

**Заявка на конкурс
Фонда образовательных инноваций 2022 года**

Номинация:
Вольная номинация

Программировать играя:
практика по разработке игры на Python
(начальный уровень)

Автор:
Сенина Анна Васильевна,
преподаватель кафедры гуманитарных
дисциплин НИУ ВШЭ - Пермь
avsenina@hse.ru

1. Общая идея и цель практики

Дисциплины блока Data Culture, реализуемые на всех образовательных программах НИУ ВШЭ, а также независимые экзамены по проверке освоения цифровых компетенций позволяют студентам овладеть навыками человека XXI века, готовят наиболее компетентных и конкурентоспособных сотрудников для рынка труда. Цифровизация образования стала практически общим местом для вузов, вошедших в программу «Приоритет-2030». В Вышке активно развивается направление Digital Humanities, разработаны и внедрены дисциплины, направленные на овладение культурой работы с данными, значительная часть которых реализуется в формате blended. Однако в учебных планах по-прежнему сложно установить баланс между классическим гуманитарным знанием и цифровыми дисциплинами. В частности, постоянный сбор обратной связи от студентов в рамках онлайн-курсов по дисциплинам цифровая грамотность, анализ данных и программирование на Python, а также студенческая оценка преподавателей показывают, что студенты положительно оценивают новизну знаний, однако по-прежнему стабильно низко определяют полезность курсов для будущей карьеры.

На ОП «История» ([Санкт-Петербург](#), [Пермь](#)) при изучении курса Python часть студентов теряет интерес на первых занятиях, поскольку написание программы выглядит как набор простых текстовых инструкций. Содержание курсов ориентировано на решение гуманитарных задач, однако перед этим необходимо овладеть базовым синтаксисом (аналогично тому, как мы изучаем буквы и слова, прежде чем писать предложения и тексты). Кроме того, основные преимущества Python - гибкость, тонкая настройка программы под задачи исследования, работа с текстами, статистикой - лучше всего могут быть продемонстрированы только на завершающих неделях курса. Зачастую студенты в своих проектах и исследованиях вместо программирования выбирают готовые программные решения, которые находятся в открытом доступе или также предоставляются Вышкой. Ожидая найти в Python «волшебную кнопку», они не заходят дальше решения стандартных (тренировочных) задач, а значит, не используют программирование как инструмент в дальнейшей деятельности. Отчасти эта ситуация нивелируется проектным подходом, который требуется для прохождения точек контроля, однако до уровня самостоятельного нахождения проектного решения доходят не все.

При этом уже на первых встречах выясняется, что у студентов есть запрос на проектирование игр и разработку приложений, что выходит за пределы программы дисциплины. Представляется, что внедрение геймификации в образовательный процесс, обучение программированию путем погружения в сторителлинг и геймдизайн позволят повысить мотивацию студентов.

Следовательно, цель практики по геймдизайну - повышение уровня владения программированием на Python у студентов гуманитарных направлений подготовки.

Используя модель геймификации Ю-Кай Чоу – модель Октализ, можно выделить следующие цели геймификации:

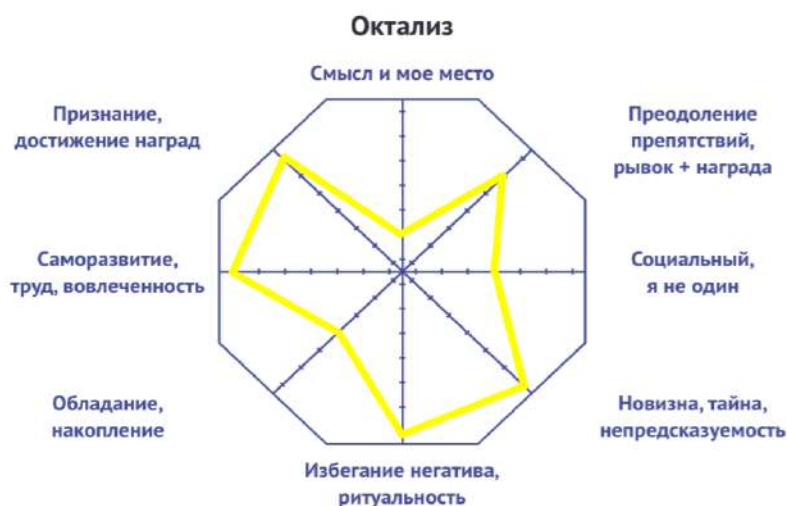


Рис. 1.
Модель Октализ

Возможность создать продукт – разработать игру, с одной стороны, направлена на повышение вовлеченности, освоение новых знаний, которые выходят за рамки нарезивания задач по программированию. С другой, одна из важнейших задач такой практики – избегание негатива, веселый и творческий способ учиться программированию, признание достижений – высокий балл (особенно для оценок 9 и 10) среди учебной группы и своего потока.

2. Формат практики

Практика по геймдизайну реализуется на семинарах или размещается в SmartLMS, может быть использована также качестве домашнего задания. Результат работы может быть включен студентом в точку контроля (проектное предложение).

Практика состоит из 3 частей:

- 1) знакомство с примерами игр на Python - текстовые квесты, интерактивные рассказы, элементы визуальных новелл;
- 2) разработка сценария по выбранной теме (в качестве темы может использоваться исторический сюжет);
- 3) реализация сценария игры на Python.

Дополнительные материалы, инструкции и примеры размещены в [Colaboratory](#) (приложение 1).

3. Нарратив - сценарий - сторителлинг

Студенты-историки работают с историческими источниками (которые могут иметь нарративный характер, подобно летописям, мемуарам, дневникам, письмам и др.) и историографией, зачастую представляющей из себя конструируемую автором интерпретацию прошлого. Поскольку нарратив как тип повествования обязательно имеет сюжет, на его основе можно реализовывать различные публично-исторические проекты, среди которых игры по определенной исторической тематике. Задача студента - выбрать подходящий сюжет на интересующую студента тему и разработать по нему сценарий: правила игры, возможные действия игрока, систему квестов, альтернативные концовки и т.д. После выбора сюжета, определения логики повествования студент переходит к сторителлингу - ищет способы рассказывания историй. В игре важно привлечь и удержать внимание игрока, вызвать эмоции, чувство сопричастности, показать развитие персонажа.

4. Типы задач профессиональной области

Поскольку наши выпускники работают в учреждениях образования и культуры, разработка игр станет вкладом в их профессиональную деятельность, поможет решать профессиональные задачи - элементы геймификации остаются востребованными в образовании, а интерактивность стала нормой для музеев, галерей и библиотек. Профессиональные траектории выпускников ОП «История» также убеждают в том, что и после выпуска они продолжают работать с нарративами, пишут тексты или создают визуальные продукты, для которых требуются навыки сторителлинга: музейный работник, педагог в школе, журналист, куратор выставки или менеджер социокультурных проектов – это те профессии, для которых важен поиск новых форматов трансляции знания, привлечение слушателей, поддержание внимания и интереса.

