## Лабораторная 1. Простая анимация

### Цель работы

Основная цель решения задач из данного списка - освоить приемы работы с графической библиотекой, а именно: построение графических примитивов, создание анимации, управление главным циклом работы программы.

### Порядок выполнения

Задания в этой лабораторной работе предусматривают индивидуальное самостоятельное выполнение. Преподаватель распределяет задания между студентами случайным образом. На семинаре преподаватель может по запросу студентов пояснить задание в части требований к выполнению или стоящей за заданием математической концепцией.

Все задания подразумевают творческий подход к их выполнению. Задание описывает только общую канву программы. В процессе выполнения студент может столкнуться с рядом дизайнерских решений - выбор скорости анимации, размера элементов, цветового решения и так далее. Все это ложится на студента как на разработчика программы. Преподаватель вправе оценить удачные решения и снизить оценку за неудачные.

По умолчанию предусматривается использование библиотеки tkinter. По желанию студента и по согласованию с преподавателем можно выбрать другой подобный инструмент решения задачи (pygame, p5 или аналогичные графические библиотеки).

К каждой задаче предлагаются дополнительные задания (пронумерованы), которые расширяют или углубляют данное задание. Студент может самостоятельно предложить расширение функционала программы. Дополнительные задания оцениваются преподавателем с учетом их сложности.

Дополнительные задания предусматривают самостоятельное освоение студентом некоторых инструментов - обработки событий, работы с элементами пользовательского интерфейса, работу с мышью и клавиатурой. Следует помнить, что все эти навыки пригодятся для выполнения дальнейших заданий.

После самостоятельного выполнения задания преподаватель может попросить студента провести публичную демонстрацию выполненного задания, включая устные комментарии написанного хода и алгоритмических решений, примененных студентом при решении задачи.

