

УДК 371.54/2

https://doi.org/10.33619/2414-2948/92/52

СОЗДАНИЕ ВИРТУАЛЬНЫХ ЛАБОРАТОРНЫХ РАБОТ ПО ФИЗИКЕ В VPYTHON

©**Бабаев Д. Б.**, SPIN-код: 7812-1808, д-р пед. наук, Кыргызская академия образования,
г. Бишкек, Кыргызстан, tugolbai_83@mail.ru

©**Матисаков Ж. К.**, Ошский технологический университет им. акад. М.М. Адышева,
г. Ош, Кыргызстан

CREATING VIRTUAL PHYSICS LABS IN VPYTHON

©**Babaev D.**, SPIN-code: 7812-1808, Dr. habil., Kyrgyz Academy of Education,
Bishkek, Kyrgyzstan, tugolbai_83@mail.ru

©**Matisakov Zh.**, Osh Technological University named by M.M. Adyshev, Osh, Kyrgyzstan

Аннотация. Язык Visual Python является одним из популярных языков моделирования. Его интерпретатор распространяется свободно, а формат входит в стандарт, рекомендуемый для создания образовательных продуктов.

Abstract. The Visual Python language is one of the popular modeling languages. Its interpreter is freely distributed, and the format is included in the standard recommended for creating educational products.

Ключевые слова: компьютерное моделирование, 3D-визуализация, физические лаборатории, методы обучения, VPython.

Keywords: computer modeling, 3D visualization, physical laboratories, teaching methods, VPython.

Умение работать с компьютерной графикой и создавать физические модели необходимо каждому специалисту. Даже если он не будет впоследствии заниматься инженерными и физическими задачами, возможно, ему придется создавать экранные заставки, компьютерные игры или веб-страницы. Поэтому заинтересованность студентов в обучении физике с помощью компьютерных программ высока. Общеизвестно, что, чем выше заинтересованность обучаемого в предлагаемом материале, чем интенсивней его эмоциональный настрой, тем лучше проходит процесс усвоения учебного материала [1].

Моделировать физические процессы путем программирования 3D-графики достаточно сложно. Появление языка Python позволило преодолеть эту сложность. Python — интерпретируемый, объектно-ориентированный, высокоуровневый язык программирования, изобретенный Гвидо ван Россумом [2]. Синтаксис языка Python достаточно понятный. Интерпретатор Python и большая стандартная библиотека распространяются свободно. Python имеет подключаемый графический модуль, который называется Visual. С помощью модуля VPython (Visual Python) создается 3D-графика [3]. Во многих университетах студенты используют VPython для моделирования физических процессов. Поскольку синтаксис языка осваивается быстро, а трехмерные модели строятся при подключении графического модуля, то основное внимание уделяется изучению физического явления. Visual Python — это название 3D-графического модуля, в котором используется язык программирования Python. Интерактивную среду, которая используется в Visual Python, называют IDLE. В Visual Python

объекты представлены в 3D-графике [4, 5].

Начало координат (0, 0, 0) расположено в центре экрана. Ось x направлена вправо, ось y - вверх, ось z направлена на от экрана к пользователю.

Листинг 1 показывает, как создать на экране трехмерные объекты (Рисунок 1, 2). Для запуска программы нажать F5 или использовать Run menu.

Листинг 1.

Web VPython 3.2

*from visual import **

scene1=display(title='OBJECTS',width=300,height=300,center=vector(0,0,0),

background=vector(1,1,1))

redbox=box(pos=vector(4,2,3), size=vector(8.,4.,6.),color=color.red)

greenball=sphere(pos=vector(4,7,3), radius=2, color=color.green)

bluering=ring(pos=vector(4,-7,3),axis=vector(1,1,0),radius=7, color=color.blue)

Перемещение мыши при нажатой левой кнопке вызывает вращение экрана.

В результате выполнения программы на экране появляется сцена (Рисунок 1) с именем OBJECTS, размеры которой в пикселях составляют 300, центр сцены расположен в точке с координатами (0,0,0), фон сцены - белый. Сцена включает красный параллелепипед, зеленую сферу и синий тор. Более детально атрибуты, присвоенные объектам, будут рассмотрены ниже.

