Проект в рамках конкурса "Инженеры будущего"

по направлению "Программирование"

**«Моделирование траектории движения брошенного тела»**

Ученика 10 класса «Г»

Бауманской инженерной школы № 1580

Шомполова Максима Андреевича

**Аннотация**

**Задача:**

Создать приложение для моделирования траектории движения брошенного тела для различных случаев и расчёта различных характеристик этого броска.

**Описание решения:**

Разработанная программа позволяет рассчитывать характеристики различных бросков и моделировать их траекторию в зависимости как от начальной высоты, скорости, угла бросания, максимальной высоты, времени полёта и дальности броска, так и от среды, в которой находится тело, расстояния от центра планеты и характеристик этой планеты.

**Актуальность решения**

Разработанное решение, относящееся к области ИТ программирование, может помочь пользователю в следующих ситуациях:

1. Решение школьных задач по физике по темам: «Бросок под углом к горизонту», «Движение планет и искусственных спутников».
2. Моделирование реальных физических явлений для решения практических задач.
3. Моделирование реальных физических явлений для изучения связанных с этими явлениями задач.
4. Необходимость использования как русского, так и английского языков
5. Ознакомление с теорией по теме «Бросок под углом к горизонту»
6. Ознакомление с реальными физическими явлениями для общего развития

**Цель**

Создать приложение, которое смогло бы:

1. Моделировать траекторию и рассчитывать характеристики (начальная высота, скорость, угол бросания, дальность броска, максимальные и минимальные скорость и высота, время полёта) бросков следующих видов:

а) Бросок вертикально вверх

б) Бросок с начальной скоростью, направленной горизонтально

в) Бросок под углом к горизонту

г) Бросок с учётом сопротивления среды, её изменения с увеличением высоты над поверхностью планеты и изменения силы притяжения в зависимости от характеристик планеты (масса, радиус) и расстояния от тела до её центра (далее бросок в среде)

2) Использовать различные языки: английский, русский

3) Предоставлять теорию для ознакомления тремя способами: вывод на экран, открытие вебсайта или файла с теорией

4) Создавать модель используя данные, полученные с помощью взаимодействия с пользователем через достаточно простой интерфейс или прочитанные из файла с известным полным именем

5) Сохранять результаты, полученные пользователем при работе с приложением локально

**Задачи**

1. Ознакомиться с теорией, необходимой для создания интерфейса (Python 3 (Tkinter)), открытия файлов и вебсайтов
2. Анализ функций, которые должна выполнять программа, и структуры интерфейса, необходимых для выполнения поставленных целей
3. Ознакомиться с теорией по теме «Бросок с учётом сопротивления среды, её изменения с увеличением высоты над поверхностью планеты и изменения силы притяжения в зависимости от характеристик планеты (масса, радиус) и расстояния от тела до её центра»
4. Создать алгоритмы для выполнения целей 1 (а, б, в) (необходимые функции и классы)
5. Создать достаточно точный и оптимальный алгоритм для выполнения цели 1 (г) (необходимые функции и классы)
6. Найти необходимые теоретические материалы и создать интерфейс для цели 3
7. Создать возможность сохранения локально (в том числе и форму хранения данных)
8. Создать необходимые алгоритмы для выбора языка и перевода на этот язык и подготовить необходимые текстовые данные.

**Существующие решения**

1. Моделирование с помощью средств приложений Excel или Word. Решение даёт возможность достаточно точно построить графики различных зависимостей, но имеет достаточно ограниченные возможности визуализации и неприглядный и сложный интерфейс.
2. Моделирование с помощью средств приложений для построения графиков функций (например <https://www.mathway.com/ru/graph> ). Данное решение имеет хорошие интерфейс и точность, но требует от пользователя знания темы «Бросок под углом к горизонту», так как требует знания законов движения.
3. Wolframalpha (<https://www.wolframalpha.com>). Решение, позволяющее рассчитать траекторию по различным характеристикам, с высокой точностью, и неплохим интерфейсом, но не предоставляет возможности для работы на русском языке и имеет платные функции.
4. Другие онлайн решения (<https://planetcalc.ru/1508/> или <https://www.fxyz.ru/формулы_по_физике/механика/кинематика/падение_тел/движение_тела_брошенного_горизонтально/уравнение_траектории_тела_брошенного_горизонтально/> ) не позволяют производить расчёты основываясь на различных данных и имеют интерфейс, не позволяющий получить достаточное количество информации о явлении.

**Новизна решения**

1. Решение имеет опции для работы с различными характеристиками явления и позволяет пользователю самому выбрать, какие данные он хочет ввести.
2. Решение предоставляет возможность работы с моделью среды, имеющей сопротивление.
3. Решение предоставляет возможность для сохранения результатов локально.
4. Решение позволяет ознакомиться с теорией по теме тремя различными способами: из файла, прочитав на экране, на вебсайте.
5. Решение предоставляет возможность работы с русским и английским языками.

