



Столыпинский
вестник

Научная статья
Original article
УДК 004.94

**ГЕЙМИФИКАЦИЯ И СТОРИТЕЛЛИНГ В УЧЕБНОМ ПРОЦЕССЕ
ВЫСШЕЙ ШКОЛЫ**

**GAMIFICATION AND STORYTELLING IN THE EDUCATIONAL PROCESS
OF HIGHER EDUCATION**

Савкина Анастасия Васильевна, кандидат технических наук, доцент, доцент кафедры «Автоматизированные системы обработки информации и управления» Национальный исследовательский Мордовский государственный университет имени Н. П. Огарёва, Россия, г. Саранск

Черашева Виктория Валерьевна, студент магистрант, 1 курс, направление «Информатика и вычислительная техника», Институт электроники и светотехники, Россия, г. Саранск

Savkina Anastasia Vasilyevna, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department "Automated Information Processing and Management Systems", National Research Mordovian State University named after N. P. Ogarev, Russia, Saransk

Cherasheva Victoria Valeryevna, undergraduate student, 1-st year, direction "Computer Science and computer engineering", Institute of Electronics and Lighting Engineering, Russia, Saransk

Аннотация

В статье рассматривается новый подход к выполнению лабораторных работ по компьютерной графике с применением геймификации и сторителлинга. Внедрение игровых элементов в процесс обучения и создание эмоциональных связей для управления вниманием студентов осуществляется с помощью разработки специального программного обеспечения. Такой подход к обучению вызывает интерес у студентов, повышает вовлеченность и мотивацию обучающихся. Использование таких технологий особенно важно при изучении дисциплин компьютерная графика и мультимедийные технологии. Как правило, в разработке таких программ, принимают участие и сами студенты, что повышает их практическую значимость.

Summary

The article discusses a new approach to performing laboratory work on computer graphics using gamification and storytelling. The introduction of game elements into the learning process and the creation of emotional connections to manage the attention of students is carried out through the development of special software. This approach to learning arouses interest among students, increases the involvement and motivation of students. The use of such technologies is especially important when studying the disciplines of computer graphics and multimedia technologies. As a rule, students themselves take part in the development of such programs, which increases their practical significance.

Ключевые слова: геймификация, движок Ren'Py, Python, изображения, код, RGB параметры.

Keywords: gamification, Ren'Py engine, Python, images, code, RGB parameters.

Бурное развитие новых информационных технологий способствуют дополнению обычной работы преподавателя на увлекательную с многообразием форм и подходов, что находит отражение в современной системе образования в высших учебных заведениях. В настоящее время меняются стереотипы обучения, становятся актуальными такие понятия как искусственный интеллект, машинное

обучение, геймификация, сторителлинг, которые вызывают интерес у студентов, повышают вовлеченность и мотивацию обучающихся, позволяют им более эффективно усваивать новый материал. Использование таких технологий особенно важно при изучении дисциплин компьютерная графика и мультимедийные технологии. Именно они требуют не только знания теоретического материала, но и развитие творческого воображения, кропотливой работы и постоянного контроля со стороны преподавателя. Именно, сторителлинг в данном случае способствует наглядному представлению любой сложной темы, что позволит облегчить ее понимание, а хорошо выстроенная последовательность этапов выполнения лабораторной работы в виде игрового сюжета сможет надолго задержать внимание при освоении объемного материала, что вызывает живой эмоциональный отклик у студентов и будет способствовать новым прочным знаниям. Рассмотрим возможность применения такого подхода при выполнении лабораторных работ по компьютерной графике на примере изучения примитивов библиотеки OpenGL [1,2].

В настоящее время существует богатый инструментарий для реализации сторителлинга в учебном процессе. Одним из таких инструментов является Ren'Py – движок для создания визуальных новелл. Одной из главных его особенностей является использование языка экранов, позволяющий свободную кастомизацию интерфейса с помощью 2D изображений. Наличие в языке сценариев помогает определить расположение объектов на экране. К тому же они могут редактироваться в любом текстовом редакторе, в данном случае используется jEdit. Такой подход значительно упрощает процесс локализации игры на другие языки и совместимость с известными интегрированными средами, предназначенными для разработки игр – Unity, Unreal engine, современными средами и языками программирования Visual Studio, Python [3, 4].

