют друг друга. Настройка функционала не является приоритетом этого рабочего дня, поэтому пока нужно просто скрыть группу переключателей с вариантами ответа (вопрос и кнопка остаются)». Перечислите необходимые виджеты. Спросите разработчиков: есть ли в этой форме группа виджетов, как раньше? Какие виджеты должны в неё включаться? Позвольте разработчикам самим предложить расположение виджетов по лэйаутам. Затем покажите своё решение.

2.4. Приложение Memory Card. Часть 2

Разработчики ProTeam продолжают работать над заказом от культурного центра «Человек мира». Чтобы специалисты центра не теряли квалификацию и хорошо ориентировались в традициях народностей и их языках, центр заказал приложение Memory Card. Приложение должно задавать пользователю вопросы с вариантами ответов, хранящиеся в памяти программы.

Цель урока — применить умение обрабатывать события средствами PyQt для создания приложения Memory Card. В первой половине занятия ученики обрабатывают нажатие на кнопку «Ответить» и программируют переключение между формами вопроса и результата. Во второй половине занятия программируют функцию, отображающую вопрос и варианты ответов по виджетам.

Примечание. В течение четырёх уроков все ученики работают в одном задании — «VSC. Приложение Memory Card».

Объявите, что сейчас вы будете разбирать задачи, связанные с обработкой всевозможных нажатий на кнопку «Ответить». Напомните, что клик по виджету можно обработать как готовой, так и собственной функцией-обработчиком. Анализ правильности ответа, данного пользователем, пока не проводится. 1. Начните с базовой задачи — перейти от формы вопроса к форме ответа. Попросите разработчиков перечислить действия, направленные на изменение интерфейса, которые нужно включить в функцию-обработчик. Затем попросите назвать команды, которые необходимо использовать в функции show_result(). 2. Аналогично попросите назвать необходимые действия и соответствующие им команды для перехода от формы ответа к форме вопроса. Если разработчики назовут те же действия, то задайте наводящий вопрос: «Достаточно ли их?». Обратите внимание, что мы не подумали о переключателях — при возврате к старой форме они не сбросятся.

Продемонстрируйте новые команды, необходимые для сброса кнопок-переключателей. Покажите, что их нужно включить не только в функцию show_question(), но и в интерфейс. Группа переключателей QButtonGroup не выделяется визуально, как QGroupBox, но позволяет управлять всеми элементами группы. После описания функции поставьте вопрос о разделении обязанностей между show_question() и show_result().

Перейдите к задачам, связанным с отображением конкретного вопроса и проверки правильности данного ответа. Отметьте, что до этого вы просто устанавливали подписи виджетам при их создании. Теперь нужно написать функцию, задающую определённый вопрос. После этого шага можно будет перейти к задаванию вопросов из набора. Для решения этой задачи нужно запрограммировать три функции: ask (данные вопроса передаются как параметры и случайно отображаются в виджетах), check_answer (вызывается при нажатии на «Ответить» и определяет результат, старая функция start_test удаляется) и show_correct (вызывается из check_answer, когда нужно показать правильный ответ). Начните с функции ask(). Продемонстрируйте её смысл — отображение передаваемого ей вопроса. Обратите внимание, что просто разместить варианты ответа по переключателям не получится, иначе правильный ответ всегда будет на первом месте. Чтобы поставить в соответствие каждому переключателю вариант ответа случайным образом, перемешаем сами переключатели. Первой из перемешанных кнопок поставим в соответствие правильный ответ, а остальным — неправильные. Перейдите к функции check_answer(). Обратите внимание, что она будет обрабатывать нажатие на кнопку «Ответить» и одновременно проверять выбранный ответ, поэтому старую функцию start_test() нужно скрыть. Т. к. на на предыдущем шаге первому из перемешанных переключателей мы поставили в соответствие правильный ответ, то проверка правильности данного ответа сводится к проверке нажатия на первый переключатель. Обратите внимание разработчиков, что отображение формы с результатом

(«Правильно» или «Неправильно и правильный ответ) можно запрограммировать прямо тут, но удобнее вынести в отдельную функцию show_correct(). Обобщите сказанное. Ещё раз назовите назначение каждой функции и покажите, как они применяются при запуске программы. Опишите ожидаемый результат работы и переходите к заданию.

2.5. Приложение Memory Card. Часть 3

Разработчики ProTeam продолжают работать над заказом от культурного центра «Человек мира». Чтобы специалисты центра не теряли квалификацию и хорошо ориентировались в традициях народностей и их языках, центр заказал приложение Memory Card. Приложение должно задавать пользователю вопросы с вариантами ответов, хранящимися в памяти программы.

Цель урока — применить знания объектно-ориентированного программирования, навыков создания функций и обработки событий в создании приложения Memory Card. В первой половине занятия ученики переходят от хранения данных о вопросе в разрозненных переменных к хранению в экземплярах класса, собранных в список. Во второй половине занятия ученики запрограммируют переход от одного вопроса списка к другому. Таким образом, к концу урока базовый функционал Memory Card должен быть готов.

Обсудите устройство структуры данных. Тут возможны разные технические решения, но в рамках данного проекта предлагается запрограммировать класс Question, полями которого будут формулировка вопроса, правильный ответ и три неправильных ответа. 1. Начните с перечисления программных действий над вопросом. Многие из них уже запрограммированы, поэтому разработчики уверенно должны назвать внесение вопросов в программу, отображение вопроса и вариантов ответа, проверку ответа пользователя и отображение правильного ответа. Затем перечислите данные, необходимые для выполнения этих действий. Сформулируйте вопрос: какая структура подошла бы для комплексного и удобного хранения этих данных и работы с ними? 2. Предложите создать класс Question. Обсудите: Какими свойствами должен быть наделён любой объект типа Question? Как задать экземпляру класса эти свойства при его создании? Какой метод нужно создать? Приведите примеры создания экземпляра класса Question с помощью данного конструктора. 3. Перечислите шаги для внедрения класса Question в программу. Добавление нового вопроса должно осуществляться не через введение новых строковых переменных, а через создание экземпляров класса. Соответственно, в виджетах отображаются данные не из переменных, а из полей класса. Функция ask(), задающая вопрос, также должна работать не с переменными, а со свойствами объекта типа Question. Обобщите сказанное и переходите к введению класса Question в VSC. Отметьте, что внешне программа не должна измениться.

Задача базового уровня подразумевает последовательный перебор вопросов. Отображение случайного вопроса из списка является темой для обсуждения следующего рабочего дня. 1. Начните с перечисления задач при переходе от одного вопроса к нескольким. Необходимо: выбрать структуру для хранения набора вопросов; описать механику отображения вопроса (в том числе, перехода к следующему вопросу). Процесс отображения вопроса нужно свернуть в функцию; описать выбор следующего действия: отображение нового вопроса или проверка ответа на текущий вопрос (в прошлых версиях проекта была функция-обработчик test()). 2. Перейдите к обсуждению задач. Обсудите, какую структуру данных удобно использовать для хранения объектов-вопросов и их последовательного перебора. Придите к необходимости использовать список. Перед обсуждением функции для отображения следующего вопроса предположите, что она уже написана, и порассуждайте, как она впишется в проект в целом. В Memory Card будет две ключевые функции: next_question(), задающая вопрос, и check_answer(),

проверяющая ответ. Вызывать подходящую функцию в зависимости от подписи на кнопке удобно из функции-посредника click_ok(). Когда положение next_question() в программе прояснится, переходите к описанию её функционала. Обратите внимание на необходимость введения счётчика вопросов. От лица разработчика Кости предложите сделать его свойством окна приложения window. Это логично, ведь вопрос отображается в виджетах окна. Обобщите сказанное, покажите, как новый функционал дополнит уже написанный код. Опишите ожидаемое решение и переходите к программированию.

2.6. Приложение Memory Card. Часть 4

Цель урока — завершить работу над приложением Memory Card, настроив отображение случайного вопроса из списка и сбор статистики ответов. В первой половине занятия ученики программируют отображение случайного вопроса из списка и сбор статистики ответов. Во второй половине занятия ученики заполняют приложение данными, тестируют и оформляют результат своей работы в коллаж. Примечание. В зависимости от уровня группы данный урок может использоваться как для доработки базового функционала, так и для совершенствования приложения и формированию навыка презентации результата работы. Сформулируйте два пожелания по доработке приложения, которые пришли от центра «Человек мира». Требуется изменить порядок отображения вопросов на случайный и дополнить приложение системой сбора статистики и подсчёта рейтинга пользователя. Обсудите изменение порядка вопросов. Напомните, что раньше вопросы задавались по порядку. Номер текущего вопроса по списку хранился в параметре cur_question, который был определён как свойство окна приложения. Теперь необходимость этого параметра отпала. Как задать случайный порядок вопросов? Можно ли отказаться от свойства окна приложения (с глобальной видимостью) и ввести обычную переменную (локально в функции)? В какой функции должен меняться номер вопроса? Как задать в переменной cur_question номер случайного вопроса? Какое максимальное и минимальное значение она может принимать? Продемонстрируйте, как озвученные изменения нужно встроить в общий код программы. Обсудите сбор статистики ответов на вопросы и подсчёт рейтинга пользователя. Продемонстрируйте ожидаемый функционал. Нужно ли производить изменения в оконном интерфейсе приложения или достаточно вывода данных в консоль? Какие параметры нужно отслеживать для печати статистики и рейтинга? Как определить (подсчитывать) текущее число заданных вопросов и правильных ответов? Будут ли эти значения изменяться только в одной функции? Как обеспечить доступ к этим параметрам из разных функций? Предложите определить данные счётчики как свойства объекта window (окна приложения), как раньше задавался номер текущего вопроса. В каких функциях нужно менять эти значения? Когда они должны печататься? Где будет рассчитываться рейтинг? Продемонстрируйте, как озвученные изменения нужно встроить в общий код программы. Обобщите сказанное и переходите к завершению проекта в VS Code.

Напомните разработчикам, что с завершением программы работа над проектом не заканчивается. Анонсируйте два важных этапа: тестирование и презентацию. 1. Начните с тестирования продукта. Сообщите, что количество и глубина этапов тестирования зависит от масштаба продукта и ресурсов, имеющихся у компании. В данном случае, механизм тестирования уже определил ведущий разработчик Костя, оформил как таблицу и передал для ознакомления. Тестирование будет состоять из трёх этапов. Сначала разработчик заполняет приложение расширенным набором данных по тематике заказчика. Затем он запускает приложение и работает с ним как пользователь. Есть ли случайный порядок вопросов? Перемешиваются ли варианты ответов? Отображается ли статистика и рейтинг, и соотносятся ли они с реальными результатами пользователя? Если самостоятельное тестирование прошло успешно, то проект может быть передан другому специалисту для проверки.

Методический комментарий. При онлайн занятиях третий этап тестирования не проводится. 2. Перейдите к обсуждению презентации результата работы. Презентация проделанной работы может проходить в разных форматах. Т. к. культурный центр работал с ProTeam в удалённом режиме, то презентация работы также будет виртуальной. Предложите разработчикам составить презентацию-отчёт в любом известном им сервисе. Если разработчик не умеем составлять презентации, то предложите ему простой и бесплатный сервис для составления коллажей.

3. Работа с текстовыми файлами(4 часа)

3.1. Основы работы с файлами

Цель — узнать, какими способами программа на Python может взаимодействовать с текстовыми файлами, и применить полученные знания для решения практических задач. На занятии разработчики осознают практическую необходимость использования файлов для долговременного хранения информации. Ученики научатся работать с текстовыми файлами в разных режимах, считывать и обрабатывать полученную информацию. Технический комментарий. Чтобы файлы проекта корректно отображались на разных операционных системах, файлам задана кодировка utf-8. Для правильной обработки данных вашим компьютером используйте дополнительный параметр encoding=’utf-8’.

Обратите внимание разработчиков, что заказ подразумевает долгосрочное хранение данных. При выключении приложения программа должна «запоминать» информацию, а не удалять.

Одним из мест хранения данных является текстовый файл.

Текстовый файл можно создать с помощью текстового редактора «Блокнот» или среды разработки VS Code. С помощью схем объясните, что файл должен находиться в одной папке с программой. Сама программа может использовать файл по-разному: как источник информации, как хранилище данных или и то, и другое. Рассмотрите различные режимы доступа к файлу и их атрибуты. Продемонстрируйте, что для того, чтобы прочитать информацию, нужно открыть файл в режиме чтения, то есть указав атрибут “r”.

«Начнём с получения информации из файла, то есть с его чтения. Для того, чтобы прочитать файл, нужно знать следующие команды: Для открытия файла используется конструкция с оператором with. После оператора нужно указать имя файла и атрибут доступа к нему. Это делается в функции open(“notes.txt”, “r”). После этого указывается название переменной (например, file), которая будет хранить ссылку на него. Итого получается строка с командой: with open(“notes.txt”, “r”) as file: В блоке после команды описываются действия с файлом. Читать информацию из файла можно по-разному. Например, можно прочитать всю информацию сразу. Для этого достаточно написать file.read(). Также можно прочитать только часть информации. Для этого методу read() нужно передать как параметр количество символов чтения, например, read(10). Есть и другие способы чтения, мы обсудим их позже. После окончания блока оператора with файл закрывается автоматически». После обсуждения методов для чтения рассмотрите примеры. Аналогично обсудите запись информации в файл с помощью атрибута “w”.

Техническое примечание. Перейдите к записи информации в файл. Заострите внимание учеников на том, что атрибут “w” для записи данных удаляет прошлое содержимое файла и устанавливает курсор на начало пустого документа. Для дозаписи информации к уже имеющейся используется атрибут “a”.

