**Слайд 1**

Представляем игру, разработанную в рамках практики программировани с использованием Python. Ее основу составляет движение тела под действием силы тяжести по поверхности, состоящей из плоских участков с различным наклоном, длиной и коэффициентом трения.

**Слайд 2**

Цель игры: меняя положение, длину и коэффициент трения участков поверхности, поразить телом все цели, находящиеся в различных точках игрового поля, за минимальное время.

Основные условия, в которых действует игрок, представлены на слайде.

**Слайд 3**

Диаграмма классов, использованных при создании игры представлена на следующем слайде.

Игра создавалась на базе библиотеки tkinter, т.к. для создания удобного оконного интерфейса требовались имеющиеся в библиотеке виджеты, при этом сложная графика не планировалась. Основные интерфейсные объекты классов tkinter, использованные для игры представлены в верхней части диаграммы.

В нижней части диаграммы представлены основные классы, реализующие физику игры: поверхность и ее точки (Surface и Point), тело (body), цели (target) и глаза Пупы (Puppa\_eye). Все представленные классы агрегируются классом Game, который по сути и является реализацией игры.

**Слайд 4.**

Основные модули, использованные в программе представлены на следующем слайде. Программа состоит из 6 модулей, включая главный, на базе которых решаются различные задачи: описание основных классов тел, поверхности, алгоритмов моделирования движения тела по поверхности и его взаимодействия с целями, визуализации основных объектов, работы с файлами.

Также на слайде представлены использованные дополнительные библиотеки.

**Слайд 5.**

Распределение задач и соответствующих модулей в команде представлено на следующем слайде.

**Слайд 6.**

Далее каждый из членов команды немного приведет пример реализации одной из подзадач.

Наиболее сложным с точки зрения реализации был алгоритм моделирования движения тела по участкам поверхности, т.к. возникали проблемы с определением точек касания, пробиванием телом поверхности из-за неверно выбранного временного шага и т.д.

Основные этапы алгоритма и поясняющая схема представлены на слайде. После вычисления новых координат тела также производилась проверка необходимости его коррекции, т.к., например, при свободном падении тела на следующем временном шаге оно могло оказаться за верхней границей поверхности.

**Слайд 7.**

Основные принципы визуализации объектов игры на экране представлены на следующем слайде.

При инициализации физического объекта игры (тела, цели, участка поверхности, векторов и т.д.) создается также и соответствующий ему графический объект на экране.

Во время игрового процесса для перемещаемых или изменяемых объектов на каждом шаге производится перевод физических координат в соответствующие экранные и последующая перерисовка объекта. Канвас библиотеки tkinter позволяет перерисовывать объект без его затирания.

При деструкции физического объекта происходит удаление соответствующего графического объекта на экране.

**Слайд 8.**

Для хранения лучших результатов, графиков скоростей и уровней используются внешние текстовые файлы. На слайде представлены общие принципы загрузки уровней из таких файлов.

Каждый файл с уровнем имеет структуру, представленную на слайде. Указатель на то, что данный файл содержит уровень: хеш-puppagame, название уровня, список координат точек участков поверхностей, целей и начального положения тела.

При запуске игры производится поиск в корневой директории игры всех текстовых файлов, в которых есть #puppagame. Из них считываются названия уровней и записываются в выпадающее меню игры.

При выборе соответствующего уровня в меню игры загрузка точек траектории, координат целей, тела

**Слайд 9.**

На следующем слайде представлено главное окно игры с указанием основных элементов.

Основное игровое поле содержит тело (синий шар) с векторами, действующих на него сил, поверхность, по которой оно двигается (участки в зависимости от коэффициента трения имеют различную текстуру), цели, которые необходимо поразить, и Пупу, который пристально следит за телом.

Справа находится панель управления, которая позволяет при остановленном моделировании движения тела, менять его начальное положение, а также коэффициенты трения участков поверхностей. Выбор участка, а также перемещение его краев, производится мышью. Также справа содержатся элементы управления уровнями (выбор из выпадающего меню и загрузка), элементы управления игровым процессом (запуск, остановка, пауза, пошаговое движение, сброс к исходному состоянию уровня). И там же находятся кнопки загрузки траектории, на которой было получено лучшее время на текущем уровне, и вызова окна просмотра графиков скоростей.

В нижней части экрана находится информационная панель, на которую выводятся различные параметры: текущее модельное время, количество пораженных целей, координаты тела, пройденное расстояние, скорости и силы.

**Слайд 10.**

На следующем слайде представлено окно вывода графиков зависимости модуля скорости тела от модельного времени. Графики выводятся для последнего запуска тела и запуска с лучшим временем.

**Слайд 11.**

При поражении всех целей за минимальное время появляется окно для ввода имени игрока, чтобы сохранить соответствующие результаты.

**Слайд 12.**

Для игры было разработано 5 демонстрационных уровней с различными концепциями, которые представлены на слайде.

Игровой процесс далее будет продемонстрирован непосредственно в самой игре.

**Запуск игры и демонстрация:**

1. **Запуск**

2. По умолчанию открывается первый ознакомительный уровень. При нажатии на старт, запускается моделирование движения тела. **Нажать на старт**

3. Векторы сил меняются, тело двигается, Пупа за ним следит. Изменяются цифры в информационной панели.

4. **Нажать на стоп.**

5. Нажать на **Графики скоростей.** Можно посмотреть графики скоростей текущий и с лучшим временем.

6. **Закрыть графики скоростей**. **Подвигать участки поверхности и их коэффициенты трения**. Можно настроить участки поверхности. В зависимости от коэффициентов трения меняется текстура участка.

7. **Нажать на старт. Прокомментировать попытку (собрали цели или нет).**

**8. Загрузить лучшую траекторию. Запустить.**

9. В некоторых случаях лучшее время достигается при использовании отрицательных коэффициентов ускорения, т.к. тело при это набирает большую скорость. Но такая стратегия работает не всегда. Например, в уровне Баланс. **Загрузить уровень Баланс**.

10. Для того, чтобы поразить последнюю цель, необходимо иметь минимальную горизонтальную скорость, т.е. тело нужно замедлить. **Загрузить лучшую траекторию.** **Запустить.** Как вариант можно выстроить траекторию так, чтобы тело при движении слева направо на большой скорости собрало все цели, кроме последней, а на обратном пути – последнюю.

**Перейти к презентации**

**Слайд 13.**

Основные результаты, полученные в работе над проектом, представлены на слайде.

* Была спроектирована и реализована на Python несложная игра.
* Закреплены на практике навыки работы с оконными виджетами, файлами и т.д.
* Освоена самостоятельная работа над разработкой приложения в команде

**Слайд 14.**

На этом все. Спасибо за внимание.