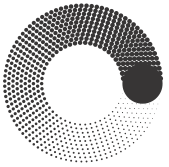
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.04.02 «Информационные системы и технологии»,**

**ЛАБОРАТОРНАЯ РАБОТА №5**

**Дисциплина:** Искусственный интеллект в мобильных системах

## **Тема:** Разработка интеллектуальных мобильных приложений с использованием пакета Kivy

**Выполнил(а): студент(ка) группы 224-371** Мусихин Ж.И

(Фамилия И.О.)

**Проверил(а):**

(Фамилия И.О. )

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

## **Задание**

Изучить внимательно[теорию по разработке интеллектуальных мобильных приложений](https://online.mospolytech.ru/mod/scorm/view.php?id=292256). Выполнить все пункты задания.

# **Ход работы**

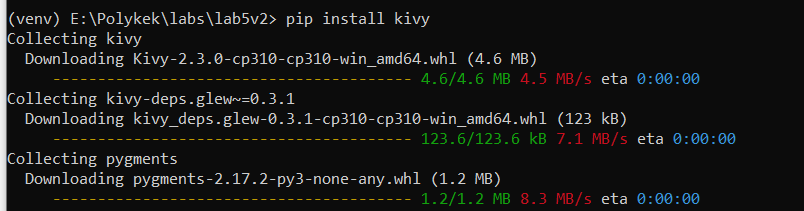
1. Создайте виртуальную среду для Kivy. Для этого задайте имя среды kivy\_venv в вашем текущем каталоге:  **python -m venv venv**



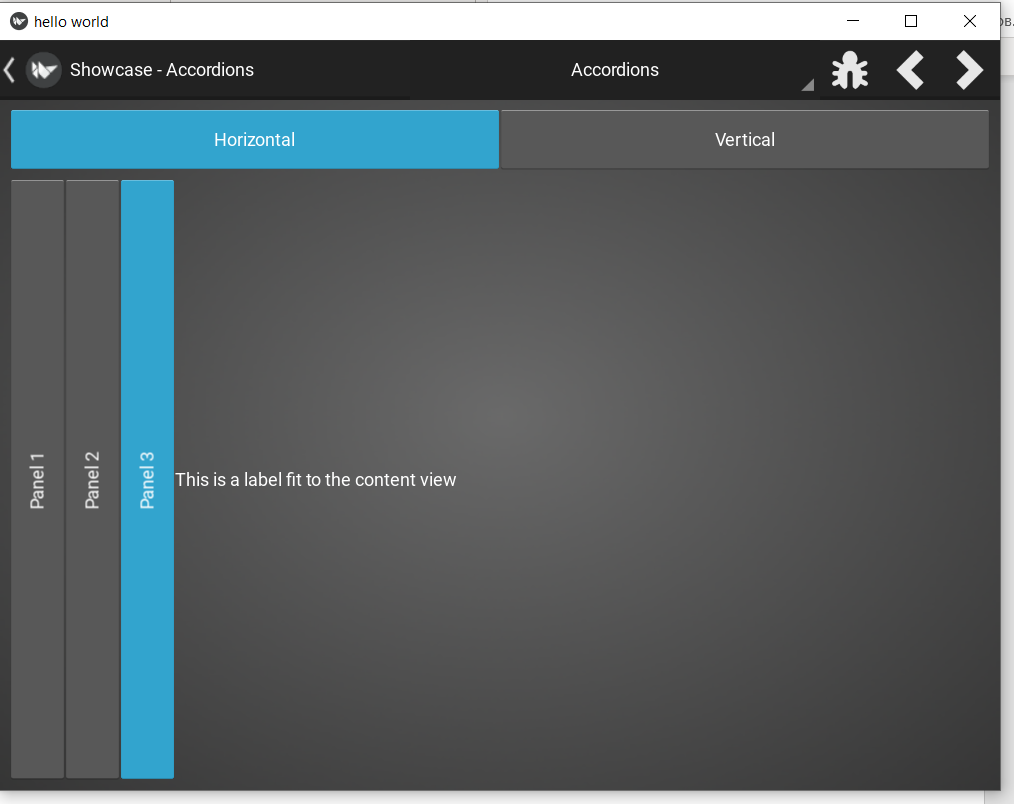
1. Активируйте виртуальную среду. Для этого из текущего каталога каждый раз, когда вы запускаете новый терминал в командной строке выполните: **kivy\_venv\Scripts\activat**



1. Установите Kivy в виртуальной среде **pip install kivy и pip install kivy\_examples. Были установлена библиотека kivy и примеры её использования.**



1. Запустите самый общий пример со всеми интерфейсными элементами, который находится по адресу: kivy\_venv\share\kivy-examples\demo\showcase\main.py (предполагается, что ваш текущий путь – это корневая папка проекта). Результат представлен на скриншоте ниже:



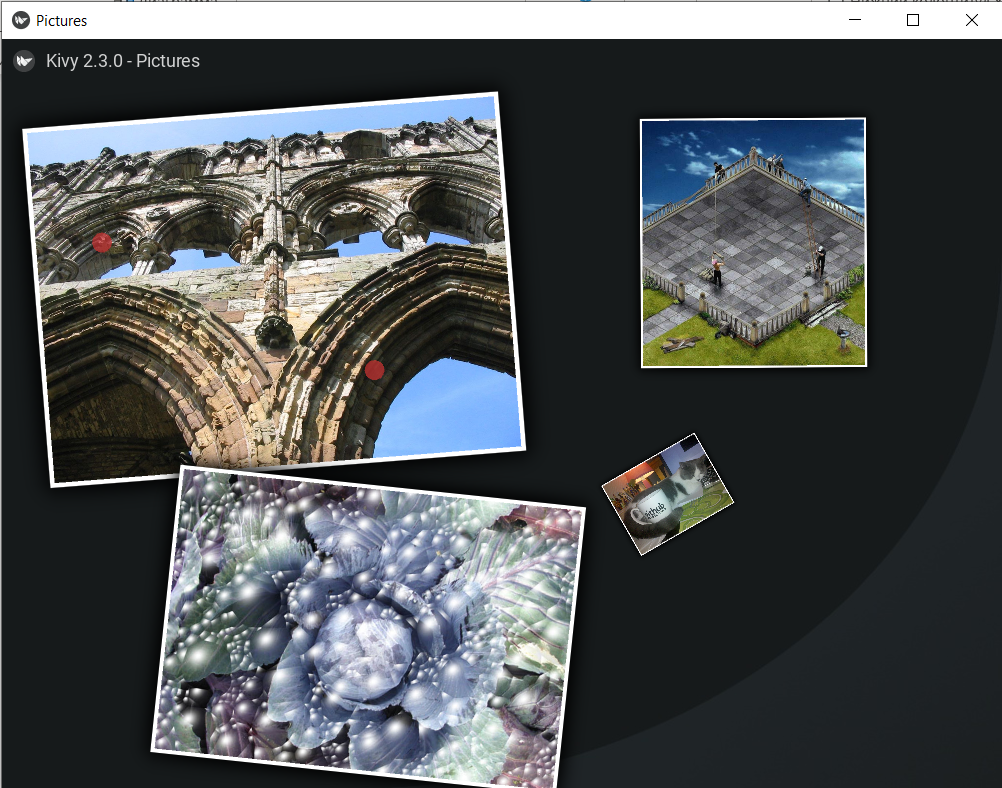
1. Составьте перечень изученных элементов: название, скриншот и описание для чего он нужен. Сведите результат в таблицу. В таблице должно быть не менее 20ти интерфейсных элементов.

|  |  |  |
| --- | --- | --- |
| Название | Скриншот | Назначение |
| Accordion |  | Форма меню, в которой параметры расположены вертикально или горизонтально, а элемент в фокусе (при касании) открывается для отображения его содержимого. |
| Bubble |  | Представляет собой форму меню или небольшого всплывающего окна со стрелкой, расположенной на одной стороне его содержимого. |
| Button |  | Кнопка — это метка со связанными действиями, которые запускаются при нажатии кнопки (или отпускании после щелчка/прикосновения). Для настройки кнопки используются те же свойства (padding, font\_size и т. д.) и система размеров, что и для класса Label |
| Carousel |  | представляет собой классический удобный для мобильных устройств вид карусели, в котором вы можете перемещаться между слайдами. Вы можете добавить в карусель любой контент и перемещать его по горизонтали или по вертикали. Карусель может отображать страницы последовательно или циклично. |
| CheckBox |  | это специальная кнопка с двумя состояниями, которую можно либо отметить, либо снять. Если флажок находится в группе, он становится переключателем |
| CodeInput |  | Поле для ввода и отображения текста (программного кода) |
| DropDown |  | Раскрывающийся список, который позволяет отображать другие элементы. |
| FileChooser |  | Отображает файлы системы в виде списка или иконок |
| Popup |  | Виджет Popup используется для создания модальных всплывающих окон. По умолчанию всплывающее окно охватывает все «родительское» окно. |
| ProgressBar |  | Отображает прогресс |
| RstDocument |  | удобный для чтения синтаксис разметки открытого текста и система синтаксического анализатора «что видишь, то и получаешь». |
| Scatter |  | используется для создания интерактивных виджетов, которые можно переводить, вращать и масштабировать двумя или более пальцами в мультитач-системе. |
| ScreenManager |  | предназначенный для управления несколькими экранами вашего приложения. |
| Slider |  | Он поддерживает горизонтальную и вертикальную ориентацию. Используется для установки различных значений |
| Spinner |  | виджет, который позволяет быстро выбрать одно значение из набора. В состоянии по умолчанию счетчик показывает выбранное в данный момент значение. Прикосновение к счетчику отображает раскрывающееся меню со всеми другими доступными значениями, из которых пользователь может выбрать новое. |
| Splitter |  | Используется для разделения контента на части. |
| Switch |  | Кнопка, которая сохраняет состояние при переключении |
| TabbedPanel |  | Используется для создания вкладок |
| TextInput |  | Поле для ввода текса с клавиатуры |
| ToggleButton |  | действует как флажок. Когда вы касаетесь или щелкаете ее, состояние переключается между «нормальным» и «нажатым» |