Рассмотрите пример с построчным чтением файла. Покажите, что чтение файла целиком зачастую неудобно или приводит к увеличению времени работы программы. Анонсируйте, что в следующем блоке заданий есть специальная задача на сравнение времени работы программы с построчным чтением и обработкой информации и с чтением файла целиком.
Рассмотрите реальную задачу. Дан файл с данными успеваемости учеников класса по информатике. В одной строке файла хранится фамилия и имя ученика и оценка. Нужно проанализировать данные и понять, какой в классе средний балл и составить список отличников. Для построчного анализа данных удобно использовать построчное считывание в цикле for. Полученную строку нужно разбить на подстроки с помощью метода split. Предложите разработчикам объединить полученные данные в экземпляры класса Pupil. Хранение информации в списке не очень удобно, т. к. уже через несколько минут будет тяжело вспомнить, что и под каким номером хранится в data. Обратите внимание разработчиков, что в данной задаче достаточно создать только один экземпляр класса. Среднюю оценку и список отличников можно накапливать в заранее подготовленных переменных (на каждой итерации будет пополняться сумма оценок класса, число учеников класса и будет производиться анализ, является ли данный ученик отличником).

3.2. Приложение «Умные заметки». Ч. 1

Цель — применить знания и умения по хранению данных в файлах и разработке приложений на PyQT для создания программы «Умные заметки». На занятии ученики приступят к разработке «Умных заметок» в среде программирования VS Code. Работа над приложением будет идти три урока. На данном уроке учащиеся спроектируют интерфейс «Умных заметок», придумают структуру для хранения заметок с тегами и запишут первую заметку в файл с расширением json.
Технический комментарий. Чтобы файлы проекта корректно отображались на разных операционных системах, файлам задана кодировка utf-8. Для правильной обработки данных вашим компьютером используйте дополнительный параметр encoding=’utf-8’.
Поприветствуйте разработчиков. Напомните, что в прошлый раз была разобрана теория для долговременного хранения данных. Спросите, является ли хранение информации о заметке в текстовом файле (с расширением .txt) оптимальным? Если нет, то какой инструмент мог бы оптимизировать работу с данными?
Рассмотрев разные варианты, придите к необходимости использовать файл с готовой внутренней структурой — json файл.
Сформулируйте цель рабочего дня и анонсируйте его содержание. В первой половины будет запрограммирован интерфейс приложения «Умные заметки», а во второй — разобрано и внедрено хранение заметок в json файле.
Вернитесь к обсуждению «Умных заметок». Обсудите первую задачу этого задания: создать интерфейс приложения.
Рассмотрите mind map проекта, составленную ведущим разработчиком Костей. Перечислите виджеты, необходимые для представления этого функционала. Уже по mind map можно заметить похожие элементы интерфейса, для которых подойдут одни и те же типы виджетов. Разберите новые типы виджетов для полей ввода текста (поле-строка — QLineEdit, большое поле — QTextEdit, кликабельный список — QListWidget).
Продемонстрируйте возможный вариант, похожий на словарь словарей. Покажите, что такая структура заметок довольно понятна в использовании. Обсудите, тяжело ли хранить такой словарь в файле? На самом деле, нет, потому что именно так устроены файлы с расширением json, которые многие программисты используют для хранения данных. У json файлов есть методы, позволяющие в одну строку прочитать и записать данные, главное, придерживаться структуры. В целом json файл

можно воспринимать как текстовый файл, со строго определённой структурой. Продемонстрируйте методы для чтения и записи информации в json файлы. Заострите внимание на некоторых полезных параметрах, которые пригодятся позже. Например, sort_keys, позволяющий отсортировать ключи json файла по алфавиту. Он будет крайне полезен при добавлении новых заметок. Также отметьте, что создать json файл можно через VS Code, создав в папке проекта новый пустой файл с расширением json. Перейдите к решению текущих задач с помощью json файлов. Предложите следующий способ задания структуры заметок в файле: описать словарь словарей notes в программе, задать в notes одну заметку-приветствие и записать notes в json файл.

Организуйте работу VS Code. Предложите ученикам открыть среду разработки и выполнить вторую задачу задания «VSC. Приложение “Умные заметки”». Задание представляет собой организацию хранения заметок в json файле и их представления в виджетах приложения.

3.3. Приложение «Умные заметки». Ч. 2

Цель — применить знания и умения по хранению данных в файлах и разработке приложений на PyQT для создания программы «Умные заметки». На занятии ученики продолжают разрабатывать «Умные заметки» в среде программирования VS Code. В первой половине занятия учащиеся запрограммируют создание, удаление и сохранение заметок. Во второй половине будут добавлены возможности добавления тега к заметке, его удаление и поиск заметок по указанному тегу. Технический комментарий. Чтобы файлы проекта корректно отображались на разных операционных системах, файлам задана кодировка utf-8. Для правильной обработки данных вашим компьютером используйте дополнительный параметр encoding=’utf-8’.

Поприветствуйте разработчиков. Напомните, что в прошлый раз был создан интерфейс «Умных заметок». Также разработчики создали json файл для хранения данных, записали в него первую заметку и отобразили информацию о ней в интерфейсе. Сегодня разработчиками будет запрограммирована работа с набором заметок и их тегами. Для этого необходимо создать шесть функций: три — для создания, удаления и сохранения заметок и три — для добавления и удаления тегов и поиска по тегу. Перейдите к рассмотрению функций для работы с заметками. Покажите, что при нажатии на «Создать заметку» должна создаваться пустая заметка. За работу с текстом и тегами отвечают другие элементы интерфейса. Название новой заметки должно отобразиться в общем списке-виджете list_notes, следовательно, нужно запросить название у пользователя. Чтобы не перегружать приложение ещё одним виджетом QLineEdit, предложите сделать ввод названия в QInputDialog (отдельное всплывающее окно с полем ввода). Он похож на изученный ранее виджет QMessageBox, который показывал сообщение в отдельном окне. Продемонстрируйте метод getText() для создания окна виджета и считывания строки текста. Покажите, что заголовок окна и комментарий к тексту задаются как параметры. Подведите итог сказанному. Покажите предполагаемый вид функции add_note. Обратите внимание, что в презентации имеется скрытый от демонстрации слайд с полным текстом add_note. Используйте его при работе со слабыми учениками. Техническое примечание. Рекомендуйте разработчикам выводить в консоль промежуточные результаты работы программы. Например, при добавлении новой заметки можно не только показать её в виджетах, но и напечатать полный словарь notes (она должна оказаться и там). Сравнение выводов в консоль и представлений в интерфейсе поможет избежать ошибок. Аналогично рассмотрите функции del_note для удаления заметки и save_note для сохранения заметки. Техническое примечание. Обратите внимание, что функции сохранения и удаления вносят

изменения и в json файл. Соответственно, после внесения изменений словарь notes нужно перезаписать в файл. После обсуждения сформулируйте задачи для программирования в первой половине рабочего дня и переходите к программированию.
Вновь организуйте разбор новой темы с использованием презентации. Сформулируйте задачи на вторую половину занятия. Покажите, что для завершения приложения осталось запрограммировать поиск по тегам. Перейдите к обсуждению функций. Функции-обработчики add_tag и del_tag аналогичны функциям для заметок. Наибольшее внимание уделите разбору поиска заметок по тегу (search_tag). Разбейте задачу по программированию поиска на две подзадачи: во-первых, поиск заметок и отображение результата; во-вторых, сброс результатов поиска.
Предложите разработчикам протестировать получившееся приложение. Не требуйте глубокого тестирования (подробно оно будет разобрано дальше на курсе), достаточно ответить на вопросы: Все элементы интерфейса активны? Корректно ли создаётся заметка? А удаляется? Можно ли установить заметке тег? А несколько тегов? Получается ли искать заметки по введённому тегу? Удаётся ли сбросить результаты поиска?

3.4. Приложение «Умные заметки». Ч. 3

Цель — оценить оптимальность и удобство работы с данными программы «Умные заметки» и сравнить текстовые файлы и json файлы.. На занятии ученики завершают разработку приложения «Умные заметки». В первой половине урока ученики обобщают знания по работе с файлами и выполняют проверочную работу на платформе. Во второй половине урока ученики организовывают хранение «Умных заметок» в текстовых файлах. Технический комментарий. Чтобы файлы проекта корректно отображались на разных операционных системах, файлам задана кодировка utf-8. Для правильной обработки данных вашим компьютером используйте дополнительный параметр encoding=’utf-8’.
Поприветствуйте разработчиков. Напомните, что на прошлом занятии была завершена основная работа над «Умными заметками». Анонсируйте содержание рабочего дня. Сообщите, что по желанию заказчика необходимо определить, является ли хранение заметок в json файлах оптимальным. Удобно ли пользоваться json заметками на компьютере без приложения? Легко ли обмениваться своими мыслями с другими учеными, печатая заметки или посылая их на электронную почту?
Организуйте обсуждение с использованием презентации. Сравните структуры json файла и текстового файла. В обсуждении придите к проблеме: в текстовых файлах нет автоматического чтения и распознавания данных. Чтобы получить список заметок notes, как в основной версии приложения, заметке нужно дополнить построчное чтение файла парсингом — выделением частиц информации из общего блока данных. Отдельно обсудите способы хранения заметок в текстовых файлах.
В результате обсуждения остановитесь на программировании демо-версии «Умных заметок» с хранением данных по типу «одна заметка — 1 файл».
«Начнём с того, как изменится хранение данных. Представим, что приложение работает только с одним файлом, в котором хранится одна заметка. Структура файла представлена на слайде. Как считать данные — название, текст и теги — из такого файла? (Ответы учеников). Верно, в целом, работу приложения можно описать следующим алгоритмом. Аналогично прошлой версии “Умных заметок” создаётся структура — список note — для хранения данных о заметке. Затем файл открывается на чтение и три строки файла записываются в note: 0 элемент — название, 1 элемент — текст заметки, 2 элемент — теги. Обратите внимание, что каждый элемент завершается символом перехода на новую строчку. Его нужно удалить. Теперь с данными всё в порядке? (Ответы учеников). На самом деле, нет. Теги, находившиеся в одной строке, считались одной строкой.

Их нужно разделить. Для этого алгоритм нужно усовершенствовать: Вернёмся к изначальной задаче. Что, если программа работает не с одной заметкой, а с несколькими? Как организовать перебор файлов с заметками? (Ответы учеников). Способов может быть несколько. Рассмотрим простейший. Пусть файлы с заметками имеют однотипные перечисляемые названия. Например, 0.txt, 1.txt или note0.txt, note1.txt и т. д. Тогда имя файла для чтения будем получать в цикле сложением строк. Полученные данные будем добавлять в список notes. Цикл будет выполняться до тех пор, пока получается открывать файлы с указанными именами. Какая конструкция поможет предпринять попытку открыть файл, а в случае неудачи (ошибки IOError) завершит работу цикла? (Ответы учеников). Да, чтение данных будет происходить в обработчике исключений try… except. Как только программе не удастся открыть файл для чтения, цикл завершится». Аналогично обсудите изменения в функционале приложения. Сформулируйте задачи для следующего этапа рабочего дня и переходите к программированию.
Технический комментарий. Используйте параметр encoding=’utf-8’ при чтении файла. Такая кодировка (Unicode) гарантирует, что данные текстового файла корректно отобразятся в виджетах на любых платформах. Иначе при открытии текстового файла на Windows система может попытаться прочитать его в кодировке Win-1251 (ansi).
Подведите итоги работы в VS Code. Спросите разработчиков, какую версию приложения они находят более удобной. Сформулируйте достоинства и недостатки json файлов и текстовых файлов. Попросите привести примеры данных, которые удобно хранить каждым из способов.

4. Автоматическая обработка изображений(4 часа)

4.1. Основы обработки изображений

Цель — изучить и применить на практике возможности библиотеки PIL для программной обработки изображений. На занятии разработчики осознают практическую необходимость обработки графических файлов инструментами Python. Ученики научатся открывать, редактировать и сохранять изображения и использовать полученные знания в решении сюжетных задач.
Продемонстрируйте разработчикам слайд с ожидаемым интерфейсом программы. Отметьте, что оно работает с графическими объектами. Исследуйте элементы интерфейса и перечислите элементы функционала приложения.
С помощью презентации расскажите о способах обработки изображений с помощью PIL. Отметьте, что инструменты библиотеки работают с объектами растровой графики. Продемонстрируйте, что важными частями PIL являются модули: Image и ImageFilter. При загрузке и обработке изображений программист работает с объектами типа Image. Перечислите поля класса Image, соответствующие параметрам изображения.
Технический комментарий. На этом занятии ученики работают только с картинками, лежащими в папке проекта. Со следующего занятия при переходе на проект «Фоторедактор» речь будет идти о произвольной папке компьютера. Обсудите задачу с загрузкой картинки, её отображением в отдельном окне и печати её параметров в консоли. Затем перечислите команды для обработки изображения. Обратите внимание разработчиков, что некоторые методы используют константы.
Обсудите ещё одну задачу с обработкой фотографии, требующей поворота и чёрно-белой цветовой палитры. Обратите внимание, что результат работы желательно сохранять как новый файл. Он создаётся автоматически при попытке сохранить картинку как файл с несуществующим именем.

для удобной обработки нескольких фотографий стоит ввести в работу собственный класс. Действительно, линейная обработка уже одного изображения несколькими фильтрами выглядит объёмно. Таким образом, разработчикам нужно создать класс ImageEditor с возможностью загрузки, обработки и сохранения его экземпляров. Продемонстрируйте наполнение класса на mind map. Сделайте техническое замечание к блоку «Загрузить фотографию» и продемонстрируйте альтернативный способ открытия картинки из папки проекта. Особенностью данного способа является обработка исключения, связанного с отсутствием файла с таким именем (иначе программа аварийно завершает работу). Обсудите схему класса ImageEditor. Предложите разработчикам заполнить пропуски с помощью предыдущих слайдов. Затем обсудите две задачи с применением и доработкой класса. Задайте вопросы на понимание программы и переходите к заданию на программирование.

