

Министерство науки и высшего образования Российской Федерации  
ФГАОУ ВО «Северо-Восточный федеральный университет им. М.К. Аммосова»

Педагогический институт

Кафедра информатики и вычислительной техники

**РАЗРАБОТКА ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО ЯЗЫКУ  
ПРОГРАММИРОВАНИЯ PYTHON НА ОСНОВЕ МОДУЛЯ PYGAME**

**БАКАЛАВРСКАЯ РАБОТА**

Направление подготовки: 44.03.04 Профессиональное обучение (по  
отраслям), профиль: Информатика и вычислительная техника

Выполнил: студент 4 курса группы  
БА-ИВТ-16 ПИ СВФУ

Белолобский Михаил Михайлович

Руководитель: доцент кафедры ИВТ,  
канд. пед. наук, Сорочинский Максим  
Анатольевич



---

Якутск – 2020

## Содержание

ВВЕДЕНИЕ .....	3
ГЛАВА 1. ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON КАК СРЕДСТВО ОБУЧЕНИЯ ДЕТЕЙ ПРОГРАММИРОВАНИЮ .....	5
1.1 Обучение детей на основе игровых технологий .....	5
1.2 Основы языка программирования Python и сравнение с другими языками программирования .....	11
1.3 Сравнение и основные возможности модулей Python .....	22
ГЛАВА 2. РАЗРАБОТКА ПРАКТИЧЕСКИХ ЗАНЯТИЙ НА ОСНОВЕ МОДУЛЯ PYGAME ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON .....	28
2.1 Анализ модуля Pygame и создание планов занятий .....	28
2.2 Разработка практических занятий .....	33
2.3 Апробация результатов исследования .....	55
Заключение .....	57
Список использованной литературы .....	59
Приложения .....	63

## ВВЕДЕНИЕ

Актуальность разработки игр становится все более популярной, игры увеличиваются каждый день. Самые удачные проекты продаются миллионными тиражами. Зарождение игровой индустрии началось в 1970-х годах одновременно с продажей первых компьютеров, рассчитанных на массового потребителя. За 40 с небольшим лет индустрия компьютерных игр развилась до таких масштабов, что уже во многом опережает своих ближайших конкурентов: киноиндустрию, музыкальную индустрию, шоу-бизнес [26].

Для решения данного направления мы предлагаем использовать язык программирования Python так как он в данное время является:

- имеет предельно простой синтаксис, на нем легко писать, его легко читать.
- нескончаемое количество справочной литературы: книг, сайтов, платных и бесплатных курсов, готовых шаблонов и исходников.
- множество доступных сред разработки, сервисов и фреймворков [3].

Python актуален для программистов, которые хотели бы связать свою профессиональную деятельность с Data Science (Наука о данных) или Machine Learning (Машинное обучение) — наиболее перспективных направлениях IT [27].

По данным Tiobe.com в апреле 2019 Python занимал 4 место с рейтингом 8%. До сегодняшнего времени Python обогнал язык программирования C++ и занимает 3 место с рейтингом 9.31% [29].

Согласно информации с hh.ru на начало 2019 года, в России ~4500 вакансий для Python-разработчиков, из них ~2000 в Москве и ~700 в Санкт-Петербурге. Это меньше, чем по запросу «Java» (~5500), но больше, чем по запросу «PHP» (~3600), — можно заметить тенденцию, что Python медленно забирает позиции PHP с рынка веб-разработки. Хотя на PHP все еще написано около 80% всех сайтов в интернете [14].

Многие разработчики рассматривают индустрию компьютерных игр лишь как сектор экономики, в котором можно неплохо заработать. Но, на самом деле, создание игр — это ещё и инструмент культуры [1].

Python представляет популярный высокоуровневый язык программирования, который предназначен для создания приложений различных типов. Это и веб-приложения, и игры, и настольные программы, и работа с базами данных. Довольно большое распространение Python получил в области машинного обучения и исследований искусственного интеллекта [2].

Объектом выступает образовательный процесс в системе дополнительного образования детей.

Предметом исследования – обучение детей модулю Pygame языка программирования Python.

Цель: разработать практические занятия для детей по языку программирования Python на основе модуля Pygame.

Задачи:

1. Проанализировать основы игровых технологий
2. Изучить основные понятия по Python
3. Сделать обзор инструментов Pygame
4. Разработать практические занятия

Гипотеза: создание практических занятий по разработке игр с помощью модуля Pygame в языке программирования Python будет способствовать быстрому обучению программированию.

Структура исследования: бакалаврская работа состоит из введения, двух глав, заключения, списка использованной литературы и приложения.

# ГЛАВА 1. ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON КАК СРЕДСТВО ОБУЧЕНИЯ ДЕТЕЙ ПРОГРАММИРОВАНИЮ

## 1.1 Обучение детей на основе игровых технологий

Б.Т. Лихачев подчеркивает о педагогической технологии как о совокупности психолого-педагогических установок, определяющих специальный набор и компоновку форм, методов, способов, приемов обучения, воспитательных средств; что она есть организационно-методический инструментарий педагогического процесса [9].

По определению Г. К. Селевко, игровая технология – это вид деятельности в условиях ситуаций, направленных на воссоздание и усвоение общественного опыта, в котором складывается и совершенствуется самоуправление поведением. Понятие «игровые педагогические технологии» включает достаточно обширную группу методов и приёмов организации педагогического процесса в форме различных педагогических игр. Характеризуя педагогические игры, Г.К. Селевко указывает на их отличие от игр «вообще» – наличием чётко поставленной цели обучения и соответствующего ей педагогического результата. Наряду с термином «игровые технологии», в педагогике используется термин «игровое обучение», смысл которого заключается в следующем: это игра по форме и учение по содержанию. Можно сказать, что «игровые технологии» и «игровое обучение» – близкие понятия, поскольку оба подразумевают осуществление воспитания и обучения в форме игры. Игровые технологии и игровое обучение призваны поднять стратегию образования на качественно новую основу [10].

Игровые технологии являются лучшим способом для организации деятельности человека. Этот метод используют для того, чтобы процесс обучения был интересным и насыщенным.

Игровая обстановка трансформирует и позицию учителя, который балансирует между ролью организатора, помощника и соучастника общего игрового действия [19].

Итоги игры выступают в двойном плане – как игровой и как учебно познавательный результат. Дидактическая функция игры реализуется через обсуждение игрового действия, анализ соотношения игровой ситуации как моделирующей, ее соотношения с реальностью. Важнейшая роль в данной модели принадлежит заключительному ретроспективному обсуждению, в котором учащиеся совместно анализируют ход и результаты игры, соотношение игровой (имитационной) модели и реальности, а также ход учебно-игрового взаимодействия [33].

В арсенале педагогики начальной школы содержатся игры, способствующие обогащению и закреплению у детей бытового словаря, связной речи; игры, направленные на развитие числовых представлений, обучение счету, и игры, развивающие память, внимание, наблюдательность, укрепляющие волю [32].

Современная дидактика рассматривает игру как основу, как источник развития ребенка. Следуя логике Й. Хейзинга, утверждавшего, что понятие «человек играющий» выражает такую же существенную функцию как «человек созидающий», С.А. Шмаков приходит к убеждению, что «в игре ребенок самовыражается как личность, как индивид, получая разнообразную информацию о мире и о себе от взрослых и сверстников, прежде всего в предметной и вербальной деятельности, в общении...» [31].

Еще в 20-х годах прошлого столетия Л.С. Выготский обратил внимание на изменение содержания и динамики детской игры. Одна из глав этой книги Л.С. Выготского «Педагогическая психология» содержит исследование педагогического значения игры. «Давно обнаружено, – пишет Л.С. Выготский, – что игра не представляет из себя чего-либо случайного, она неизменно возникает на всех стадиях культурной жизни у самых разных народов и представляет неустранимую и естественную особенность человеческой природы. ... Они (игры) организуют высшие формы поведения, бывают связаны с разрешением довольно сложных задач поведения, требуют от

играющего напряжения, сметливости и находчивости, совместного и комбинированного действия самых разных способностей и сил» [35].

Место и роль игровой технологии в учебном процессе, сочетание элементов игры и учения во многом зависят от понимания учителем функций и классификации педагогических игр.

При рассмотрении классификаций учебных игр, предложенных различными исследователями, чаще всего их разделяют по месту и времени проведения. Это игры на воздухе и в помещениях, игры на воде и спортивной площадке, зимние и летние игры. Все это подводит к некоторому разделению игр:

- по области деятельности (физические, интеллектуальные, трудовые, социальные, психологические);

- по характеру педагогического процесса (обучающие, тренинговые, контролируемые, обобщающие, познавательные, воспитательные, развивающие, репродуктивные, продуктивные, творческие, коммуникативные, диагностические);

- по игровой методике (предметные, сюжетные, ролевые, деловые, имитационные, соревновательные, драматизации);

- по предметной области (математические, экологические, физические, искусствоведческие, музыкальные, литературные, прикладные, производственные, спортивные, народные, туристические, управленческие, экономические, педагогические, досуговые и др.);

- по игровой среде (с предметами и без предметов, компьютерные, пространственно-временные) [39].

По классификации Е.А. Крюковой учебные игры подразделяются на ситуационные, ролевые и деловые. Объединяет все эти игры то, что они личностно ориентированы.

Игровые технологии в обучении использовались как эффективный способ передачи информации в целях образования и воспитания детей уже издавна. Это традиционный метод обмена знаниями от старшего поколения к

младшему. Как интерактивный, его использовали в народной педагогике. В дошкольных заведениях и внешкольных учреждениях наиболее часто встречаются случаи использования этих методов.

Наиболее популярным является применение технологии, как одного из методов внеклассной работы. Игровая тематика развивалась для максимально эффективного воздействия на учеников. Дети, которые погружены в определенные заданные условия, проявляют больший интерес к процессу обучения и конкретной теме [33].

Игровая ситуация в педагогике – это один из важнейших аспектов интерактивного обучения ребенка. Система применима не только по отношению к детям дошкольного возраста, но и в старшей школе. Со временем она трансформируется из учебной деятельности в общественно-полезную и профессиональную. Она оставляет отпечаток на процессе развития и формирования ребенка, позволяя охватить и запомнить больший объем информации.

Педагогу бывает сложно оптимально сочетать метод игры с другими направлениями в обучении. Эффективность состоит в том, что в игровой технологии сочетаются несколько значащих факторов. В ней есть свобода действий и четкое распределение обязанностей, напряженные моменты и развлечение, реальность и мистика, эмоции и рациональное мышление.

Игровые технологии в педагогике позволяют ребенку, будучи лично заинтересованным, отрабатывать навыки работы в команде, тем самым воспитывая в себе ответственность. Одна из задач педагога – выработка мотивации. Ребенок в процессе игры мотивирован собственной заданной целью, то есть, он в любом случае будет запоминать материал, поданный в ходе игры, ведь это нужно ему самому [32].

Понятие «игровые технологии» включает достаточно обширную группу приемов организации педагогического процесса в форме разных педагогических игр. В отличие от игр вообще, педагогическая игра обладает существенным признаком – четко поставленной целью обучения и



соответствующим ей педагогическим результатом, которые могут быть обоснованы, выделены в явном виде и характеризуются учебно-познавательной направленностью. Игровая форма занятий создается на уроках при помощи игровых приемов и ситуаций, выступающих как средство побуждения, стимулирования к учебной деятельности. Деятельность учащихся должна быть построена на творческом использовании игры и игровых действий в учебно-воспитательном процессе с младшими школьниками, наиболее удовлетворяющей возрастные потребности данной категории учеников [19].

В общественной деятельности игра выполняет следующую миссию:

- развлекательную (это главное назначение игры — развлечь, удовлетворить, вдохновить, пробудить интерес);
- коммуникативную: усвоение разновидностей общения; самовыражения в игре как полигоне человеческой практики;
- игротерапевтическую: переселение разных трудностей, появляющихся в других аспектах жизнедеятельности;
- распознавательная: обнаружение изменений нормативных поступков, самоизучение в игровой деятельности;
- корректировочная: введение положительных перемен в устройство индивидуальных данных;
- межнационального общения: освоение неделимых для всех людей социально-культурных ценностей; социализации: вовлечение в систему общественных взаимоотношений, освоение стандартов человеческого общежития.

Игровые виды деятельности воссоздаются на занятиях с помощью игровых способов и условий, которые выдвигаются как средство воздействия, подталкивание учеников к учебным занятиям. Осуществление игровых способов и ситуаций при занятиях в форме урока происходит по таким ведущим назначениям:

- наставническая задача ставится перед учениками в игровой форме;

- учебная работа зависит от правил игры;
- учебным материалом пользуются в качестве её средства, в учебную работу вносятся элементы соперничества, которая меняет дидактическую задачу в игровую;
- благополучное исполнение дидактического задания сплетается с результатом игры.

Виды образовательных игр делятся по видам деятельности на:

- физические (двигательные);
- мыслительные (умственные);
- рабочие, общественные и психологические.

По виду педагогического процесса распознаются следующие группы игр:

- учебные, тренировочные, осматривающие и обобщающие;
- познавательные, воспитательные, развивающие, социализирующие;
- репродуктивные, продуктивные, творческие;
- коммуникативные, диагностические, профориентационные, психотехнические и другие [39].

Можно сделать вывод, что игровая технология – это вид деятельности в условиях ситуаций, направленных на воссоздание и усвоение общественного опыта, в котором складывается и совершенствуется самоуправление поведением. Она строится как целостное образование, охватывающее определенную часть учебного процесса, объединенное общим содержанием сюжетом, персонажем. В неё включаются последовательно игры и упражнения, формирующие умение выделять основные, характерные признаки предметов сравнивать, сопоставлять их; группы игр на обобщение предметов по определенным признакам; группы игр, в процессе которых у младших школьников развивается умение отличать реальные явления от нереальных; группы игр, воспитывающих умение владеть собой, быстроту реакции на слово, фонематический слух смекалку и др. При этом игровой сюжет развивается параллельно основному содержанию обучения, помогает

активизировать учебный процесс, осваивать ряд учебных элементов. Составление игровых технологий из отдельных игр и элементов – забота каждого учителя начальной школы.

## 1.2 Основы языка программирования Python и сравнение с другими языками программирования

Python является не единственным достойным вниманием из языков программирования. Есть много и других языков на выбор, в том числе:

- Golang;
- Node.js;
- Java;
- Ruby;
- PHP.

### *Популярность Python в 2020 году*

С 2012 года популярность Python неуклонно растет, и эта тенденция, скорее всего, с каждым годом увеличивается [7].

По данным Stack Overflow Developer Survey 2020 Python занимает 1 место в рейтинге:

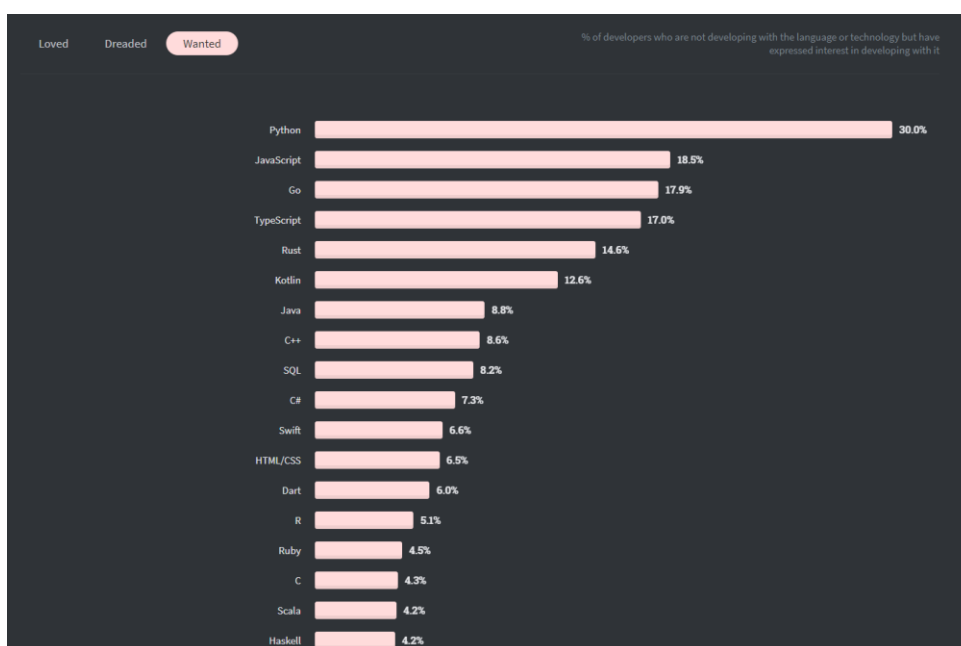


Рисунок 1 – Рейтинг Stack Overflow Developer Survey

Спрос и поддержка Python также находятся на подъеме, и, если верить прогнозам, Python еще на три раза обгонит преследующих.