1. Запустите следующие примеры

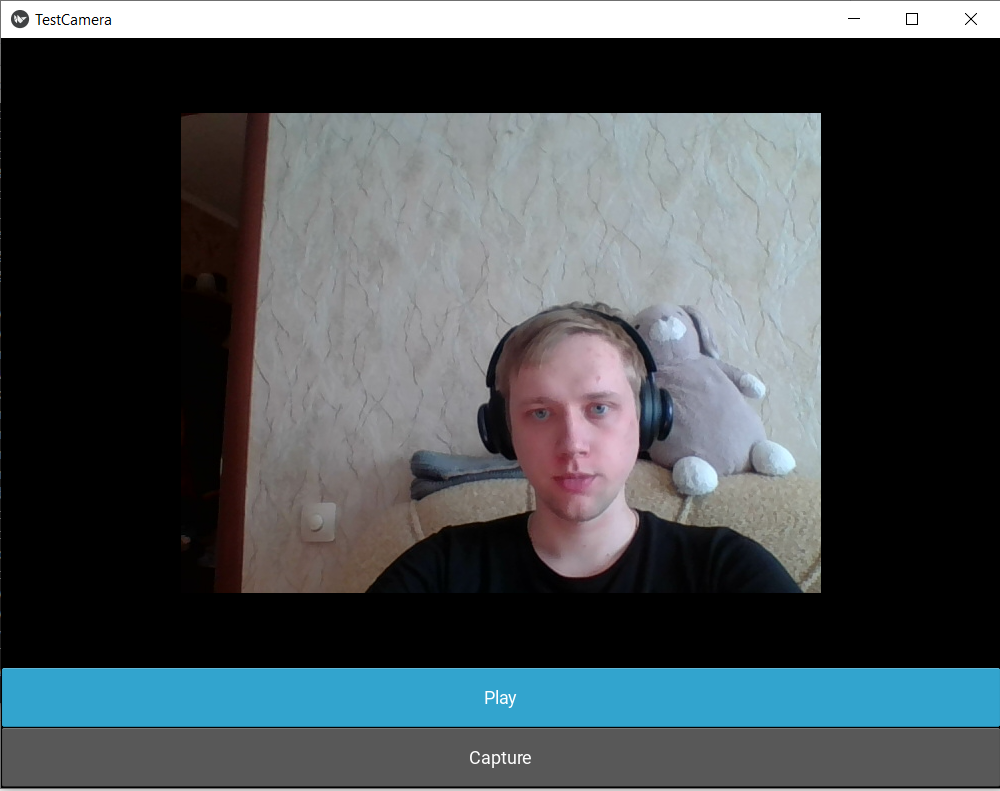
 PICTURES – вращение и изменение размера фото

В примере представлено 4 фото, которые можно вращать и изменять размер. Для применения нескольких касаний на пк нужно нажать клавишу ctrl



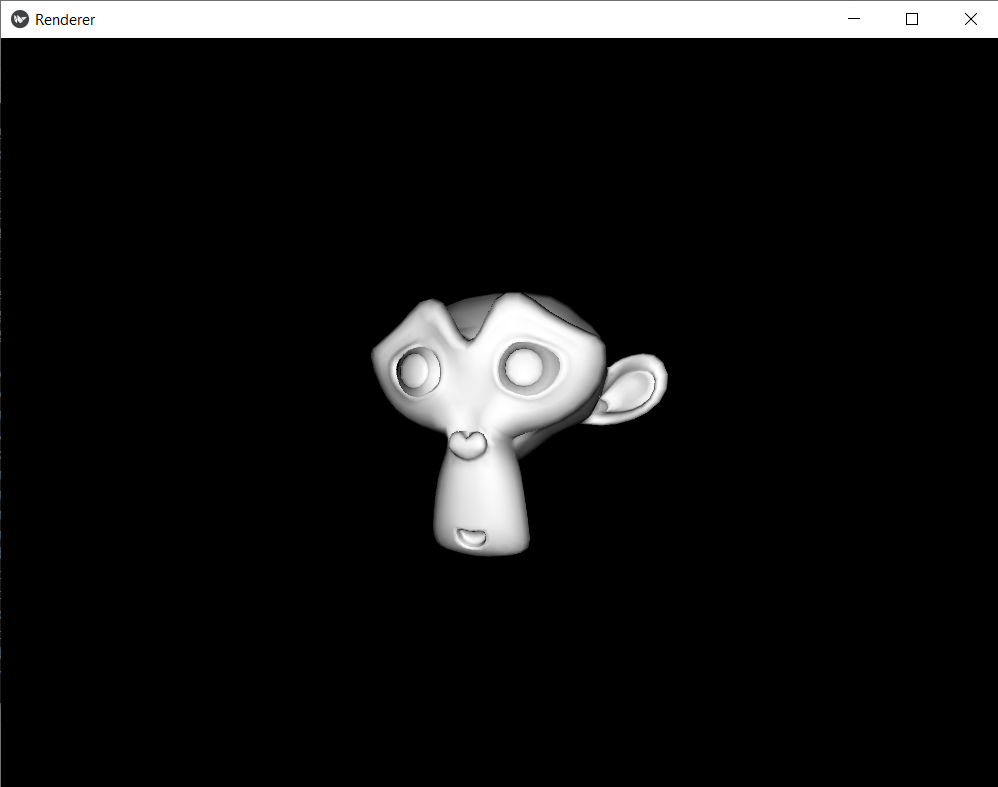
CAMERA -  работа с камерой

Пример, использующий дополнительную библиотеку OpenCV для отображения видео с веб-камеры.