4.2. Приложение Easy Editor. Ч. 1

Цель — запрограммировать интерфейс приложения с возможностью отображения списка графических файлов. В первой половине урока учащиеся программируют интерфейс приложения Easy Editor. Во второй половине — дополняют класс методами для загрузки и отображения списка изображений для обработки. Техническое примечание. Вся работа над приложением Easy Editor организована в одном задании в VS Code.

С помощью презентации разберите устройство интерфейса приложения Easy Editor. Отметьте, что отображение превью картинки возможно только после загрузки списка графических файлов из папки компьютера, поэтому для начала достаточно надписи «Картинка». Какие виджеты есть на скриншоте ожидаемого интерфейса приложения? Каким виджетам нужно дополнительно задать свойства, например, размер или надпись (не забывайте, что окно приложения — это тоже виджет)? Как разместить виджеты по лэйаутам? Опишите минимум два способа размещения. Как запустить и оценить интерфейс приложения? Разработчики уже имеют опыт разработки интерфейсов, поэтому обсуждать все детали окна приложения не нужно. Подведите итоги обсуждения и переходите к работе в VS Code.

«Как отобразить названия картинок из папки компьютера (произвольной) в интерфейсе приложения?)». Перейдите к схеме связывающей действия, которые должен предпринять пользователь, и исполнение команд компьютером. Отметьте, что для реализации некоторых возможностей, например, для возможности выбора папки, нужно изучить новые команды. Расскажите, как запрограммировать выбор рабочей папки. Виджет PyQt5 QFileDialog может вызвать окно «Проводника» и позволить пользователю выбрать директорию. Метод getExistingDirectory() возвращает в программу путь к выбранной папке. Загрузить в программу имена всех файлов папки можно с помощью команды модуля os. Кратко поясните, что os — это встроенный модуль стандартной библиотеки Python. Для работы потребуются команды для обращения к файлам и папкам по пути.

Продемонстрируйте функцию listdir() и отметьте необходимость отсортировать и оставить только графические файлы. Отбор файлов по расширениям можно обернуть в отдельную функцию filter(). Обратите внимание разработчиков: получить часть имени файла с расширением можно как с помощью среза, так и функцией endswith(). Технический комментарий. В Python есть стандартная функция filter, но написание собственной функции отбора файлов по расширениям является хорошей учебной задачей. Такая задача помогает лучше понять, когда функцию необходимо определять как метод класса, а когда нет.

Озвученные технические решения можно объединить в функции-обработчике showFilenamesList(). Процесс отображения готового списка строк в виджете аналогичен задаче из прошлого проекта «Умные заметки». Покажите внедрение описанных функций в программу. Обратите внимание разработчиков на необходимость объявления переменной workdir как глобальной.

В противном случае после завершения работы функции-обработчика путь к текущей рабочей папке будет потерян.
Технический комментарий. При желании на следующем занятии переменную workdir можно будет сделать полем класса ImageProcessor. В рамках базовой версии программы workdir останется глобальной переменной. Сформулируйте задачу следующего этапа и переходите к программированию.

4.3. Приложение Easy Editor. Ч. 2

Цель — запрограммировать отображение картинки из списка изображений, переключение между картинками и наложение чёрно-белого фильтра. В первой половине урока учащиеся приступают к созданию класса обработки фотографий ImageProcessor и реализуют отображение выбранной картинки. Во второй половине — дополняют ImageProcessor первым методом обработки изображения и сохранением результата. Техническое примечание. Вся работа над приложением Easy Editor организована в одном задании в VS Code.
Может показаться, что сегодняшний объём работы гораздо меньше совокупности задач прошлого рабочего дня. Отметьте, что задача отображения превью картинки сложнее, чем кажется, поскольку один и тот же набор функций будет показывать как оригиналы картинок (при переключении между именами файлов в виджете), так и их обработанные копии (при нажатии на кнопки «Лево», «Право», «Резкость» и «Ч/б»). Сообщите, что если разработчикам удастся уложиться в отведённое время, то уже сегодня их приложения смогут накладывать на фотографии чёрно-белый фильтр. Покажите задачи для реализации в течение текущего рабочего дня. Затем сформулируйте его цель и содержание.
Методический комментарий. На первом занятии модуля рассматривались только картинки, лежащие в папке проекта. Для них не требовалось знание полного пути к папке или самой картинке. Поскольку приложение Easy Editor работает с произвольной папкой компьютера, к изображениям необходимо обращаться по полному пути. Поэтому на этапе повторения можно отметить, что метод open() работает как с кратким путем к файлу (если он лежит в папке проекта), так и с полным (если он размещён в другой папке). Возможно, найдутся ученики, которые плохо ориентируются в иерархии директорий компьютера. В этом случае выберите на своём компьютере папку и продемонстрируйте, как соотносится её расположение в системе и полный путь к ней. Запишите полный путь на доске и обращайтесь к этому примеру при последующих затруднениях.
С этого этапа начинается работа над классом ImageProcessor. Он объединит в себе и данные по текущей картинке и методы для загрузки, отображения, обработки и сохранения картинки. Продемонстрируйте общую схему класса и перейдите к обсуждению отображения превью картинки в приложении. С помощью схемы разберите устройство метода loadImage(), загружающего картинки в программу. Для загрузки объекта Image (и сохранения его в соответствующее поле экземпляра ImageProcessor) необходимо сформировать полный путь к файлу. Это легко сделать, зная путь к рабочей папке (глобальная переменная workdir) и имя файла. Получение имени файла из виджета-списка разбирается ниже. Введите новую функцию модуля os раздела path — join(). Покажите её работу на схеме. Затем сопоставьте схемам эталонный код метода loadImage(). Технический комментарий. При желании на этом занятии переменную workdir можно сделать полем класса ImageProcessor. В рамках базовой версии программы workdir остаётся глобальной переменной. Работу следующего метода showImage() представьте в форме мастер-класса от разработчика Кости. Для красивого отображения картинки в интерфейсе приложения требуется введение объекта вида QPixmap. Проблематизация класса QPixmap не входит в тематику данного занятия. Тем не менее, можно отметить, что благодаря QPixmap изображение произвольного размера можно красиво «вписать» в окно приложения. Чтобы

применить написанный функционал в программе, создадим функцию-обработчик клика по элементу списка имён картинок showChosenImage(). Методы для работы с виджетом-списком уже знакомы разработчикам по проекту «Умные заметки». Опишите работу функции с помощью блок-схемы и покажите эталонный код. Затем продемонстрируйте, как описанный функционал можно внедрить в проект. Подведите итоги обсуждения, сформулируйте задачу следующего этапа и переходите к работе в VS Code.

Перейдите к программированию обработки изображений. Напомните разработчикам, что при изучении обработки изображений с помощью PIL они программировали класс ImageEditor. Обработка в ImageProcessor будет устроена так же, только к ней добавится метод, сохраняющий отредактированную копию в подпапку Modified рабочей папки. Таким образом, требуется запрограммировать метод saveImage(), метод do_bw() (наложение чёрно-белого фильтра). Обработка нажатия на кнопку «Ч/б» тоже будет производиться методом do_bw(). По очереди разберите устройство новых функций и методов. Особое внимание уделите методу saveImage() и новым командам модуля os. Напомните, что по техническому заданию требуется сохранять картинки в подпапку Modified выбранной рабочей директории. Если эта папка отсутствует, её нужно создать. Имена сохраняемых файлов должны совпадать с изначальными. Продемонстрируйте, как дополнить код проекта написанными методами, сформулируйте задачи следующего этапа и переходите к программированию.

4.4. Приложение Easy Editor. Ч. 3

Цель — запрограммировать функционал приложения, связанный с обработкой графических файлов. В первой половине урока учащиеся программируют обработку фотографий приложением Easy Editor. На этом базовая версия приложения готова. Во второй половине дети знакомятся с профессией «тестировщик программного обеспечения» и тестируют приложение с помощью собственных тест-кейсов. Техническое примечание. Вся работа над приложением Easy Editor организована в одном задании в VS Code.

С помощью презентации организуйте подтверждение квалификации разработчиков перед началом работы. В этот раз оно проходит по темам: «Модуль os» и «Обработка изображений средствами PIL». Методический комментарий. Уже на этапе повторения ученикам можно показать новый способ применения метода Image.transpose(). С константой FLIP_LEFT_RIGHT он отображает картинку зеркально слева направо.

Откройте следующий слайд и конкретизируйте рабочие задачи данного этапа. Покажите, что большая часть приложения уже написана, осталось дополнить класс ImageProcessor методами do_left(), do_right(), do_flip(), do_sharpen() (повернуть изображение влево/вправо, отобразить его зеркально и навести резкость). При этом один метод обработки уже реализован — do_bw(). На самом деле, новые методы будут отличаться от него только командами библиотеки PIL. На примере метода do_flip() (отобразить изображение зеркально) напомните устройство таких функций-обработчиков. Объясните, что отзеркаливание изображения происходит с помощью знакомого метода transpose (использовался при повороте изображений) и новой константы объекта Image — FLIP_LEFT_RIGHT. Остальные методы обработки хорошо знакомы разработчикам по тренировочному классу ImageEditor, который программировался при знакомстве с PIL. Их нужно запрограммировать самостоятельно. Покажите, как написанные методы можно внедрить в код проекта. Подведите итоги обсуждения, сформулируйте задачу по завершению приложения Easy Editor и переходите к работе в VS Code.

При разработке большого проекта становится трудно заметить недочёты и ошибки. Также зачастую сложно предсказать все варианты действий пользователя и корректно обработать их в коде. Это чревато неожиданными поломками продукта уже после его релиза (публикации). Для избежания таких недоразумений каждый проект обязательно проходит этап тестирования. Для этого даже существует отдельная профессия — тестировщик, задача которого — смоделировать все ситуации, которые могут возникнуть при использовании ПО и убедиться, что программа будет корректно себя вести в этих условиях
Расскажите, что для тестирования программного обеспечения составляются тест-кейсы — документы, содержащие информацию о проверяемой функции, действиях пользователя и поведении программы. Предложите разработчикам составить три тест-кейса для приложения Easy Editor и протестировать его по ним.

5. Продвинутая разработка игр на Pygame(9 часов)

5.1. Основы создания игр

Цель — вспомнить и применить на практике основы создания игр с помощью библиотеки PyGame. В первой половине занятия ученики вспоминают понятия «спрайт», «сцена», «игровой цикл» и применяют их в решении тренировочных задач. Во второй половине занятия учащиеся вспоминают обработку событий клавиатуры и работают над мини-проектом «Догонялки».
Примечание. Некоторые ученики могут быть уже знакомы с библиотекой PyGame. Тем не менее, модуль рекомендуется начать с основ. Опытным ученикам представьте первый урок как тренинг, который сделает работу над предстоящими проектами эффективнее.
Обсудите с разработчиками, что им известно об играх и их создании: в какие игры они играют на домашних компьютерах? Какие жанры игр им известны? Знают ли они, чем геймдизайнер отличается от разработчика игр? Отметьте, что в предстоящей работе они будут заниматься именно разработкой игр и наладкой игрового цикла. Кратко поясните, что игровой цикл — это цикл, на каждом шаге (в каждом «кадре») которого происходит обработка событий (как внешнего мира, так и самой игры, отрисовка персонажей и сцены, отсчёт времени). Продемонстрируйте слайд с ожидаемым интерфейсом интерактивной игры «Лабиринт». Сообщите разработчикам, что для создания такой игры, нужно уверенно владеть основами pygame. Для этого в течение сегодняшнего рабочего дня будет реализован тренировочный проект «Догонялки». Сформулируйте цель рабочего дня и анонсируйте его содержание.
Предложите разработчикам изучить внешний вид игры «Догонялки» и перечислить элементы функционала приложения. Подробно обсудите наполнение игры с помощью mind map. Например, базовыми компонентами mind map могут быть «Сцена» (внешнее представление игры) и «Функционал». Порассуждайте, какие компоненты библиотеки pygame следует изучить в первую очередь? Перейдите к разбору наполнения pygame. Сообщите, что как и многие библиотеки, pygame имеет иерархическую структуру и состоит из множества модулей. Приведите примеры модулей: transform и event. Обсудите с разработчиками первый шаг создания игры «Догонялки» — создание заготовки с фоном. Напомните, что используемые картинки обязательно должны лежать в папке проекта. «В Pygame точка начала отсчёта располагается в левом верхнем углу окна. Размер окна определяет разработчик». Продемонстрируйте разработчикам методы для создания окна игры и фона в виде картинки. Отметьте, что размеры картинки могут быть адаптированы под размер окна. Обратите внимание, что даже простейший код с созданием окна игры и установкой фона не будет работать корректно без игрового цикла (окно откроется и