Разработка игры позволяет сочетать научно-исследовательскую составляющую обучения (поиск и анализ материалов исторических источников, работа с научной литературой) с культурно-просветительской, выраженной в виде игры как проекта из области public history.

Разработка простых игр востребована и на рынке обучения программированию: ее используют в курсе [Python-разработчика](#) от Яндекс.Практикума и в курсах от [Skillbox](#).

5. Планируемые результаты обучения, критерии оценивания

В результате разработки собственной игры студент:

- работает с историческими источниками, историографией (ПК 7, ПК 9);
- приобретает навыки программирования, сторителлинга, используя проблематику гуманитарных наук (ПК 12);
- создает новый цифровой продукт, направленный на широкую аудиторию, в популярной форме представляющую историческую эпоху или ситуацию (ПК 16, ПК 24).

В таблице ниже представлен один из возможных вариантов использования разработки игры в качестве проектного предложения. Поскольку эта точка контроля предполагает вариативность – особенно в плане использования различных цифровых технологий (таймлайны, карты, сети, викторины, боты, компьютерные игры и другое), в таблице ниже уточнены критерии оценивания. Для решения задач предметной области, в особенности, исследовательских задач студента, можно выбрать любой цифровой инструмент, включая разработку игры. Важно, что студенты осознанно выбирают такой формат работы, могут работать в безопасной среде – среди единомышленников, а также получают поощрение в баллах. Даже при возникновении технических сложностей, которые студенты не разрешили в процессе работы над проектом, их наработки также оцениваются как проектное предложение, не дошедшее до стадии реализации – в случае наличия интересной идеи и продуманного алгоритма по воплощению ее в жизнь, можно получить оценку 8 (подробнее об этом в таблице).

Критерии оценивания из ПУД	Критерии оценивания разработки игры в качестве проектного предложения	Ожидаемый эффект геймификации	ПРО
<p>Проектное предложение: цифровая публичная история</p> <p>Студент готовит проектное предложение по следующему техническому заданию:</p> <ol style="list-style-type: none"> 1. Цель проекта, исследовательские задачи. 2. Предполагаемые источники (их состояние: традиционные или машиночитаемые). 3. Возможные методы и технологии реализации проекта. Программные средства. 4. Ожидаемые результаты. <p>Проектное предложение может быть связано с темой исследовательской работы студента; студент видит возможность применения цифровых технологий к</p>	<p>Корректировка критериев оценивания применительно к разработке компьютерных игр:</p> <ol style="list-style-type: none"> 1. Цель, задачи. 2. Предполагаемые источники (их состояние: традиционные или машиночитаемые), включая историографические. 3. Подготовленный сценарий компьютерной игры, описание игровых персонажи, игровая механика, возможные действия игрока. 4. Техническая реализация: разработанный алгоритм реализации 	<p>Сбор материалов, работа с источниками и литературой выходит на качественно новый уровень: студенты привлекают разнообразные источники для того, чтобы обеспечить иммерсивность, погружение в игровую ситуацию. Для создания убедительного повествования приходится привлекать дополнительные источники, обращаться к истории повседневности. Эффект вызван большей мотивированностью и заинтересованностью студента.</p> <p>В плане обучения программированию студенты отрабатывают темы: условный оператор if, циклы, функции, требующиеся для внесения в игру альтернативных выборов и концовок, а также знакомятся с дополнительными библиотеками Python, не входящими в онлайн-курс (Pillow, Ren'Py и др). представляющая</p>	<p>ПК 7, ПК 9</p> <p>ПК 12</p>

<p>своей исследовательской задаче; умеет работать с имеющимися источниками, в том числе цифровыми; предлагаемое цифровое решение найдено самостоятельно; цифровой проект разработан с учетом интересов широкой аудитории; проектное предложение оформлено в виде текста с презентацией (8 баллов).</p> <p>Обязательный критерий для получения отличной оценки – алгоритм реализации проекта на Python. Работы, выполненные без использования Python, могут быть зачтены только с понижением баллов.</p>	<p>игры, обоснование использования необходимых библиотек Python, отражена логика написания программы.</p> <p>5. Перспективы использования в качестве публично-исторического проекта</p> <p>Для получения 9 и 10 баллов студентам необходимо продемонстрировать пробную версию игры. В этом случае также оценивается ее техническая реализация: код, визуализация при ее наличии.</p>	<p>собой движок для создания визуальных новелл, другими словами, объединяющая текстовый квест и графику.</p> <p>При разработке сюжета студенты работают для решения просветительских задач: так, в одном из проектов на тему революции 1917 года ваш герой беседует с Керенским о текущей политической ситуации, обсуждает министров-социалистов и другие проблемные вопросы, тем самым, получая историческую справку о периоде. От выбора игрока зависят его политические взгляды, которые проведут его через события 1917 года. Тем самым, студенты работают в поле public history – рассказывают историю широкой публике, адаптируют сюжет под ее запросы, учитывают интересы различных групп потенциальных пользователей.</p>	<p>ПК 16, ПК 24</p>
--	--	---	-------------------------

Практикум по геймификации, таким образом, сочетается с проектным подходом: во многом идея проектирования игры на основе исторической тематики возникла в ходе разработки проектов студентов-историков 2 курса в Санкт-Петербургском кампусе. В ходе проектного задания некоторые из них вышли на проектирование ботов в telegram (примеры: [Lemon Day](#) (чат-бот по творчеству Э. Лимонова), [бот-священник](#), общая доска [miro](#), на которой проходила защита проектов в формате постерной сессии) или небольших программ с вычислениями (к заявке приложен скриншот кода на Python, определяющий дату Пасхи по различным календарям, Приложение 2).

Часть студентов при работе над проектом встречалась со сложностями: приходится программировать не для автопроверки (как в онлайн-курсе), нужно самостоятельно определить, какой результат должен получиться в результате выполнения программы. Несколько проектов по этой причине не дошли до стадии реализации. Для текстового квеста достаточно начального уровня владения Python и обычного блокнота в Colab, в некоторой степени, можно разнообразить его картинками (хотя и для этого потребуется специальная библиотека Pillow, которая не входит в курс Python начального уровня). Однако несколько проектов дошли и до уровня полноценных визуальных новелл – для этого студентам пришлось освоить библиотеку Ren'Py, представляющую собой движок для объединения текстового квеста и графики.

В 4 модуле студентов выбирали разработку игр для сдачи точки контроля (выбор разработки игры как итогового проектного задания определялся исключительно интересом студентов к такому формату), 2 лучших проекта, получивших 10 баллов, приложены к заявке (Приложение 3). Студенты разработали пилотный выпуск – то, что в визуальных новеллах или интерактивных сериалах считается 1 серией. Несмотря на завершение курса, ребята планируют продолжить свои истории, а также представить их на конференциях.

Курс «Культуры работы с данными (Data Culture)» на ОП «История» (НИУ ВШЭ - Санкт-Петербург) реализован в 3 и 4 модуле, поэтому итоги СОП на текущий момент не

подведены. Тем не менее, сократилось количество работ с плагиатом в рамках этой точки контроля, и общий уровень проектов выше, чем в прошлом году.