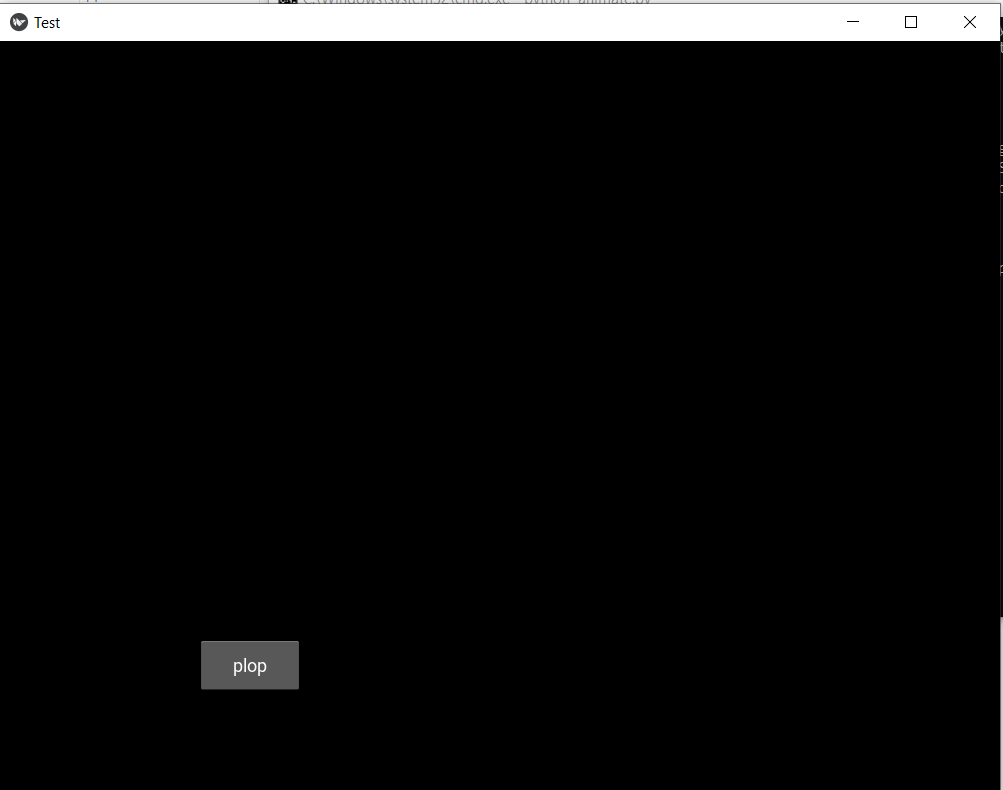
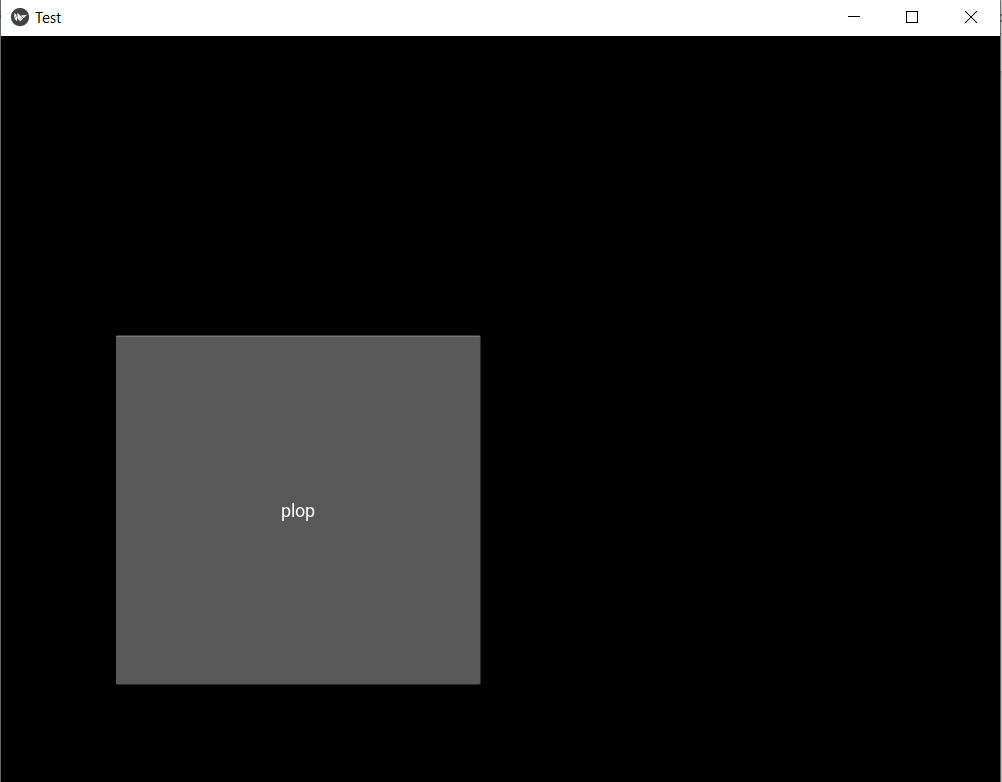
 3Drendering – рендеринг трехмерной модели