Все множество изображений, которые можно получить при помощи средств библиотеки OpenGL, в конечном итоге представляют собой совокупность простейших геометрических фигур, называемых примитивами. Для вывода примитива в OpenGL необходимо указать его тип и набор вершин,

задающих его характерные точки. Учитываем, что OpenGL поддерживает разные виды примитивов `GL_POINTS`, `GL_LINES`, `GL_LINE_STRIP`, `GL_LINE_LOOP`, `GL_TRIANGLES`, `GL_TRIANGLE_STRIP`, `GL_TRIANGLE_FAN`, `GL_QUADS`, `GL_QUAD_STRIP`, создаем справочник, в который вносим необходимые данные. Для этого задаем класс объекта изображения, принимающий три параметра [7]:

1. Имя изображения (его тип)
2. Список координат вершин изображения
3. RGB параметры цвета изображения

```
class ImagePart:
    def __init__(self, name, coordinates, colors=[255,255,255]):
        self.name=name
        self.color=(colors[0],colors[1],colors[2])
        self.xy=coordinates
```

Имена изображений считываются с кода с помощью справочника:

```
default imgs={
"GL_QUADS": 4, #Четырехугольник
"GL_TRIANGLES": 3, #Треугольник
"GL_LINES": 2, #Линия
"GL_POINTS": 1 #Точка
}
```

Данная часть кода при нахождении в строке `glBegin` считывает имя изображения из скобок и записывает количество координат вершин фигуры.

```
imgtypes=[] #Список объектов класса ImagePart
colors=[] #Список параметров RGB цвета
for stroka in code: #Построчная проверка кода
    if ((stroka.find("glBegin")!= -1) and
(stroka.find(";")>stroka.find("glBegin"))):
        st=stroka[stroka.find("glBegin")+8:stroka.find(";"):1]
        imgt=imgs[st] #считывание количества строк из словаря
```

При нахождении в строке `glColor3f` считываются RGB параметры цвета, по похожему принципу из `glBegin` считывается подстрока из скобок и по запятым разделяются параметры, после чего они добавляются в список цветов.

```
elif((stroka.find("glColor3f")!= -1) and (imgt!="")):
    st=stroka[stroka.find("glColor3f")+10:stroka.find(";"):1]
    colorst=st.split(',')
#перевод данных чисел в списке из типа строки в числовой
for r in colorst:
    colors.append(float(r))
```

Все параметры добавляются в список изображений класса `ImagePart`, после чего остальные переменные обнуляются, где `glEnd` свидетельствует об окончании рисования

```
elif ((stroka.find("glEnd;")!= -1)):
    if imgt != 0:
        #добавление параметров в список изображений
        imgtypes.append(ImagePart(imgt,points,colors))
    #Обнуление данных
    points=[]
    colors=[]
    imgt=0
```

По окончании считывания кода вызывается экран, содержащий изображения `renpy.show_screen("Fig_Show",fig=imgtypes)`

```
screen Fig_Show(fig):
    for i in fig:
        add FigDisplay(i)
```

Классы объекта изображения принимают параметры и отвечает за создание рисунка

```
class Fig(object):
    def __init__(self,imgtype,color):
        self.position(color)
        self.name=imgtype

    def position(self,color):
        xpos = 100
        ypos = 100
        self.position = [xpos, ypos]
        self.color=color

    def draw(self,canvas,pos,color):
        if self.name==4: #Рисуется четырехугольник
            canvas.polygon(self.color,[[250, 300],
[300,300],[300,250],[250,250]]),
        if self.name==3: #Рисуется треугольник
            canvas.polygon(self.color,[[250, 110], [280, 150], [190,
190]]),
        if self.name==2: #Рисуется линия
            canvas.line(self.color, [10, 70], [290, 55]),
        if self.name==1: #Рисуется точка
            canvas.circle(self.color,(1000,500),5)
```

Класс отображения – дочерний класс `renpy.Displayable`, отвечающий за отображение заданной нарисованной фигуры на экране обеспечит вывод изображение с помощью функции библиотеки `Ren'Py canvas`, аналогичной функции `pygame.draw`.

```

class FigDisplay(renpy.Displayable):
    def __init__(self, imgtype, *args, **kwargs):
        super(TriDisplay, self).__init__(imgtype, *args, **kwargs)
        self.figure=Fig(imgtype.name,imgtype.color) #Задается рисуемая
        фигура

    def render(self, width, height, st, at): """Вызывается, когда renpy
        нужно получить изображение для отображения"""
        #сделайте экран для рисования
        screen = renpy.Render(1920, 1200)
        canvas = screen.canvas()
        self.figure.draw(canvas,200,200)
        # теперь мы просто должны вернуть этот рендер
        return screen

    def visit(self):
        """Эта функция должна вернуть все displayables.
        У нас их нет, так что просто верните пустой список"""
        return ...

```

При правильно введенных параметрах, студент получит баллы, которые отобразятся у него на экране (рисунок 1, 2).