6. Перспективы

Игровой формат позволяет обучать и обучаться: он обеспечивает глубокое погружение в курс, развивает творческое мышление, помогает собрать единомышленников. В результате выполнения задания студент создает продукт, который останется с ним после завершения курса. Поскольку большое внимание уделяется обратной связи, обсуждению с однокурсниками и преподавателем, студент собирает единомышленников, а в обсуждении рождаются новые идеи. Так, студент, разработавший игру на английском, впоследствии узнал, как создать игру на русском языке с помощью Ren'Py, и то, что не удалось сделать самостоятельно, получилось с помощью обмена опытом.

Обладая начальным уровнем программирования, чего зачастую недостаточно, чтобы считаться хотя бы junior-разработчиком в сфере IT, студенты-историки могут использовать программирование для просветительских проектов, создавать игры-квесты для партнеров Вышки в рамках проектной работы. Тем самым обеспечивается «мост» между цифровыми навыками и классическим историческим (гуманитарным) знанием, навыки программирования из категории «нового, но бесполезного знания» переходят в разряд инструментов для решения исторических задач.

Работа такого типа легко трансформируется в синхронный / асинхронный форматы, возможно расширение тематики задания на другие дисциплины, а также проведение различных обучающих мастер-классов по геймдизайну с примерами проектов студентов - как внутри Вышки, так и для внешних слушателей или абитуриентов.

Перечень приложений:

1. Блокнот в [Colaboratory](#) с материалами, инструкциями и примерами.
2. Пример решения исторической задачи на Python.
3. Игры, разработанные студентами:
 - Самая революционная новелла по мотивам 1917 года
 - Открытая общага

▼ Программировать играя:

практика по геймдизайну на Python (начальный уровень)

Приложение 1



Мы всегда говорим, что Python - это не "волшебная кнопка", которая сделает все за вас. Но стоит придумать алгоритм...

Автор изображения [dundik678](#)

▼ Разнообразие игр и жанров

Текстовый квест - один из жанров интерактивной литературы. Для его создания нужно написать тексты и соединить их переходами.

Можно добавить вариативность - выборы, зависящие от игрока, и несколько альтернативных концовок.

Для реализации на Python достаточно блокнота, условного оператора if (дополнительно можно упростить программу, используя циклы while или for, а также функции

Текстовый квест с простейшей графикой. Придется воспользоваться специальной библиотекой Pillow для работы с изображениями. Их можно найти в интернете или загрузить на свой GitHub

```
pip install Pillow
```

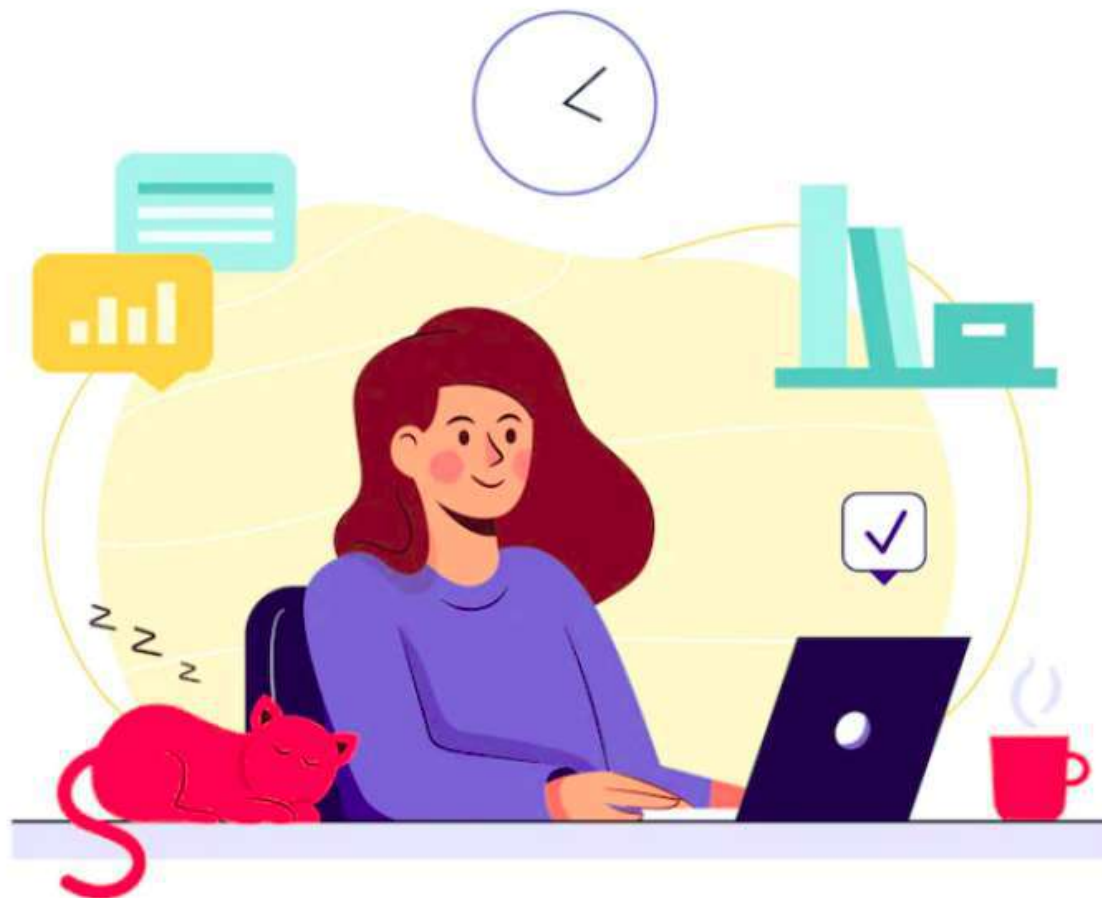
Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (7.1.2)

```
from PIL import Image
from io import BytesIO
import requests
```

```
url = 'https://raw.githubusercontent.com/AnnSenina/Other/main/%D0%A1%D0%BD%D0%B8%D0%BC%D0%BE%D0%BA%20%D1%8D%D0%BA%D1%80%D0%B0%D0%BD%D0%B0%20%D0%B
```

```
response = requests.get(url)
img = Image.open(BytesIO(response.content))
```