Пример отображающий рендер 3D модели



Animation – анимация объектов

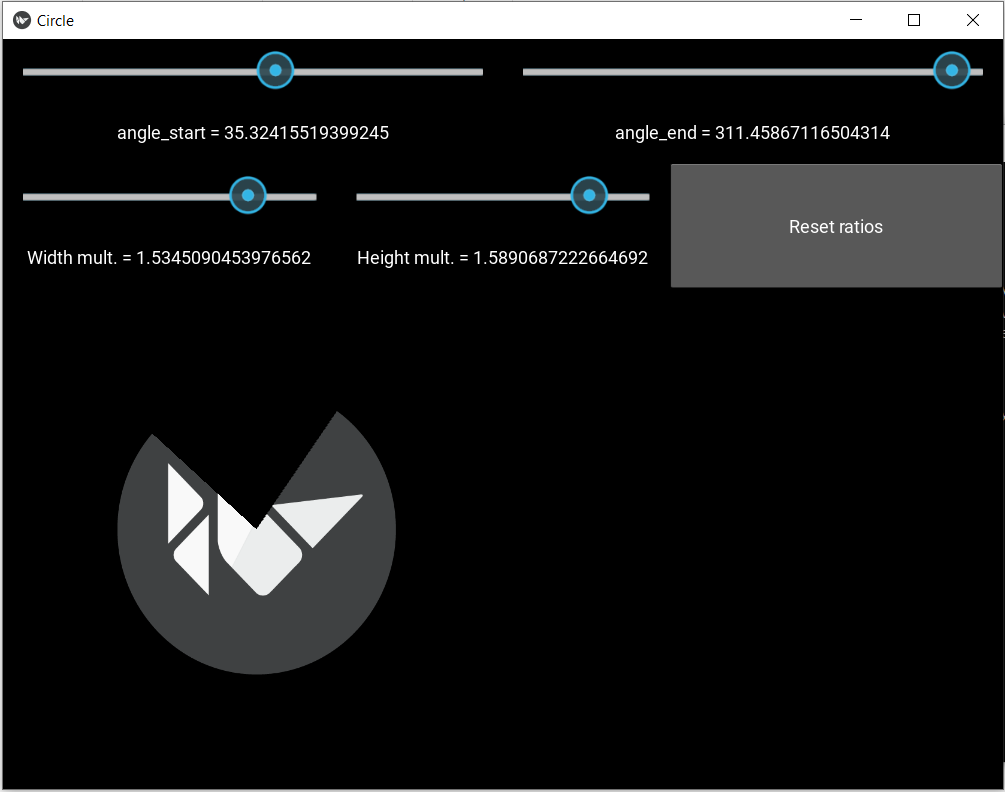
В примере происходит анимация изменения размера кнопки при нажатии.

Далее по заданию нужно было выбрать 3 дополнительных примера.

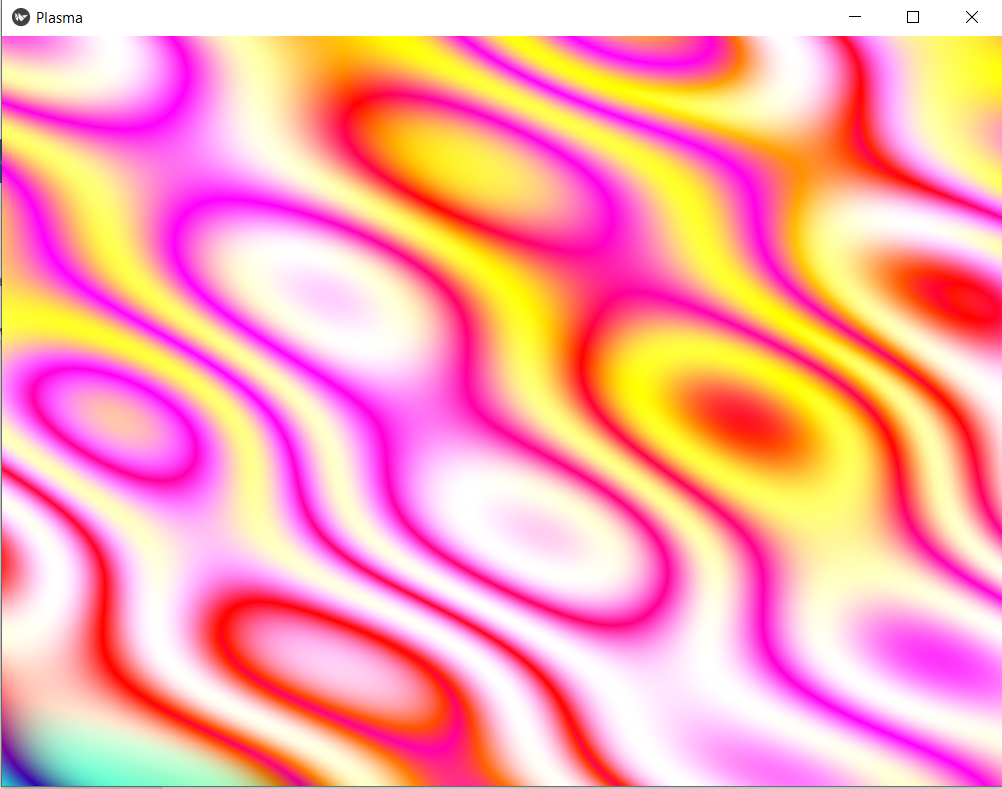
CANVAS(CIRCLE) – отображение фигур и их изменение.