*Является ли Python популярным?*

Да, благодаря популярности Python мы, скорее всего, найдем готовое решение для любой проблемы, с которой мы можем столкнуться. Сообщество энтузиастов Python велика, и они неустанно работают над улучшением языка каждый день [6].

Python также имеет ряд корпоративных спонсоров, стремящихся еще больше популяризировать этот язык. Среди них есть такие технологические гиганты, как Google, который сам использует Python.

Python предназначен для того, чтобы быть доступным. Это делает написание кода Python очень легким, а разработку программного обеспечения на Python качественным. Меньше времени тратится на изучение языка и больше времени тратится на создание вашего продукта [5].

*Многочисленные библиотеки и фреймворки*

Огромным преимуществом Python является широкий выбор библиотек и фреймворков, которые он предоставляет. Рынок библиотек улучшается, если мы будем использовать их, так как не будем кодировать функции вручную.

Библиотеки Python можно использовать во всем:

- data visualization;
- machine learning;
- data science;
- natural language processing;
- complex data analysis.

Фреймворки, которые помогают поднять наш проект с земли и сэкономить наше время и усилия. Существует множество фреймворков на выбор, в зависимости от наших потребностей, таких как:

- Django;
- Flask;
- Pyramid;

- Twisted;
- Falcon.

Одним из самых больших критических замечаний к Python является время выполнения, относительно медленное по сравнению с другими языками. Однако существует обходной путь для решения этой конкретной проблемы [8].

Когда производительность имеет приоритет, Python дает нам возможность интегрировать другие, более эффективные языки в наш код.

### *Простой синтаксис*

Python интуитивно понятен для чтения, потому что все коды написаны на английском языке. Кроме того, Python имеет четкий синтаксис и не требует такого количества строк кода, как Java или C, чтобы дать нам сопоставимые результаты.

### *Каковы преимущества высокой читабельности Python?*

Простота Python особенно полезна при чтении кода, поскольку код Python имеет меньше строк, поэтому просмотр занимает гораздо меньше времени. Это один из больших преимуществ [12].

## **Python vs. Golang**

### *Развитие без усилий*

Python настолько гибок и удобочитаем, что его можно понять без каких-либо предварительных знаний языка. Ни один из них не требует даже чтения одного учебника, чтобы следовать своему коду.

### *Сходство между Python и Golang*

Основное сходство между Python и Golang заключается в высокоуровневых типах.

Срезы и карты Golang напоминают списки Python, только статически типизированных.

Кроме того, мы перечислим в Python функции как диапазон в Golang.

### *Различия между Python и Golang*

Есть гораздо больше различий, чем сходства между Python и Golang, некоторые из них, вероятно, шокируют разработчиков Python.

Например, у Golang нет try-except, вместо этого позволяет функциям возвращать тип ошибки вместе с результатом. Поэтому перед использованием функции необходимо проверить, была ли возвращена ошибка.

Однако самое большое различие между этими двумя языками заключается в типизации. Python имеет динамическую типизацию, а Golang статическую. Python также является интерпретируемым языком, в отличие от Golang, который является компилируемым языком [13].

У Golang есть целый ряд других сюрпризов в запасе для разработчиков Python, чтобы узнать, в том числе:

- каналы (отправка сообщений между goroutines)
- структуры (составные типы).

#### *Высокая читабельность как Python, так и Golang*

Причина, по которой разработчики Python могут понять Golang без особых проблем, заключается в том, что дизайн Python и дизайн Golang основаны на схожих принципах.

Если мы сравним Python с руководящими принципами Golang, то заметим, что оба языка отдают приоритет простоте и минимизации беспорядка:

1. Python ценит читабельность, а синтаксис Golang приводит именно к высокой читабельности.
2. "Простой лучше, чем сложный" для Python, и то же самое достигается благодаря простоте Golang.
3. Статическая типизация

#### *Какой язык лучше: Python или Golang?*

Python — это отличный выбор для data science и для web. Между тем, поскольку Golang компилируется и статически типизируется, его производительность намного выше, чем у интерпретируемого и динамически типизируемого языка, такого как Python [15].

Самый оптимальный подход — это использовать Python и идти вместе.

*Вывод*

Сходство дизайна между Python и Golang делает переход с одного к другому понятным и удобным.

### **Python vs. Node.js**

Очень важно знать, что Node.js — это среда выполнения для JavaScript, а не язык программирования, как Python, Node.js использует один и тот же язык на фронтенде и бэкэнде.

#### *Преимущества Python над Node.js*

##### 1. Более подходит к новичкам

На более продвинутом уровне JavaScript может быть трудно понять для разработчиков с меньшим знанием над Node.js. Они могут совершать довольно распространенные ошибки, замедляя развитие в этом процессе.

##### 2. Больше приложений

Node.js в основном используется для интернета, а Python для всего.

Универсальность Python является одними из главных причин, почему этот язык отлично подходит для трендовых технологий, таких как data science.

##### 3. Хорошая реализация

Среды выполнения JavaScript и фреймворки реализуют этот язык по-разному. Node.js не является исключением. Экосистема JavaScript немного запутана — хотя и не так плохо, как раньше. У Python нет этой проблемы, поэтому он простой в использовании. Это также делает язык более быстрым для записи [16].

#### *Преимущества Node.js над Python*

##### 1. Более продвинутые разработчики

Надо знать JavaScript, если мы хотим использовать Node.js, поскольку мы имеем программу с одним и тем же языком на фронтенде и бэкэнде.

##### 2. Менее самоуверенная экосистема

Пакеты для Node.js часто просты и предназначены только для одной цели. Это подталкивает разработчиков к более тщательному размышлению о том, что они хотят использовать, и когда они хотят это использовать.

Из-за этого, Node.js требует, чтобы разработчики были более продвинутыми. Например, написание кода Python в Django далеко не так требовательно.

### 3. Быстрый рост и большое сообщество

С 2012 года Python постоянно хвалят за его большое сообщество и поддержку—и это правильно. Но времена его огромного преимущества перед фреймворками и библиотеками уже прошли.

В наши дни JavaScript так же хорошо поддерживается. Он продолжает расти без каких-либо признаков остановки и остается лидирующим среди наиболее динамично развивающихся языков в бизнесе.

#### *История развития Python и JavaScript*

Следовательно, растущая популярность JavaScript в последние годы принесла с собой бесконечные обновления и значительные проблемы с документацией. Опять же, Python здесь оказывается лучшим. Документация и охват Python превосходят Node.JS. Когда дело доходит до надежности, Python всегда был впереди JavaScript.

#### *Трендовые технологии*

Хаотичная экосистема JavaScript также делает Node.js слишком нестабильным и непредсказуемым, чтобы полагаться на трендовые технологии.

Из-за значительной показательности тенденций JavaScript технологии устаревают гораздо быстрее. Вот почему Node.js-рискованный выбор для новых технологических тенденций.

Python, с другой стороны, не представляет такого риска, поскольку он очень медленно вносит существенные изменения. Этот язык идеально подходит для современных технологий, таких как машинное обучение или Data science, с библиотечной поддержкой [17].



### *Производительность и скорость*

Node.js может бороться с выполнением большого количества задач одновременно. Если код написан не очень хорошо, то продукт будет работать плохо и работать медленно.

Это также может произойти с Python, но фреймворки Python, такие как Django, поставляются с готовыми решениями, которые помогают программному обеспечению выдерживать высокую нагрузку.

Это еще один пример того, как Python облегчает жизнь вашим разработчикам.

### *Вывод*

Учитывая все обстоятельства, чаша весов склоняется в пользу Python в одном отношении: он гораздо дружелюбнее для младших или неопытных разработчиков.

Но настоящая разница заключается в нашей команде разработчиков, а не в языке. Именно они решают успех или неудачу проекта, поэтому должны выбрать тот вариант, который лучше подходит.

### **Python vs Java**

Python — это интерпретируемый и динамически типизированный язык, в то время как Java — это компилируемый и статически типизированный язык.

Код Python не обязательно нужно компилировать перед запуском. С другой стороны, код Java с запуска должен быть скомпилирован из кода.

Проще говоря, это обычно означает, что Python имеет более быстрое время запуска и более медленное время выполнения, а Java имеет более медленное время запуска и более быстрое время выполнения [18].

Python идеально подходит для новичков. Этот язык чрезвычайно удобен для пользователя. Java требует высокого изучения над языком.

В двух словах, начало работы на Python занимает недели, а Java занимает месяцами.

### *Стабильность*

Существует предположение, что Java — это корпоративное решение для разработки программного обеспечения.

Корпоративы считают Java сильным и надежным языком из-за его большого объема кода. Они считают, что это делает язык более стабильным и безопасным, чем Python.

Однако это совсем неверно. Python также имеет обработку программных продуктов для крупных предприятий—трейдинга, в частности.

Называть Python нестабильным было бы несправедливо и ложно.

### *Скорость*

Создание продукта в Java может занять месяцы из-за его высокой сложности кода и объема. Следовательно, большие проекты, написанные на Java, часто длятся годами и требуют больше разработчиков в команде.

Python не имеет ни одной из этих проблем, благодаря своей молниеносной скорости разработки. Можно построить продукт с Python всего за несколько недель, закончить весь проект за несколько месяцев для этой работы нужно только малое количество разработчиков [20].

### *Ресурсы*

Разработка на Java — это более крупная инвестиция во всем мире; она требует больше времени и денег.

Python стоит дешевле, поэтому для большинства проектов он является предпочтительным выбором.

### *Трендовые технологии*

Ни один язык программирования не подходит лучше для трендовых технологий, чем Python.

Будь то искусственный интеллект или машинное обучение, особенности Python дают преимущество перед всеми другими языками.

Основная причина, по которой Python был принят в качестве языка перехода к трендовым технологиям, заключается в его обширной поддержке библиотек AI/ML.

### *Вывод*

Python легко читать, легко писать. С другой стороны, Java прекрасно подходит для решения действительно сложных задач.

## **Python vs Ruby**

### *Простота и творческий талант*

Python и Ruby позволяют разработчикам достигать схожих результатов при создании веб-приложений. Ruby в первую очередь используется для веб-приложений, Python способен на гораздо большее.

Эти два языка также различаются по своей философии и подходам к решению проблем. Использование Ruby сокращалось в течение последнего десятилетия, в то время как популярность Python резко возросла [22].

### *Сходство между Python и Ruby*

Сходство:

- серверная часть, высокий уровень и сценарии;
- динамически типизированный, кроссплатформенный и универсальный;

### *Различия между Python и Ruby*

Популярность

Python пользуется гораздо более высокими показателями среди разработчиков, чем Ruby.

Octoverse GitHub обнаружил, что популярность Ruby с каждым годом снижается. Он опустился с 5-го места до 10-го с 2014 до 2018.

Stack Overflow назвал Python "самым быстрорастущим основным языком программирования".

### *Философия*

Подход к решению задач — это самое большое различие между Python и Ruby. В то время как первый предлагает простые, единичные решения, второй обычно предлагает более одного способа и добиться чего-то.

Хотя это можно рассматривать как преимущество Ruby, на самом деле это может поставить под угрозу читаемость и простоту, а также затруднить отладку ошибок.

## *Вывод*

Все, что можно сделать на Ruby, можно и сделать на Python. Однако это правило не применяется в обратном порядке. Существует множество областей—такие как, машинное обучение и анализ данных—где Python имеет явное преимущество перед Ruby.

Несмотря на то, что популярность Ruby снижается, этот язык все еще может многое предложить, когда речь заходит о веб-разработке.

Однако, учитывая множество вариантов использования Python, выбор между Python и Ruby это не проблема. Динамичный рост Python, его применение во многих различных отраслях промышленности и простота использования явно делают его лучшим выбором.

## **Python vs. PHP**

### *Для чего используется PHP?*

PHP — это широко используемый язык с открытым исходным кодом, который стал технологией веб-сервера по умолчанию. Часто называют его “сценарии для интернета”, PHP поддерживает около 80% серверов веб-сайтов.

Хотя этот язык часто используется для “традиционных” веб-проектов, которые не требуют большого количества вычислений или новейших функций, он применяется повсеместно, от аутентификации пользователей через СУБД и до приложений реального времени.

PHP чрезвычайно прост в освоении для новичка, но он предлагает множество продвинутых функций, которые приводят к высшим результатам.

### *Для чего используется Python?*

Несмотря на то, что Python в настоящее время используется преимущественно для веб-разработки, эта область изначально не была предназначена.

Диапазон вариантов использования Python широк и продолжает расти. Он может быть успешно применен в области машинного обучения, анализа данных, статистики, науки и интернета вещей.

### *Преимущества Python над PHP*

## 1. Многосторонность

От анализа данных через модели машинного обучения и до мощных веб-приложений может сделать Python. С точки зрения универсальности он превосходит PHP.

## 2. Структура

Поскольку выпусков Python было меньше, чем PHP, он имеет тенденцию быть более организованным, безопасным и простым в обслуживании.

## 3. Популярность

Учитывая его применение в таких областях, как искусственный интеллект, машинное обучение и интернет вещей, а также широкий спектр применений, которые остаются за рамками PHP, Python в последние годы пользуется огромным всплеском популярности.

Несмотря на то, что PHP традиционно был более популярен, чем Python, в последнее время постепенно теряет свою популярность [24].

### *Преимущества PHP перед Python*

#### 1. Особенности

PHP устанавливается с большим количеством готовых функций, чем Python. Однако у Python библиотеки надо устанавливать вручную.

#### 2. Простота установки

PHP легче установить на любой платформе, чем Python. Кроме Linux системы.

### *Python или PHP для программного проекта?*

Несмотря на то, что и Python, и PHP являются зрелыми и хорошо зарекомендовавшими себя языками, которые пользуются широким использованием и поддержкой.

Однако, выбор между ними, решение всегда должно сводиться к индивидуальным требованиям проекта.

Если будет веб-сайт, блог или простой веб-сервис, конечный результат, скорее всего, будет одинаковым, независимо от того, используете ли вы Python.

Однако если объем вашего проекта более разнообразен и включает, например, машинное обучение, анализ данных или интернет вещей, то следует выбрать Python.

В общем, вы не можете ошибиться с Python. Поскольку он отлично подходит для веб-разработки и имеет множество других применений, не нужно будет беспокоиться о том, что в будущем вы не сможете расширить сферу своего проекта [25].

Кроме того, с его стремительно растущей популярностью в последние годы, Python останется в течение длительного времени в отличие от PHP.

Python:

- он популярен, быстр, удобочитаем, интуитивно понятен, прост и понятен;
- он предлагает массу полезных фреймворков и библиотек;
- он пользуется огромным и постоянно растущим сообществом программистов и энтузиастов.

### 1.3 Сравнение и основные возможности модулей Python

Python это высокоуровневый язык программирования, простой в изучении, объектно-ориентированный, модульный и подчеркнуто легко читаемый. Python широко применяется в образовательной сфере, для научных вычислений, больших данных и машинного обучения, в веб- и интернет-разработке, графике, GUI, играх и других направлениях [23].

Поскольку экосистема Python огромна, существуют бесчисленные библиотеки, упрощающие программирование на этом языке. Благодаря таким библиотекам становится возможным более легкое выполнение определенных задач без необходимости написания излишнего кода.

Python имеет множество библиотек для разработки программных решений - Django и Flask широко используются для веб-разработки, а NumPy и SciPy необходимы для решения задач Data Science. Разработчики

программных решений, владеющие Python, востребованы на рынке труда и занимают высокооплачиваемые позиции в крупнейших IT-корпорациях [28].