img



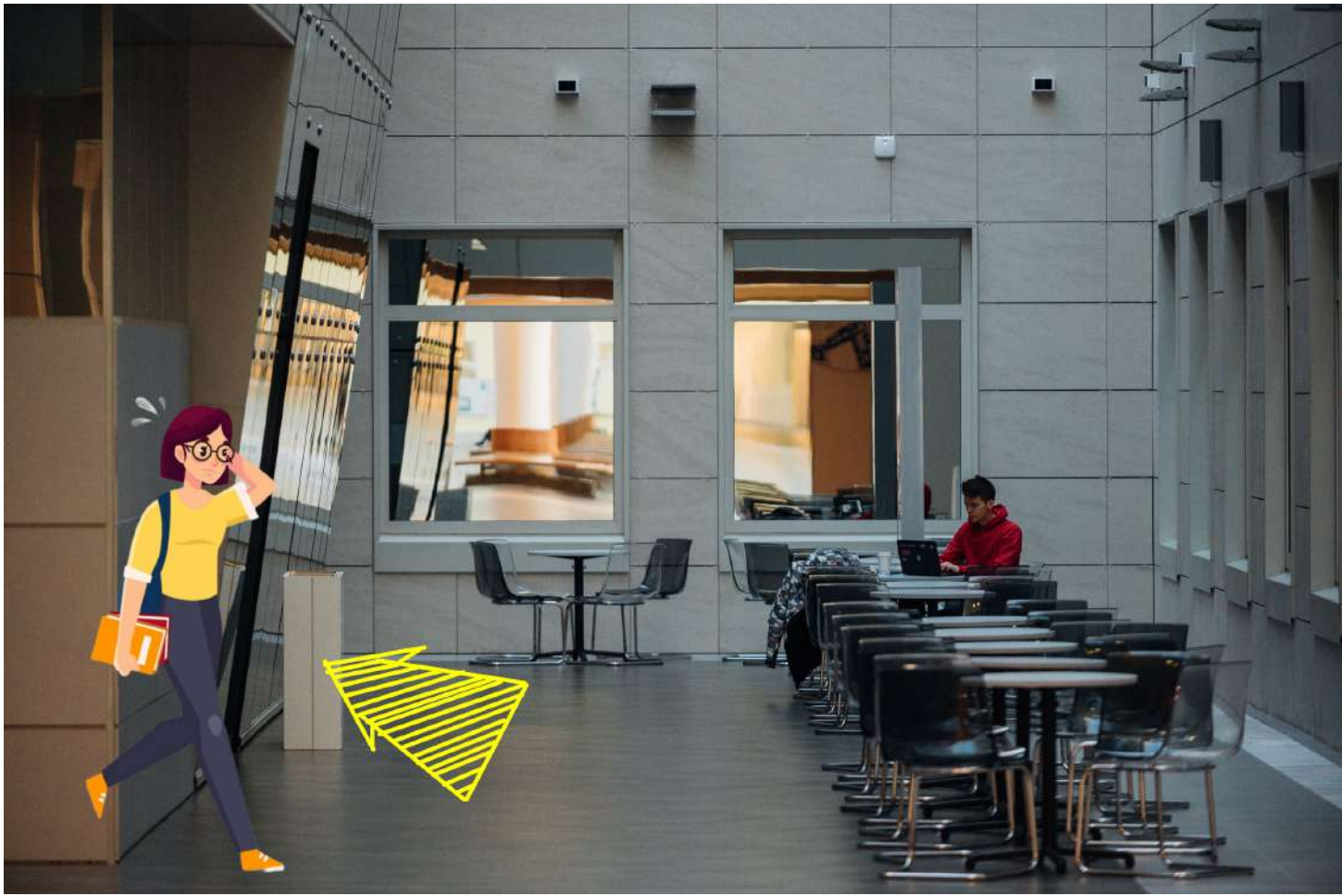
Здесь и далее используются бесплатные изображения с <https://ru.freepik.com/>

Визуальные новеллы - визуализированный текстовый квест. Для полноценной визуализации требуются фоны, персонажи и текстовая история. Библиотека Pillow позволяет накладывать текст на изображение, однако более удобный вариант - библиотека RenPy для создания полноценных визуальных новелл.



```
url3 = 'https://raw.githubusercontent.com/AnnSenina/Other/main/%D0%9F%D0%BE%D0%BA%D1%80%D0%BE%D0%B2%D0%BA%D0%B0-18.jpg'
```

```
response = requests.get(url3)
img3 = Image.open(BytesIO(response.content))
img3
```



В примерах выше бесплатные изображения персонажей наложены на виртуальные фоны для Zoom - фотографии Вышки

▼ Как разработать свою игру?

Шаг 1. Тема

За игру на историческую тематику вы получите дополнительные баллы!

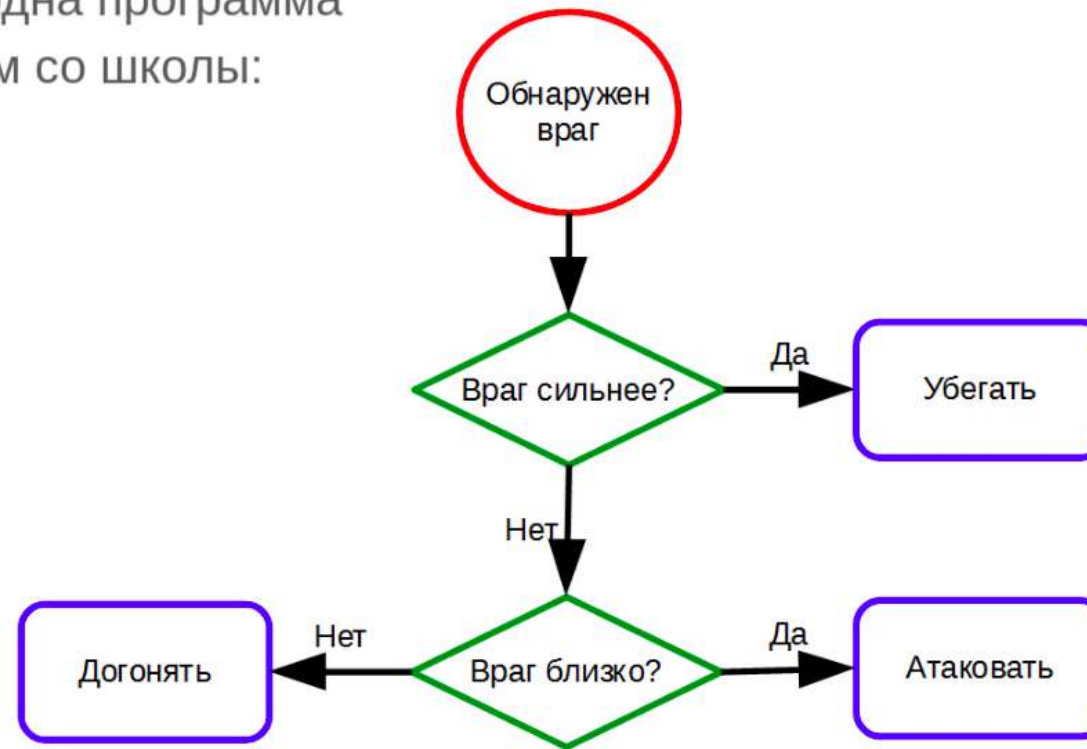
Шаг 2. Сюжет

Желательно сразу продумать начало и конец повествования. Используйте нелинейное повествование и альтернативные концовки.