сразу закроется)! Придите к необходимости запрограммировать простейший игровой цикл, отображающий окно с фоном до тех пор, пока пользователь не нажмёт на кнопку «Закрыть окно» (красный крестик). Обсудите синтаксис необходимых методов из модуля event библиотеки pygame. Уточните, что кадры, возникающие на экране с каждым шагом цикла, нужно обновлять с помощью команды display.update(). Отметьте, что спрайты в форме картинок (например, игровые персонажи) можно также адаптировать под нужный размер (например, 100x100 пикселей) и располагать в произвольной точке окна по координатам. Объедините разобранные фрагменты в программу, подведите итоги обсуждения и переходите к работе в VS Code.
Перейдите к обсуждению следующего этапа создания игры — задание FPS и управление спрайтами с помощью клавиатуры. Продемонстрируйте слайд со сравнением работы двух игр. «Отмечу, что скорость перемещения спрайтов одинаковая. Но почему в игре, представленной справа, спрайт перемещается более плавно и быстро? (Ответы учеников). На самом деле, это связано в разной частотой обновления кадров игры в секунду». Расскажите разработчикам о важной характеристике игры — частоте кадров, отображаемых за одну секунду. Также принято сокращение FPS (frames per seconds). При прочих равных, высокая частота обновлений выглядит лучше, но требует большей производительности компьютера. Комфортная для человеческих глаз частота кадров для игр — 50–60 кадров/сек. Для задания частоты кадров требуется введение нового объекта Clock() — «часов», отслеживающих время. Желаемое FPS задаётся в игровом цикле методом tick(). Технический комментарий. На самом деле, метод tick() не задаёт именно FPS. В действительности в каждом кадре секунда будет разделена на 60, и следующий кадр появится с задержкой на 1/60 секунды. Перейдите к обсуждению перемещения спрайта с помощью клавиатуры. Оказывается, обработка всех возможных событий «в лоб» будет слишком громоздкой. Особенно трудно будет запрограммировать ситуацию, в которой при зажатии клавиши со стрелкой спрайт будет продолжать двигаться. Покажите синтаксис метода get_pressed(), возвращающего набор нажатых клавиш. Тогда обработка событий клавиатуры сведётся к анализу: есть ли в полученной в данном кадре структуре определённая клавиша или нет. Технический комментарий. На самом деле, команда keys.get_pressed() возвращает кортеж из нулей и единиц. При этом каждое число всегда соответствует одной и той же клавише, например, «стрелка вправо» — 275. Но помнить эти числа наизусть не обязательно, т. к. к клавишам можно обращаться по константам, например, K_UP. Отметьте, что перемещение спрайта должно ограничиваться размерами экрана. Подведите итоги обсуждения и переходите к работе в VS Code.

5.2. Игра «Лабиринт». Часть 1

Цель — запрограммировать класс GameSprite и разместить его экземпляры (спрайты) на сцене с помощью игрового цикла. В первой половине урока учащиеся планируют работу над проектом и создают заготовку для игры с фоном и музыкой. Во второй половине — программируют класс GameSprite, создают экземпляры класса для спрайтов и располагают их на сцене.
Техническое примечание. Вся работа над приложением «Лабиринт» организована в одном задании в VS Code.
Методическая рекомендация. Заранее сохраните себе на компьютер эталонное решение проекта. Продемонстрируете его во время обсуждения ожидаемого вида игры, чтобы разработчики увидели не только графику, но и звуковое сопровождение.
Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Нашему отделу поступил новый заказ на создание игры “Лабиринт”. Работа над этим проектом будет длиться три рабочих дня. Коллеги из других отделов уже продумали логику игры и нарисовали картинки для фона и спрайтов. Вам же предстоит запрограммировать функционал игры». Продемонстрируйте разработчикам слайд с техническим заданием и ожидаемым видом игры. Отметьте,

что в игре присутствуют новые механики, например, разное управление спрайтами (один персонаж — с клавиатуры, а другой — автоматически) и звуковое сопровождение. Вместе с разработчиками попытайтесь составить mind map игры. Обсудите какие задачи должны быть реализованы в первый рабочий день, и составьте по ним чек-лист. Этот этап рекомендуется выполнять за компьютерами в соответствующих заданиях на платформе. По мере выполнения заданий демонстрируйте mind map и чек-лист задач, предложенные разработчиком Костей. Планируя проект, укажите на неизвестные особенности игры, например, необходимость обработки столкновений спрайтов. Есть ли в pygame готовый инструмент? Сформулируйте цель рабочего дня и анонсируйте его содержание.

С помощью презентации расскажите о модулях работы с музыкой. Отметьте, что для добавления фоновой музыки и добавления звуковых эффектов используются разные модули библиотеки pygame и разные методы соответственно. Обсудите, как внедрить в игровой цикл проигрывание фоновой музыки. Уточните, что файлы со звуками должны иметь форматы ogg или wav, так как их поддерживают все операционные системы, и находиться в папке проекта. Технический комментарий. Фоновую музыку можно задать и формате .mp3, но при запуске такой программы на операционной системе Linux возможны проблемы. Подведите итоги обсуждения и переходите к работе в VS Code.

Методический комментарий. На данном уроке крайне важно добиться осознанного использования наследования для создания игрового класса. На двух следующих уроках дети продолжат работать с этой темой и будут создавать новые классы, например, Enemy как наследника от GameSprite. С учетом данной особенности и выбран невысокий темп урока. Напомните разработчикам, что по техническому заданию игрок проигрывает, если сталкивается с врагом или со стенами лабиринта. Снова продемонстрируйте игру: «Как запрограммировать столкновение игрока со спрайтом-врагом? (Ответы учеников). Возможное решение — сравнивать по координатам текущее местоположение спрайтов игрока и врага. Догадайтесь, в чём недостаток такого подхода? (Ответы учеников). Действительно, обработка столкновения со стенами будет очень громоздкой. Требуется другое решение, и в pygame оно есть — инструменты готового класса Sprite!» Объясните разработчикам, что фундамент для создания спрайтов уже реализован в классе Sprite, но создавать персонажей как экземпляры Sprite не получится: одни методы универсальны для всех спрайтов, а другие — реализуются по-своему в зависимости от типа спрайта. В обсуждении сформулируйте выход из затруднения: можно взять из готового класса Sprite полезные свойства и методы и дополнить их новыми деталями с помощью наследования. Для этого можно создать класс-наследник GameSprite готового класса Sprite и в нём уточнить свойства и методы под особенности «Лабиринта». Обсудите схему класса GameSprite. Предложите разработчикам заполнить пропуски с помощью предыдущих слайдов. Обсудите внедрение класса в готовый код программы. Подведите итоги обсуждения и переходите к работе в VS Code.

Организуйте работу в VS Code. В задании разработчикам требуется реализовать класс GameSprite и создать экземпляры этого класса для спрайтов: игрок, враг и сокровище. Разместить спрайты на сцене.

5.3. Игра «Лабиринт». Часть 2

Цель — запрограммировать классы Player и Enemy как наследники суперкласса GameSprite и реализовать в них управление спрайтами игрока и врага. В первой половине урока учащиеся планируют работу над проектом и создают класс Player с управлением игроком с клавиатуры. Во второй половине — программируют класс Enemy с автоматическим перемещением врага по сцене.

Выделите главную задачу — запрограммировать перемещение спрайтов. Важно отметить, что перемещение будет реализовано по-разному для двух типов спрайтов: игрок должен управляться клавишами, а враг — перемещаться автоматически. В связи с этим, у каждого типа спрайта должны быть дополнительные методы, которые присущи только ему. В обсуждении придите к необходимости создания двух классов-наследников от суперкласса GameSprite. Тогда в наследниках — Player и Enemy — нужно будет дописать лишь методы перемещения. Сформулируйте цель рабочего дня и анонсируйте его содержание.
Начните обсуждение с разбора взаимосвязи классов Sprite, GameSprite и Player. Разберите схему, где класс GameSprite наследуется от класса Sprite из библиотеки pygame, а класс Player наследуется от GameSprite. Получается, что фундаментальным является класс Sprite. Затем он уточняется под нужды игры «Лабиринт», и становится классом GameSprite. И в заключение GameSprite дополняется свойствами и методами, необходимыми конкретным типам спрайтов и возникают классы Player и Enemy. В качестве примера покажите, что класс-наследник Player дополняется только методом update(), в котором будет реализовано «перемещение» по сцене (на самом деле, обновление местоположения), а все остальные свойства и методы уже описаны в классе GameSprite. Разберите синтаксис команд метода update() класса Player. В методе должно быть реализовано изменение координат спрайта при нажатии на управляющие клавиши. Методический комментарий. Классы Player и Enemy настоятельно рекомендуется предложить для самостоятельной реализации. В ходе «мозговых штурмов» обсуждаются идеи и наполнение, а не разбирается синтаксис. Покажите, как ввести класс Player в код игры «Лабиринт». Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.
Перейдите к обсуждению класса Enemy для врага, охраняющего сокровище. Данный класс, также как и класс Player, дополняется только методом update(), но с другой реализацией — метод будет отвечать за автоматическое обновление положения спрайта на сцене. «Но тогда в обоих классах будет присутствовать метод update(). Не возникнет ли ошибки из-за одинаковых названий методов для в классах Player и Enemy?» Объясните разработчикам, что ошибки возникать не будет. Дело в том, что метод будет применяться к конкретному объекту, класс которого известен интерпретатору. Напомните, что и раньше разработчики использовали для разных типов объектов методы с одинаковыми названиями, например, show() для разных типов виджетов в PyQt. Разберите блок-схему, описывающую механизм автоматического перемещения спрайта-врага влево-вправо между двумя точками. Аналогично обсудите идеи разработчиков, а реализацию предложите выполнить самостоятельно.
Обсудите, как ввести класс Enemy в код игры «Лабиринт». Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

5.4. Игра «Лабиринт». Часть 3

Цель — запрограммировать класс Wall и условия победы и проигрыша в игре «Лабиринт». В первой половине урока учащиеся планируют работу над проектом, создают класс Wall и размещают его экземпляры на сцене. Во второй половине — программируют подписку на события, связанные с победой и проигрышем, и их обработку. Техническое примечание. Вся работа над приложением «Лабиринт» организована в одном задании в VS Code.
Покажите разработчикам техническое задание и ожидаемый вид игры, попросите назвать ещё нереализованные пункты в проекте. Обратите внимание разработчиков на то, как должны быть реализованы препятствия в игре. По техническому заданию стены должны быть заданы прямоугольниками, залитыми цветом. Подойдёт ли картинка такого прямоугольника? Или стоит исследовать возможности построения геометрических фигур с помощью pygame? Отметьте, что в перспективе объекту

Стена потребуется распознавание столкновений, реализованное в классе Sprite. Придите к необходимости запрограммировать класс Wall по аналогии с уже реализованным классом Sprite.
Напомните разработчикам, что на этапе обсуждения задач они пришли к выводу, что класс Wall должен быть создан как наследник суперкласса Sprite. От лица разработчика Кости сообщите, что для внешнего вида стены подойдёт объект типа Surface. Расскажите, что Surface представляет из себя объект типа поверхность. Она может создаваться и в качестве фона сцены (при этом можно регулировать прозрачность поверхностей), и как самостоятельный спрайт. Подробнее рассмотрите заполнение объекта Surface цветом. Для заливки объекта произвольным цветом удобно использовать цветовую палитру RGB (red, green, blue). Рассмотрите примеры получения различных цветов на слайдах или, если необходимо, перейдите по ссылке на RGB-калькулятор и продемонстрируйте смешивание цветов в нём. Обсудите, какие свойства и методы должны присутствовать в классе Wall. Предоставьте разработчикам возможность заполнить пропуски в коде. Подведите итоги обсуждения, покажите, как ввести класс Wall в игру, сформулируйте задачи следующего этапа и переходите к работе в VS Code.
Перейдите к обсуждению завершающего этапа игры — оформлению условий победы и проигрыша. Заметьте, что пользователь обязательно должен увидеть результат — выиграл он или проиграл. При этом окно игры закрывать не нужно, на случай если пользователь захочет пройти уровень заново (при наличии такого функционала). Рассмотрите блок-схему работы игрового цикла игры «Лабиринт» с переменными game (отвечает за работу игры как программы) и finish (отвечает за прохождение игры) и фрагмент возможного кода этого цикла. «Давайте вспомним, что является условием победы в игре? (Ответы учеников). Правильно, это момент, когда спрайт-игрок касается сокровища. А каковы условия проигрыша? (Ответы учеников). Это касание стены или столкновение с врагом». Рассмотрите синтаксис нового метода sprite.collide_rect() и напомните понятие «подписка на событие». В зависимости от исхода игры должны воспроизводиться разные звуки (звон монет или звук удара) и появляться соответствующие надписи «YOU WIN!» или «YOU LOSE!». Для отображения текста в pygame есть готовый модуль font. Кратко разберите команды нового модуля. Покажите, как новые условия и команды дополняют игровой цикл. Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

5.5. Игра «Шутер». Часть 1

Цель — создать заготовку игры «Шутер» с фоном, музыкой и спрайтом-игроком, управляемым с клавиатуры. В первой половине урока учащиеся планируют работу над проектом и создают заготовку для игры с фоном и музыкой. Во второй половине — программируют класс Player и создают управляемого с клавиатуры спрайта-игрока и располагают его на сцене.
Техническое примечание. Вся работа над игрой «Шутер» организована в одном задании в VS Code.
Методическая рекомендация. Заранее сохраните на свой компьютер эталонное решение проекта. Продемонстрируете его во время обсуждения ожидаемого вида игры, чтобы разработчики увидели не только графику, но и звуковое сопровождение.
Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Нашему отделу поступил новый заказ на создание игры “Шутер”. Как и в прошлый раз, геймдизайнеры продумали игровые сценарии, а художники подготовили картинки для фона и спрайтов. Вам же предстоит запрограммировать функционал игры». Продемонстрируйте слайд с техническим заданием и ожидаемым видом игры. Отметьте, что многие механики в игре уже известны разработчикам по прошлому проекту. Вместе с разработчиками составьте mind map игры. Обсудите, какие задачи должны быть реализованы в первый рабочий день, и составьте по ним чек-лист. Этот этап рекомендуется выполнять за компьютерами в соответствующих

заданиях на платформе. Затем продемонстрируйте mind map и чек-лист задач, предложенные разработчиком Костей. Планируя проект, укажите на неизвестные особенности игры, например, необходимость создания множества врагов и пуль и управление ими. Сформулируйте цель рабочего дня и анонсируйте его содержание.