Начнем с библиотек общего назначения, таких как Requests, Simplejson, SQLAlchemy, NumPy, а затем обратимся к более “продвинутым” библиотекам для машинного обучения, научных данных, построения GUI и научных вычислений [37].

## 1. Pygame

Pygame предназначена для создания мультимедиа-приложений с играми включительно. Она построена на мощной Simple Directmedia Library (SDL). Эта библиотека среди прочего используется для обучения детей разработке игр. Имеет открытый исходный код.

Сила Pygame в том, что она не требует OpenGL, позволяет использование множественных CPU для работы в многоядерной системе для супервысокой производительности, а также использование оптимизированных кодов на C и Assembly для внутренних функций [37].

Pygame весьма портативна и может запускаться практически на любых платформах и операционных системах. Эту библиотеку скачали миллионы людей и очень вероятно, что это одна из лучших игровых библиотек.

## 2. Simplejson

Это еще одна очень популярная библиотека и одна из самых скачиваемых. Simplejson это быстрый, точный и расширяемый json кодировщик и декодировщик, который поддерживает Python 3.3+ с обратной совместимостью с Python 2.5.

Simplejson написан на Python и не имеет внешних зависимостей. Он также включает расширения, написанные на C для представлений с особо высоким быстродействием. Помимо дефолтной UTF-8, декодер Simplejson может оперировать строками JSON, поступающими в любой кодировке.

С дефолтным Kwargs для дампов кодировщик Simplejson могут быть использован для обеспечения сериализации даже неподдерживаемых объектов.

Библиотека Simplejson может осуществлять заключительную обработку JSON- объектов с помощью кваргов `object_hook` или `object_pairs_hook`. Реализация таких протоколов как JSON-RPC существенно облегчается с использованием Simplejson [34].

### 3. SQL Alchemy

Базы данных — неотъемлемая часть разработки приложений, а в мире Python SQLAlchemy — необходимая библиотека для работы с базами данных. Предлагая полный комплект устойчивых моделей предметных областей для высокоэффективного доступа к базам данных, она действует как набор инструментов SQL и объектно-реляционное отображение.

Вызывает интерес модель отображения данных, в которой различные классы отображаются в базу данных различными способами, что позволяет объектной модели и схеме базы данных быть изначально развязанными.

Разработчики получают полный контроль и обзор конструкции SQL, ничто не прячется за стенами упаковщиков.

Подход этой библиотеки гораздо более современный и эффективный, если сравнивать с доступными инструментами SQL / ORM, а потому в списке самых полезных библиотек Python она занимает высокое место.

После выхода последнего релиза SQLAlchemy выбивается в лидеры; ее используют такие организации как Freshbooks, Survey Monkey, Mozilla, reddit, Yelp и многие другие [38].

### 4. TensorFlow

TensorFlow идет за основными библиотеками и погружает вас мир машинного интеллекта. Это опенсорсная библиотека Python, успешно применяемая для произведения расчетов с использованием графов потоков данных.

Вычисление изначально представляется в форме графов, где каждый узел графа предназначен для выполнения математических операций. Фактическое вычисление, тем не менее, производится по запросу, что позволяет повысить продуктивность сложных расчетов.



Библиотека учитывает нужды комплексных вычислений: она обслуживает вычисления, распределенные на CPU/GPU и несколько систем, заботясь о дублировании [36].

TensorFlow это бесплатная библиотека с открытым исходным кодом. Она была разработана инженерами Google, работающими над проектом Google Brain.

## 5. Scrapy

Scrapy это фреймворк, использующийся для извлечения структурированных данных с вебсайтов. Это быстрая, простая, расширяемая, легкая в использовании библиотека. Ее использование включает добычу данных, обработку информации, поисковый робот, извлечение данных с помощью API и многое другое.

Scrapy «из коробки» обеспечивает поддержку извлечения данных из HTML/XML с использованием выражений Xpath и селекторов CSS, интерактивную оболочку консоли, экспорт данных в json, csv, xml, бэк-энд хранилище ftp, s3, локальное хранилище, управление cookies и сессиями, сжатие, кэширование, аутентификацию, user-agent спуфинг.

Scrapy позволяет вам прописывать правила для извлечения данных. Библиотека написана на Python, ее конструкция предполагает возможность расширения за счет нового функционала, и вы можете перенести ее для запуска на Linux, Windows, Mac и BSD.

## 6. Matplotlib

Библиотека двумерных числовых построений. Она отлично подходит для анализа данных и создания высококачественных фигур различного формата, включая интерактивные графики и схемы.

Matplotlib можно использовать, а скриптах, написанных на Python, в оболочке IPython и на множестве разнообразных серверов веб-приложений. Это значительно упрощает сложные задачи на построение. Написав небольшой кусочек кода, вы сможете генерировать графики, гистограммы, диаграммы разброса данных.

Модули `pyplot` и `Ipython combo` предлагают `MATLAB` в качестве интерфейса для построения простых графиков. Вы можете полностью контролировать вид линий, свойства системы координат и тому подобное. Это, несомненно, одна из лучших библиотек для числовых построений, доступных для `Python`.

Такие инструменты как `basemap`, `cartopy`, `mplot3d`, `axes_grid` увеличивают возможности `Matplotlib`.

Эта библиотека – детище Джона Хантера. Она применяется в тысячах исследовательских, научных и издательских проектов.

## 7. Scikit-Learn

Высокоуровневая библиотека, содержащая алгоритмы, подобные `random forest`, готовые к использованию в проектах, связанных с машинным обучением [40].

Языком `Scikit-learn` по большей части является `Python`, но также задействуется `Cython` для улучшения производительности в некоторых внутренних алгоритмах. Оболочка `Cython` вокруг `LIBSVM` используется для поддержки векторной реализации, а `LIBLINEAR` – для вычислительной регрессии и линейной опоры векторных машин. `Scikit-learn` также использует `CBLAS`, который является `C`-интерфейсом для пользования библиотекой `Basic Linear Algebra Subprograms (CBLAS)`.

Эта библиотека построена на `SciPy` и распространяется на условиях лицензии `3-Clause BSD` с открытым исходным кодом для исследований, а также для коммерческого использования.

## 8. Requests

Имея больше 26 тысяч звезд на `GitHub` и тысячи скачиваний ежедневно, `Requests` занимает первую строчку рейтинга самых популярных и полезных библиотек для разработки на `Python`. Эта библиотека используется в проектах `Spotify`, `Microsoft`, `NPR`, `Heroku`, `Amazon`, `BuzzFeed`, `Reddit`, `Twitter`, `Mozilla` (список можно продолжать еще долго).

Requests уменьшает тяжелую ручную работу и автоматизирует такие задачи как добавление строки запроса к URL, шифрование форм отправки данных, постоянное HTTP-соединение, объединение HTTP-соединений и т. д. путем использования urllib3.

Для облегчения разработки также обеспечивается автоматическое дешифрование контента. Библиотека предлагает загрузку других функций, таких как журнал аутентификации, элегантные key/value cookies, Unicode-представление для атрибута body объекта response, потоковое скачивание, тайм-аут соединения, поддержка netrc.

#### 9. Arrow

Arrow предназначена для преодоления недостатков встроенного в Python функционала даты и времени, который не совсем понятен и легок. Может быть использована в качестве прекрасной замены для модулей datetime и time в Python.

#### 10. wxPython

wxPython входит в тройку пользующихся наибольшей популярностью GUI-библиотек для Python (остальные две – PyQt и Tkinter). wxPython интуитивна для разработчиков python и проста в использовании, это прекрасная смесь из C++ wxWidget-ов и программирования на Python.

wxPython применяется в качестве расширяющего модуля Python и является кроссплатформенным набором инструментов, запускаемым на многих платформах без нужды в модификациях. Поддерживается на многих платформах, включая Unix, Macintosh OS X и Microsoft Windows (32 bit).

wxPython предлагает тысячи элементов, позволяющих с легкостью создавать трудоемкие и функциональные GUI приложения на Python [41].

Пакеты и библиотеки Python играют жизненно важную роль в карьере разработчика. Будь то Data Science, машинное обучение или любые другие аспекты мира программирования. Однако, в дополнение к нашему библиотек python, есть также много других библиотек и пакетов.

## ГЛАВА 2. РАЗРАБОТКА ПРАКТИЧЕСКИХ ЗАНЯТИЙ НА ОСНОВЕ МОДУЛЯ PYGAME ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

### 2.1 Анализ модуля Pygame и создание планов занятий

Pygame – это библиотека модулей для языка Python, созданная для разработки 2D игр был написан Питом Шиннерсом (Pete Shinnners). Начиная примерно с 2004/2005 года поддерживается и развивается сообществом свободного программного обеспечения. Также Pygame могут называть фреймворком. В программировании понятия "библиотека" и "фреймворк" несколько разные. Но когда дело касается классификации конкретного инструмента, не все так однозначно.

В любом случае, фреймворк является более мощным по-сравнению с библиотекой, он накладывает свою специфику на особенности программирования и сферу использования продукта. С точки зрения специфики Pygame – это фреймворк. Однако его сложно назвать "мощным инструментом". По своему объему и функционалу это скорее библиотека [37].

Pygame обладает высокой портативностью и работает практически на любой платформе и в любой операционной системе.

Она бесплатна выпущенная по лицензии LGPL, мы можем создавать с ее помощью открытые, бесплатные, условно-бесплатные и коммерческие игры.

Можно использовать pygame из командной строки, если нужна только обработка изображений и ввод данных с помощью джойстика или воспроизведения звуков.

Также существует понятие "игрового движка" как программной среды для разработки игр. По своему назначению Pygame можно считать игровым движком. В то же время, с точки зрения классификации программного обеспечения, Pygame является API для Питона к API библиотеки SDL.

API – это интерфейс (в основном набор функций и классов) для прикладного (часто более высокоуровневого) программирования, который

предоставляет, например, та или иная библиотека. SDL – это библиотека, которая работает с мультимедийными устройствами компьютера.

Игры событийно-ориентированны, также как любое приложение с графическим интерфейсом пользователя. Поэтому какие-никакие, но игры можно было бы писать с помощью Tkinter, в частности на его экземплярах холста. Но поскольку основное назначение библиотеки графического пользовательского интерфейса совсем другое, то пришлось бы изобретать велосипеды. В то время как библиотека, специально предназначенная для написания игр, уже содержит необходимые объекты, что упрощает разработку [30].

Например, чтобы определить, столкнулись ли два объекта, надо написать код, проверяющий совпадение координат. Это может быть непростой задачей, так как надо учесть области перекрытия, форму объектов и др. В то же время игровой движок может включать готовую функцию проверки коллизии (столкновения) с необходимыми опциями настройки.

При всем этом Pygame достаточно низкоуровневый игровой движок, если его можно так называть. Это значит, что многое в нем не остается за кадром, а дается программисту на доработку, вынуждает его понимать, как работают "шестеренки". Так в Pygame отсутствует эмуляция физических явлений. Если вам надо смоделировать движение с ускорением или по дуге, программируйте это сами, предварительно взяв из курса физики соответствующую формулу.

Игры относятся к мультимедийным приложениям. Однако, в отличие от других приложений этой группы, для них характерна сложная программная логика и нередко много математики, хотя достаточно простой, плюс эмуляция физических явлений. В играх программируется подобие искусственного интеллекта. В многопользовательской игре, хотя пользователи играют друг с другом, а не с ИИ, создаются виртуальные миры, существующие по законам, заложенными разработчиками [41].

В программном коде игры выделяют три основных логических блока:

- Отслеживание событий, производимых пользователем и не только им.
- Изменение состояний объектов, согласно произошедшим событиям.
- Отображение объектов на экране, согласно их текущим состояниям.

Эти три этапа повторяются в цикле бесчисленное количество раз, пока игра запущена [4].

Модули Pygame, несмотря на свою простоту и функциональные возможности, не подходят для создания сложных компьютерных игр, однако как часть игры, или реализацию простой игры – подходят отлично. Pygame является необходимым и доступным инструментом для ознакомления с процессом создания игр, игровых скриптов.

Пример анимации мяча с 7 строками кода:

```
import sys, pygame
pygame.init()
size = width, height = 320, 240
speed = [2, 2]
black = 0, 0, 0
screen = pygame.display.set_mode(size)
ball = pygame.image.load("intro_ball.gif")
ballrect = ball.get_rect()
while 1:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: sys.exit()
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]
    screen.fill(black)
    screen.blit(ball, ballrect)
    pygame.display.flip()
```

### Модули Pygame:

- `cdrom` — управление устройствами компакт-дисков и воспроизведения звука
- `cursors` — загрузка изображений курсора, включает стандартные курсоры
- `display` — управление окном или экраном
- `draw` — рисование графических примитивов на поверхности (класс `Surface`)
- `event` — управление событиями и очередью событий
- `font` — создание и отображение шрифтов TrueType
- `image` — сохранение и загрузка изображений
- `joystick` — управление джойстиком
- `key` — управление клавиатурой
- `mouse` — управление мышью
- `movie` — проигрывание mp3 клипов
- `sndarray` — управление звуками с помощью класса `Numeric`
- `surfarray` — управление изображениями с помощью класса `Numeric`
- `time` — управление таймерами

`transform` — изменение размеров, вращение и изменение ориентации изображений.

Практические занятия разработаны как дополнительное образование для школьников.

Объем учебного времени за весь курс составляет 7 часа.

Курс основан на практической части для детей, знакомых с языком программированием «Python» версии 3. Обучающиеся займутся собственно технологией создания игр. Общей функцией урока является целостное формирование и развитие личности школьника на основе развивающего и воспитывающего обучения.

На практических занятиях обучающиеся освоят язык программирование «Python» версии 3, на основе которых должны будут создать свою игру.

На основе курса домашние задания не задаются. Учащиеся выполняют самостоятельные работы на занятии.

Приобретенные знания, умения и навыки на данном курсе, учащиеся смогут применить их на других дисциплинах (предметах).

Детям будет интересна создание 2D-игры на языке программирования Python (пайтон), используя модули библиотеки PyGame. Для каждого урока в данном образовательном модуле представлен подробный краткосрочный план, который содержит задания, практические работы, раздаточные материалы. В уроках используются активные методы обучения и разноуровневые задания (дифференциация).

Тематика практических занятий охватывает широкий круг вопросов в области информационных технологий и технологий программирования: программирование в Python, программирование в модуле Pygame, мультимедийные технологии, компьютерная графика.

Таблица 1 – Календарно-тематический план

<b>Общая тема</b>	<b>Темы, содержание</b>	<b>Количество во часов</b>	<b>Цели обучения</b>
Создание 2D игры на языке программирования Python (пайтон)	Библиотека PyGame (пайгейм)	1	<ul style="list-style-type: none"><li>– подключать библиотеку PyGame (пайгейм)</li><li>– использовать готовые модули библиотеки PyGame (пайгейм)</li></ul>
	Задний фон и персонажи игры	2	<ul style="list-style-type: none"><li>– создавать задний фон игры</li><li>– загружать готовые персонажи для игры</li></ul>



	Анимирование персонажей	1	<ul style="list-style-type: none"> <li>– программировать движение персонажа</li> <li>– управлять персонажами с клавиатуры</li> </ul>
	Программирование условий	3	<ul style="list-style-type: none"> <li>– разрабатывать игру по готовому сценарию</li> <li>– реализовывать алгоритм подсчета результатов игры</li> </ul>

## 2.2 Разработка практических занятий

### Первое занятие

<b>Тема урока</b>	<b>Библиотека PyGame</b>
<b>Цели обучения, которые достигаются на данном уроке</b>	Подключать библиотеку PyGame (пайгейм); Использовать готовые модули библиотеки PyGame (пайгейм) для создания окна для игры.