Пример из [презентации](#) Даниила Скоринкина ниже:

Условные инструкции

- Без них не пишется ни одна программа
- Интуитивно понятны нам со школы:
 - Если
 - То (Да)
 - Иначе (Нет)



Шаг 3. Персонаж

Вы можете описать его или выбрать подходящее изображение. У персонажа может быть легенда, которую вы рассказываете в ходе повествования. Прежде чем переходить к следующему пункту, ответьте на вопрос: что персонаж будет делать? Общаться и

дружить с другими персонажами, сражаться с боссом? Какими характеристиками он должен обладать? Можно ли их измерить в какой-либо шкале?

Хороший пример: тест "Квалификационно-оценочный задачник администрации" из игры "Fallout 3". Ответы связаны с получением в дальнейшем различных навыков, влияющих на игру. Мы можем ввести переменные для различных характеристик и в дальнейшем (в зависимости от выборов игрока) повышать или понижать его характеристики. Улучшение характеристик может требоваться для получения определенной концовки.

Еще один хороший пример - простой и увлекательный тест от Арзамас по [истории Великой французской революции](#). Пример интерактивной истории, который уже стал классикой российской public history. Вы тоже так можете!

Где взять бесплатные изображения для персонажа? Можно на [freepik](#), также рекомендую эту [статью](#) с GeekBrains. Поэкспериментируйте с портретами исторических личностей, если это уместно для вашего сюжета.

Шаг 4. Игровая механика

Другими словами, что персонаж делает в вашей игре. Передвигается? Взаимодействует с предметами? Сражается? Сложно для визуальной новеллы, но возможно даже для текстового квеста.

Вам может потребоваться библиотека random для выбора случайного события (она есть в материалах онлайн-курса), ниже написан

```
import random
def dice():
    return random.randint(1, 6)

print(dice())
```

4

Рекомендуемый пример для понимания игровых механик: игра [Ozaria](#) - обучающая игра по программированию на Python. Ниже несколько скриншотов из нее:

Chapter 1: Sky Mountain



Одинокая башня



Я могу подойти к двери и открыть ее используя `hero.use("door")`.



справочник

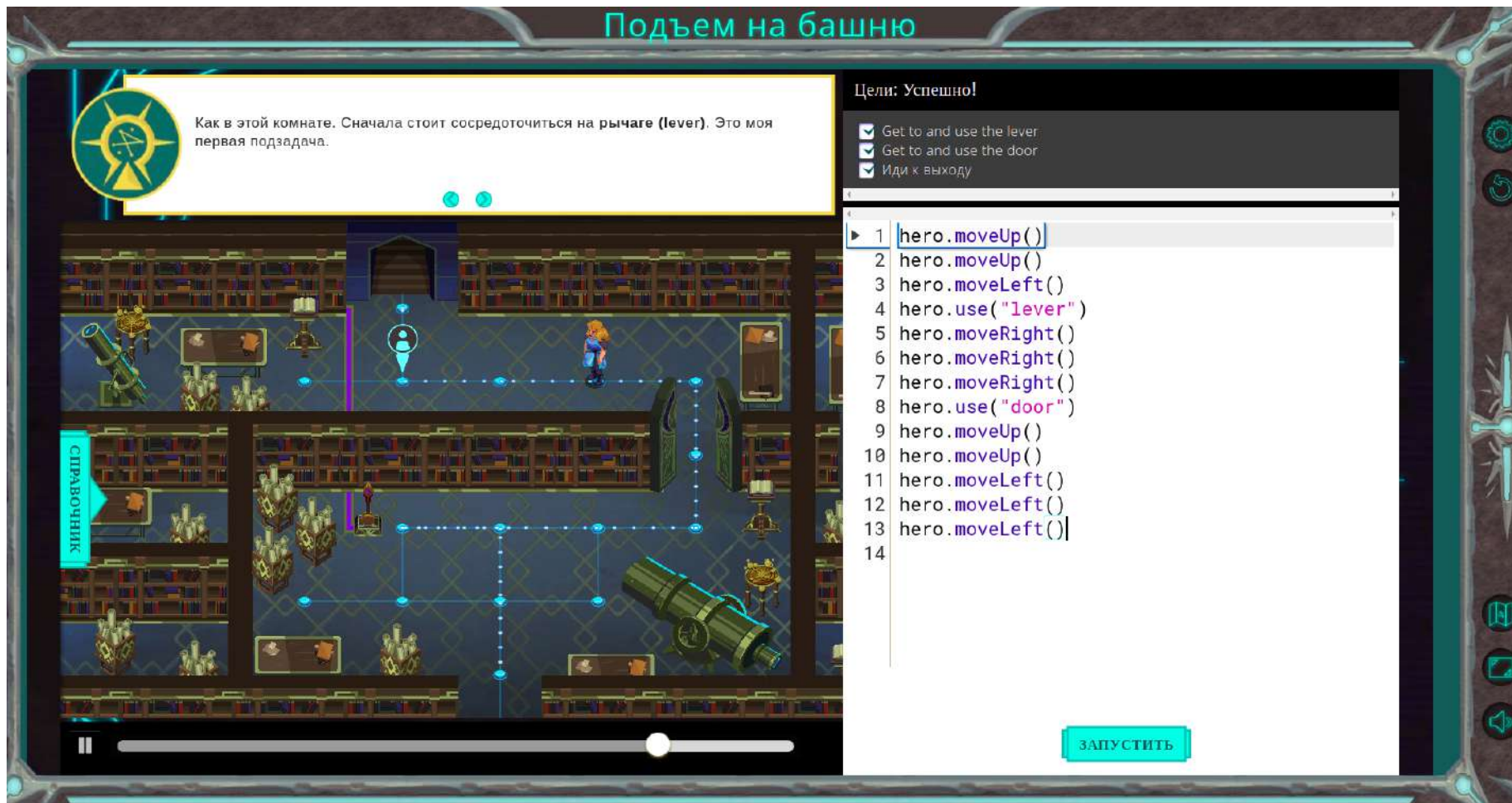


Цели: Успешно!

- ☒ Подойди к двери.
- ☒ Используй дверь (door).

```
✓1 hero.moveLeft()  
✓2 hero.moveLeft()  
✓3 hero.moveUp()  
✓4 hero.moveUp()  
✓5 hero.moveRight()  
▶6 hero.moveRight()  
7 hero.use("door")  
8
```