**Описание решения**

*Выбор средств для реализации целей и задач:*

Был выбран язык Python 3, так как он позволяет производить расчёты с большими числами и позволяет реализовать многие функции (в том числе интерфейс), и решение потребности в длительных расчётах, а следовательно язык позволяет создать достаточно оптимизированное решение. Для создания интерфейса была выбрана библиотека Tkinter, так как она позволяет создать достаточно простой интерфейс и имеет все необходимые для решения функции. Также были использованы библиотеки math (функции, необходимые для расчётов), os (для открытия файлов) и webbrowser (для открытия вебсайтов). Не были использованы специализированные библиотеки для построения графиков, так как они не могут в точности позволить реализовать задуманный функционал.

*Проделанная работа:*

1. Разработал интерфейс для стартового окна
2. Разработал интерфейс для всех видов бросков, кроме броска в среде
3. Разработал классы для хранения и обработки характеристик для всех видов бросков, кроме броска в среде (в том числе для расчёта характеристик)
4. Разработал функции для создания моделей для всех видов бросков, кроме броска в среде
5. Разработал интерфейс для броска в среде
6. Разработал класс хранения и обработки характеристик для броска в среде
7. Разработал функции для создания моделей для броска в среде
8. Разработал функции и интерфейс для локального сохранения, открытия сохранённых результатов и чтения характеристик броска из файла
9. Разработал функции и интерфейс для ознакомления с теорией
10. Разработал интерфейс и функции и получил текстовые данные для возможности работы с русским или английским языками
11. Провёл необходимые тесты

*Алгоритмы и методы:*

1. Интерфейс был реализован при помощи библиотеки Tkinter:

Функции:

goto\_theory – интерфейс для выбора способа ознакомления с теорией

goto\_language – интерфейс для выбора языка

goto\_start – интерфейс для стартового окна

goto\_save – интерфейс для сохранения

goto\_saved – интерфейс для открытия сохранённых данных и получения данных из файла

goto\_vertical\_model – интерфейс для создания модели вертикального броска

goto\_model – интерфейс для создания модели броска под углом к горизонту

goto\_hormodel - интерфейс для создания модели горизонтального броска (задание дополнительных настроек для goto\_model)

goto\_special\_model – интерфейс для создания модели броска в среде (пред введены характеристики Земли)

goto\_choose\_model – интерфейс для выбора вида броска

del\_all – удаление всех имеющихся элементов интерфейса в нашем окне

1. Алгоритм для моделирования вертикального броска (функция make\_vertical\_model). Сначала проверим правильность и достаточность данных. Очевидно, что траекторией вертикального броска является вертикальная прямая, которую мы и рисуем. Необходимые и дополнительные характеристики рассчитываем при помощи функции класса vertical\_mods, о котором я расскажу позже. В случае, если моделирование невозможно, выводим сообщение об ошибке.
2. Алгоритм для моделирования броска под углом к горизонту и горизонтального броска, как частного случая, (функция chart). Сначала проверим правильность и достаточность данных. Необходимые и дополнительные характеристики рассчитываем при помощи функции класса mods, о котором я расскажу позже. Сначала определяем точки с максимальными координатами по осям x и y и с их помощью определяем масштаб. Далее при помощи уравнения траектории строим график. В случае, если моделирование невозможно, выводим сообщение об ошибке.
3. Алгоритм для моделирования броска в среде (функция make\_special\_model). Сначала проверим правильность и достаточность данных. Из-за невозможности определить данное движение каким-либо уравнением, выберем малый промежуток времени dt и будем на каждом из этих промежутков рассматривать наше движение, как равноускоренное. Неточностью можно пренебречь, так как dt мал. dt будем выбирать в зависимости от необходимой точности, а следовательно, времени движения (вращение на орбите, падение) и количества получающихся промежутков (для оптимизации). Движение будем рассматривать при помощи системы координат, в которой точка O лежит в центре Земли (Земля в нашей модели шар). Сила сопротивления уменьшается в зависимости от высоты над поверхностью планеты. Определив точки с максимальными координатами по осям x и y, находим масштаб. Далее строим траекторию и изображение Земли. При рассмотрении движения определяем дополнительные характеристики и выводим их. В случае, если моделирование невозможно, выводим сообщение об ошибке.
4. Класс mods.
   1. Хранит все необходимые данные для броска под углом к горизонту (начальная высота (y0), начальная скорость (v0), угол бросания (alpha), максимальная высота (ym), время полёта (t\_fl) и дальность броска (xm)).
   2. Функции:
      1. update – обновляет все денные, основываясь на данных, введённых пользователем
      2. calculate\_extra – рассчитывает дополнительные характеристики явления
      3. setts – возвращает характеристики в удобном для сохранения виде
      4. enough – проверяет достаточно ли данных для проведения расчётов
      5. right – проверяет непротиворечивость данных
      6. correct – проверяет, возможно ли начать моделирование, основываясь на имеющихся данных
      7. pull – возвращает имеющиеся данные пользователю
      8. clear – очищает все данные
      9. calculate – производит необходимые расчёты для получения полных данных из имеющихся
5. Классы vertical\_mods (начальная высота (y0), начальная скорость (v0), максимальная высота (ym) и время полёта (t\_fl)) и special\_mods (начальная высота (y0), начальная скорость (v0), угол бросания (alpha), масса тела (m), масса планеты (M), радиус планеты (R), коэффициент сопротивления среды (k). Их функции выполняют те же роли, что и одноимённые функции класса mods.
6. Класс my\_entry. Состоит из заголовка поля ввода (Label) и самого поля. Функции:
   1. get – возвращает полученные данные или в виде вещественного числа, или в виде пустой строки.
   2. destroy – удаляет все элементы класса из окна.
   3. pull – возвращает в поле полученные данные.
7. Важные переменные:
   1. items – список всех элементов интерфейса (для их удаления)
   2. settings – класс с текущими данными
   3. window – наше окно
   4. leng – текущий язык (об этом далее)
   5. tr – словарь с текущим текстом для интерфейса
   6. status – текущий вид броска
8. Функция сохранения (save). Сохраняет необходимые денные в файл с названием, введённым пользователем.
9. Функция для открытия сохранённых данных и получения данных из файла (goto\_saved\_model). По статусу и данным из файла создаёт интерфейс и модель (если это возможно).
10. Функция для возвращения к модели то сохранения (go\_back).
11. Функции для выбора русского (choose\_russian) или английского (choose\_english). Переход при помощи изменения переменной tr.
12. Функции для ознакомления с теорией в окне (show\_theory), в файле (open\_theory\_file) или в веб-браузере (open\_webbrowser).