<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>

<p>Начало урока 0-5 мин.</p>	<p>Организационный момент. Проверка присутствующих. В начале урока учитель задает наводящие вопросы по теме, разделу урока и использует метод «<b>Мозгового штурма</b>», дает возможность сформулировать разнообразные идеи для решения поставленных проблем.</p> <ul style="list-style-type: none"> <li>– Как создаются компьютерные игры (<i>2D или 3D</i>)?</li> <li>– Какие вы знаете конструкторы для создания игр?</li> <li>– Как можно создать простую игру на языке <i>Python</i>?</li> </ul> <p>После ответов учащихся учитель знакомит с темой «<b>Библиотека PyGame</b>» и целями урока. Объяснение учителя, что такое Pygame. PyGame – это кроссплатформенная библиотека языка Python, предназначенная для написания компьютерных игр и мультимедиа-приложений. Скачать пакет установки Pygame</p> <ol style="list-style-type: none"> <li>1. Главным при установке PyGame является соответствие его версии с версией самого Python. Например, версия PyGame – 1.9.1 подходит для версии 2.x – 2.6, например, интерпретатор языка Python 2.6.</li> <li>2. Устанавливаем Python 3.2, следуя командам установщика, скачиваем пакет PyGame 1.9.1.</li> </ol>	
--------------------------------------	--	--

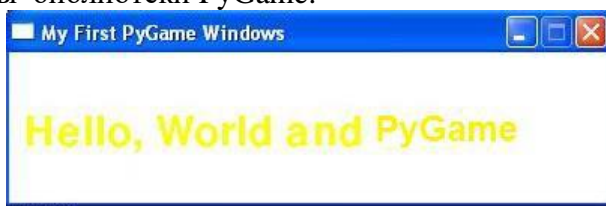
17-37 мин.

```
game_display = pygame.display.set_mode((display_width,  
display_height))
```

```
import pygame  
from pygame.locals import  
* pygame.init()  
game_display = pygame.display.set_mode((800,600))  
game_display =  
pygame.display.set_mode((display_width,  
display_height))
```

*ФО1 – Практическая работа на компьютере*  
**Первая программа на PyGame «Hello, World and PyGame»**

1. Создайте свою первую программу, используя готовый сценарий и библиотеку PyGame языка Python.
2. Инициализируйте библиотеку PyGame.
3. Создайте простой экран, используя готовые методы библиотеки PyGame.



Дескрипторы:

- ✓ подключает установленную библиотеку PyGame;
- ✓ создает простой экран, используя соответствующие методы;
- ✓ выводит текст на экран.

Тема урока	Задний фон и персонажи игры	
Цели обучения, которые достигаются на данном уроке	Создавать задний фон игры; Загружать готовые персонажи для игры	
Конец урока 37-40 мин.	<p>Учащиеся проводят рефлексию достижения целей урока на стикерах.</p> <p><b>Рефлексия</b></p> <p><b>Мы умеем...</b></p> 	

### Второе занятие

Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы

<p>Начало урока 0-5 мин.</p>	<p>Организационный момент. Проверка присутствующих. В начале урока учитель продолжает знакомство с библиотекой PyGame и озвучивает тему и цели урока. Затем учитель предлагает вспомнить материал предыдущего урока и провести мини-игру «Решение кроссворда».</p> <ol style="list-style-type: none"> <li>1. Кроссплатформенный язык программирования</li> <li>2. Пропущенное слово в следующей команде <b>PyGame._____set_mode()</b></li> <li>3. Пропущенное слово в методе <b>PyGame.event._____()</b></li> <li>4. Функция используемая с flip, которая обновляет только часть изображения</li> <li>5. Импорт библиотеки PyGame начинается с этого слова.</li> <li>6. Функция используемая в модуле display</li> </ol>	<p>Приложение 1</p>
<p>5-12 мин.</p>		

	<p style="text-align: center;">1. <b>P</b> y t h o n</p> <p>2. d i s p l a <b>y</b></p> <p style="text-align: center;">3. g e t</p> <p>4. u p d a t e</p> <p style="text-align: center;">5. i m p o r t</p> <p style="text-align: center;">6. s e t m o d e</p> <p>По завершении решения кроссворда и проверки ответов учитель задает наводящий вопрос по теме урока, показывая следующий код программы:</p> <pre><b>import pygame</b> <b>pygame.init()</b> <b>screen = pygame.display.set_mode((800, 600))</b> <b>clock = pygame.time.Clock()</b> <b>Что выводит данный код?</b></pre> <p>Ответ: Создается экран игры. Для отображения экрана нужно создать цикл: Основной <i>цикл</i>, который постоянно заполняет экран однотонным серым цветом и вызывает метод таймера <i>tick()</i> с частотой кадров.</p> <pre>while True:           #цикл для отображения экрана     screen.fill((192, 192, 192)) #заливает экран серым цветом     pygame.display.update()      #обновляет экран     clock.tick(60)              #частота кадра</pre>	
<p>Середина урока 12-20 мин.</p>	<p><b>Объяснение учителя</b> <b>Использование фонового изображения</b> PyGame очень хорошо работает с изображениями.</p> <pre>import pygame PyGame.init() screen = PyGame.display.set_mode((800, 600)) clock = PyGame.time.Clock() <b>background_image</b> = <b>PyGame.image.load('images/background.jpg')</b> while True: <b>screen.blit(background_image, (0, 0))</b> PyGame.display.update() clock.tick(60)</pre> <p>*PyGame.image.load – загружает фоновое изображение с помощью функции screen.blit(background_image,(0,0)) – затем выполняется блиттинг (копирование битов) изображения на экран в позицию (0,0)</p> <p><b>Результат вывода кода:</b></p>	<p>Приложение 2</p> <p>Приложение 3</p>

20-37 мин.



Для закрепления материала предлагается выполнить практическую работу.

### Практическое задание

Используя изученный материал урока, создайте экран игры размера на 500\*300 пикселей и загрузите изображение под названием «game\_background.jpg». Назовите ваше окно «New background». Задержка экрана 30 кадров в секунду.

Дескрипторы:

- ✓ использует нужные модули и функции библиотеки PyGame;
- ✓ создает простой экран, используя готовый фон.

### ОТВЕТ

```
File Edit Format Run Options Window Help
import pygame
pygame.init()
win = pygame.display.set_mode((500, 300))
pygame.display.set_caption("New background")
bg = pygame.image.load('game_background.jpg')
clock = pygame.time.Clock()


run=True
while run:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            run=False
    win.blit(bg, (0,0))
    pygame.display.update()
pygame.quit()
```



<p>Конец урока 37-40 мин.</p>	<p>Учащиеся проводят рефлексию достижения целей урока на стикерах.</p> <p><b>Рефлексия</b></p> <hr/> <p>Напишите на цветном стикере, где вам была не понятна тема урока, были ли вопросы; или все было понятно и интересно</p> 	<p>Цветные стикеры</p>
-----------------------------------	---	------------------------

Третье занятие

<p><b>Тема урока</b></p>	<p><b>Задний фон и персонажи игры (продолжение)</b></p>
<p><b>Цели обучения, которые достигаются на данном уроке</b></p>	<p>Создавать задний фон игры; Загружать готовые персонажи для игры.</p>

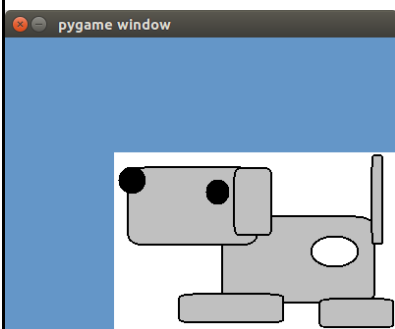
<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>
<p>Начало урока 0-4 мин.</p>     <p>4-14 мин.</p>	<p><b>Объяснение учителя</b> Загрузка готового персонажа</p>  <p>Функция <code>load ()</code> модуля <code>PyGame.image</code> загружает изображение и создает экземпляр <code>Surface</code>, на котором отображено это изображение.</p> <p>Основным форматом является <code>BMP</code>, однако если функция <code>PyGame.image.get_extended()</code> возвращает</p>	



истину, то можно загружать ряд других форматов: PNG, GIF, JPG и др.

Учитель объясняет возможности загрузки готового персонажа на примере:

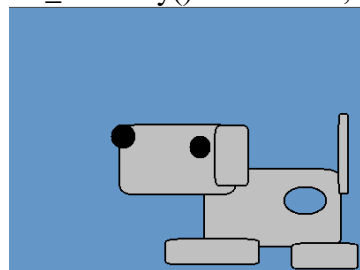
```
import pygame
pygame.init()
sc = pygame.display.set_mode((400, 300))
sc.fill((100, 150, 200))
dog_surf = pygame.image.load('dog.bmp')
dog_rect = dog_surf.get_rect(bottomright=(400, 300))
sc.blit(dog_surf, dog_rect)
pygame.display.update()
while 1:
for i in pygame.event.get():
if i.type == pygame.QUIT:
exit()
pygame.time.delay(20)
```



Если у изображения нет прозрачного слоя, но он необходим, то следует воспользоваться методом **set\_colorkey()** класса Surface:

```
dog_surf = pygame.image.load('dog.bmp')
dog_surf.set_colorkey((255, 255, 255))
```

Все пиксели, цвет которых совпадает с переданным в **set\_colorkey()** значением, станут прозрачными.






### Геометрические фигуры

В качестве персонажей можно использовать разные фигуры, которые легко можно нарисовать в PyGame.

В модуле **PyGame.draw** есть функции для отрисовки следующих фигур:

- ✓ прямоугольника (rect);
- ✓ многоугольника (polygon);
- ✓ круга (circle);
- ✓ эллипса (ellipse);
- ✓ дуги (arc);
- ✓ отрезка (line);
- ✓ отрезков (lines);
- ✓ сглаженного отрезка (anti-aliased line);

14-17 мин.

<p>Середина урока 17-37 мин.</p>	<p>✓ сглаженных отрезков (anti-aliased lines).</p> <p>Для закрепления материала предлагается выполнить практическую работу.</p> <p><b>Практическое задание</b></p> <ol style="list-style-type: none"> <li>Используя готовый экран игры с фоном, созданную в предыдущем уроке, добавьте готовый персонаж девочки.</li> <li>Дополнительное задание. Добавьте ее передвижение вправо. Дескрипторы: <ul style="list-style-type: none"> <li>✓ умеет добавлять в игру готовый персонаж;</li> <li>✓ создает его движение.</li> </ul> </li> </ol>  	<p>Приложение 1</p>
<p>Конец урока 37-40 мин.</p>	<p>Учащиеся проводят рефлекссию достижения целей урока на стикерах.</p> <p><b>Рефлексия</b></p> <hr/> <p>Напишите на цветном стикере, где вам была не понятна тема урока, были ли вопросы; или все было понятно и интересно</p> 	<p>Цветные стикеры</p>

Четвертое занятие

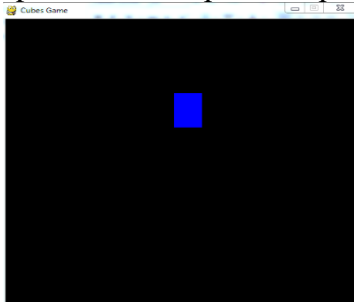
<b>Тема урока</b>	Анимирование персонажей
<b>Цели обучения, которые достигаются на данном уроке</b>	Программировать движение персонажа; Управлять персонажами с клавиатуры.

<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>
Начало урока 0-7 мин.	<p>Организационный момент. Проверка присутствующих. Учитель знакомит учащихся с темой и целями урока.</p> <p>Для повторения пройденного материала учитель задает вопросы. Назовите метод, используемый для загрузки фона для игры. Назовите модуль и его функцию для загрузки персонажа игры в PyGame. Назовите используемый модуль для создания геометрических фигур.</p> <p>Учащиеся коллективно отвечают на вопросы. После повторения учитель продолжает тему библиотеки.</p> <p><b>Объяснение учителя</b> <b>Создайте персонаж игры (прямоугольник синего цвета):</b></p> <ol style="list-style-type: none"> <li>1. Создаем экран, расположение и размер</li> </ol>	

7-17 мин.

прямоугольника, его скорость передвижения:

```
import pygame
pygame.init()
win = pygame.display.set_mode((500, 500)) #создаем
экран
pygame.display.set_caption("Cubes      Game")
#название игры
x=50 # координаты x оси
y=50 #координаты y оси
width=40 # ширина прямоугольника
height=60 # высота прямоугольника
speed=5 #скорость передвижения
```



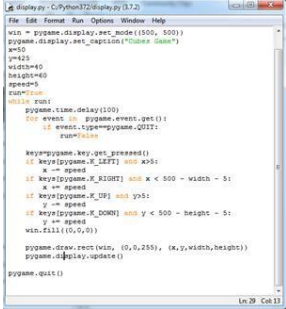
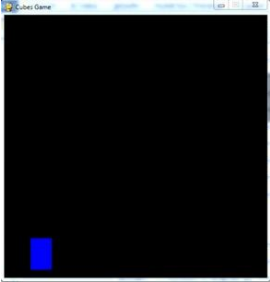
17-22 мин.

1. Создаем цикл для работы экрана и обработки событий:

```
run=True
while run: #цикл обработки экрана
    pygame.time.delay(100)
    for event in pygame.event.get(): #цикл перебора
    всех событий которые возможны
        if event.type==pygame.QUIT: #если
пользователь закрыл окно
            run=False # закрытие экрана
```

2. Рисуем персонаж и создаем его движение по нажатию на клавишу:

```
keys=pygame.key.get_pressed() #создаем
переменную для движения
if keys[pygame.K_LEFT]: #создаем
условия, если нажимаем стрелку влево
    x -= speed #если да минус один
шаг
if keys[pygame.K_RIGHT]: #создаем условия,
если нажимаем стрелку вправо
    x += speed #если да плюс один
шаг
if keys[pygame.K_UP]: #вверх
    y -= speed
if keys[pygame.K_DOWN]: #вниз
    y += speed
win.fill((0,0,0)) #заполняем экран
черным цветом
pygame.draw.rect(win, (0,0,255), (x,y,width,height))
#рисуем прямоугольник
pygame.display.update() #обновляем
```

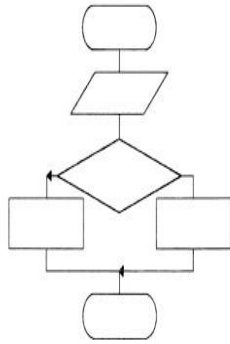
	<p>экран</p> <p style="text-align: center;">pygame.quit() #выход из игры</p>	
<p>Середина урока 22-37 мин.</p>	<p>Для закрепления материала предлагается выполнить практическую работу.</p> <p><b>Практическое задание</b></p> <p>Задание (уровень А):</p> <ol style="list-style-type: none"> <li>Используя готовый сценарий (раздаточный материал), создайте персонаж прямоугольника.</li> <li>Запрограммируйте его на движение с помощью клавиш.</li> <li>Добавьте код границ экрана, чтобы персонаж не выходил за границы.</li> </ol> <p>Задание (уровень В):</p> <ol style="list-style-type: none"> <li>Дополнительно к движению (уровень А) добавьте прыжки персонажа. (Раздаточный материал)</li> </ol> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>Дескрипторы:</p> <ul style="list-style-type: none"> <li>✓ создает простой персонаж игры;</li> <li>✓ использует нужные модули и функции;</li> <li>✓ пишет код для передвижения и прыжков персонажа.</li> </ul>	<p>Приложение 1</p>

Конец урока 37-40 мин.	Для проведения рефлексии предлагается игра «Продолжите фразу и передайте слово соседу». <p style="text-align: center;"><b>Продолжите фразу и передайте слово соседу</b></p> <div style="text-align: center;"> </div>	
---------------------------	--	--

Пятое занятие

Тема урока	Программирование условий (Часть 1)
Цели обучения, которые достигаются на данном уроке	Разрабатывать игру по готовому сценарию; Реализовывать алгоритм подсчета результатов игры

Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
Начало урока 0-4 мин.          4-9 мин.	<p>Организационный момент. Проверка присутствующих. Для повторения понятия алгоритмов и условий учитель задает вопросы и дает возможность учащимся вспомнить знания предыдущих классов.</p> <ul style="list-style-type: none"> <li>- Что такое алгоритмы?</li> <li>- Какие виды алгоритмов вы знаете?</li> <li>- Для чего необходимы блок-схемы?</li> </ul> <p>После повторения озвучивается тема урока.</p> <p>Урок разделен на три части для лучшего усвоения материала и достижения целей.</p> <p>Чтобы вспомнить построения блок-схемы, проводится «Разминка».</p> <p>Составьте блок-схему возведения четного числа в квадрат, нечетное число умножить на 10.</p>	



### Объяснение учителя

Для того чтобы учащиеся вспомнили условные операторы из 8 класса, учитель задает проблемную задачу.