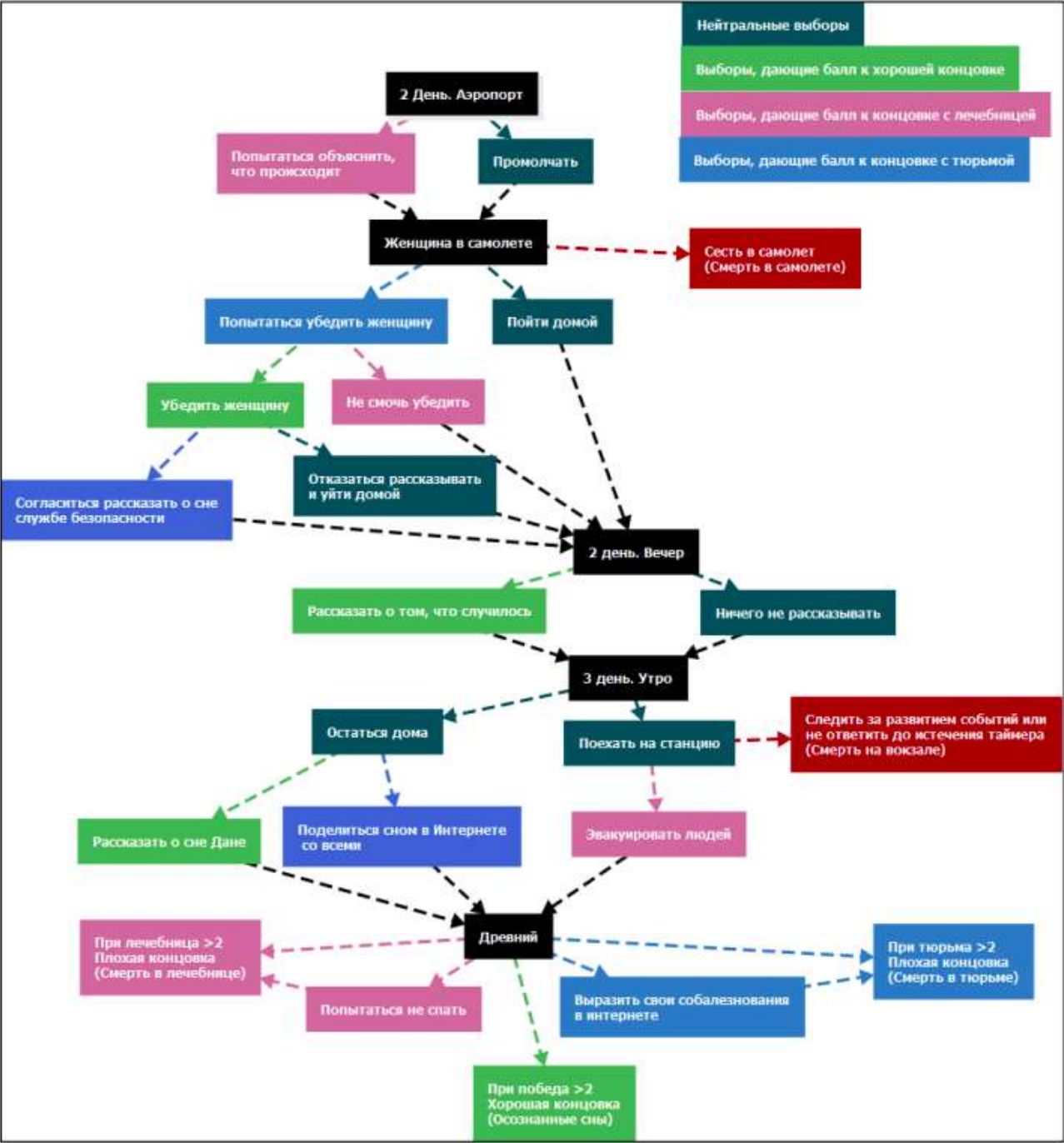
ЗАПУСТИТЬ



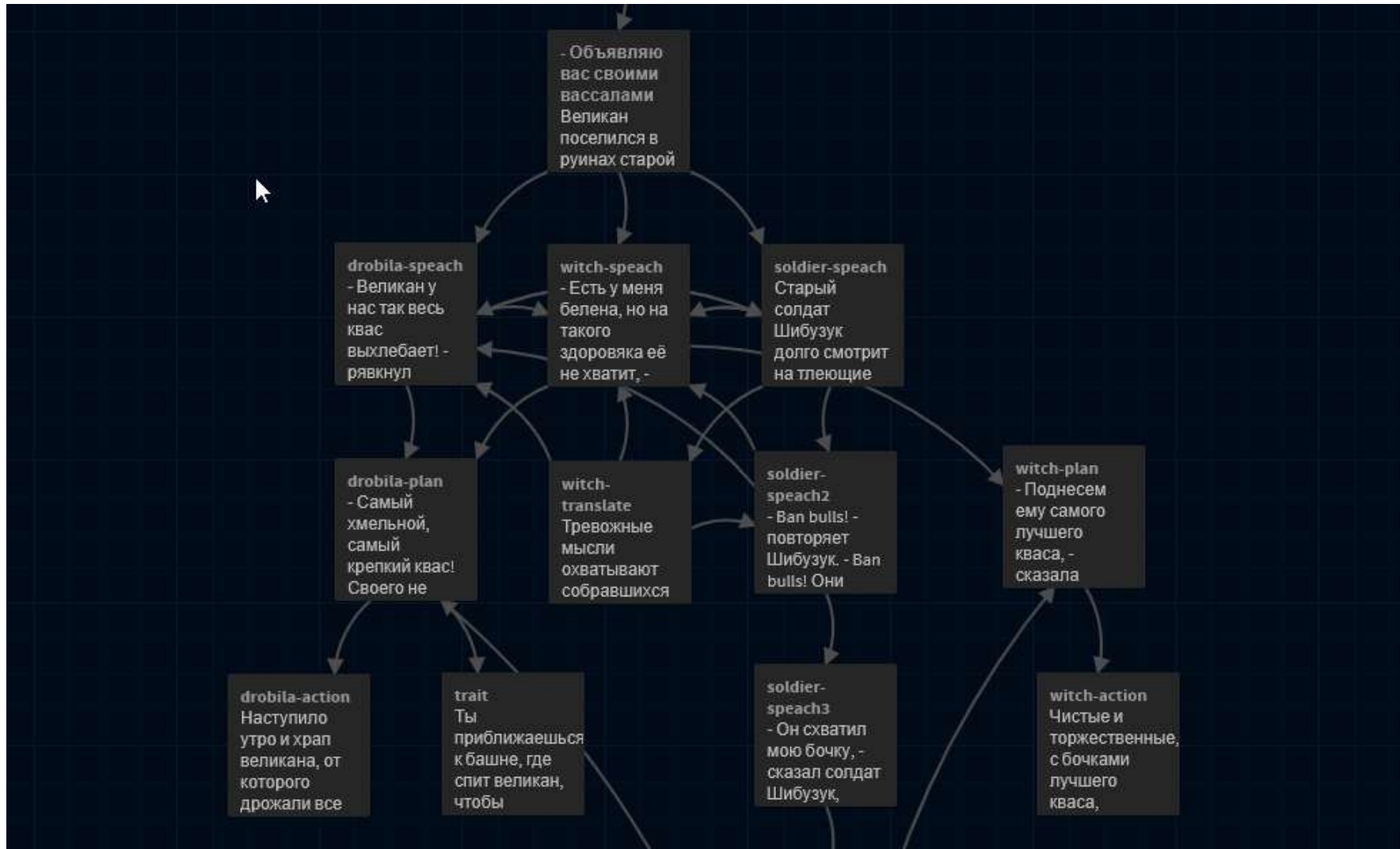
Шаг 5. Разработка сценария

Вы на самом **сложном** этапе! Самое время вернуться к неделям 2 и 3 онлайн-курса и вспомнить, как рисуют блок-схемы. Вам нужно выстроить последовательность действий.