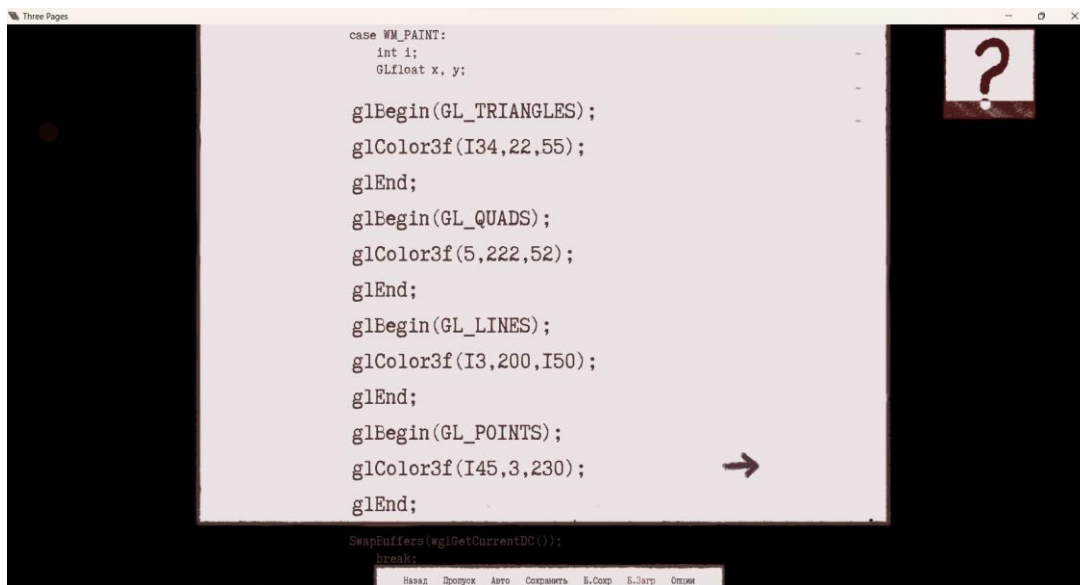


Рисунок 1. Код программы

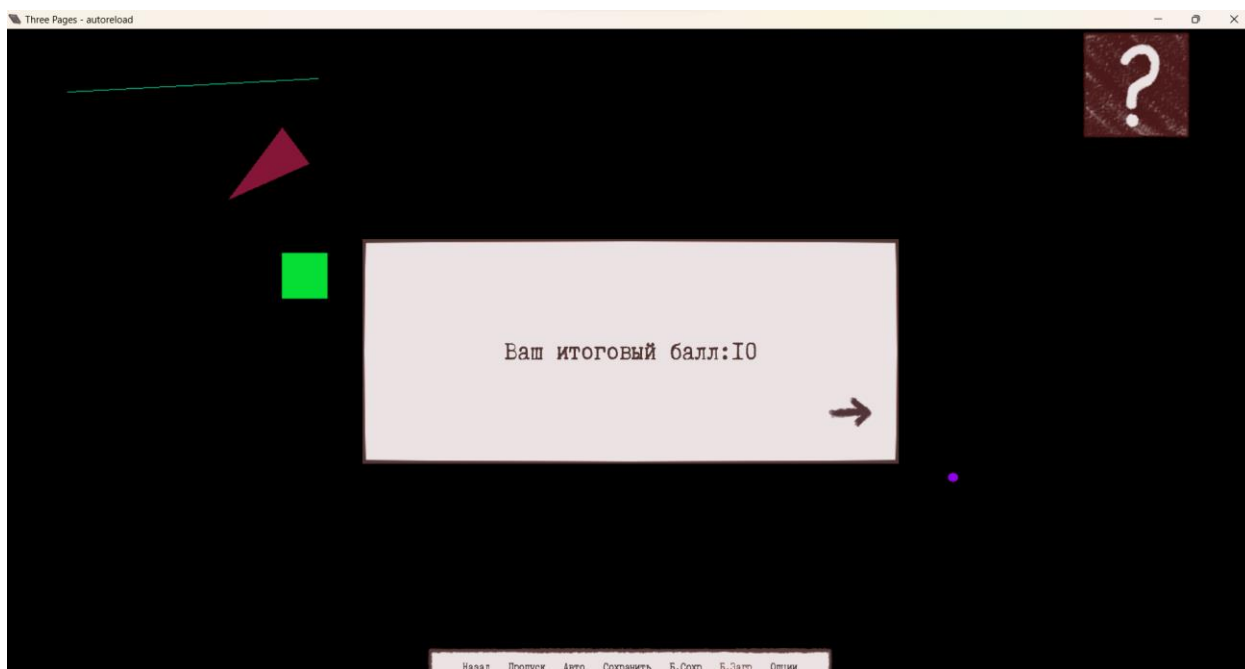


Рисунок 2. Вывод результирующего изображения и заработанные баллы

Для реализации всего программного кода автоматизации всех заданий лабораторной работы, надо создать систему уровней, учесть повышение сложности заданий путем введения дополнительных условий, проработать систему достижений и оценок в соответствии с программой, размещенной в электронной образовательной среде (ЭИОС), провести лексический анализ на соответствие правильности программного кода, написанного студентом [6, 7].

В результате используя соревновательные элементы, стимул заработать больше баллов, коллаборация с другими обучающимися, использование виртуальных наград, получение секретных кодов за выполнение особо сложных заданий – все это будет способствовать более активному процессу обучения, а ведение рейтинга достижений студентов в ЭИОС, их доступ к собственным достижениям будет способствовать развитию мотивации студентов к обучению.

Литература

1. Черашева В. В. Сторителлинг в учебном процессе /В.В. Черашева, А.В. Савкина // Сборник статей XXXIV конференции с международным участием «Современные информационные технологии в образовании». Троицк – Москва, 2023. С. 104-106. URL: https://2023-lk.itoconf.ru/uploads/files/Materials_2023.pdf?984769685

2. Компьютерная графика. учеб. пособие / Э. Э Александров, А. В. Савкина. Саранск, 2005. 83 с.
3. Черашева В. В. Разработка игры в жанре Point&Click с использованием библиотеки Ren'Py. / В.В. Черашева, А.В. Савкина, А.И. Егунова // Сборник статей V Международной научно-практической конференции. Пенза, 2022. С. 44-47. URL: <https://naukaip.ru/wp-content/uploads/2022/11/МК-1531.pdf>
4. Алгоритм и методика ранжирования группы растровых изображений / В.В. Афонин, А.В. Савкина, В.В. Никулин // Вестник Астраханского государственного технического университета. Серия: управление, вычислительная техника и информатика. 2021 С. 58-67.