9-22 мин.

В зависимости от возраста пользователя ему рекомендуются определенные фильмы. При этом выделяют группы от 3 до 6 лет, от 6 до 12, от 12 до 16, 16+. Итого 4 диапазона.

- Как бы вы решили задачу, используя конструкцию if-else?

Когда учащиеся дают ответ на вопрос, учитель демонстрирует свой вариант ответа.

Самый простой способ – последовательно проверять вхождение введенного числа-возраста в определенный диапазон с помощью следующих друг за другом условных операторов:

```

old = int(input('Ваш возраст: '))
print('Рекомендовано:', end=' ')

if 3 <= old < 6:
    print("Заяц в лабиринте")

if 6 <= old < 12:
    print("Марсианин")

if 12 <= old < 16:
    print("Загадочный остров")

if 16 <= old:
    print("Поток сознания")
  
```

```

Old = int(input('Ваш возраст: '))
print('Рекомендовано:', end=' ')
if 3 <= old < 6:
    print("Заяц в лабиринте")
if 6 <= old < 12:
    print("Марсианин")
if 12 <= old < 16:
    print("Загадочный остров")
if 16 <= old:
    print("Поток сознания")
  
```

Но он не эффективен. Почему?

Не эффективен потому, что каждый if в нем – это отдельно взятый оператор, не связанный с другими if. Процессор тратит время на обработку каждого из них, даже если в этом уже нет необходимости.

Затем учитель показывает и объясняет следующий вариант решения.

Вложение условных операторов друг в друга:

```
old = int(input('Ваш возраст: '))

print('Рекомендовано:', end=' ')

if 3 <= old < 6:
    print("Заяц в лабиринте")
else:
    if 6 <= old < 12:
        print("Марсианин")
    else:
        if 12 <= old < 16:
            print("Загадочный остров")
        else:
            if 16 <= old:
                print("Поток сознания")
```

```
old = int(input('Ваш возраст: '))
print('Рекомендовано:', end=' ')
if 3 <= old < 6:
    print("Заяц в лабиринте")
else:
    if 6 <= old < 12:
        print("Марсианин")
    else:
        if 12 <= old < 16:
            print("Загадочный остров")
        else:
            if 16 <= old:
                print("Поток сознания")
```

Поток выполнения этого варианта кода:

Сначала проверяется условие в первом if.

Если здесь было получено True, то тело этого if выполняется, а в ветку else мы даже не заходим, так как она срабатывает только тогда, когда в условии if возникает ложь.

**Множественное ветвление на одном уровне вложенности реализуется с помощью веток elif (иначе если).**

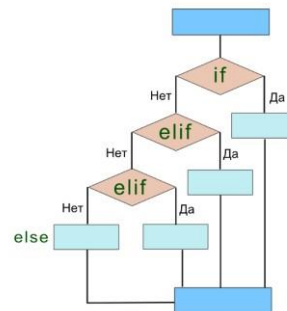
В отличие от else в заголовке elif обязательно должно быть логическое выражение так же, как в заголовке if. Перепишем нашу программу, используя конструкцию множественного ветвления:

```
old = int(input('Ваш возраст: '))

print('Рекомендовано:', end=' ')

if 3 <= old < 6:
    print("Заяц в лабиринте")
elif 6 <= old < 12:
    print("Марсианин")
elif 12 <= old < 16:
    print("Загадочный остров")
elif 16 <= old:
    print("Поток сознания")
```

```
old = int(input('Ваш возраст: '))
print('Рекомендовано:', end=' ')
if 3 <= old < 6:
    print("Заяц в лабиринте")
```





	<pre>elif 6 &lt;= old &lt; 12: print('Марсианин ') elif 12 &lt;= old &lt; 16: print ('Загадочный острое') elif 16 &lt;= old: print('Поток сознания')</pre>	
<p>Середина урока 22-37 мин.</p>	<p>Для закрепления материала предлагается выполнить парную работу.</p> <p><b>Практическое задание</b></p> <ol style="list-style-type: none"> <li>1. Дополните предыдущий код веткой else, которая обработает те случаи, когда пользователь вводит числа, не входящие в заданные четыре диапазона.</li> <li>2. Усовершенствуйте предыдущую программу, обработав исключение ValueError, возникающее, когда вводится не целое число.</li> <li>3. Напишите программу, которая запрашивает на ввод число. Если оно положительное, то на экран выводится цифра 1. Если число отрицательное, выводится – 1. Если введенное число это 0, то на экран выводится 0. Используйте в коде условный оператор множественного ветвления.</li> </ol> <p>Дескрипторы:</p> <ul style="list-style-type: none"> <li>✓ умеет использовать условные операторы;</li> <li>✓ пишет корректный код программы.</li> </ul>	<p>Приложение</p>
<p>Конец урока 37- 40 мин.</p>	<p>Для проведения рефлексии учащимся выдаются листочки с мясорубкой, чемоданом и корзиной. Нужно заполнить соответствующий рисунок в зависимости от усвоения материала. Мясорубка – информация переработана.</p> <div data-bbox="518 1473 1252 1848" style="text-align: center;"> <p><b>Feedback</b></p> <p><b>Чемодан, мясорубка, корзина</b></p> <p>Чемодан – всё, что пригодится в дальнейшем</p> <p>Мясорубка – информацию переработано</p> <p>Корзина – всё выброшу</p> </div>	<p>Раздаточны й материал</p>

## Шестое занятие

<b>Тема урока</b>	<b>Программирование условий (Часть 2)</b>
<b>Цели обучения, которые достигаются на данном уроке</b>	Разрабатывать игру по готовому сценарию; Реализовывать алгоритм подсчета результатов игры

<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>
Начало урока 0-4 мин.	<p>Организационный момент. Проверка присутствующих. Продолжение предыдущего урока</p> <p>Для привлечения внимания учащихся к теме урока урок начинается с вопроса. - При каком условии выполняется данный алгоритм? Обоснуйте ответ. Назовите условные операторы языка Python.</p> <p>If You find the key then Open the lock Else The lock can't be opened</p>	Приложение
4-9 мин.	<p>Приведите пример разветвляющейся модели встречающейся в повседневной жизни.</p> <p>Ученики дают свои варианты ответов. Учитель дает возможность обосновать свой ответ, приведя пример. Далее учитель раздает задания с условиями для создания игры в карты.,</p>	

<p>9-25 минут.</p>	<p>Карты делятся на 4 вида:</p> <ul style="list-style-type: none"> <li>✓ «кубики»;</li> <li>✓ «крестики»;</li> <li>✓ «ромбики» ;</li> <li>✓ «кружочки».</li> </ul> <p>Каждый вид имеет 9 достоинств:</p> <ul style="list-style-type: none"> <li>✓ «шесть»;</li> <li>✓ «семь»;</li> <li>✓ «восемь»;</li> <li>✓ «девять»;</li> <li>✓ «десять»;</li> <li>✓ «красный»;</li> <li>✓ «синий»;</li> <li>✓ «желтый»;</li> <li>✓ «зеленый».</li> </ul> <p>Два игрока выбирают карты наугад. Найти игрока, у которого карта «больше», при условиях их увеличения по порядку:</p> <p style="text-align: center;"><b>«кубики»&lt;«крестики»&lt;«ромбики»&lt;«кружочки»</b> и <b>«шесть»&lt;«семь»&lt;«восемь»&lt;«девять»&lt;«десять»&lt;«красный»&lt;«синий»&lt; «желтый» &lt; «зеленый»</b></p> <p>Присвоены условные номера: вид «кубики» – 1, вид «крестики» – 2, вид «ромбики» – 3, вид «кружочки» – 4; достоинствам: «шестерка» – 6, «семерка» – 7, ..., «десятка» – 10, «красный» – 11, «синий» – 12, «желтый» – 13, «зеленый» – 14.</p>	
<p>Середина урока 25-37 мин.</p>	<p>Далее задание выполняется в группах (деление на группы осуществляется путем выбора палочки с определенным цветом).</p> <p><b>Практическое задание</b></p> <ol style="list-style-type: none"> <li>1. Используя раздаточный материал с условиями игры, распишите алгоритм действий, происходящих в игре при данных условиях.</li> <li>2. Создайте код программы, реализовав разработанный алгоритм.</li> <li>3. Разработайте также вариант программы с неоднократным выбором карты участниками игры и подсчетом результатов.</li> <li>4. Презентуйте свой вариант реализации алгоритма остальным группам.</li> </ol> <p>Дескрипторы:</p> <ul style="list-style-type: none"> <li>✓ разбирается в разновидностях алгоритмов;</li> <li>✓ пишет код игры, используя операторов условия;</li> <li>✓ пишет алгоритм подсчета результатов игры с операторами условия;</li> </ul>	<p>Приложение</p>

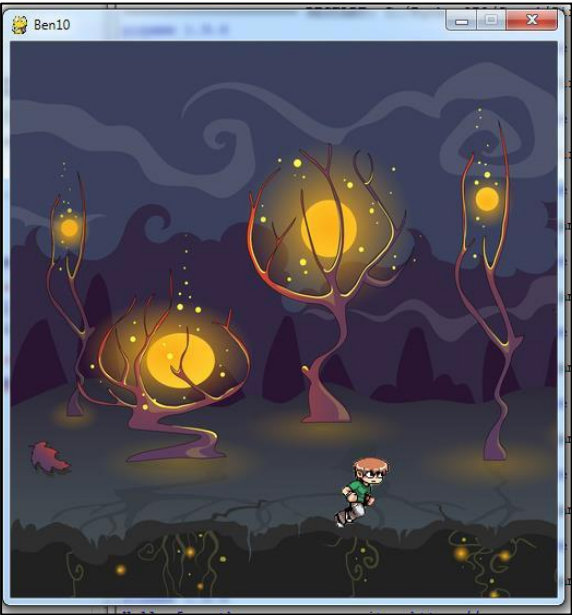
	✓ грамотно обосновывает свой вариант решения.	
Конец урока 37-40 мин.	<p>В качестве рефлексии ученикам предлагается «Синквейн».</p> <p>1 строка – заголовок, в который вносится ключевое слово, понятие, тема синквейна, выраженное в форм существительного.</p> <p>2 строка – два прилагательных. 3 строка – три глагола.</p> <p>4 строка – фраза, несущая определенный смысл. 5 строка – резюме, вывод, одно (два) слова, существительное (обычно).</p> <p>Например.</p> <p>Дружба Крепкая, верная Объединяет, поддерживает, лечит Дружба – великая сила. Согласие</p>	В тетрадях

#### Седьмое занятие

<b>Тема урока</b>	<b>Программирование условий (Часть 3)</b>
<b>Цели обучения, которые достигаются на данном уроке</b>	Разрабатывать игру по готовому сценарию; Реализовывать алгоритм подсчета результатов игры

<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>

<p>Начало урока 0-2 мин.</p> <p>2-7 мин.</p>	<p>Организационный момент. Проверка присутствующих. Продолжение предыдущего урока</p> <p>Так как это последний урок раздела, учитель предлагает ученикам заполнить таблицу «ЗИУ» (<b>З</b>наю. <b>И</b>нтересуюсь. <b>У</b>мею (<b>У</b>же научился)). Отвечая на вопросы таблицы, учащиеся вспомнят все пройденные темы раздела и легко определят, что им нужно повторить.</p>																	
	<table border="1"><thead><tr><th data-bbox="517 1173 769 1240"></th><th data-bbox="769 1173 861 1240">Знаю</th><th data-bbox="861 1173 1031 1240">Интересуюсь</th><th data-bbox="1031 1173 1273 1240">Умею (Уже научился)</th></tr></thead><tbody><tr><td data-bbox="517 1240 769 1469">Как подключить и инициализировать библиотеку PyGame ?</td><td data-bbox="769 1240 861 1469"></td><td data-bbox="861 1240 1031 1469"></td><td data-bbox="1031 1240 1273 1469"></td></tr><tr><td data-bbox="517 1469 769 1621">Как создать простой экран игры?</td><td data-bbox="769 1469 861 1621"></td><td data-bbox="861 1469 1031 1621"></td><td data-bbox="1031 1469 1273 1621"></td></tr><tr><td data-bbox="517 1621 769 1697">Как создать фон игры?</td><td data-bbox="769 1621 861 1697"></td><td data-bbox="861 1621 1031 1697"></td><td data-bbox="1031 1621 1273 1697"></td></tr></tbody></table>		Знаю	Интересуюсь	Умею (Уже научился)	Как подключить и инициализировать библиотеку PyGame ?				Как создать простой экран игры?				Как создать фон игры?				
	Знаю	Интересуюсь	Умею (Уже научился)															
Как подключить и инициализировать библиотеку PyGame ?																		
Как создать простой экран игры?																		
Как создать фон игры?																		

<p>7-22 мин.</p>	<table border="1" data-bbox="512 58 1246 349"> <tr> <td data-bbox="512 58 767 188">Как добавить готового персонажа игры или создать его из геометрических фигур?</td> <td data-bbox="767 58 863 188"></td> <td data-bbox="863 58 959 188"></td> <td data-bbox="959 58 1054 188"></td> </tr> <tr> <td data-bbox="512 188 767 349">Как добавить движение персонажа? Как передвигать персонажа с помощью клавиатуры?</td> <td data-bbox="767 188 863 349"></td> <td data-bbox="863 188 959 349"></td> <td data-bbox="959 188 1054 349"></td> </tr> </table> <p>После заполнения таблицы повторяется тот материал, который они не запомнили.</p>	Как добавить готового персонажа игры или создать его из геометрических фигур?				Как добавить движение персонажа? Как передвигать персонажа с помощью клавиатуры?				
Как добавить готового персонажа игры или создать его из геометрических фигур?										
Как добавить движение персонажа? Как передвигать персонажа с помощью клавиатуры?										
<p>Середина урока 22-37 мин.</p>	<p><b>Суммативная работа за раздел «Создание 2D-игры на языке программирования Python»</b></p> <p><b>Практическое задание</b></p> <p>Создайте мини-игру «Ben10», используя раздаточный материал из готового спрайта Бена.</p>  <p>Дескрипторы:</p> <ul style="list-style-type: none"> <li>✓ умеет добавлять в игру готовый персонаж;</li> <li>✓ создает его движение;</li> <li>✓ загружает фон экрана.</li> </ul>	<p>Раздаточный материал</p>								
<p>Конец урока 37-40 мин.</p>	<p><b>Рефлексия.</b> «Оцени себя сам»</p> <p>Вопросы</p> <p>Что было трудным?</p> <p>В чем требуется помощь?</p> <p>Что интересного было?</p>									

## 2.3 Апробация результатов исследования

Апробация учебно-методических материалов была проведена в МБУ ЦТТ «Центр технического творчества», мной было проведено 14 уроков по монтажу видео и создание игр через модуль Pygame в языке программирования Python.

В группе 10 обучающихся: 6 мальчиков и 4 девочки. Работоспособность, познавательный уровень развития учащихся средние, внимание устойчивое.

Уровень детей до занятий и после развивающих уроков существенно отличается. После проведения работы с учащимися показатели творческих способностей существенно изменился. Обучающиеся продемонстрировали позитивные изменения.

Следовательно, сделанные проекты, нацеленная на развитие творческих способностей имеет свои положительные результаты.

Представленная система уроков и созданные условия способствуют развитию творческих способностей детей школьного возраста. Эти данные позволяют сделать вывод о том, что программирование игр положительно влияют на развитие творческих способностей.

Первый проект которые сделали дети это пиксель которое осуществляется управлением через клавиатуры.