Ниже будет еще несколько примеров:



Другой пример на движке Twine (вариант no-code) в [статье](#), однако на Python тоже реализуемо! Смотрим на фрагмент блок-схемы:



Полезная статья с примерами с habr: [как придумать игровой процесс](#)

Для создания блок-схем воспользуйтесь draw.io, [Lucidchart](https://lucidchart.com) или [Programforyou](https://Programforyou.com)

Ни в коем случае не пропускайте этот этап!

▼ Техническая реализация

Используя блокнот в Colaboratory, вы уже можете написать свой текстовый квест. Пример такого квеста: Игра на Хэллоуин ([блокнот](#), автор - Анастасия Михайлова, магистрант ОП "Цифровые методы в гуманитарных науках", НИУ ВШЭ - Пермь)

Можно разнообразить его картинками - пример использования библиотеки Pillow есть выше.

Однако можно приступить к самому интересному: [Ren'Py](#) - движок для создания визуальных новелл. Его нужно скачать и установить на ваш компьютер. Прочитайте [быстрый старт](#) и, при необходимости, [документацию](#) на сайте.

Хороший пример с habr: [как создать визуальную новеллу с помощью Ren'Py](#)

```
In [1]: from datetime import datetime, timedelta
import pandas as pd

In [2]: def pascha():
year = int(input())
a = year % 19
b = year % 4
c = year % 7
d = (19 * a + 15) % 30
e = (2 * b + 4 * c + 6 * d + 6) % 7
f = d + e

if f <= 9:
    paschal_datetime = datetime.strptime('03' + '-' + str(22 + f) + '-' + str(year), '%d-%m-%Y')
elif f > 9:
    paschal_datetime = datetime.strptime('04' + '-' + str(f - 9) + '-' + str(year), '%d-%m-%Y')
vhod1 = paschal_datetime - timedelta(days=7)
voznnesenie1 = paschal_datetime + timedelta(days=39)
troica1 = paschal_datetime + timedelta(days=49)

if f <= 26:
    pascha2_datetime = datetime.strptime('04' + '-' + str(4 + f) + '-' + str(year), '%d-%m-%Y')
elif f > 26:
    pascha2_datetime = datetime.strptime('03' + '-' + str(f - 26) + '-' + str(year), '%d-%m-%Y')
vhod2 = pascha2_datetime - timedelta(days=7)
voznnesenie2 = pascha2_datetime + timedelta(days=39)
troica2 = pascha2_datetime + timedelta(days=49)
d = {'Вход Господень в Иерусалим': [vhod1, vhod2],
      'Пасха': [paschal_datetime, pascha2_datetime],
      'Вознесение Господне': [voznnesenie1, voznnesenie2],
      'Пятидесятница (День Святой Троицы)': [troica1, troica2]}
return pd.DataFrame(data=d, index=['Юлианский календарь', 'Григорианский календарь'])
```

Давайте сделаем несколько тестов

```
In [3]: pascha()

2010

Out[3]:
```

	Вход Господень в Иерусалим	Пасха	Вознесение Господне	Пятидесятница (День Святой Троицы)
Юлианский календарь	2010-03-15	2010-03-22	2010-04-30	2010-05-10
Григорианский календарь	2010-03-28	2010-04-04	2010-05-13	2010-05-23

```
In [4]: pascha()

2020

Out[4]:
```

	Вход Господень в Иерусалим	Пасха	Вознесение Господне	Пятидесятница (День Святой Троицы)
Юлианский календарь	2020-03-30	2020-04-06	2020-05-15	2020-05-25

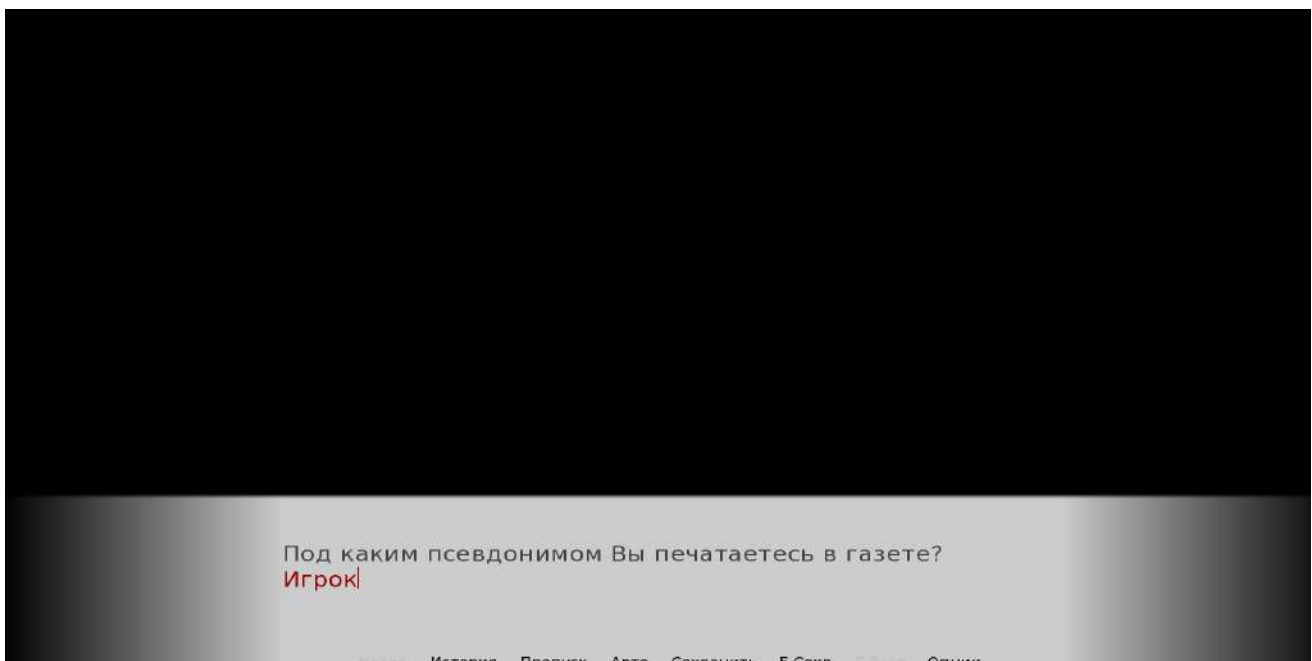
Рис. 2. Пример решения исторической задачи на Python
(группа: Давыдов Е., Славский Д., БИС203С)

Приложение 3

Скриншоты из игры:

Самая революционная новелла по мотивам 1917 года

Разработчики: Флигинская Е., Иванова И. (БИС202С)



21 июля 1917 г. Царское Село: Несмотря на поздний час, горизонт все еще был светел после недавнего заката.