Выделите задачу на первую половину рабочего дня — создание окна игры с картинкой и фоновой музыкой. Обсудите игровой цикл проекта «Шутер». Обратите внимание, что похожий цикл был запрограммирован для игры «Лабиринт», поэтому есть возможность использовать готовые фрагменты кода. Разберите схему и соответствующие им фрагменты кода. Сообщите разработчикам, что несмотря на то, что другие сотрудники ProTeam уже продумали внешний вид «Шутера», они тоже могут предложить свои идеи для фона, спрайтов и музыки. Уточните, что несмотря на важность следования техническому заданию, предоставление выбора прототипов для заказчиков всегда подчёркивает мастерство разработчика. «Настоящие разработчики в ответ на техническое задание готовы не только реализовать его, но и предложить свои идеи по доработке. Это бывает интересно не всем клиентам, но в случае с данным заказом свежие и оригинальные идеи будут только приветствоваться». Предложите быстрым разработчикам усовершенствовать внешний вид игры и подобрать свои картинки для заставки и персонажей. Используйте одни из предложенных или другие известные ресурсы с материалами, бесплатными для частного использования. Подведите итоги обсуждения и переходите к работе в VS Code.

Обсудите с разработчиками, какие типы спрайтов должны быть представлены в игре. Перечислите подробнее методы каждого спрайта и порассуждайте, требуется ли создание соответствующих классов. Пример. Спрайт-игрок управляется с клавиатуры и распознаёт столкновение с врагом. Управлять перемещением можно и со спрайтом-картинкой, но для столкновений удобно использовать метод из готового класса Sprite. Следовательно, стоит запрограммировать игрока как экземпляр собственного класса-наследника Sprite. Пример. Спрайт-счётчик не перемещается, надписи меняются при взаимодействии других спрайтов. Этого наверняка можно добиться просто перерисовкой текста. Следовательно, дополнительный класс не нужен. Можно обойтись готовыми объектами pygame. Придите к необходимости создать три класса: Player, Enemy и Bullet. Обратите внимание, что несмотря на отличия, у этих классов есть много общего. Например, внешний вид (картинка), позиционирование спрайта по координатам, отрисовка спрайта на сцене и др. Вместе с разработчиками подумайте над оптимизацией создания этих классов. Напомните, что раньше разработчики уже искали пути оптимизации в игре «Лабиринт». Это достигалось: созданием класса-наследника GameSprite от Sprite с необходимыми полями и методами (требования к GameSprite повторялись на этапе квалификации); созданием классов-наследников GameSprite для различных типов спрайтов. Предложите разработчикам воспользоваться наработками предыдущего проекта — классом GameSprite. Уточните, что использование собственных наработок — частая для программистов практика. Главное, не опускаться до бездумного копирования чужого кода и использовать только то, в чём хорошо разбираешься сам. Перейдите к обсуждению класса Player. Объект данного класса должен перемещаться при нажатии на клавиши со стрелками и стрелять по врагам при нажатии на пробел. Все методы, кроме создания объекта-пули, разработчикам уже известны, поэтому подробный разбор команд не требуется. Создание класса Bullet для реализации пуль отложим на следующие рабочие дни. Рассмотрите с разработчиками изменённый игровой цикл, обратите внимание, что его стоит дополнить новой переменной finish, которая отвечает за состояние игры, но не вызывает закрытие приложения. Обсудите внедрение класса в готовый код программы. Подведите итоги обсуждения и переходите к работе в VS Code.

5.6. Игра «Шутер». Часть 2

Цель — дополнить «Шутер» спрайтами-врагами и счётчиком упущенных врагов. В первой половине урока учащиеся планируют работу над проектом, создают класс Enemy и группу спрайтов-врагов. Во второй половине — создают счётчик для подсчёта упущенных врагов. Техническое примечание. Вся работа над игрой «Шутер» организована в одном задании в VS Code.

Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Сегодня мы продолжаем разработку игры “Шутер”. В прошлый раз мы спланировали работу над проектом и составили mind map и чек-лист. Также мы уже создали заготовку игры с фоном и персонажем, управляемым с клавиатуры. Определим задачи сегодняшнего рабочего дня и занесём их в чек-лист». Выделите главную задачу — дополнить игру спрайтами-врагами и создать счётчик пропущенных врагов. Напомните, что в прошлый раз было принято решение сделать врагов экземплярами класса Enemy (наследника класса GameSprite). Обсудите действия, которые должны выполнять враги игры «Шутер» и их программную реализацию. Например, автоматическое перемещение спрайта-врага вниз стоит реализовать в методе update() (враг, пролетевший мимо игрока, увеличивает счётчик упущенных врагов). Обратите внимание разработчиков на то, что спрайтов-врагов в их игре много и это требует дополнительных программных решений. Разберите несколько идей работы с однотипными спрайтами. Обсудите преимущества и недостатки каждого решения и придите к необходимости создания группы спрайтов. Перечислите задачи, которыми необходимо дополнить чек-лист. Сформулируйте цель рабочего дня и анонсируйте его содержание.

Напомните задачу первой половины рабочего дня — запрограммировать класс Enemy и группу спрайтов-врагов. Начните обсуждение с того, что класс Enemy, как и Player, является наследником класса GameSprite и дополняется только методом перемещения. Перемещение спрайтов-врагов происходит автоматически и выполняется в игровом цикле. Разберите устройство метода update(), который будет изменять координаты спрайта или группы спрайтов при смене кадров игрового цикла. Обратите внимание, что на сцене всегда находится 5 врагов. После того как враг достигает нижней части экрана, он же появляется в случайном месте наверху (то есть генерировать бесконечное число спрайтов и тратить мощность компьютера не нужно). Обсудите с разработчиками, как программно реализовать автоматическое перемещение спрайта врага сверху вниз и его исчезновение, если он дошёл до конца экрана, и появление в случайной точке наверху экрана. Продемонстрируйте, что работать с группой спрайтов очень удобно, во многом благодаря тому, что ею можно управлять как одним спрайтом. В частности, метод update() можно применять сразу к группе спрайтов, а не к каждому из них по очереди. Разберите с разработчиками синтаксис создания группы объектов и работу с ней. Покажите на схеме, как ввести класс Enemy в код игры «Шутер». Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

Организуйте работу в VS Code над игрой «Шутер». В задании разработчикам требуется запрограммировать класс-наследник Enemy, создать группу спрайтов-врагов и внести изменения в игровой цикл: на каждом шаге цикла враги должны перемещаться по экрану.

Выделите задачу на вторую половину рабочего дня — создание счётчика статистики пропущенных врагов. Объясните разработчикам, что счётчик должен увеличиваться на единицу с каждым пропущенным кораблём. Для реализации счётчика необходимо создать все сущности: переменную-накопитель числа пропущенных врагов; текстовый объект, отображающий значение переменной на экране. Обсудите с разработчиками, где в коде необходимо реализовать проверку того, достиг ли враг

нижней части окна. Предложите поместить данную проверку в метод класса Enemy. «Будет ли работать такая программа? (Ответы учеников). На самом деле, нет. Счётчик будет «висеть» на отметке 0, а в классе будет ошибка! В классе Enemy своя область видимости, он не может взаимодействовать с переменной lost. Есть ли в программировании возможность сделать переменную lost видимой в методе класс? (Ответы учеников). Да, такая возможность есть. Надо сделать lost глобальной переменной». Введите понятие «глобальная переменная» и разберите синтаксис создания таких переменных. Покажите, как внедрить полученные знания в код программы. Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

Отвлеките разработчиков от компьютеров и организуйте техническое собеседование по материалам «мозговых штурмов». Предложите разработчикам теоретическую документацию. Анонсируйте, что в следующий рабочий день разработчки продолжат программировать игру «Шутер» и научат корабль игрока стрелять по врагам!

5.7. Игра «Шутер». Часть 3

Цель — дополнить «Шутер» спрайтами-пулями и запрограммировать выстрел по спрайтам-врагам. В первой половине урока учащиеся планируют работу над проектом, создают класс Bullet и группу спрайтов-пуль. Во второй половине — дополняют класс Player методом fire() и рассуждают о настройках трудности игры. Техническое примечание. Вся работа над игрой «Шутер» организована в одном задании в VS Code.

Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Сегодня мы продолжаем разработку игры “Шутер”. Вспомним техническое задание и реализованные задачи с помощью mind map и дополним чек-лист задачами на сегодня». Напомните разработчикам, на каком этапе разработки игры «Шутер» они находятся в данный момент. В прошлый рабочий день были созданы враги, автоматически перемещающиеся навстречу игроку. Задача на сегодня — запрограммировать стрельбу пулями, при этом каждый выстрел должен сопровождаться звуковым эффектом. Напомните, что для реализации пуль потребуется создать класс Bullet — наследник класса GameSprite, который в свою очередь является наследником описанного в pygame класса Sprite. Выделите действия, характерные для спрайта-пули, которые нужно включить в Bullet: появление при нажатии на «пробел», автоматическое перемещение наверх, исчезновение при столкновении с врагом. Кратко сформулируйте возможные способы реализации данных действий. Отметьте, что в игре будет много пуль (в отличие от спрайтов-врагов каждое нажатие на «пробел» создаёт новый экземпляр класса, а не перемещает уже существующий, ведь их число не ограничено). Придите к необходимости добавлять все создаваемые пули в группу. Так с помощью одной команды можно будет выполнять действия со всеми пулями, например, перемещать их вверх или проверять их столкновение с врагами. С другой стороны, число пуль будет постоянно расти, ведь большая часть из них просто улетает за пределы экрана. Может получиться, что большинство проверок столкновений пуль и врагов будет осуществляться для неактуальных спрайтов. Предложите разработчикам сформулировать возможное решение этой проблемы. Ответ от разработчика Кости будет дан на этапе «мозгового штурма». Отметьте уже выполненные задачи в чек-листе и перечислите задачи, которыми необходимо дополнить чек-лист на текущий рабочий день. Сформулируйте цель рабочего дня и анонсируйте его содержание.

Напомните задачу первой половины рабочего дня — запрограммировать класс Bullet и спрайты-пули. Начните обсуждение с того, что класс Bullet, как и Enemy, является наследником класса GameSprite и дополняется только методом перемещения. Перемещение спрайтов-пуль будет происходить в игровом цикле с помощью метода update(). Разберите устройство update(),

обновляющее координаты группы при смене кадров игрового цикла. Обратите внимание, что на сцене может находиться любое число пуль. С каждым нажатием на «Пробел» создаётся новый экземпляр класса Bullet. В отличие от спрайтов-врагов после пересечения верхней границы экрана пуля никуда не исчезает и не появляется в новом месте. Получается, количество пуль будет расти с каждым нажатием на «Пробел». В будущем это может усложнить работу, ведь в игровом цикле будут проверяться столкновения врагов с пулями, которых давно нет на сцене. Придите к решению данной проблемы: удалять из группы пули, «улетевшие» за границы сцены. Продемонстрируйте разработчикам команды удаления ненужных спрайтов из рабочей группы. Обсудите с разработчиками программную реализацию автоматического перемещения спрайта-пули снизу вверх и его исчезновение по достижении верхней границы окна. Покажите на схеме, как ввести класс Bullet в код игры «Шутер». Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.
Выделите задачу на вторую половину рабочего дня — запрограммировать «выстрел» объектом-пулей. С точки зрения игровых объектов, выстрел производит спрайт-игрок. Заметьте, что в рамках объектно-ориентированного подхода будет разумным описать выстрел в методе класса Player. Тогда обработка нажатия на «Пробел» должна быть описана в методе fire() класса Player. Примечание. Если быстрые ученики уже запрограммировали обработку событий прямо в игровом цикле, её необходимо перенести в класс Player. «Пуля должна появляться в верхней точке корабля игрока. При этом положение корабля меняется при каждом нажатии на клавишу со стрелкой влево или вправо. Как связать точку появления пули с текущим положением игрока? (Ответы учеников). Вы знаете, что каждый спрайт «вписан» в невидимый прямоугольник типа Rect. Положение прямоугольника мы считаем положением спрайта. Пуля должна появляться наверху посередине этого прямоугольника. Задать точку появления можно с помощью новых полей класса Rect». Продемонстрируйте разработчикам команды для получения текущей координаты спрайта, координаты X середины спрайта и координаты Y верхней точки спрайта. Покажите на схеме, как ввести метод fire() класса Player в код игры «Шутер». Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.
Организуйте работу в VS Code над вторым этапом создания игры «Шутер». В задании требуется перенести функционал выстрела в класс Player и изменить обработку события «нажатие на пробел» в игровом цикле. Разработчики, у которых осталось время, могут запрограммировать надписи «YOU WIN» и «YOU LOSE». Они пригодятся им на следующем уроке.
Отвлеките разработчиков от компьютеров и организуйте техническое собеседование по материалам «мозговых штурмов». Предложите разработчикам теоретическую документацию. Анонсируйте, что в следующий рабочий день разработчки завершат создание игры «Шутер» и запрограммируют самую важную часть игры: условия победы и проигрыша!