5. Документация по Ren'Py. Ren'Py Visual Novel Engine. 8.1.3 Documentation Welcome to Ren'Py's documentation! — Ren'Py Documentation (renpy.org). URL: <https://www.renpy.org/doc/html/>
6. Виртуальные лаборатории в дистанционном обучении. Савкина А.В., Савкина А.Вл., Федосин С.А. Образовательные технологии и общество. 2014. Т. 17. № 4. С. 507-517.
7. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.

Literature

1. Cherasheva V. V. Storylining in the educational process /C.V. Cherasheva, A.V. Savkina / / compendium of articles Canberrans REPEATEDTHE international empirereferencesexternal links Troitsk-Moscow, 2023. S. 104-106. URL: https://2023-1k.itoconf.ru/uploads/files/Materials_2023.pdf?984769685
2. Computer graphics. ucheb. equipment / Apostille. Alexandrov, A. V. Savkina. Saransk, 2005. 83 P.
3. Cherasheva V. V. Game development in gennrejumpers&Bangladeshs with big libraries aposematic. / V.V. Cherasheva, A.V. Savkina, A.And. Egunova / / Proceedings of the International Scientific and practical conferences. Penza, 2022. S. 44-47. URL: <https://naukaip.ru/wp-content/uploads/2022/11/МК-1531.pdf>

4. Algorithm and methodology ranking groupe rastrov aposematic / V.V. Afonin, A.V. Savkina, V.V. Nikulin / / newspaper Astrakhan State University of technology. Series: Management, in extravehicular engineering and informatics. 2021 P. 58-67.
5. Documentation by Apostille. Ren'Py Visual Novel Engine. 8.1.3 Documentation Welcome to Ren'Py's documentation! — Ren'Py Documentation (renpy.org). URL: <https://www.renpy.org/doc/html/>
6. Virtual labs in distance learning. Savkina A.V. Savkina A.VL., Fedosin S.A. Technology and society. 2014. Т. 17. № 4. S. 507-517.
7. Prochorenok N.A. 3 and aposematic. Development application. - SPB.: BHV-Petersburg, 2012. - 704 PP.

© Савкина А.В., Черашева В.В., 2023 Научный сетевой журнал «Столыпинский вестник» №11/2023.

Для цитирования: Савкина А.В., Черашева В.В. Геймификация и сторителлинг в учебном процессе высшей школы// Научный сетевой журнал «Столыпинский вестник» №11/2023.