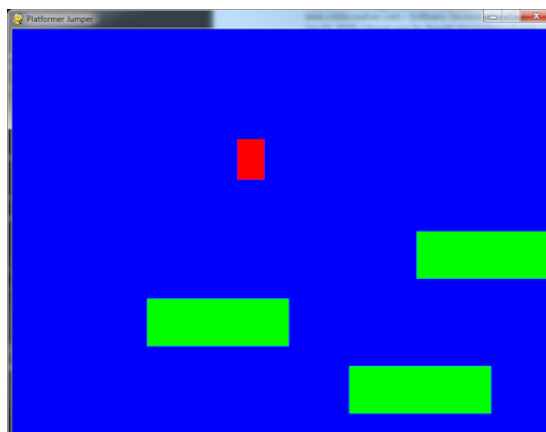


Рисунок 2 – Движущиеся пиксели

Во втором проекте обучающиеся использовали задний фон, добавили готового персонажа и создали его движение и создали простой персонаж игры,

использовали нужные модули и функции, написали код для передвижения и прыжков персонажа.



Рисунок 3 – Мини игра Ben10

```
game2.py - C:/Python372/Game2/game2.py (3.7.2)
File Edit Format Run Options Window Help
import pygame
pygame.init()
win = pygame.display.set_mode((500, 500))
pygame.display.set_caption("Ben10")

walkRight = [pygame.image.load('right_1.png'),
pygame.image.load('right_2.png'), pygame.image.load('right_3.png'),
pygame.image.load('right_4.png'), pygame.image.load('right_5.png'),
pygame.image.load('right_6.png')]

walkLeft = [pygame.image.load('left_1.png'),
pygame.image.load('left_2.png'), pygame.image.load('left_3.png'),
pygame.image.load('left_4.png'), pygame.image.load('left_5.png'),
pygame.image.load('left_6.png')]

bg = pygame.image.load('picture.jpg')
playerStand = pygame.image.load('idle.png')

clock = pygame.time.Clock()

x=50
y=200

```

Рисунок 4 – Кодинг мини игры Ben10

Можно сделать вывод, что представленные учебно-методические материалы, подготовленными с использованием модуля Pygame на языке программирования Python, доказали эффективность при обучении школьников созданию игр.



## Заключение

Игровая технология – это вид деятельности в условиях ситуаций, направленных на воссоздание и усвоение общественного опыта, в котором складывается и совершенствуется самоуправление поведением. Она строится как целостное образование, охватывающее определенную часть учебного процесса, объединенное общим содержанием сюжетом, персонажем. В неё включаются последовательно игры и упражнения, формирующие умение выделять основные, характерные признаки предметов сравнивать, сопоставлять их; группы игр на обобщение предметов по определенным признакам; группы игр, в процессе которых у младших школьников развивается умение отличать реальные явления от нереальных; группы игр, воспитывающих умение владеть собой, быстроту реакции на слово, фонематический слух смекалку и др. При этом игровой сюжет развивается параллельно основному содержанию обучения, помогает активизировать учебный процесс, осваивать ряд учебных элементов. Составление игровых технологий из отдельных игр и элементов – забота каждого учителя начальной школы.

Итоги игры выступают в двойном плане – как игровой и как учебно познавательный результат. Дидактическая функция игры реализуется через обсуждение игрового действия, анализ соотношения игровой ситуации как моделирующей, ее соотношения с реальностью. Важнейшая роль в данной модели принадлежит заключительному ретроспективному обсуждению, в котором учащиеся совместно анализируют ход и результаты игры, соотношение игровой (имитационной) модели и реальности, а также ход учебно-игрового взаимодействия

Представленные учебно-методические материалы, подготовленными с использованием модуля Pygame на языке программирования Python, доказали эффективность при обучении школьников созданию игр.

Python представляет популярный высокоуровневый язык программирования, который предназначен для создания приложений

различных типов. Это и веб-приложения, и игры, и настольные программы, и работа с базами данных.

Представленная система уроков и созданные условия способствуют развитию творческих способностей детей школьного возраста. Эти данные позволяют сделать вывод о том, что программирование игр положительно влияют на развитие творческих способностей.

## Список использованной литературы

1. Mertens, Nick Top 4 Reasons Python is So Popular in 2020 / Nick Mertens. — Текст : электронный // go skills : [сайт]. — URL: <https://www.goskills.com/Development/Articles/Why-is-Python-so-popular> (дата обращения: 15.03.2020).
2. Top 10 Reasons Why Python is So Popular With Developers in 2020. — Текст : электронный // upGrad blog : [сайт]. — URL: <https://www.upgrad.com/blog/reasons-why-python-popular-with-developers/> (дата обращения: 03.05.2020).
3. Бейдер, Дэн Чистый Python. Тонкости программирования для профи / Дэн Бейдер. — Санкт Петербург : Питер, 2018. — 320 с. — Текст : непосредственный.
4. Белолобский, М. М. Изучение модуля для создания игр «Pygame» языка программирования Python/ М. М. Белолобский, М. А. Сорочинский // Современное образование: традиции и инновации. — 2020. — № 2. — С. 220.
5. Бизли, Дэвид Python. Подробный справочник. 4-е издание / Дэвид Бизли. — Москва : Символ-Плюс, 2018. — 140 с. — Текст : непосредственный.
6. Бриггс, Джейсон Python для детей. Самоучитель по программированию: [перевод с англ. Станислава Ломакина] / Джейсон Бриггс. — Москва : Манн, Иванов и Фербер, 2017. — 320 с. — Текст : непосредственный.
7. Бубнов, Илья 5 причин, почему Python всё еще крут / Илья Бубнов. — Текст : электронный // Geek brains : [сайт]. — URL: [https://geekbrains.ru/posts/why\\_love\\_python](https://geekbrains.ru/posts/why_love_python) (дата обращения: 15.03.2020).
8. Васильев, А. Н. Python на примерах / А. Н. Васильев. — Москва : Наука и техника, 2016. — 432 с. — Текст : непосредственный.
9. Гайнуллова, Ф. С. Игровые технологии в образовании дошкольников и младших школьников / Ф. С. Гайнуллова, И. С. Сергеева. — Москва : КноРус, 2018. — 48 с. — Текст : непосредственный.

10. Герасимова, О. Е. Игровые технологии как средство развития ориентировки в пространстве у детей 4-6 лет / О. Е. Герасимова. — Москва : Учитель, 2016. — 63 с. — Текст : непосредственный.

11. Грамаков Д.А. Обучение программированию бакалавров педагогического образования в эпоху цифровых трансформаций // Педагогическое образование и наука. №6. 2019. С. 30-32.

12. Гуреев, Александр Python становится самым популярным языком программирования в мире / Александр Гуреев. — Текст : электронный // Хабр : [сайт]. — URL: <https://habr.com/ru/post/436626/> (дата обращения: 15.03.2020).

13. Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. — Москва : ДМК Пресс, 2017. — 284 с. — Текст : непосредственный.

14. ИТ: обзор рынка вакансий и топ-15 специальностей. — URL: <https://hh.ru/article/24562> (дата обращения: 12.02.2020).

15. Кудлай, Андрей Python — популярный язык программирования / Андрей Кудлай. — Текст : электронный // Web for Myself : [сайт]. — URL: <https://webformyself.com/python-populyarnyj-yazyk-programmirovaniya/> (дата обращения: 15.03.2020).

16. Лутц М. Изучаем Python: четвертое издание // М. Лутц. - СПб.: Питер, 2011. – 1280с.

17. Лутц, Марк Python. Карманный справочник / Марк Лутц. — 5-е изд.; [перевод с англ. И.В. Берштейна] — Москва : И.Д.Вильямс, 2016. — 320 с. — Текст : непосредственный.

18. МакГрат, Майк Программирование на Python для начинающих: [перевод с англ. М.А. Райтмана] / Майк МакГрат. — Москва : Эксмо, 2016. — 192 с. — Текст : непосредственный.

19. Михайленко, Т. М. Игровые технологии как вид педагогических технологий / Т. М. Михайленко. — Текст: непосредственный // Педагогика: традиции и инновации: материалы I Междунар. науч. конф. (г. Челябинск, октябрь 2011 г.). — Т. 1. — Челябинск: Два комсомольца, 2011. — С. 140-146.

— URL: <https://moluch.ru/conf/ped/archive/19/1084/> (дата обращения: 27.05.2020).

20. Модули деления. — Текст : электронный // foobarcom : [сайт]. — URL: <https://foobar.com/questions/80306/find-the-division-remainder-of-a-number> (20.02.2020).

21. Шлюссер, Таня Автостопом по Python / Таня Шлюссер, Кеннет Рейтц. — Санкт-Петербург : Питер, 2017. — 336 с. — Текст : непосредственный.

22. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. — Санкт – Петербург : БХВ - Петербург, 2016. — 832 с. — Текст : непосредственный.

23. Пэйн, Брайсон Python для детей и родителей. Играй и программируй / Брайсон Пэйн. — 1-е изд. — Москва : Э, 2017. — 352 с. — Текст : непосредственный.

24. Рашка, С. Python и машинное обучение / С. Рашка. — Москва : ДМК Пресс, 2017. — 418 с. — Текст : непосредственный.

25. Роберт, Седжвик Программирование на Python: базовый курс / Седжвик Роберт, Уэйн Кевин, Дондеро Роберт. — Москва : Вильямс, 2020. — 736 с. — Текст : непосредственный.

26. Сайт «Abitu.net». – URL: <https://abitu.net/conference/1781> (Дата обращения: 20.04.2020)

27. Сайт «Kgeu.ru». – URL: <https://kgeu.ru/News/Item/17/8851> (Дата обращения 17.04.2020)

28. Сайт «Python.org». – URL: <https://www.python.org> (дата обращения: 30.02.2020).

29. Сайт «Tiobe.com». – URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 02.05.2020).

30. Сайт «Skillbox.ru». – URL: [https://skillbox.ru/media/code/dlya\\_chego\\_nuzhen\\_python/](https://skillbox.ru/media/code/dlya_chego_nuzhen_python/)

31. Сайт «spravochnick.ru». – URL: [https://spravochnick.ru/pedagogika/teoriya\\_obucheniya/igrovyte\\_tehnologii\\_obucheniya/](https://spravochnick.ru/pedagogika/teoriya_obucheniya/igrovyte_tehnologii_obucheniya/)
32. Сайт «Zaochnik.com». – URL: <https://zaochnik.com/spravochnik/pedagogika/teoriya-obucheniya/igrovyte-tehnologii-obucheniya/>
33. Сергеева, М. Е. Игровые технологии на уроках и во внеурочной деятельности. 5-9класс / М. Е. Сергеева. — Москва : Учитель, 2017. — 94 с. — Текст : непосредственный.
34. Силен, Дэви Основы Data Science и Big Data. Python и наука о данных / Дэви Силен, Арно Мейсман, Мохамед Али. — Санкт Петербург : Вильямс, 2018. — 336 с. — Текст : непосредственный.
35. Скуратова, Е.А. Игровые технологии / Скуратова Екатерина. — Москва : Издательские решения, 2020. — 164 с. — Текст : непосредственный.
36. Сысоева, М. В. Программирование для нормальных с нуля на языке Python / М. В. Сысоева, И. В. Сысоев. — 1-е изд. — Москва : Базальт СПО, 2018. — 180 с. — Текст : непосредственный.
37. Хеллман, Даг Стандартная библиотека Python 3. Справочник с примерами / Даг Хеллман. — Москва : Вильямс, 2018. — 1376 с. — Текст : непосредственный.
38. Холден, Стив Python. Справочник. Полное описание языка / Стив Холден, Анна Рейвенскрофт, Алекс Мартелли. — Москва : Диалектика, 2018. — 896 с. — Текст : непосредственный.
39. Хуторской, А. В. Педагогика. Учебник / А. В. Хуторской. — Санкт-Петербург : Питер, 2019. — 608 с. — Текст : непосредственный.
40. Что такое Python: чем он хорош, где пригодится и как его выучить. — Текст : электронный // Медиа нетологии : [сайт]. — URL: <https://netology.ru/blog/python> (дата обращения: 24.04.2020)
41. Шелудько В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие // Южный федеральный университет. – Ростов-на Дону; Таганрог: Издательство Южного федерального университета. 2017. 107с.

# Приложения

## Приложение 1

### Приложение 1. Практическая работа Первая программа на PyGame «Hello, World and PyGame»

1. Запускаем программу для создания игры (Python).
2. Импортируем библиотеку PyGame в наш код:

```
import pygame, sys  
from pygame.locals import *
```

3. Инициализируем библиотеку:

```
PyGame.init()
```

4. Нужно создать окно, на котором будет формироваться кадр, используя метод:

```
PyGame.display.set_mode(resolution=(0,0), flags=0, depth=0)
```

где **resolution** – это список чисел, в котором хранится размер окна по оси **x** и **y**; **flags** – флаг окна (полный экран, изменяемые размеры и т. д);  
**depth** – глубина цвета.

В нашем коде инициализируется окно размером 400x100 с глубиной цвета – 32 следующим кодом:

```
screen = PyGame.display.set_mode((640,360),0,32)
```

5. Зададим заголовок для окна. Делается это функцией

```
PyGame.display.set_caption(title, icontitle=None)
```

где **title** – строка с заголовком окна;

**icontitle** – иконка в заголовке.

Иконку используем только в следующем тексте в заголовке окна.

```
PyGame.display.set_caption(«My First PyGame Windows»)
```

6. На этом инициализация данных еще не закончена. Для того чтобы вывести текст в окне, надо знать несколько значений – сам текст, его координаты, размер шрифта.

Для текста определим переменную **helloText**, а для координат – переменные **x** и **y**, для размера шрифта – **fontSize**. Вот код инициализации этих переменных:

```
helloText = «Hello, World and  
PyGame» (x,y, fontSize) =  
(10,40,14)
```

7. Теперь нужен шрифт. Шрифт создается методом

```
PyGame.font.SysFont(name, size, bold=False,  
italic=False) где name – название шрифта,  
установленного в системе; size – размер;
```

**bold/italic** – свойства шрифта.

Используем обычный шрифт с названием «None» и размером, который хранится в переменной **fontSize**. Код следующий:

```
myFont = PyGame.font.SysFont(«None», fontSize)
```

8. Цвет шрифта. Создается он не сложно, это последовательность из трех чисел, каждое число – это вектор цвета в системе RGB, минимальное значение для него – 0, а максимально – 255. Создадим желтый цвет:

```
fontColor = (255,255,0)
```

Так же нам понадобится еще цвет фона окна. Назовем его **bgColor**. Определяется он так же, как и цвет шрифта, только он будет уже белым, а не желтым:

```
bgColor = (255,255,255)
```

9. Нельзя на окне взять и написать текст. Сначала его надо преобразовать в изображение и лишь только потом рисовать это изображение в окне. Для хранения изображения мы определим переменную с именем **fontImage**. А метод объекта **myFont**, который преобразует текст в картинку, выглядит следующим образом:

```
render(text, antialias, color, background=None)
```

где **text** – текст;

**antialias** – сглаживание шрифта;

**color** – цвет текста;

**background** – цвет фона (не обязательный параметр).

Зная это, мы можем написать следующий код, в котором определяем изображение с текстом:

```
fontImage = myFont.render(helloText, 0, (fontColor))
```

10. Последний пункт в инициализации данных – создание цикла отображения кадров. Флаг будет иметь имя **mainLoop** и содержать в себе логическое значение True, а пока код инициализации флага цикла выглядит следующим образом:

```
mainLoop = True
```

11. Теперь нужно проверить флаг главного цикла. Если этот флаг имеет значение True, то продолжается цикл, в котором идет обработка событий, формирование кадра и отображение этого кадра, иначе будет происходить освобождение ресурсов и выход из программы. Флаг проверяется следующим кодом:

```
while mainLoop:
```

12. Для обработки событий будет введен итератор **event**. Он будет проходить по списку, который возвращает метод **PyGame.event.get()**, и в зависимости от значения итератора будет обрабатываться определенное событие, например, если итератор содержит значение QUIT, то это значит, что пользователь закрыл окно и следует поменять флаг главного цикла, чтобы освободить ресурсы. Код следующий:

```
for event in  
pygame.event.get(): if  
event.type == QUIT:  
mainLoop = False
```