### Задания для выполнения

|  |  |
| --- | --- |
| Звездное поле В произвольном месте на экране появляются звезды, которые движутся по направлению от центра экрана. Должно складываться ощущение движения наблюдателя сквозь звездное поле. Сделайте звезды разного размера и цветов.   1. Реализуйте движение в 3D, где размер звезды зависит от расстояния до нее. 2. Скорость движения регулируется с клавиатуры. 3. Мышкой можно регулировать направление движения. |  |
| Дождь На экране постоянно генерируются контрастные “капли”, равномерно движущиеся вниз.   1. Поэкспериментируйте с плотностью генерации капель. 2. Капли должны создавать ощущение трехмерности - дальние движутся медленнее и меньше в размерах. |  |
| Солнечная система В центре экрана находится неподвижное солнце. Вокруг него вращаются несколько планет по круговым орбитам с разной скоростью и на разном расстоянии.   1. Попробуйте добавить прозрачность. 2. У планет могут быть собственные спутники. 3. Добавьте пояс астероидов. |  |
| Аттрактор Лоренца Двумерная проекция движения точки в соответствии с [системой](https://ru.wikipedia.org/wiki/%D0%90%D1%82%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%9B%D0%BE%D1%80%D0%B5%D0%BD%D1%86%D0%B0) дифференциальных уравнений (для примера возьмите начальное положение точки (0;0;0,1), где центр координат - это центр экрана и параметры σ=10, r=28 и b=8/3).   1. Поэкспериментируйте с различными наборами значений параметров. 2. Сделайте их интерактивными при помощи слайдеров. |  |
| Простой фрактал Нарисуйте фрактальное дерево по следующему алгоритму: рисуется вертикальная линия снизу примерно до центра экрана; от ее концов рисуются две линии поменьше под определенным углом к ней (с двух сторон); от концов этих линий строятся еще по две, и так далее. Десяти итераций будет достаточно.   1. Поэкспериментируйте с углом и коэффициентом убывания длины. Должны получаться фигуры как на анимации справа. 2. Сделайте величину угла интерактивной при помощи слайдера |  |
| Суперэллипс Реализуйте отрисовку [суперэллипса](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BF%D0%B5%D1%80%D1%8D%D0%BB%D0%BB%D0%B8%D0%BF%D1%81) с заданной величиной n. Для начала сделайте полуоси суперэллипса равными. Попробуйте построить фигуру с разными значениями n. Должны получаться изображения как справа на анимации.   1. Сделайте величину n интерактивной при помощи слайдера. 2. Постройте обощенный суперэллипс по параметризациям из Википедии. |  |
| Множество Мандельброта Реализуйте отрисовку [множества Мандельброта](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE_%D0%9C%D0%B0%D0%BD%D0%B4%D0%B5%D0%BB%D1%8C%D0%B1%D1%80%D0%BE%D1%82%D0%B0). Возьмите область радиусом два вокруг начала координат. Для каждой точки из области отрисовки необходимо проверить ее поведение при итерации особой функции комплексного переменного. Преобразуйте эту функцию в действительные числа.   1. Реализуйте цветную визуализацию. 2. Реализуйте возможность зума. |  |
| Спираль Реализуйте отрисовку кругов по спирали так, что каждый из них отстоит от центра на определенное расстояние и на определенный угол, как в [филлотаксисе](https://ru.qwe.wiki/wiki/Three-gap_theorem). Определите приемлемую скорости анимации.   1. Сделайте симуляцию роста листов растения вокруг стебля. 2. Поэкспериентируйте с различной величиной иррационального угла. |  |
| Случайный путь Реализацией визуализацию случайного пути точки. Точка сначала находится в центре экрана. На каждой итерации цикла случайно выбирается направление движения и точка перемещается на определенный шаг вдоль этого направления. Затем интонации повторяются.   1. Запустите симуляцию с разной величиной шага. 2. Задайте точке приоритетное направление движения. 3. Сделайте интерактивную версию, в которой приоритетное направление и "сила притяжения" к нему регулируется слайдерами. |  |
| Полярная роза Реализуйте отрисовку полярной розы для различный значений параметров n и d.  Подумайте с различными значениями параметров.   1. Сделайте параметры интерактивными с помощью слайдеров. 2. Попробуйте так же реализовать другие параметрические кривые. |  |
| Часы Реализуйте визуализацию часов, как показано на анимации. Часы должны показывать текущее время компьютера и идти в реальном времени.   1. Сделайте разные варианты форм стрелок часов. 2. Реализуйте различные часовые усложнения (окно даты, фазы луны, интерактивный хронограф). |  |
| Рекурсия Реализуйте с помощью рекурсивного алгоритма отрисовку фрактала, изображённого справа.   1. Реализуйте остановку рекурсии по достижению определенного количества шагов или при достижении минимального размера элемента. 2. Попробуйте реализовать так же другие простые фракталы. |  |
| Система частиц С помощью генератора случайных чисел и прозрачности создайте анимацию наподобие изображенной справа. Каждая частица изображается кружком и имеет ограниченный срок жизни.   1. Добавьте прозрачность, зависящую от оставшегося времени жизни частицы. 2. Добавьте возможность добавлять такие же системы частиц мышкой в произвольном месте поля. |  |
| Аппроксимация пи При помощи метода Монте-Карло проведите симуляцию генерацию точек внутри квадрата и при подсчете, какая доля из них попадает внутрь круга посчитайте приблизительное значение числа пи.   1. С помощью такого же метода аппроксимируйте другие величины. |  |
| Двоичные числа Сделайте интерактивную программу, в которой пользователь щелчком мыши может устанавливать или снимать значения нескольких двоичных регистров. Программа должна считать значение получаемого двоичного числа.   1. Добавьте отображение внутри каждого регистра его цифрового значения. 2. Реализуйте так же другие системы счисления (троичную уравновешенную, систему со смешанным основанием). |  |
| Треугольник Серпинского Реализуйте отрисовку треугольника Серпинского. Он представляет собой треугольник, у которого вырезана центральная часть так, чтобы получилось три подобных треугольника. У каждого из них так же вырезана центральная часть и так далее. Установите максимальное количество итераций.   1. Реализуйте подробным методом другие простые фракталы (снежинка Коха, губка Менгера). |  |
| Визуализация таблицы умножения Реализуйте рисунок, визуализировать умножение на два точек на окружности. У вас должна получиться красивая фигура - кардиоида.  [Видео](https://www.youtube.com/watch?v=qhbuKbxJsk8) в поддержку.  Реализуйте уменьшение на разные числа, чтобы посмотреть, какие фигуры получается.   1. Отнимайте несколько окружностей с наиболее примечательными узорами. 2. Сделайте множитель интерактивной временной с помощью слайдера. 3. Сделайте анимацию, как на рисунке справа. |  |
| Сердце Используя полярные координаты нарисуйте аналогичную анимацию.  Начните с простой отрисовки сердца, анимацию добавьте следующим шагом. Используйте переменное увеличение угла отрисовки.   1. Изобразите другие фигуры в полярных координатах. 2. Отделите в коде функцию отрисовки и фигуру, чтобы можно было легко менять фигуру с минимальными изменением кода. |  |
| Ряд Лейбница При помощи [ряда Лейбница](https://ru.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%9B%D0%B5%D0%B9%D0%B1%D0%BD%D0%B8%D1%86%D0%B0) отрисуйте следующую визуализацию и вычислите приближение числа пи.   1. Таким же образом визуализируйте другие известные ряды. 2. Оцените скорость сходимости разных ряда к истинному л библиотечному) значению пи. |  |
| Визуализация сортировки Реализуйте визуализацию алгоритма сортировки пузырьком числового массива.   1. Добавьте выделение цветом сравниваемых на текущем шаге элементов массива. 2. Реализуйте таким же образом другие алгоритмы сортировки. |  |

### 