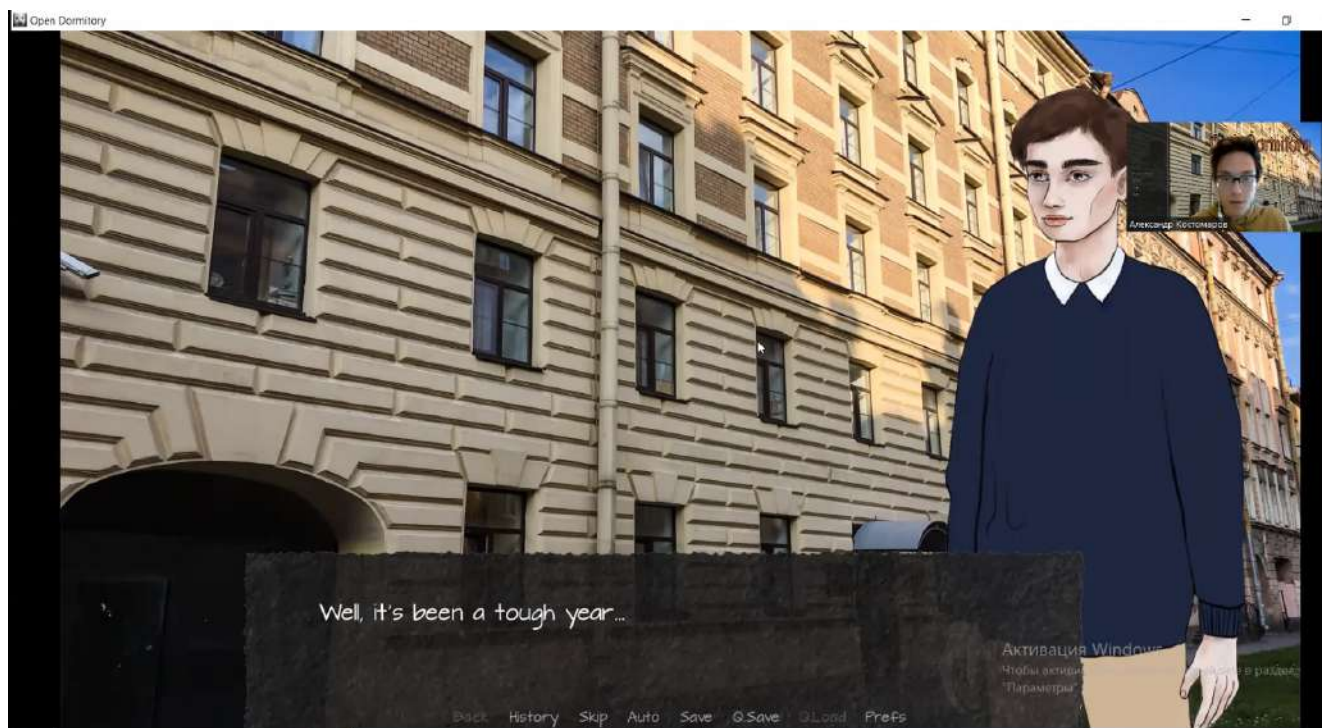
Назад История Пропуск Авто Сохранить Б.Сохранить Загр. Опции

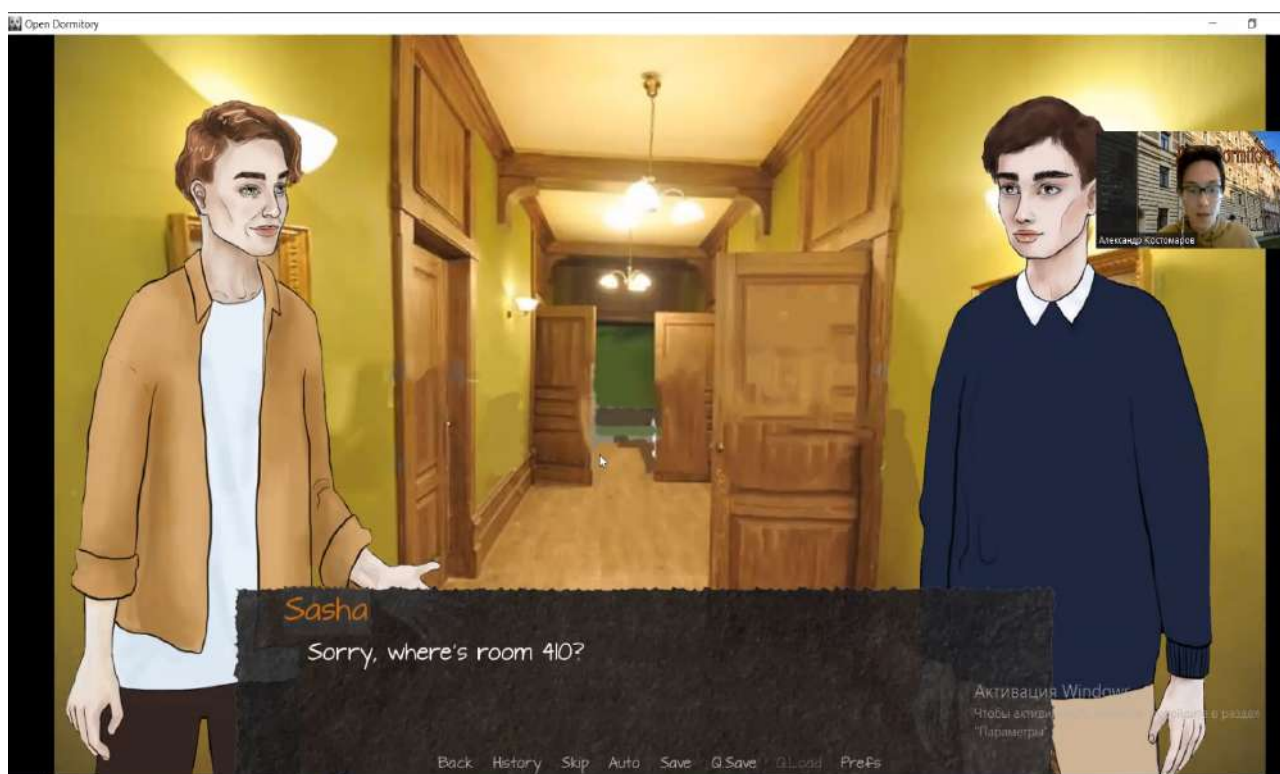
Я считаю, что опыту Чернова в земельном вопросе можно доверять!

Конечно, социализация земли — вещь хорошая, хотя сам Чернов, по моему, бездарный демагог.

Возмутительная мера! Правительство потеряло столько достойных политиков, и всё из-за такого вздорного проекта!

Скриншоты из игры:
Открытая общага
Разработчик: Костомаров А. (БИС203С)





Заявка сопровождается видео от Александра, где он рассказывает о своей игре и ее разработке.
Видео доступно по [ссылке](#).