13. Теперь следует сформировать кадр. Для начала закрасим кадр цветом заднего фона. За это отвечает метод

```
fill(color, rect=None, special_flags=0) объекта screen
```

Как видно, этот метод в параметрах принимает список из трех чисел, который у нас уже есть и называется он **bgColor**, так что код заливки выглядит так:

```
screen.fill(bgColor)
```

14. Окончательная часть цикла – это отображение кадра. За это отвечает метод **PyGame.display.update()**, не требующий никаких параметров в данном случае, поэтому просто добавляем его в код:

```
PyGame.display.update()
```

15. Чтобы освободить ресурсы, необходимо закрыть **PyGame**. Делается это методом **PyGame**



.quit(). У этого метода в качестве параметра ничего не указывается, нам нужно просто написать его в коде программы:

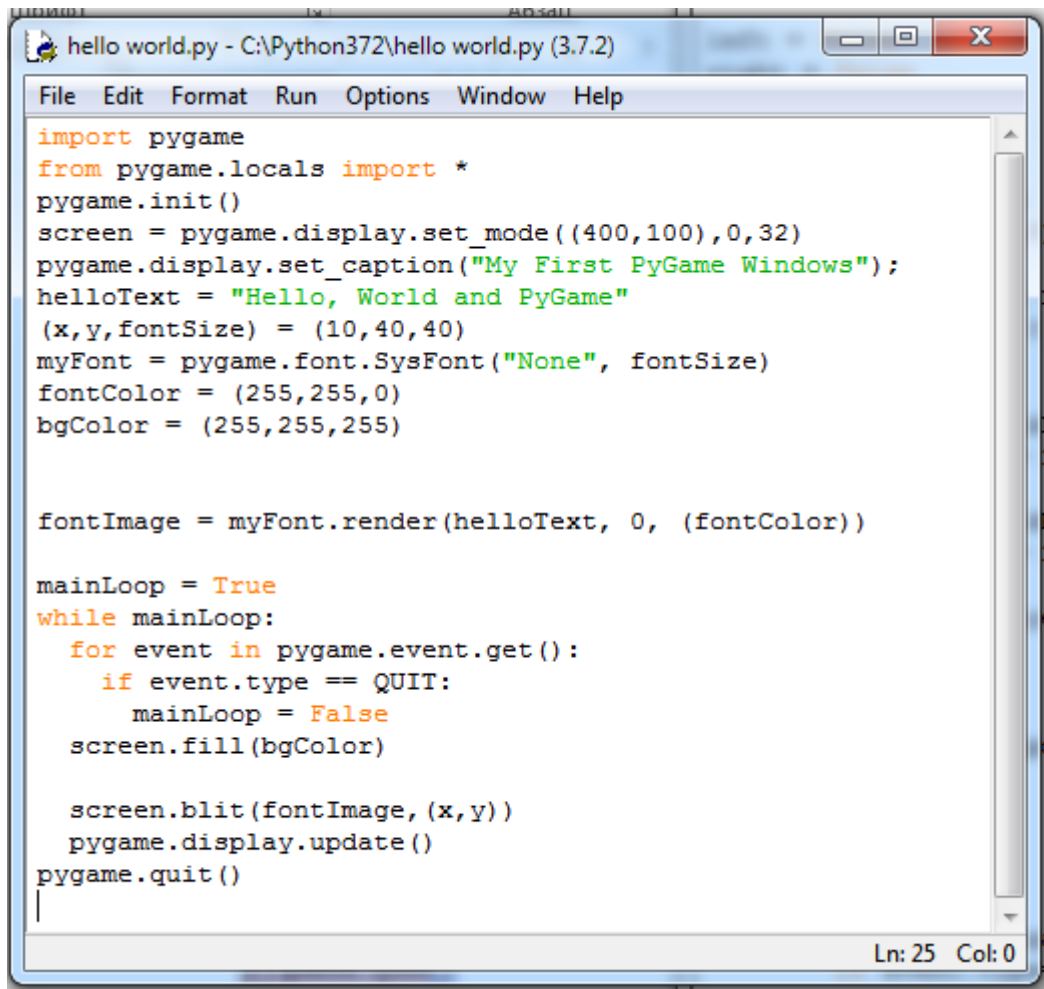
### PyGame.quit()

16. Итоговый код должен выглядеть так:

```
import pygame, sys
from pygame.locals import
* pygame.init()
screen = pygame.display.set_mode((400,100),0,32)
pygame.display.set_caption("My First PyGame Windows");
helloText = "Hello, World and PyGame"
(x,y,fontSize) = (10,40,40)
myFont = pygame.font.SysFont("None", fontSize)
fontColor = (255,255,0)
bgColor = (255,255,255)
fontImage = myFont.render(helloText, 0, (fontColor))
mainLoop = True
while mainLoop:
    for event in
    pygame.event.get(): if
    event.type == QUIT:
    mainLoop = False
    screen.fill(bgColor)
    screen.blit(fontImage,(x,y))
    pygame.display.update()
    pygame.quit()
```

17. Теперь осталось запустить этот код в интерпретаторе и посмотреть. Заходим в меню «Run» и выбираем пункт «Run Module (F5)». Появится окно с надписью (с заголовком) «My First PyGame Windows», в котором желтыми буквами написано «Hello, World and PyGame».





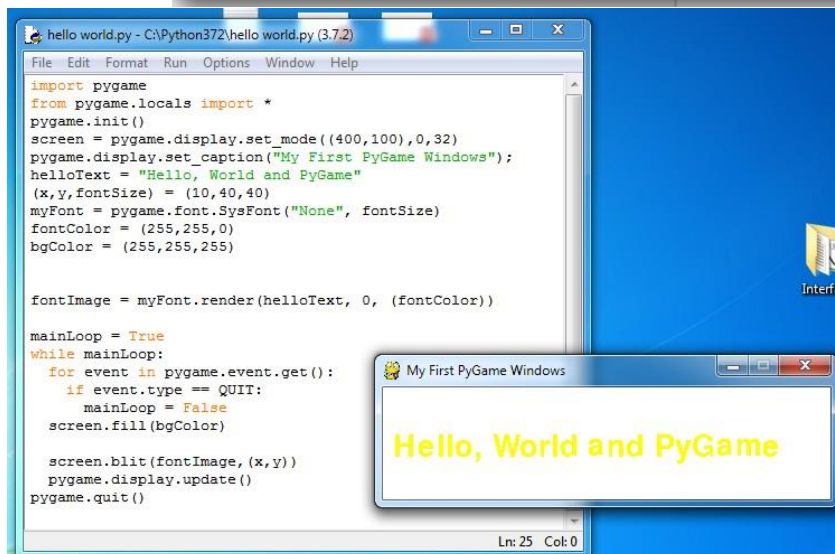
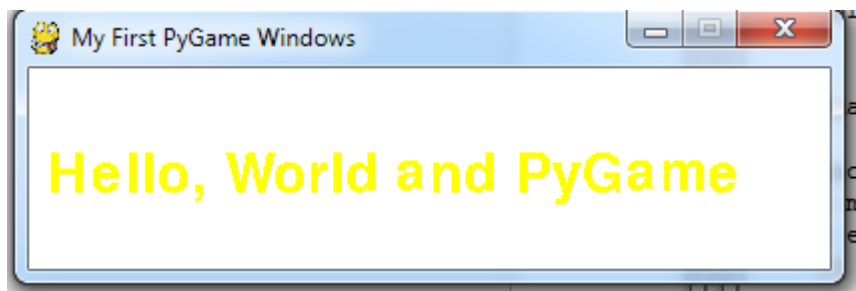
```
hello world.py - C:\Python372\hello world.py (3.7.2)
File Edit Format Run Options Window Help
import pygame
from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((400,100),0,32)
pygame.display.set_caption("My First PyGame Windows");
helloText = "Hello, World and PyGame"
(x,y,fontSize) = (10,40,40)
myFont = pygame.font.SysFont("None", fontSize)
fontColor = (255,255,0)
bgColor = (255,255,255)

fontImage = myFont.render(helloText, 0, (fontColor))

mainLoop = True
while mainLoop:
    for event in pygame.event.get():
        if event.type == QUIT:
            mainLoop = False
            screen.fill(bgColor)

        screen.blit(fontImage, (x,y))
        pygame.display.update()
pygame.quit()

Ln: 25 Col: 0
```



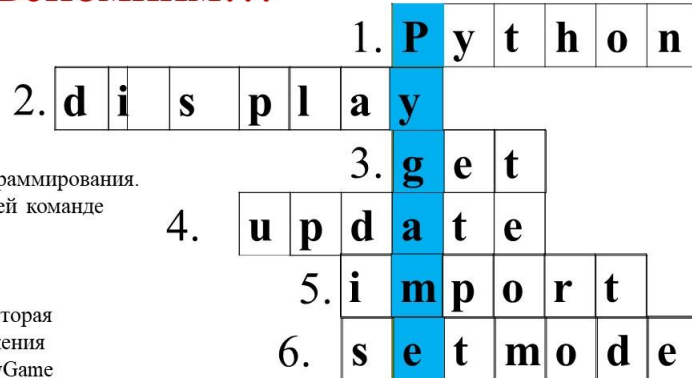
## Задний фон и персонажи игры

### Приложение 2

Кроссворд



Давайте вспомним...



1. Кроссплатформенный язык программирования.
2. Пропущенное слово в следующей команде `pygame._____.set_mode()`
3. Пропущенное слово в методе `pygame.event._____()`
4. Функция используемая с `flip`, которая обновляет только часть изображения
5. Импортирование библиотеки `PyGame` начинается с этого слова.
6. Функция используемая в модуле `display`

Давайте вспомним...

1. Python
2. display
3. get
4. update
5. import
6. setmode

1. Кроссплатформенный язык программирования.
2. Пропущенное слово в следующей команде: `pygame._____.set_mode`
3. Пропущенное слово в методе `pygame.event._____()`
4. Функция, используемая с `flip`, которая обновляет только часть изображения.
5. Импортирование библиотеки `PyGame` начинается с этого слова.
6. Функция, используемая в модуле `display`.

## Приложение

# Использование фонового изображения

Pygame очень хорошо работает с изображениями.

```
import pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
clock = pygame.time.Clock()
background_image = pygame.image.load('images/background.jpg')
while True:
    screen.blit(background_image, (0, 0))
    pygame.display.update()
    clock.tick(60)
```

\*pygame.image.load – загружает фоновое изображение с помощью функции

screen.blit(background\_image,(0,0)) - затем выполняется блиттинг (копирование битов) изображения на экран в позицию (0,0)

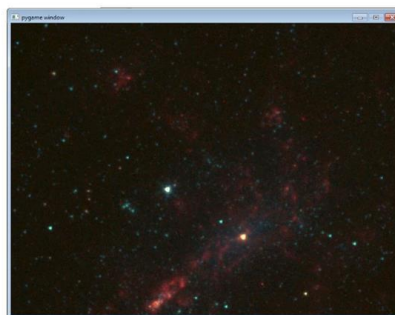
Использование фонового изображения  
PyGame очень хорошо работает с изображениями.

```
import
pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
clock = pygame.time.Clock()
background_image =
pygame.image.load('images/background.j
while True:
screen.blit(background_image, (0, 0))
pygame. display. update()
clock. tick(60)
```

\*pygame.image.load – загружает фоновое изображение с помощью функции  
screen.blit(background\_image, (0, 0)) – затем выполняется блиттинг (копирование битов) изображения на экран в позицию (0,0)

## Результат вывода кода:

```
import pygame
pygame.init()
screen = pygame.display.set_mode((800, 600))
clock = pygame.time.Clock()
background_image =
pygame.image.load('images/background.jpg')
while True:
    screen.blit(background_image, (0, 0))
    pygame.display.update()
    clock.tick(60)
```



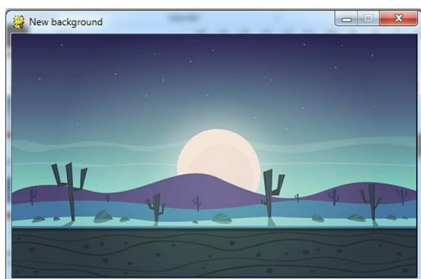
Результат вывода кода: `import`  
`pygame`  
`pygame.init()`  
`screen = pygame.display.set_mode((800, 600))`  
`clock =`  
`pygame.time.Clock()`  
`background_image = pygame.image.load('images/background.jpg')`  
`while True:`  
`screen.blit(background_image, (0,0))`  
`pygame.display.update()`  
`clock.tick(60)`

## Приложение

9.3.3.5 *создавать задний фон игры*

ФО1 - Практическое задание

### Индивидуальная практическая работа



1. Используя изученный материал урока создайте экран игры размера на 500\*300 пикселей и загрузите изображение под названием game\_background.jpg. Назовите ваше окно «New background». Задержка экрана 30 кадров в секунду.

#### Дескрипторы:

- ✓ Использовать нужные модули и функции библиотеки PyGame
- ✓ Создать простой экран используя готовый фон

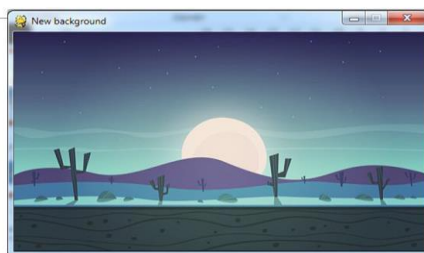
## ОТВЕТ

```
File Edit Format Run Options Window Help
import pygame
pygame.init()
win = pygame.display.set_mode((500, 300))
pygame.display.set_caption("New background")

bg = pygame.image.load('game_background.jpg')

clock = pygame.time.Clock()

run=True
while run:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            run=False
    win.blit(bg, (0,0))
    pygame.display.update()
pygame.quit()
```



Используя изученный материал урока, создайте экран игры размером 500\*300 пикселей и загрузите изображение под названием game\_background.jpg. Назовите ваше окно «New background». Задержка экрана 30 кадров в секунду.

#### Дескрипторы:

- ✓ использовать нужные модули и функции библиотеки PyGame; создать простой экран, используя готовый фон.

## Приложение 3

### Задний фон и персонажи игры (продолжение)

#### Задний фон и персонажи игры

#### Приложение 1

## Индивидуальная практическая работа



1. Используя готовый экран игры с фоном, созданную в предыдущем уроке добавьте готовый персонаж девочки.

2. Дополнительное задание:

Добавьте ее передвижение вправо.

### Дескрипторы:

- ✓ Умеет добавлять в игру готового персонажа
- ✓ создает его движение

### Практическое задание

#### Индивидуальная практическая работа

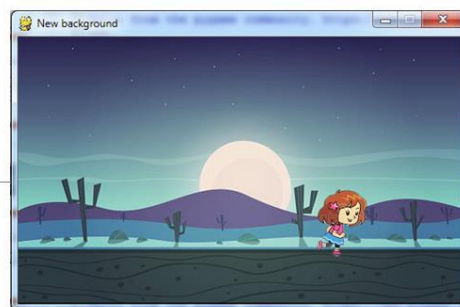
1. Используя готовый экран игры с фоном, созданную в предыдущем уроке, добавьте готовый персонаж девочки.

2. Дополнительное задание: добавьте ее передвижение вправо. **Дескрипторы:**

- ✓ умеет добавлять в игру готового персонажа;
- ✓ создает его движение.