В примере с помощью слайдеров можно изменять различные настройки отображения окружности(размер, разрез)

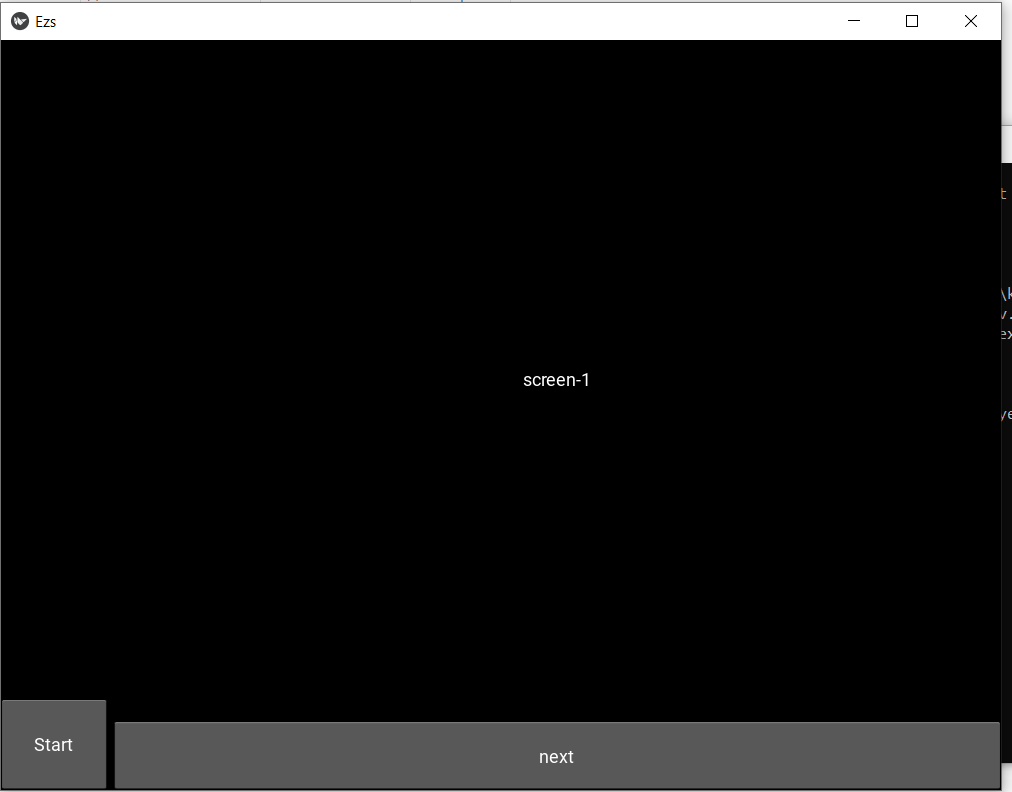
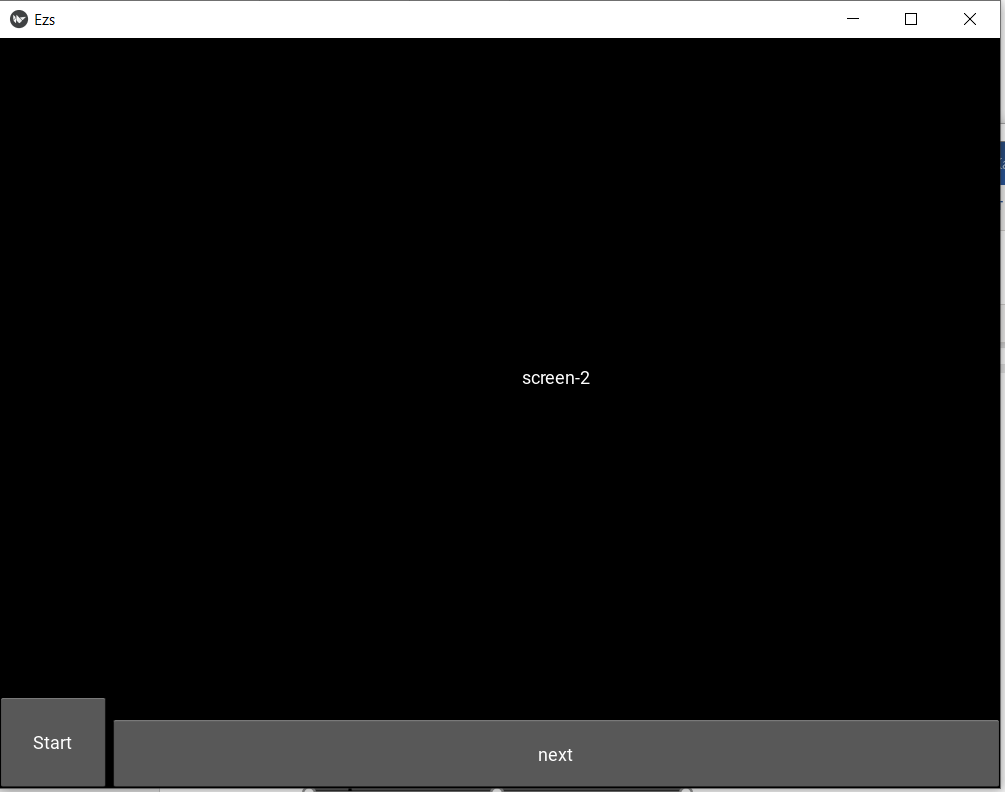