5.8. Игра «Шутер». Часть 4

Цель — завершить игру «Шутер» описанием условий победы и проигрыша, связанных со столкновениями спрайтов. В первой половине урока учащиеся завершают работу над игрой «Шутер». Во второй половине — описываются кейсы для тестирования игры. Затем разработчики тестируют готовый продукт. Техническое примечание. Вся работа над игрой «Шутер» организована в одном задании в VS Code.
Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Сегодня мы завершаем работу над основной частью игры “Шутер”. Вы уже запрограммировали большую часть функционала игры. Сегодня мы добавим в проекты условия победы и проигрыша. Также мы составим несколько тест-кейсов и оценим работоспособность получившегося

продукта. Готовы к работе? (Ответы учеников)». Напомните разработчикам, на каком этапе разработки игры «Шутер» они находятся в данный момент. В прошлый рабочий день была запрограммирована стрельба пулями. Задача на сегодня — запрограммировать столкновения и реакции на них, влияющие на исход игры. Обсудите события, которыми может быть обеспечен проигрыш и выигрыш. Обратите внимание разработчиков на то, что в случайный момент времени на сцене находится ровно шесть спрайтов-врагов и неограниченное число пуль. Все эти спрайты необходимо проверять на столкновения. Сообщите, что поскольку пули и враги были объединены в группы спрайтов, проверку можно будет задать всего одной командой, не перебирая и не проверяя возможное столкновение каждой пули с каждым врагом вручную. Напомните разработчикам о необходимости тестирования коммерческого продукта перед передачей заказчику. Сегодня разработчики не только завершат работу над «Шутером», но и проверят, получилось ли соблюсти все требования технического задания. Отметьте уже выполненные задачи в чек-листе и перечислите задачи, которыми необходимо дополнить чек-лист на текущий рабочий день. Сформулируйте цель рабочего дня и анонсируйте его содержание.

Начните обсуждение с выигрыша. Чтобы описать выигрыш, необходимо: подписаться на столкновения пуль и врагов, и сделать счётчик поверженных врагов активным. Если счётчик превысил отметку в 10 врагов, наступает победа. Продемонстрируйте команды для обработки столкновения спрайтов, принадлежащих группам. Обратите внимание, что в аргументе функции проверки столкновения необходимо указать, нужно ли удалять столкнувшиеся спрайты. Это очень удобно, ведь не нужно отдельной командой исключать спрайт из группы. Например, если враг будет повержен пулей, то при передаче аргументов True удалятся и враг, и пуля. Продемонстрируйте блок-схему обработки столкновений пуль и монстров. Поясните, на каком этапе необходимо удалять монстров, а на каком создавать новых. Кратко обсудите, почему новый счётчик победных очков score не требуется делать глобальным. Рассмотрите блок-схемы победы и проигрыша. Напомните, что переменная finish отражает работу всех игровых спрайтов и от её значения зависит, работает ли игровой цикл. Поясните, что значение переменной finish в новых алгоритмах должно меняться. Отметьте, что при желании игру можно усложнить, добавив условие, что при столкновении спрайта-игрока с любым из спрайтов-врагов игрок проигрывает. Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

Организуйте работу в VS Code над игрой «Шутер». В задании разработчикам требуется добавить в игровой цикл условия победы и проигрыша, описать набор действий в каждой из ситуаций. Разработчики, которые справились с заданием раньше остальных, могут попробовать усовершенствовать игру, добавив автоматический перезапуск пройденного уровня через 5 секунд. Ссылка на архив проекта есть в конце методических указаний.

Выделите задачу на вторую половину рабочего дня — тестирование игры. Напомните разработчикам, что они уже знакомы с профессией тестировщика ПО и этапами тестирования продукта. Напомните, почему тестирование является обязательной ступенью жизненного цикла программного обеспечения: при разработке большого проекта трудно заметить мелкие недочёты; сложно предсказать все действия пользователя и обработать их в коде. Напомните разработчикам, что программа тестируется по тест-кейсам. Каждый тест-кейс реализуется в 3 шага: Оставляется проверяемая функция. Какую цель преследуем? Описываются шаги для достижения цели. Что делает пользователь? Описывается ожидаемый результат. Что делает программа? Приведите пример тест-кейса для перемещения врага. Рассмотрите базовую ситуацию, при которой игрок не влияет на движение врага (не касается и не стреляет в него). Обсудите, что должно происходить на каждом шаге теста.

Предложите разработчикам описать не менее двух тест-кейсов к «Шутеру» с помощью mind map, а затем протестировать игру по ним. Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к тестированию.

5.9. Доработка и презентация проекта

Цель — доработать игру «Шутер» и презентовать результаты проектной работы. В первой половине урока учащиеся дорабатывают игру «Шутер» по предложенным вариантам и собственным идеям. Во второй половине — изучают основы публичного выступления и презентуют доработанный проект. Техническое примечание. Работа над игрой «Шутер» организована в VS Code, а над презентацией — в одном облачном файле-копии заготовки презентации. Методическая рекомендация. На последний этап урока с презентацией проекта можно пригласить родителей учеников.

Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! В прошлый раз вы завершили работу над игрой “Шутер”. Разработчик Костя направил бета-версию игры заказчику, и работа получила восторженные отзывы. Поздравляем! Однако заказчик выразил пожелания по доработке. Он предлагает сделать игру более сложной, например, с ограничением пуль и препятствиями в форме астероидов. Готовы к работе? (Ответы учеников)». Обозначьте задачи рабочего дня: разнообразить функционал игры «Шутер» новыми механиками и презентовать финальный результат заказчику. Сообщите разработчикам, что заказчики редко являются экспертами в программировании, поэтому проект нужно не только выполнить, но и правильно преподнести. Чтобы сделать представление результатов работы наглядным, необходимо подготовить презентацию. Поскольку мы сообща работали над проектом, презентация также будет общей. Каждый разработчик составит несколько слайдов и выступит с ними. Обобщите цель рабочего дня и анонсируйте его содержание.

Обсудите доработку игры «Шутер». Обязательна реализация предложений заказчика, но можно внести и собственные идеи. Рассмотрите два пожелания: новые препятствия в форме астероидов и перезарядку оружия игрока (непрерывно можно произвести 5 выстрелов, затем идёт «перезарядка» в течение 3 сек). Рассматривая первое дополнение, напомните, что ранее вы уже вводили группу похожих спрайтов-врагов в игру. Препятствия (астероиды) можно ввести аналогичным образом. Постарайтесь максимально вовлечь разработчиков в обсуждение: «Опишите создание группы спрайтов. Как будут появляться и перемещаться спрайты? Что будет происходить при столкновении с игроком? (Ответы учеников)». После обсуждения разберите блок-схему реализации препятствий (астероидов). Обратите внимание, что без счётчика жизней игра может оказаться сложной (разработчики смогут самостоятельно сделать этот вывод в процессе тестирования). Методический комментарий. На этапе доработок показывать готовый код крайне не рекомендуется. Учащиеся уже достаточно хорошо знают проект и библиотеку pygame, чтобы запрограммировать дополнения самостоятельно. Рассмотрите второе дополнение. С помощью оружия игрок может уничтожать монстров. Однако количество пуль и скорость стрельбы не ограничена, поэтому пользователь слишком легко может набрать необходимое число очков. Ограничим количество пуль пятью и добавим перезарядку в течение 3 секунд. Такое изменение сделает игру сложнее и замотивирует игрока экономнее расходовать боеприпасы. Для реализации второго дополнения необходимо ввести в игру две переменные: num_fire — переменная-счётчик для подсчёта совершённых выстрелов; rel_time — переменная-флаг с логическими значениями, отвечающая на вопрос: «Идёт ли перезарядка?». Рассмотрите блок-схемы перезарядки орудия игрока. Если необходимо, устно поясните, какие команды соответствуют блокам. Подведите итоги обсуждения, сформулируйте задачи следующего этапа и переходите к работе в VS Code.

Организуйте работу в VS Code над игрой «Шутер». В задании разработчикам требуется ввести в игру препятствия-астероиды и ограничить стрельбу пулями. Разработчики, которые справились с заданием раньше остальных, могут попробовать запрограммировать счётчик жизней игрока. Число жизней уменьшается при столкновении с монстрами и астероидами. Напомните разработчикам, что с завершением программы работа над проектом не заканчивается: необходимо провести тестирование, а после — презентацию проекта. Обоснуйте необходимость проведения презентации: заказчику нужно продемонстрировать разработанное программное обеспечение. При этом важно обеспечить наглядность и говорить с аудиторией «на одном языке». Продемонстрируйте разработчикам базовую структуру презентации проекта и заготовку предложений, которые можно использовать при выступлении. Отметьте, что осуществлялась общая работа над техническим заданием, поэтому доклад о проделанной работе тоже будет общим. Опорные точки доклада и иллюстрации можно разместить на слайдах с помощью Google Презентаций. Познакомьте разработчиков с основами работы с презентациями. Обратите внимание, что работа будет идти в общем документе, поэтому важно быть внимательными и не редактировать слайды коллег. Для этого будут распределены зоны ответственности, которых нужно придерживаться. Сообщите разработчикам, что для создания собственной презентации нужна регистрация, но для того чтобы редактировать чей-то документ, свой аккаунт не нужен. Продемонстрируйте основные моменты работы с Google Презентациями: добавление и размещение текста и картинок, редактирование и копирование элементов. Подведите итоги обсуждения. Откройте копию шаблона презентаций, вышлите ссылку разработчикам через марсограм или чат онлайн-платформы. Перейдите в первый скрытый слайд и распределите слайды. Затем переходите к работе над презентацией.

6. Публикация и распространение ПО(4 часа)

6.1. Сборка проекта в приложение

Цель — изучить и применить на практике возможности модуля Pyinstaller. В течение занятия ученики знакомятся с системой управления пакетами pip и расширениями с предустановленными модулями в VS Code. Затем каждый учащийся установит дополнительный модуль Pyinstaller и с его помощью упакует игру в файл с расширением .exe (.pkg).
Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! В прошлый раз вы завершили работу с Pygame и презентовали продукт заказчикам. Вы уже знаете, что завершение технического аспекта — лишь середина жизненного цикла программного обеспечения. Далее следуют новые этапы: документирование ПО; ввод в эксплуатацию; распространение и сопровождение. Правила распространения продукта определяют его создатели. Как вы думаете, любые программы можно распространять свободно? (Ответы учеников). Давайте разберёмся, какие способы распространения ПО существуют и может ли пользователь бесплатно использовать любое ПО, найденное в Интернете». Объясните разработчикам, что выделяют три категории программного обеспечения: закрытое, открытое, свободное. Кратко охарактеризуйте каждую категорию. Рассмотрите пример с конфликтом фирмы StarProg с разработчиком Петром. Конфликт возник из-за использования Петром кода приложения «Трекер здоровья», разработанного StarProg. Разберите доводы обеих сторон и придите к выводу, что теперь спор можно будет решить лишь через суд. Заметьте, что фирме стоило не допускать подобную ситуацию и чётко указывать, допустимы ли использование и доработка кода их продукта. Объясните, что во избежание споров публикация продукта должна сопровождаться его лицензированием. Лицензия — это документ, регулирующий правила распространения и

использования программного обеспечения. В настоящее время популярными являются лицензии, описанные компанией Creative Commons. Одну из них — Creative Commons (CC) Attribution — будут использовать и разработчики. В лицензии декларируется следующее правило: «Используйте мой продукт как угодно, но сохраняйте моё авторство». Рассмотрите ещё одну ситуацию. Фирма StarProg заново разместила в Интернете приложение «Трекер здоровья» с лицензией CC Attribution. Инвестор Александр хотел ознакомиться с продуктом, но не смог этого сделать из-за отсутствия VS Code на компьютере. Обсудите, была ли возможность у фирмы не упустить инвестора. На самом деле, возможность была. В основном VS Code используют разработчики, а инвесторам и пользователям он не нужен. И для запуска, например, VS Code не требуется дополнительная программная среда — всё работает «из коробки». Решением проблемы является упаковка проекта в исполняемый файл, который не требуется запускать из VS Code. Для этого необходимо изучить новый модуль Pyinstaller. Работа с модулем будет происходить через терминал (консоль), поэтому также нужно изучить команды нового типа. Сформулируйте цель рабочего дня и анонсируйте его содержание. «Мозговой штурм»: «Расширения. Установка пакетов» (15 мин) Продемонстрируйте внешний вид среды программирования VS Code. Покажите, что в среду интегрирована платформа «Алгоритмики», т. е. базовые возможности редактора кода были дополнены функционалом платформы. Расскажите, что в данном случае платформа — это расширение, т. е. программа, дополняющая возможности другой программы или приложения. Расширения могут дополнять функционал как программ (в частности, игр), так и операционных систем. Например, некоторые расширения позволяют работать с нестандартными файлами или устройствами. Сообщите разработчикам, что в настоящее время в VS Code уже стоят минимум два расширения. Предложите отыскать и назвать их. Продемонстрируйте, что расширения находятся во вкладке Extensions. В данный момент в VS Code установлены: Расширение для Python. Выделяет фрагменты кода цветами, подсказывает команды и проверяет синтаксис. Расширение платформы «Алгоритмики». Связывает создаваемые программы с уровнями платформы и автоматически сохраняет все решения на сервер «Алгоритмики». Для установки расширения необходимо выполнить следующие шаги : Найти расширение через поиск и открыть страницу с ним. Нажать на Install. Дождаться окончания установки и убедиться, что расширение появилось во вкладке слева. Расскажите разработчикам о популярных расширениях Bracket Pair Colorizer 2 (окрашивает парные скобки в одинаковый цвет), Bookmarks (позволяет расставлять в коде закладки и переключаться между ними) и других. Продемонстрируйте, как установить и удалить расширение. Расширение «Алгоритмики» позволяет работать с библиотеками Python, не входящими в стандарт языка. Но Pyinstaller не является частью расширения, поэтому нужно научиться устанавливать необходимые для работы пакеты. С помощью презентации разберите шаги для отображения в терминале списка имеющихся библиотек и установки недостающих. Установка происходит с помощью пакетного менеджера pip. Техническое примечание. При работе с pip система может предложить обновление менеджера. В таком случае обновите его, используя предложенную терминалом команду. Подведите итоги обсуждения и переходите к работе в VS Code. VS Code + платформа. Установка Pyinstaller (15 мин) Организуйте работу в VS Code. В задании разработчикам требуется установить в VS Code новое расширение и протестировать его работу на одном из существующих проектов. Затем с помощью pip необходимо отобразить и изучить список установленных пакетов и установить новый пакет Pyinstaller. Техническое примечание. Если Pyinstaller уже установлен, требуется установить его заново, т. к. предыдущие версии пакета работают иначе.