## ОТВЕТ

```
File Edit Format Run Options Window Help
import pygame
pygame.init()
win = pygame.display.set_mode((500, 300))
pygame.display.set_caption("New Background")
clock = pygame.time.Clock()
walkRight = [pygame.image.load('girl_1.png'),
pygame.image.load('girl_2.png'), pygame.image.load('girl_3.png'),
pygame.image.load('girl_4.png'), pygame.image.load('girl_5.png'),
pygame.image.load('girl_6.png')]
bg = pygame.image.load('game_background.jpg')
playerStand = pygame.image.load('girl_center.png')
x=50
y=170
width=40
height=71
speed=5
right = False
animCount = 0
def drawWindow():
    global animCount
    win.blit(bg, (0,0))
    if animCount + 1 >= 30:
        animCount = 0
    if right:
        win.blit(walkRight[animCount // 5], (x,y))
        animCount += 1
    else:
        win.blit(playerStand, (x,y))
    pygame.display.update()
run=True
while run:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            run=False
    keys=pygame.key.get_pressed()
    if keys[pygame.K_RIGHT] and x < 500 - width - 5:
        x += speed
        right=True
    drawWindow()
pygame.quit()
```



## Приложение 4

### Приложение 1. Практическая работа Анимированный персонаж (прямоугольник) в PyGame

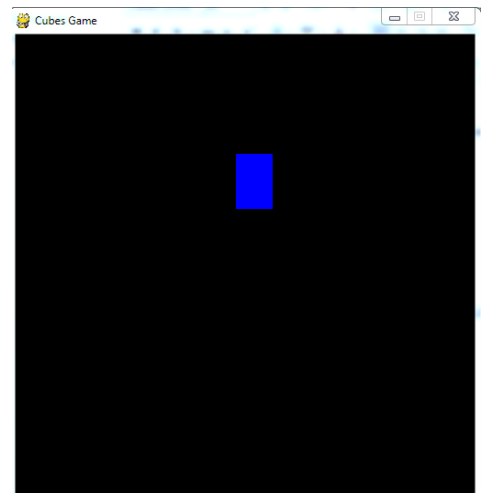
1. Запускаем программу для создания игры (Python).
2. Импортируем библиотеку PyGame в наш код:

```
import pygame, sys  
from pygame.locals import *
```

3. Инициализируем библиотеку:  
**PyGame.init()**

4. Далее нарисуем прямоугольник и запрограммируем его на передвижение (вправо, влево, вверх, вниз).

```
import  
pygame  
e  
pygame  
e.init()  
win =  
pygame.display.set_mode((500,  
500))  
pygame.display.set_caption("Cubes  
Game") x=50  
y=50  
width=  
40  
height  
=60  
speed=  
5  
run=True  
while  
run:  
    pygame.time.delay(100)  
    for event in  
        pygame.event.get(): if  
            event.type==pygame.Q  
UIT:  
            run=False  
  
    keys=pygame.key.get_pressed()  
  
    if  
        keys[pygame.K_LE  
FT]: x -= speed
```



```

    if
        keys[pygame.K_RIG
            HT]: x += speed
    if
        keys[pygame.K_
            UP]: y -= speed
    if
        keys[pygame.K_DO
            WN]: y += speed
    win.fill((0,0,0))

    pygame.draw.rect(win, (0,0,255), (x,y,width,height))
    pygame.display.update()

```

```
pygame.quit()
```

5. Теперь изменим данный выше код так, чтобы можно было указать границы экрана для того, чтобы объект не выходил за экран:

```

import pygame
pygame.init()
win = pygame.display.set_mode((500,
500))
pygame.display.set_caption("Cubes
Game") x=50
y=425
width=40
height=60
speed=5
run=True
while run:
    pygame.time.delay(100)
    for event in
        pygame.event.get(): if
            event.type==pygame.QUI
                T:
                    run=False

    keys=pygame.key.get_pressed()
    if keys[pygame.K_LEFT] and x>5:
        x -= speed
    if keys[pygame.K_RIGHT] and x < 500 - width
- 5: x += speed
    if keys[pygame.K_UP] and
y>5: y -= speed
    if keys[pygame.K_DOWN] and y < 500 - height - 5:
        y += speed
    win.fill((0,0,0))

    pygame.draw.rect(win, (0,0,255),
(x,y,width,height)) pygame.display.update()

pygame.quit()

```

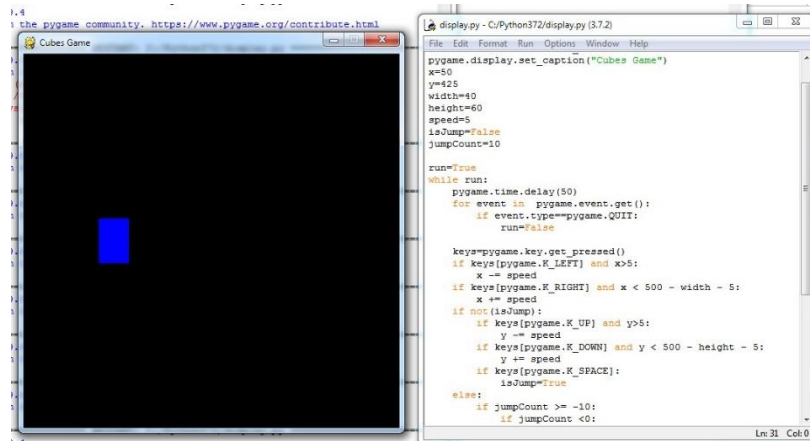


6. Дополним наш код, чтобы персонаж умел прыгать. import

```
pygame
pygame.init()
win = pygame.display.set_mode((500, 500))
pygame.display.set_caption("Cubes
Game") x=50
y=425
width=4
0
height=6
0
speed=5
is.Jump=
False
jumpCo
unt=10

run=
True
while
run:
    pygame.time.delay(50)
    for event in
        pygame.event.get():
            if
                event.type==pygame.
                    QUIT:
                        run=False

    keys=pygame.key.get_pressed()
    if keys[pygame.K_LEFT]
        and x>5: x -= speed
    if keys[pygame.K_RIGHT] and x < 500 -
        width - 5: x += speed
    if not(is.Jump):
        if keys[pygame.K_UP]
            and y>5: y -= speed
        if keys[pygame.K_DOWN] and y < 500 -
            height - 5: y += speed
        if keys[pygame.K_SPACE]:
            is.Jump=True
    else:
        if jumpCount >= -10:
            if jumpCount <0:
                v +=(jumpCount **2)/2
            else:
                y -=(jumpCount **2)/2
            jumpCount -=1
        else:
            is.Jump = False
            jumpCount = 10
pygame.quit()
```



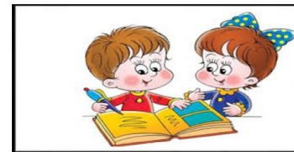
## Приложение 5 Приложение 1



ФО1 - Практическое задание

### Парная работа

1. Дополните предыдущий код веткой else, обрабатывающие случаи, когда пользователь вводит числа не входящие в заданные четыре диапазона.
2. Усовершенствуйте предыдущую программу, обработав исключение ValueError, возникающее, когда вводится не целое число.
3. Напишите программу, которая запрашивает на ввод число. Если оно положительное, то на экран выводится цифра 1. Если число отрицательное, выводится -1. Если введенное число – это 0, то на экран выводится 0. Используйте в коде условный оператор множественного ветвления.



#### Дескрипторы:

- ✓ Умеет использовать условные операторы
- ✓ Пишет корректный код программы

### Практическое задание

#### Парная работа

1. Дополните предыдущий код веткой else, которая обрабатывает те случаи, когда пользователь вводит числа, не входящие в заданные четыре диапазона.
2. Усовершенствуйте предыдущую программу, обработав исключение ValueError, возникающее, когда вводится не целое число.
3. Напишите программу, которая запрашивает на ввод число. Если оно положительное, то на экран выводится цифра 1. Если число отрицательное, выводится -1. Если введенное число 0, то на экран выводится 0. Используйте в коде условный оператор множественного ветвления.

#### Дескрипторы:

- ✓ умеет использовать условные операторы;
- ✓ пишет корректный код программы.

### Приложение 2. Раздаточный материал для рефлексии



Feedback  
 Чемодан, мясорубка, корзина  
 Чемодан – все, что пригодится в дальнейшем  
 Мясорубка – информацию переработаю  
 Корзина – все выброшу

## Приложение 6

### Приложение 1

9.3.3.10 реализовать алгоритм подсчета результатов игры

ФО1 - Практическое задание в группах

## Подсчет результатов игры пример «Игра в карты»

1. Используя раздаточный материал с условиями игры распишите алгоритм действий происходящих в игре при данных условиях.
2. Разработайте код программы, реализуя разработанный алгоритм.
3. Разработайте также вариант программы с неоднократным выбором карты участниками игры и подсчетом результатов.
4. Презентуйте свой вариант реализации алгоритма остальным группам

#### Дескрипторы:

- ✓ Разбирается в разновидностях алгоритмов
- ✓ пишет код игры используя операторы условия
- ✓ Пишет алгоритм подсчета результатов игры с операторами условия
- ✓ Грамотно обосновывает свой вариант решения

Активация Windows  
 Чтобы активировать Windows, перейдите в раздел "Параметры".

#### Практическое задание в группах

#### Подсчет результатов игры, например, «Игра в карты»

1. Используя раздаточный материал с условиями игры, распишите алгоритм действий, происходящих в игре при данных условиях.
2. Создайте код программы, реализуя разработанный алгоритм.
3. Разработайте также вариант программы с неоднократным выбором карты участниками игры и подсчетом результатов.
4. Презентуйте свой вариант реализации алгоритма остальным группам. **Дескрипторы:**

- ✓ разбирается в разновидностях алгоритмов;
- ✓ пишет код игры, используя операторы условия;
- ✓ пишет алгоритм подсчета результатов игры с операторами условия;
- ✓ грамотно обосновывает свой вариант решения.

## Приложение 2.

### Подсчет результатов игры «Игра в карты»

Карты делятся на 4 вида:

- ✓ «кубики»;
- ✓ «крестики»;
- ✓ «ромбики»;
- ✓ «кружочки».

Каждый вид имеет 9 достоинств:

- ✓ «шесть»;
- ✓ «семь»;
- ✓ «восемь»;
- ✓ «девять»;
- ✓ «десять»;
- ✓ «красный»;
- ✓ «синий»;
- ✓ «желтый»;
- ✓ «зеленый».

Два игрока выбирают карты наугад. Найти игрока, у которого карта «больше», при условиях их увеличения по порядку:

**«кубики» < «крестики» < «ромбики» < «кружочки»  
и «шесть» < «семь» < «восемь» < «девять» < «десять» < «красный» < «синий» <  
«желтый» <  
«зеленый»**

Присвоены условные номера: вид «кубики» – 1, вид «крестики» – 2, вид «ромбики» – 3, вид «кружочки» – 4; достоинствам: «шестерка» – 6, «семерка» – 7, ..., «десятка» – 10, «красный» – 11, «синий» – 12, «желтый» – 13, «зеленый» – 14.

Основные этапы моделирования игры следующие.

1. Ввод имен участников:

```
igrok1 = input('Введите имя 1-го  
игрока ') igrok2 = input('Введите имя  
2-го игрока ') 
```

2. Определение (случайным образом) номера видов и номера достоинства карты первого игрока:

```
nomer_vida_1 = randint(1, 4)  
nomer_dost_1 = randint(6, 14)
```

3. Определение соответствующего названия вида:

```
if nomer_vida_1 ==  
1: vid = 'кубик'  
elif nomer_vida_1  
== 2: vid =  
'крестик'  
elif nomer_vida_1  
== 3: vid =  
'ромбик'  
else: vid = 'круг'
```

и названия достоинства карты:

```
if nomer_dost_1 ==
```

```

6: dost =
  'Шестерка'
elif nomer_dost_1
  == 7: dost =
  'Семерка' ...
elif nomer_dost_1
  == 10: dost =
  'Десятка'
elif nomer_dost_1
  == 11: dost =
  'Красный'
elif nomer_dost_1
  == 12: dost =
  'Синий'
elif nomer_dost_1
  == 13: dost =
  'Желтый'
else: dost = 'Зеленый'

```

4. Вывод на экран полного названия карты первого игрока («красный кубик», «шестерка ромб» и т. п.):

```
print(igrok1, '- выпала карта:', dost, vid )
```

5. Определение номера вида и номера достоинства карты второго игрока.
6. Определение соответствующего названия вида и названия достоинства карты.
7. Вывод на экран полного названия карты второго игрока.
8. Определение результата игры:

```

#Сравниваем виды карт (их номера)
if nomer_vida_1 >
  nomer_vida_2:
  print('Выиграл', igrok1)
elif nomer_vida_2 > nomer_vida_1:
  print('Выиграл', igrok2)
else:
  #Виды карт игроков одинаковые
  #Сравниваем достоинства карт (их
  номера) if nomer_dost_1 >
  nomer_dost_2: print('Выиграл',
  igrok1)
  elif nomer_dost_2 > nomer_dost_1:
  print('Выиграл', igrok2)
  else:
    #Достоинства карт тоже одинаковые
    print('Ничья!')

```

Всю программу соберите самостоятельно.

Разработайте вариант программы с неоднократным выбором карты участниками игры и подсчетом результатов.

## Приложение 6

### Мини-игра «Ben10» на PyGame

1. Запускаем программу для создания игры (Python). Создайте документ с названием игры «Cubes Game».

2. Импортируйте библиотеку Pygame и инициализируйте его:

```
import pygame  
pygame.init()
```

3. Создайте экран и назовите его «Cubes

Game». *win =*

```
pygame.display.set_mode((500, 500))  
pygame.display.set_caption("Ben10")
```

4. Далее скопируйте картинки готового персонажа и фона в папку, где находится ваш проект (документ Python). Создайте переменные, чтобы ссылаться на данные изображения.



```
walkRight = [pygame.image.load('right_1.png'),  
pygame.image.load('right_2.png'), pygame.image.load('right_3.png'),  
pygame.image.load('right_4.png'), pygame.image.load('right_5.png'),  
pygame.image.load('right_6.png')]
```

```
walkLeft = [pygame.image.load('left_1.png'),  
pygame.image.load('left_2.png'), pygame.image.load('left_3.png'),  
pygame.image.load('left_4.png'), pygame.image.load('left_5.png'),  
pygame.image.load('left_6.png')]
```

```
bg = pygame.image.load('picture.jpg')  
playerStand =  
pygame.image.load('idle.png')
```

5. Создайте свой персонаж, его фон (готовый рисунок) и нужные переменные для его передвижения, а также добавьте таймер для задержки экрана:

```
clock = pygame.time.Clock()  
x=50  
y=370  
width=40  
height=71  
speed=5
```

```
isJump=False
jumpCount=10
```

```
left = False
right = False
animCount = 0
```

```
def drawWindow():
    global animCount
    win.blit(bg, (0,0))

    if animCount + 1 >= 30:
        animCount = 0

    if left:
        win.blit(walkLeft[animCount // 5], (x,y))
        animCount += 1
    elif right:
        win.blit(walkRight[animCount // 5], (x,y))
        animCount += 1
    else:
        win.blit(playerStand, (x,y))

pygame.display.update()
```

6. Добавляем переменные движения:

```
run=True
while run:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            run=False

    keys=pygame.key.get_pressed()
    if keys[pygame.K_LEFT] and
        x>5: x -= speed
        left=True
        right=False
    elif keys[pygame.K_RIGHT] and x < 500 - width - 5:
        x += speed
        left=False
        right=True
    else:
        left=False
        right=False animCount =
        0
    if not(isJump):
        if keys[pygame.K_SPACE]:
            isJump=True
    else:
        if jumpCount >= -10: if
            jumpCount <0:
```

```
    y +=(jumpCount **2)/2 else:
    y -=(jumpCount **2)/2 jumpCount -
=1
else:
    isJump = False
    jumpCount = 10
```

*drawWindow() pygame.quit()*

6. Результат должен быть следующим:



```
game2.py - C:/Python372/Game2/game2.py (3.7.2)
File Edit Format Run Options Window Help
import pygame
pygame.init()
win = pygame.display.set_mode((500, 500))
pygame.display.set_caption("Ben10")

walkRight = [pygame.image.load('right_1.png'),
pygame.image.load('right_2.png'), pygame.image.load('right_3.png'),
pygame.image.load('right_4.png'), pygame.image.load('right_5.png'),
pygame.image.load('right_6.png')]

walkLeft = [pygame.image.load('left_1.png'),
pygame.image.load('left_2.png'), pygame.image.load('left_3.png'),
pygame.image.load('left_4.png'), pygame.image.load('left_5.png'),
pygame.image.load('left_6.png')]

bg = pygame.image.load('picture.jpg')
playerStand = pygame.image.load('idle.png')

clock = pygame.time.Clock()

x=50
y=370
Ln: 22 Col: 0
```