SHADER(PLASMA)

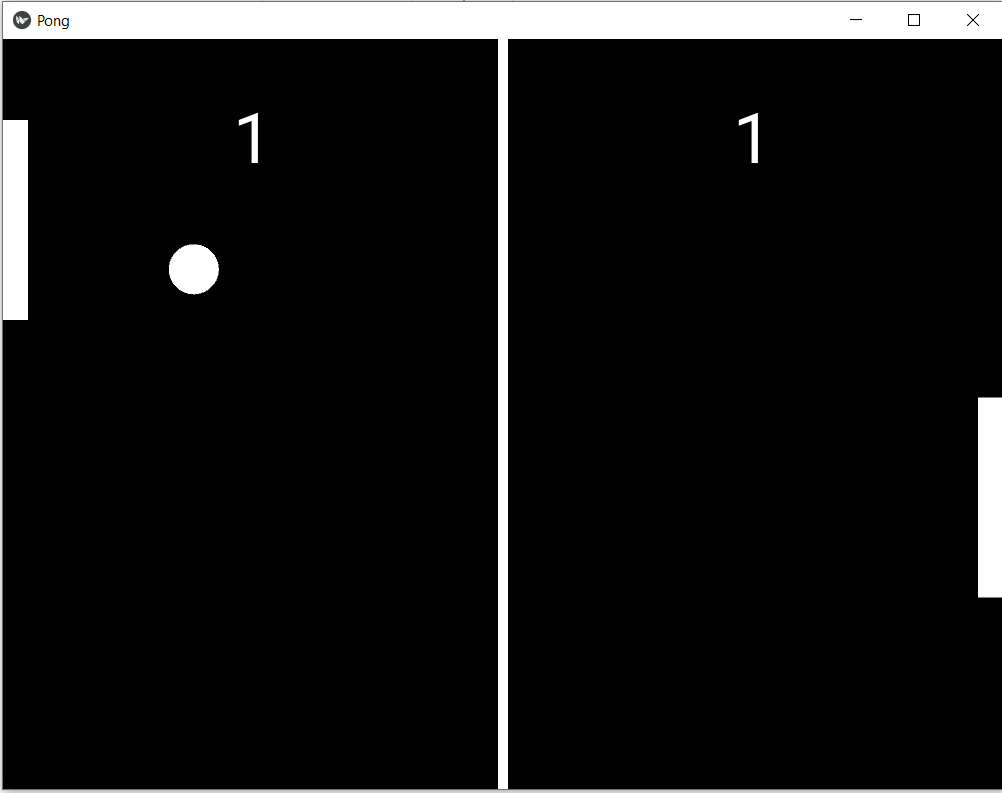
В примере демонстрирует работу шейдеров



CONTAINER – отображение контейнеров(экранов)