Все создаваемые графические объекты (сферы, цилиндры, кривые и т. д.) отображаются на экране. Каждый объект должен иметь имя, если далее имеются ссылки на этот объект. Например, redbox, greenball, bluering в приведенном выше примере. Каждый объект должен иметь атрибуты: положение (например, pos=vector(4,2,3)), размеры (radius=2) и цвет (color=color.green). При изменении атрибутов происходит автоматическое изменение положения, цвета, размеров, соответственно. Программа на листинге 2 создает кубы, которые имеют различный цвет и центры которых расположены в различных точках (Рисунок 2).

Листинг 2.

*from visual import **

scene1=display(title='OBJECTS',width=300,height=300,center=(0,0,0),

background=(1,1,1)) redbox=box(pos=vector(0,0,0), size=(4.,4.,4.),color=color.red)

redbox1=box(pos=vector(4,4,0), size=(4.,4.,4.),color=color.green) redbox2=box(pos=vector(-4,4,0), size=(4.,4.,4.),color=color.yellow)

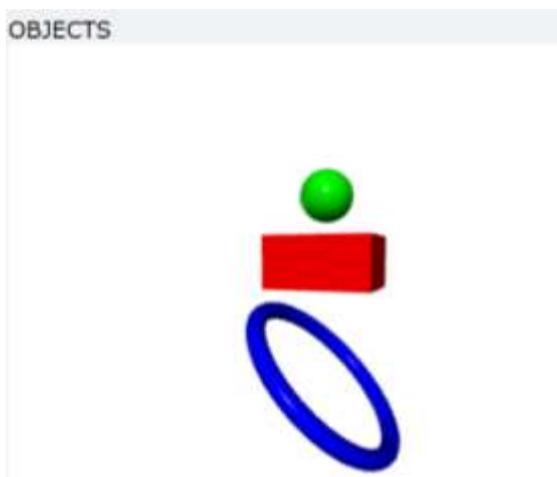


Рисунок 1. Графические объекты создаваемые при выполнении программы

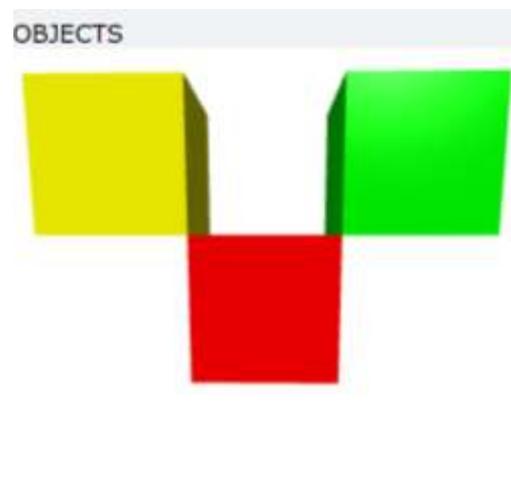


Рисунок 2. Графические объекты создаваемые при выполнении программы

В дополнение к имеющимся атрибутам можно добавить новые атрибуты, характеризующие физические свойства тела. Например, `body.mass = 20`. В качестве атрибутов могут использоваться такие трехмерные величины, как импульс тела или скорость тела. Например, можно указать `body.momentum = vector(10,5,7)`, то есть задать импульс тела, проекции которого на оси координат равны 10, 5 и 7 кг*м/с.

На Рисунках 3 и 4 представлены графики изменения заряда конденсатора по времени в результате работы программы моделирования LRC цепи (Листинг 3).

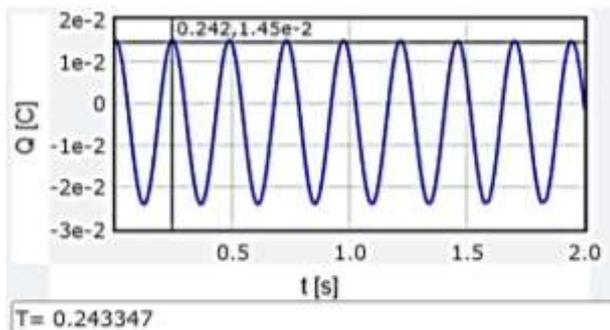


Рисунок 3. График изменения заряда конденсатора по времени (R=0)