«Мозговой штурм»: «Создание исполняемого файла» (15 мин) Технический комментарий. Проверьте, что в названиях всех файлов в пути к папке проекта используются только англоязычные символы. Особое внимание обратите на имя пользователя: если в нём используются символы русского языка, его необходимо изменить (например, Олег — Oleg): Windows. «Пуск» -> «Параметры» -> «Учётные записи» -> «Изменение имени своей учётной записи». Linux. «Терминал» -> usermod -l new_username old_username. macOS. «Системные настройки» -> «Пользователи и группы» -> Нажмите на кнопку с замком -> Выделите изменяемую учетную запись -> «Дополнительные параметры» -> «Полное имя». Задача на вторую половину рабочего дня — с помощью Pyinstaller создать исполняемый файл игры, который можно будет запускать без VS Code. Обратите внимание, на разных операционных системах процесс создания файла выполняется по-разному! Например, установка на macOS требует некоторых дополнительных шагов, а в Linux готовый файл может не запуститься двойным кликом, и к нему нужно будет обращаться через терминал. Рассмотрите схему создания исполняемого файла и изучите реализацию её шагов. Сначала необходимо открыть проект с помощью расширения «Алгоритмики» и убедиться, что игра работает. Затем все шрифты игры заменяются на стандартные, например, Arial. В противном случае при сборке могут возникнуть ошибки. Также нужно отключить расширение «Алгоритмики» и кликнуть на Reload, чтобы изменения отобразились в редакторе. Затем нужно разместить проект в удобно располагаемой рабочей папке. Для этого кликните правой кнопкой мыши по любому файлу в папке проекта и выберите «Показать в Проводнике» или Reveal in Finder. Скопируйте файлы проекта и разместите их в новой папке. Откройте эту папку в VS Code. Дополнительные действия для macOS. Исполняемый файл будет создаваться с помощью стандартного интерпретатора Python в собственном виртуальном окружении. Для этого в левом нижнем углу выбирается стандартный интерпретатор Python 3. Затем создаётся окружение: python3 -m venv my_venv. Во всплывшем предупреждении нужно одобрить замену окружения на my_venv. После этого окружение активируется с помощью команды: source my_venv/bin/activate. Теперь нужно заново установить пакеты Pygame и Pyinstall. Для этого используется команда: pip install pygame pyinstal. Дальнейшие шаги снова являются общими. Исполняемый файл создаётся с помощью команды: pyinstaller --onefile shooter_game.py. После этого нужно убедиться, что упаковка завершилась успешно (successfully). Последним шагом подготовьте итоговую папку с файлами для публикации или распространения. Создайте новую папку, например, result, и скопируйте туда созданный исполняемый файл из папки dist и все картинки и музыку, используемые в игре. Подведите итоги обсуждения и переходите к работе в VS Code. VS Code + платформа. Создание исполняемого файла (20 мин) Организуйте работу в VS Code. В задании разработчикам требуется подготовить VS Code и проект к созданию файла, а затем с помощью Pyinstaller создать исполняемый файл с игрой. Завершение урока (10 мин) С помощью презентации подведите итоги рабочего дня. Проведите техническое интервью с вопросами по материалам «мозгового штурма». Предложите дополнительно попробовать дома создать ещё один исполняемый файл для другой игры на Pygame. После упаковки проект можно скопировать на флешку и открывать на другом компьютере.

6.2. Повторение. Введение в Git

Цель — изучить функционал сервиса GitHub и взаимодействие с ним через сайт. В течение занятия ученики создают ещё один исполняемый файл (на этот раз для приложения на PyQt). Затем они знакомятся с системой контроля версий Git, изучают функционал GitHub с помощью браузера и создают репозиторий с кодом приложения. Примечание. Рекомендуйте ученикам заранее завести электронную почту на любом удобном ресурсе. Адрес почты потребуется при регистрации на GitHub.

Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! К нам поступил вопрос от младшего разработчика Екатерины, осваивающей создание приложений на PyQt. Кажется, несколько программистов, занимающихся одним приложением, никак не могут наладить рабочий процесс. Вместе с Андреем и Машей Екатерина работает над приложением Swipe and Fun. Разработка идёт очень медленно, потому что они не могут решить несколько проблем. Готовы помочь коллегам? (Ответы учеников)». Рассмотрите поступившие вопросы. Первая проблема заключается в том, что каждый разработчик работает на своём компьютере и использует флешку для обмена наработками. Это отнимает много времени и создаёт неудобства для всей команды. Вторая проблема заключается в небезопасности такого подхода. Неделю назад разработчик Андрей хотел скопировать проект с компьютера Екатерины, но перепутал комбинации клавиш и не скопировал, а вырезал папку. Затем флешка с папкой сломалась, и работа Кати была потеряна. С подобными трудностями сталкиваются многие начинающие разработчики. Надо разобраться, как избежать возникновения таких неприятных ситуаций. Придите к выводу, что во избежание повторения подобных ситуаций необходимо пересмотреть подход к разработке. При этом он должен отвечать ряду требований: быстрый доступ к проекту должен быть у нескольких разработчиков сразу; должна существовать возможность самостоятельной доработки проекта и объединения одного личного решения с решениями других программистов; проект должен оставаться целым при поломке техники. Продемонстрируйте схему с изложенными проблемами и способами их решения. Сообщите разработчикам, что решить весь перечень проблем можно с помощью системы контроля версий Git и сервиса GitHub. Система контроля версий — это подход к разработке, сохраняющий все изменения, происходящие с кодом. Git — одна из разновидностей системы. При работе с Git сохраняются все версии проекта, благодаря чему в любой момент можно «откатиться» до прошлой версии и на её основе создать свою собственную. GitHub — это сервис для публикации (хостинга) кода, быстрого доступа к его версиям и обмена информацией. На самом деле, это целая социальная сеть для разработчиков. Работать с GitHub можно с помощью браузера или прямо через терминал. Работа с системой управления версиями Git и сервисом GitHub требует определённых навыков. Нужно уметь размещать материал в хранилище, выстраивать иерархию версий проекта и настраивать параметры использования разработанного кода. Тем не менее, уже сегодня разработчики опубликуют на GitHub свой первый проект. Сформулируйте цель рабочего дня и анонсируйте его содержание. «Мозговой штурм»: «Создание исполняемого файла» (15 мин) Сообщите разработчикам, что сегодня на GitHub будет размещён один из первых больших проектов — приложение Memory Card. Правила данного сообщества не приветствуют загрузку исполняемых файлов, т.к. они много весят и отличаются для разных операционных систем, поэтому он будет использоваться только для физического распространения (например, с помощью флешки). Продемонстрируйте план работы, выделите задачи на первую и вторую половины рабочего дня. «Коллеги, вы уже имеете опыт создания исполняемых файлов. Напомните, что такое расширение? Для чего используется терминал? Каковы основные этапы создания исполняемого файла. (Ответы учеников)». С помощью схемы повторите с разработчиками основные этапы создания исполняемого файла. Затем перейдите к упаковке приложения. Технический комментарий для пользователей macOS. Создание виртуального окружения было технической необходимостью именно для PyGame. При упаковке приложения на PyQt можно обойтись и без него. Подведите итоги обсуждения и переходите к работе в VS Code. VS Code + платформа. Создание исполняемого файла (15 мин) Организуйте работу в VS Code. В задании разработчикам требуется подготовить VS Code и проект к созданию

исполняемого файла для приложения на PyQt. С помощью Pyinstaller создайте исполняемый файл и разместите его в отдельной папке. Если необходимо, используйте документацию к прошлому рабочему дню.
«Мозговой штурм»: «Основы GitHub» (15 мин) Сообщите разработчикам, что знакомство с GitHub начнётся с регистрации и публикации собственного продукта. Для регистрации требуется адрес электронной почты. Откройте сайт github.com. Введите свои электронные адреса и нажмите на «Зарегистрироваться в GitHub». Заполните необходимые поля с информацией. В GitHub можно подписываться на другие аккаунты, они будут формировать вашу ленту новостей. Если кликнуть на превью аккаунта справа наверху и перейти во вкладку «Ваш профиль», то отобразится информация о вашей активности. Необходимо заполнить информацию о себе и указать свои профессиональные интересы. Для этого кликните на вкладку «Редактировать профиль». Продемонстрируйте вкладку «Репозитории». Репозиторий — это место для хранения и поддержки данных проекта. Для начала необходимо придумать репозиторию имя, которое будет отражать суть проекта, затем настроить доступ к нему. У разработчиков будет доступ Public: репозиторий можно будет найти через Интернет. При создании репозитория стоит поставить дополнительные галочки для добавления лицензии (выберете Creative Commons) и файла ReadMe. Далее добавьте в репозиторий файл с кодом проекта. Любую загрузку файлов с изменениями проекта стоит комментировать. В данном случае напишите, что загружена первая версия приложения. Подведите итоги обсуждения и переходите к работе в VS Code. GitHub: «Основы работы» (20 мин) Организуйте работу в GitHub. В задании разработчикам требуется создать аккаунт в GitHub и заполнить информацию о себе. Затем необходимо создать репозиторий с приложением Memory Card. Завершение урока (10 мин) С помощью презентации подведите итоги рабочего дня. Проведите техническое интервью с вопросами по материалам «мозгового штурма». Предложите дополнительно попробовать дома создать ещё один репозиторий с другим проектом, например, Easy Editor.

6.3. Игра «Пинг-понг». Часть 1

Цель — применить знание Python и навыки проектной работы для реализации игры. В течение занятия ученики выполняют полное планирование работы и создают репозиторий для игры «Пинг-понг» на GitHub. Затем они реализуют первую часть игры, связанную с созданием двух ракеток и управлением ими, и отображают изменения в репозитории (делают коммиты). Примечание. Попросите учеников разместить ссылки на репозитории сразу после их создания в задании на платформе и своевременно делать коммиты. Так вы сможете своевременно отслеживать текущий прогресс. Техническое примечание. Папка и файл для проекта создаются учениками на компьютере самостоятельно (задания по реализации в VS Code нет). Для спрайтов можно использовать свои картинки или графические объекты: кружок и два прямоугольника.
Сюжетная линия. Обсуждение: «Работа над проектом» (5 мин) Откройте презентацию. Разработчикам компьютеры пока не требуются. «Нашему отделу поставлена задача по созданию прототипа мультиплеерной игры “Пинг-понг”, который будет представлен заказчику. После этого он, возможно, закажет у нашей фирмы разработку полной версии игры. Изучим возможный вид игры. Готовы? (Ответы учеников)». Сообщите разработчикам, что от них не требуется полная реализация игры. Достаточно запрограммировать: мяч, автоматически передвигающийся по сцене; две ракетки, управляемые с клавиатуры; условие проигрыша: проигрывает игрок, пропустивший мяч. При наличии времени функционал может быть расширен и дополнен. Разработчикам требуется составить полное планирование реализации проекта. Для этого им необходимо использовать все профессиональные навыки разработки программного обеспечения. Весь цикл работы над прототипом будет