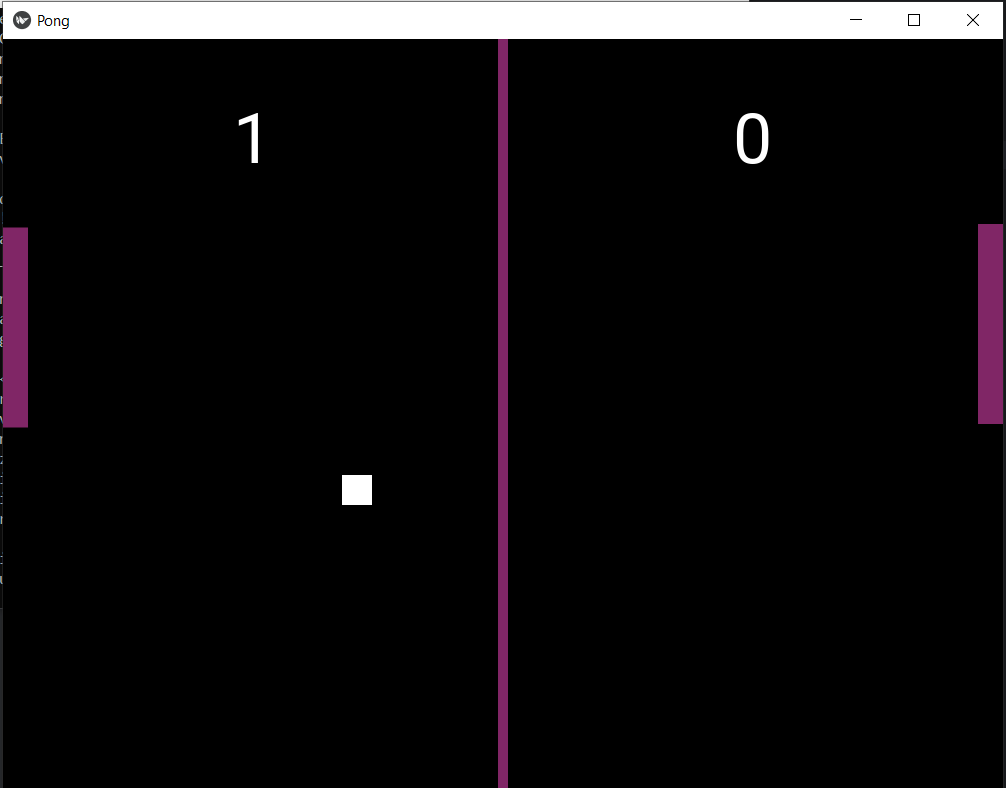
 

1. Запустите на выполнение игру PONG



1. Замените (усовершенствуйте) некоторые элементы игры. Включите в отчет, что было Вами сделано и скриншоты.

* Изменил цвет игроков и горизонтальной линии
* Заменил круг на квадрат(как в оригинальной игре)

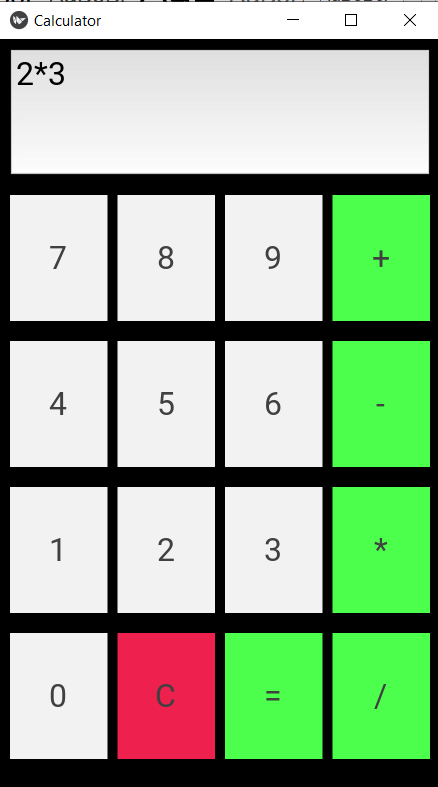


1. Используя язык дизайна KV, разработайте интерфейс и программу простого кнопочного калькулятора: 10 цифр, минимум 4 действия (+,-,\*,/), кнопка вычисления (=) и кнопка сброса (С). Включите в отчет код программы, код дизайна и скриншоты выполнения программы.

|  |
| --- |
| Main.py |
| import kivy  from kivy.app import App  kivy.require('1.9.0')  from kivy.uix.gridlayout import GridLayout  from kivy.config import Config  Config.set('graphics', 'resizable', 1)  class CalcGridLayout(GridLayout):      def calculate(self, calculation):          if calculation:              try:                  self.display.text = str(eval(calculation))              except Exception:                  self.display.text = "Error"    class CalculatorApp(App):      def build(self):          return CalcGridLayout()  calcApp = CalculatorApp()  calcApp.run() |

|  |
| --- |
| Calculator.kv |
| <CustButton@Button>:  font\_size: 32  background\_normal: ''  background\_color:[242/255.0, 242/255.0, 242/255.0, 1]  color: [64/255.0, 64/255.0 ,64/255.0 ,1]    <SignButton@Button>:  font\_size: 32  background\_normal: ''  color: [64/255.0, 64/255.0 ,64/255.0 ,1]  background\_color:[77/255.0, 255/255.0, 77/255.0,1]  <DelButton@Button>:  font\_size: 32  background\_normal: ''  color: [64/255.0, 64/255.0 ,64/255.0 ,1]  background\_color:[238/255.0, 32/255.0, 77/255.0,1]  <CalcGridLayout>:  id: calculator  display: entry  rows: 6  padding: 10  spacing: 20  BoxLayout:  TextInput:  id: entry  font\_size: 32  multiline: False  BoxLayout:  spacing: 10  CustButton:  text: "7"  on\_press: entry.text += self.text  CustButton:  text: "8"  on\_press: entry.text += self.text  CustButton:  text: "9"  on\_press: entry.text += self.text  SignButton:  text: "+"  on\_press: entry.text += self.text  BoxLayout:  spacing: 10  CustButton:  text: "4"  on\_press: entry.text += self.text  CustButton:  text: "5"  on\_press: entry.text += self.text  CustButton:  text: "6"  on\_press: entry.text += self.text  SignButton:  text: "-"  on\_press: entry.text += self.text  BoxLayout:  spacing: 10  CustButton:  text: "1"  on\_press: entry.text += self.text  CustButton:  text: "2"  on\_press: entry.text += self.text  CustButton:  text: "3"  on\_press: entry.text += self.text  SignButton:  text: "\*"  on\_press: entry.text += self.text  BoxLayout:  spacing: 10  CustButton:  text: "0"  on\_press: entry.text += self.text  DelButton:  text: "C"  on\_press: entry.text = ""  SignButton:  text: "="  on\_press: calculator.calculate(entry.text)  SignButton:  text: "/"  on\_press: entry.text += self.text |

Результат представлен на скриншоте ниже

# Вывод

Были изучены основы работы с kivy.