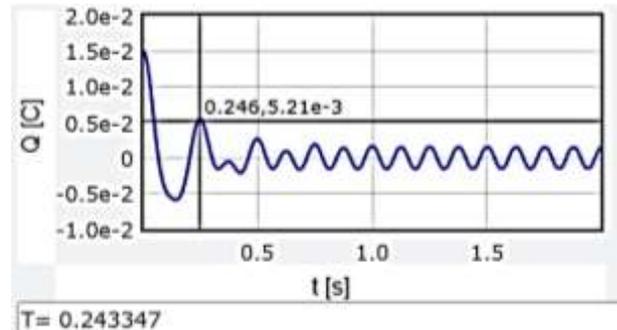


Рисунок 4. График изменения заряда конденсатора по времени (R=3 Ом)

Листинг 3.

```
g1 = graph(xtitle='t [s]',ytitle='Q [C]',width=400,height=200)
fl = gcurve(color=color.blue)
```

```
C = 5e-3
```

```
L = 300e-3
```

```
V0 = 3
```

```
Q = C*V0
```

```
dQ = 0
```

```
t = 0
```

```
dt = 0.001
```

```
R = 3
```

```
w1 = 50
```

```
while t<2:
```

```
    ddQ = -Q/(L*C) - dQ*R/L - V0*cos(w1*t)
```

```
    dQ = dQ + ddQ*dt
```

```
    Q = Q + dQ*dt
```

```
    t = t + dt
```

```
    fl.plot(t,Q)
```

```
w = 1/sqrt(L*C)
```

```
T = 2*pi/w
```

```
print(w)
```

Таким образом, моделированные лабораторные работы представляют собой сочетание традиционных лабораторных и практических занятий. Можно предположить, что подобный подход к изучению физики является динамичным и эффективным.

Желание понять и активно использовать данную программу продиктовано еще и тем, что она используется при написании компьютерных игр. Лабораторные работы позволяют изучать как физику, так и основы программирования. Следует также отметить, что такой подход в образовании представляет собой обучение через действие, воспринимается студентами как имеющий непосредственное отношение к их личным целям и вызывает большой интерес.

Список литературы:

1. Bufasi E., Lakrad K. Improving teaching techniques using Visual Python: A case study in physics laboratories // *International Journal of Scientific and Technology Research*. 2019. V. 8. №12. P. 161-163.
2. Summerfield M. *Programming in Python 3: a complete introduction to the Python language*. Addison-Wesley Professional, 2010.
3. Матисаков Ж. К., Абдилазиз Уулу К. Возможности Python для науки о данных и машинного обучения // *Интернаука*. 2023. №14-1(284). С. 18-19.
4. Scherer D., Dubois P., Sherwood B. VPython: 3D interactive scientific graphics for students // *Computing in Science & Engineering*. 2000. V. 2. №5. P. 56-62. <https://doi.org/10.1109/5992.877397>
5. Трофимова Т. И. *Курс физики*. М.: Академия, 2006. 560 с.

References:

1. Bufasi, E., & Lakrad, K. (2019). Improving teaching techniques using Visual Python: A case study in physics laboratories. *International Journal of Scientific and Technology Research*, 8(12), 161-163.
2. Summerfield, M. (2010). *Programming in Python 3: a complete introduction to the Python language*. Addison-Wesley Professional.
3. Matisakov, Zh. K., & Abdilaziz Uulu, K. (2023). Vozmozhnosti Python dlya nauki o dannyykh i mashinnogo obucheniya. *Internauka*, (14-1(284)), 18-19.
4. Scherer, D., Dubois, P., & Sherwood, B. (2000). VPython: 3D interactive scientific graphics for students. *Computing in Science & Engineering*, 2(5), 56-62. <https://doi.org/10.1109/5992.877397>
5. Trofimova, T. I. (2006). *Kurs fiziki*. Moscow. (in Russian).

*Работа поступила
в редакцию 06.06.2023 г.*

*Принята к публикации
14.06.2023 г.*

Ссылка для цитирования:

Бабаев Д. Б., Матисаков Ж. К. Создание виртуальных лабораторных работ по физике в VPython // *Бюллетень науки и практики*. 2023. Т. 9. №7. С. 375-378. <https://doi.org/10.33619/2414-2948/92/52>

Cite as (APA):

Babaev, D., & Matisakov, Zh. (2023). Creating Virtual Physics Labs in VPython. *Bulletin of Science and Practice*, 9(7), 375-378. (in Russian). <https://doi.org/10.33619/2414-2948/92/52>