включать в себя создание: mind map игры с описанием функционала; полного чек-листа с задачами по созданию прототипа; репозитория игры на GitHub (по мере реализации задач из чек-листа репозиторий должен обновляться); презентации с итогами работы и её защита перед заказчиками. Сформулируйте цель рабочего дня и анонсируйте его содержание. «Мозговой штурм»: «Планирование работы» (10 мин) Начните реализацию полного планирования проекта. Поскольку потребуется создать только прототип игры, то можно спланировать все рабочие задачи сразу. Предложите открыть задание «Игра “Пинг-понг”: планирование» и оформить карту проекта. Сообщите, что разработчики могут сами выбрать или придумать принцип выделения и разбиения частей в mind map. Далее предложите оформить чек-лист. Обратите внимание, что в него должна быть добавлена работа на GitHub (создание репозитория и коммиты реализованных фрагментов) и презентация игры. Продемонстрируйте возможное начало чек-листа. Обратите внимание, что в прошлой игре «Шутер» спрайты были похожими по функционалу: спрайт-игрок управлялся с клавиатуры (главный герой), а спрайты-враги также перемещались автоматически. Тем не менее, потребуется внести некоторые изменения, например, метод update() класса Player нужно будет разделить на два метода — для управления левой и правой ракетами. Подведите итоги обсуждения и переходите к работе на платформе и GitHub. Платформа + GitHub: создание репозитория (15 мин) Организуйте работу в VS Code. В задании разработчикам требуется завершить планирование проекта на платформе, создать на GitHub репозиторий для игры «Пинг-понг», выбрать лицензию и оформить файл ReadMe. Обратите внимание на слайд-подсказку с описанием файла ReadMe. Если необходимо, используйте документацию к прошлому рабочему дню. «Мозговой штурм»: «Реализация игры» (15 мин) Обратите внимание разработчиков на то, что функционал игры «Пинг-понг» отчасти похож на игру «Шутер»: спрайты-игроки управляются с клавиатуры и перемещаются вдоль одной линии, спрайт другого типа автоматически перемещается по сцене, также необходимо обрабатывать столкновения спрайтов и пересечение ими границ сцены. Задайте вопросы: «Получится ли использовать часть готового кода игры “Шутер” в проекте “Пинг-понг”?», «Если да, то какую?». Придите к выводу, что в игру «Пинг-понг» можно скопировать классы GameSprite и Player. Тогда ракетка будет экземпляром класса Player, а мяч — экземпляром класса GameSprite или класса, аналогичного Enemy. Также в классе Player должна быть реализована возможность управлять разными экземплярами класса с помощью разных клавиш. Продемонстрируйте, как изменить класс Player и создать два разных метода update() для левой и правой ракетки. Затем рассмотрите схему части игры с двумя ракетками. Примечание. Допускаются и другие способы управления двумя ракетками. Например, можно реализовать один метод update(), но при создании объекта Player передавать коды клавиш, с помощью которых он будет управляться (def __init__(self, K_w, ...)). Такой способ более сложный и более профессиональный. Напомните, что по мере реализации задач чек-листа необходимо делать коммиты в репозиторий игры на GitHub. Коммит — это изменение файлов, входящих в репозиторий. Первый коммит можно сделать с заготовкой игры. Дайте разработчикам несколько рекомендаций: Загружаемый на GitHub проект должен быть рабочим. Это позволит в будущем «откатиться» до прошлой версии проекта, если что-то пойдёт не так. Если проект не работает (например, при запуске возникают ошибки, которые вы хотите исправить в следующий раз), его лучше не коммитить. Не существует единого мнения, как часто нужно коммитить проект (сохранять его версии в репозитории). Один из разумных подходов — делать коммит при успешном решении новой задачи (например, при успешном выполнении очередного пункта из чек-листа). При создании коммита можно выбрать рабочую ветку. По умолчанию коммиты совершаются в главной ветке (the main branch). Создание новой отдельной ветки может быть полезно при совместной работе над одним проектом. При желании

дополнительные ветки можно будет слить с главной. Разберите процесс создания первого коммита с заготовкой игры. Так как в проекте ещё нет файлов, добавьте программный файл с игрой. После этого внесённые изменения должны отобразиться в репозитории с написанным комментарием. Далее представьте, что игра была дополнена классом GameSprite и был создан и отображён на сцене спрайт-мяч: Откройте файл, в который нужно внести изменения и перейдите к его редактированию. Внизу появится поле для комментирования коммита. Вставьте в поле с кодом изменённый код. Внизу прокомментируйте изменения и нажмите на Commit Changes. При необходимости можете проследить все изменения проекта. Для этого воспользуйтесь вкладкой History с историей коммитов. Подведите итоги обсуждения и переходите к работе в VS Code и GitHub.
VS Code + GitHub: «Игра “Пинг-понг”» (30 мин) Организуйте работу в GitHub. В задании разработчикам требуется запрограммировать часть прототипа игра «Пинг-понг» с двумя управляемыми ракетками в VS Code и по мере выполнения задач делать коммиты в репозиторий проекта на GitHub. Технический комментарий. Если разработчики задают спрайты с помощью картинок, то их тоже нужно добавить в репозиторий. Завершение урока. Коммит наработок (10 мин) С помощью презентации подведите итоги рабочего дня. Проведите техническое интервью с вопросами по материалам «мозгового штурма». Напомните ученикам о необходимости загрузить в задание на платформе ссылку на свой репозиторий и проверить перед уходом, закоммитили ли они последние изменения в своих проектах.

6.4. Игра «Пинг-понг». Часть 2

Цель — завершить реализацию игры «Пинг-понг» и презентовать результат работы. В течение занятия ученики реализуют вторую часть игры, связанную со столкновениями и условием проигрыша, отображают изменения в репозитории (делают коммиты) и презентуют результат работы.
Сюжетная линия. Обсуждение: «Планирование работы» (5 мин) Откройте презентацию. Разработчикам компьютеры пока не требуются. «Здравствуйте, коллеги! Нашему отделу поставлена задача по созданию прототипа мультиплеерной игры “Пинг-понг”. В прошлый раз мы запрограммировали интерфейс игры и управление ракетками с клавиатуры. Осталось реализовать автоматическое перемещение мяча по сцене и условие проигрыша. В нашем прототипе проигрывает игрок, пропустивший мяч. Готовы? (Ответы учеников)». Напомните разработчикам, что для реализации прототипа используются профессиональные навыки разработки программного обеспечения, в частности: создание mind map игры с описанием функционала; создание полного чек-листа с задачами по созданию прототипа; создание и обновление репозитория игры на GitHub. В течение рабочего дня разработчикам предстоит завершить работу над прототипом, составить и представить заказчику презентацию проекта. Напомните общую структуру проекта, отметьте задачи, выполненные в предыдущий рабочий день, и выделите задачи, которые предстоит реализовать сегодня. Затем предложите открыть чек-лист и отметить выполненные задачи. После этого изучите задачи на сегодня. Сформулируйте цель рабочего дня и анонсируйте его содержание. «Мозговой штурм»: «Завершение игры» (15 мин) Обсудите задачи на день: Запрограммировать автоматическое перемещение мяча и столкновение мяча с границами сцены. Обработать столкновение мяча и ракетки (отскок мяча от ракетки). Описать условие завершения игры. Обсудите реализацию первой задачи — запрограммировать автоматическое перемещение мяча можно по-разному, например, описать в новом классе Ball метод update() или добавить новые команды в игровой цикл. Первый способ является более долгим, но и более профессиональным решением. Второй вариант — это быстрое решение, не подходящее для долгосрочного проекта (игровой цикл перегружать не рекомендуется). Но поскольку работа идёт над прототипом игры, второе решение является более

выгодным. Если прототип будет утверждён заказчиком, то позднее можно будет обернуть решение в метод класса. Продемонстрируйте слайд с траекторией движения мяча. Чтобы запрограммировать такое перемещение, можно задать изменение координат x и y. Пусть объект ball типа GameSprite уже задан, тогда сдвиг спрайта по горизонтальной и вертикальной оси можно задать через изменение значений переменных. Движение мяча будет продолжаться, пока работает игровой цикл. При столкновении с верхней и нижней границами сцены мяч должен отскакивать в противоположном направлении. Для этого при приближении мяча к верхней или нижней границе, необходимо поменять направление скорости по вертикали на противоположное. Задавать такое условие для боковых границ не нужно, т. к. их пересечение будет означать проигрыш. Реализация отскока мяча от ракетки будет другой. В отличие от границ, ракетка постоянно перемещается и задать отскок просто по координатам невозможно. Зато ракетка является спрайтом, и столкновение с ней можно обработать с помощью метода collide_rect(). В этом случае изменим скорость по горизонтали на противоположную. Перейдите к обсуждению условия проигрыша. В игре соревнуются два игрока, поэтому условие проигрыша для каждого из них будет своё. Первый игрок проигрывает, если мяч улетел за левую границу сцены (оказался сзади ракетки), а второй игрок — если мяч улетел за правую границу. Необходимо создать текстовые объекты для оповещений о завершении игры. Делать проверку на сближение мяча с левой и правой границами сцены нужно постоянно. Если касание стены всё же произошло, то игра считается оконченной и появляется текстовое оповещение. Подведите итоги обсуждения и переходите к работе в VS Code и GitHub. VS Code + GitHub: «Игра “Пинг-понг”» (25 мин) Организуйте работу в VS Code. В задании разработчикам требуется завершить игру «Пинг-понг», делая коммиты реализованных шагов в репозиторий на GitHub. Если необходимо, используйте документацию к прошлому рабочему дню.

Напомните разработчикам, что с завершением программы работа над техническим заданием не заканчивается. Далее проводится тестирование, а затем презентация проекта. Этап тестирования сегодня опускается, поскольку разрабатывался прототип, а не полноценный продукт. Напомните базовую структуру презентации проекта, состоящую из трёх компонентов. Отметьте, что второй шаг может быть расширен. Как и в прошлый раз, обеспечим выступлению наглядность с помощью общей презентации. Опорные точки доклада и иллюстрации разместим на слайдах с помощью Google Презентаций. Скопируйте шаблон себе на диск, откройте доступ на редактирование и отправьте ссылку разработчикам. Перейдите в первый скрытый слайд и распределите работу. Подведите итоги обсуждения и переходите к работе над презентацией. Google Презентации: «Создание презентации» (25 мин) Организуйте работу в Google Презентациях. Разработчикам требуется подготовить общую презентацию о прототипе игры «Пинг-понг». Каждому необходимо сделать несколько слайдов в соответствии с таблицей и быть готовым выступить с ними. Презентация проекта. Рефлексия (10 мин) С помощью презентации подведите итоги рабочего дня. Проведите техническое интервью с вопросами по материалам «мозгового штурма». После завершения презентации, от лица Алёны сообщите о завершении и сдаче игры «Пинг-понг». Отметьте, что руководство ProTeam приняло решение направить команду разработчиков на хакатон по программированию на Python. Среди гостей мероприятия будут инвесторы, поэтому компанию должны представлять настоящие профессионалы. Если хакатон пройдёт успешно, то разработчикам будет предложено присоединиться к бизнес-инкубатору стартаперов.

7. Итоговое повторение(3 часа)

7.1. Приложение Text-to-speech

Project description
pyttsx3 is a text-to-speech conversion library in Python.
Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.
Installation - pip install pyttsx3
If you recieve errors such as No module named win32com.client, No module named win32, or No module named win32api, you will need to additionally install pypiwin32.
Usage :

```
import pyttsx3
engine = pyttsx3.init()
engine.say("I will speak this text")
engine.runAndWait()
```

Changing Voice , Rate and Volume :

```
import pyttsx3
engine = pyttsx3.init()
# object creation """ RATE""" rate = engine.getProperty('rate')
# getting details of current speaking rate print (rate)
#printing current voice rate engine.setProperty('rate', 125)
# setting up new voice rate """VOLUME""" volume = engine.getProperty('volume')
#getting to know current volume level (min=0 and max=1) print (volume)
#printing current volume level engine.setProperty('volume',1.0)
# setting up volume level  between 0 and 1 """VOICE""" voices = engine.getProperty('voices')
#getting details of current voice #engine.setProperty('voice', voices[0].id)
#changing index, changes voices. o for male engine.setProperty('voice', voices[1].id)
#changing index, changes voices. 1 for female engine.say("Hello World!") engine.say('My current speaking rate is ' + str(rate))
engine.runAndWait() engine.stop() """Saving Voice to a file"""
# On linux make sure that 'espeak' and 'ffmpeg' are installed
engine.save_to_file('Hello World', 'test.mp3')
engine.runAndWait()
```

7.2. Приложение Level Editor

Создание редактора уровней для платформера с набором текстур и спрайтов персонажей.
Разработка при помощи библиотеки Pygame с возможностью сохранять созданный уровень

7.3. Игра Платформер

Создание игры на базе редактора уровней

Приложение№1

Календарно-тематическое планирование кружка программирования на Python(второй год)

Класс: 7 - 11 класс

Количество часов за год всего 34 **часа, в неделю** 1 **час.**

Планирование составлено на основе:

Методическое пособие: Алгоритмика Python Start (второй год)

№ урока	Дата проведения		Тема урока
	План	Факт	
Раздел 1. Продвинутый курс			
Тема 1 СТРУКТУРЫ ДАННЫХ			
1			1.1. Повторение. Обработка исключений
2			1.2. Повторение. Списки
3			1.3. Словари
4			1.4. Вложенные структуры данных
Тема 2 РАЗРАБОТКА ОКОННЫХ ПРИЛОЖЕНИЙ			
5			2.1. Классы. Введение в PyQt
6			2.2. Проектирование интерфейса
7			2.3. Приложение Memory Card. Часть 1
8			2.4. Приложение Memory Card. Часть 2
9			2.5. Приложение Memory Card. Часть 3
10			2.6. Приложение Memory Card. Часть 4
Тема 3 РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ			
11			3.1. Основы работы с файлами
12			3.2. Приложение «Умные заметки». Ч. 1
13			3.3. Приложение «Умные заметки». Ч. 2
14			3.4. Приложение «Умные заметки». Ч. 3
Тема 4 АВТОМАТИЧЕСКАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ			
15			4.1. Основы обработки изображений
16			4.2. Приложение Easy Editor. Ч. 1
17			4.3. Приложение Easy Editor. Ч. 2
18			4.4. Приложение Easy Editor. Ч. 3
Тема 5 ПРОДВИНУТАЯ РАЗРАБОТКА ИГР НА PYGAME			
19			5.1. Основы создания игр

20			5.2.	Игра «Лабиринт». Часть 1
21			5.3.	Игра «Лабиринт». Часть 2
22			5.4.	Игра «Лабиринт». Часть 3
23			5.5.	Игра «Шутер». Часть 1
24			5.6.	Игра «Шутер». Часть 2
25			5.7.	Игра «Шутер». Часть 3
26			5.8.	Игра «Шутер». Часть 4
27			5.9.	Доработка и презентация проекта
Тема 6 ПУБЛИКАЦИЯ И РАСПРОСТРАНЕНИЕ ПО				
28			6.1.	Сборка проекта в приложение
29			6.2.	Повторение. Введение в Git
30			6.3.	Игра «Пинг-понг». Часть 1
31			6.4.	Игра «Пинг-понг». Часть 2
Тема 7. ИТОГОВОЕ ПОВТОРЕНИЕ				
32			7.1.	Приложение Text-to-speech
33			7.2.	Приложение Level Editor
34			7.3.	Игра Платформер