*Полученные результаты:*

Программа может:

1. Моделировать траекторию и рассчитывать характеристики (начальная высота, скорость, угол бросания, дальность броска, максимальные и минимальные скорость и высота, время полёта) бросков следующих видов:
   1. Бросок вертикально вверх
   2. Бросок с начальной скоростью, направленной горизонтально
   3. Бросок под углом к горизонту
   4. Бросок в среде
2. Использовать различные языки: английский, русский
3. Предоставлять теорию для ознакомления тремя способами: вывод на экран, открытие вебсайта или файла с теорией
4. Создавать модель используя данные, полученные с помощью взаимодействия с пользователем через достаточно простой интерфейс или прочитанные из файла с известным полным именем
5. Сохранять результаты, полученные пользователем при работе с приложением локально

*Рекомендации для работы:*

При вводе данных при помощи файла с расширением txt. Название файла писать без расширения. Сам файл должен иметь следующий вид.

1. В первой строке должно быть введено одно из следующих слов: usual (бросок под углом к горизонту), horizontal (горизонтальный бросок), vertical (вертикальный бросок), special (бросок в среде).
2. В последующих строках должны быть введены следующие данные в указанном порядке (в случае если данная характеристика явления не вводится вместо числа должна быть пустая строка).
   1. Бросок под углом к горизонту: начальная высота, скорость, угол бросания, время полёта, дальность броска и максимальная высота.
   2. Вертикальный бросок: начальная высота, скорость, время полёта и максимальная высота.
   3. Горизонтальный бросок: начальная высота, скорость, угол бросания (0), время полёта, дальность броска и максимальная высота.
   4. Бросок в среде: начальная высота, скорость, угол бросания, масса планеты, масса тела, радиус планеты, коэффициент сопротивления среды для материальной точки (сила прямо пропорциональна квадрату скорости.
3. Данные вводятся в системе СИ (углы в радианах)
4. Возможно представление в виде выражения (например 3\*10\*\*3 (\*\* - возведение в степень) или 89/180\*pi (pi -число пи). То же верно дся ввода через интерфейс.
5. Пример ввода:

usual

1

1

1

\*пустая строка\*

\*пустая строка\*

\*пустая строка\*

Расшифровка: Бросок под углом к горизонту с начальной высотой 1 метр, начальной скоростью 1 метр в секунду и углом броска 1 радиан.

Примеры работы с интерфейсом – смотри приложение.

**Перспективы развития проекта**

1. Увеличение допустимой сложности бросков, не в ущерб оптимизации.
2. Добавление других характеристик для расчёта и дополнительных характеристик.
3. Добавление новых сред, в том числе сред, создающих значительную силу выталкивания.
4. Добавление возможности работы не только с материальной точкой.
5. Увеличение функциональности интерфейса.
6. Добавление возможности работы при помощи веб-сайта.
7. Добавление дополнительных языков.
8. Улучшение и добавление других способов сохранения результатов.
9. Улучшение теоретического материала.
10. Ввести дополнительные единицы измерения.

**Библиографический список**

1. <https://www.wolframalpha.com>
2. <https://planetcalc.ru/1508/>
3. <https://www.fxyz.ru/формулы_по_физике/механика/кинематика/падение_тел/движение_тела_брошенного_горизонтально/уравнение_траектории_тела_брошенного_горизонтально/>
4. <https://ido.tsu.ru/schools/physmat/data/res/virtlab/text/m2_1.html>
5. <https://www.matematicus.ru/en/mechanics-and-physics/movement-of-a-body-thrown-at-an-angle-to-the-horizon>
6. <https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter>
7. <https://riptutorial.com/Download/tkinter-ru.pdf>
8. <https://wiki.fenix.help/fizika/sila-soprotivleniya>

**Приложение – примеру работы с интерфейсом**

