|  |  |
| --- | --- |
| **Предмет** | Технология разработки программного обеспечения |
| **Курс** | 4 |
| **Семестр** | 1 |
| **Работа** | 9 |
| **Дата** | 05.12.2020 |
| **Группа** | 474 |
| **Фамилия** | Алексеев |
| **Имя** | Игорь |
| **Отчество** | Алексеевич |

1. Воссоздайте интерфейс приложения.

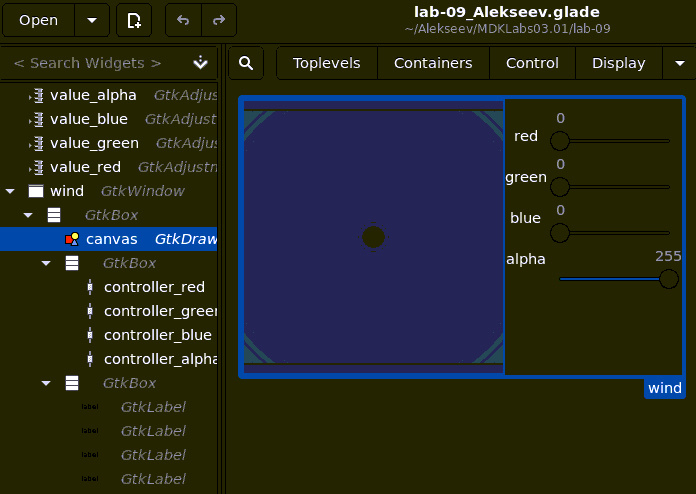


Рисунок 1 – Приложение в конструкторе

1. Для компонента GtkDrawingArea нужно разрешить получение следующих событий, представленных на рисунке 2, изменив свойство Events на вкладке Common.

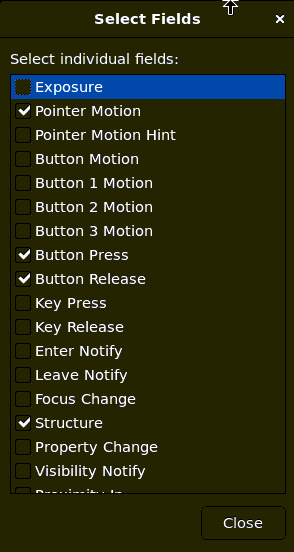


Рисунок 2 – Получение следующих событий

1. Создайте элементы GtkAdjustment для каждого GtkScale и настройте отображение текущего значения, как показано на рисунке. Для всех GtkAdjustment value\_red, value\_green, value\_blue задайте параметры Maximum Value равное 255, Step Increment – 1 и Page Increment – 0.1, так же для value\_alpha задайте Value значение Maximum Value.

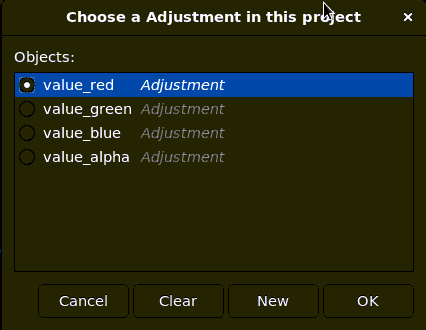


Рисунок 3 – Приложение в конструкторе

1. CSS файл:



Рисунок 4 – CSS файл

1. Код программы:

*local* *lgi* *=* require 'lgi'

*local* *gtk* *=* *lgi*.*Gtk*

*local* *gdk* *=* *lgi*.*Gdk*

*gtk*.*init*()

*local* *bld* *=* *gtk*.*Builder*()

bld:*add\_from\_file*('lab-09\_Alekseev.glade')

*local* *prov* *=* *gtk*.*CssProvider*()

prov:*load\_from\_path*('styles.css')

*context* *=* *gtk*.*StyleContext*()

*screen* *=* *gdk*.*Screen*.*get\_default*()

*context*.*add\_provider\_for\_screen*(*screen*, *prov*,

*gtk*.*STYLE\_PROVIDER\_PRIORITY\_APPLICATION*)

*local* *ui* *=* *bld*.*objects*

*local* *x* *=* 0

*local* *y* *=* 0

*local* *btn\_check* *=* false

*local* *r* *=* 0

*local* *g* *=* 0

*local* *b* *=* 0

*local* *a* *=* 255

*function* *btnl\_check*(ev, str, b)*-- mouse down*

*if* *ev*.*button* *==* 1 *then*

        print(*str*)

*btn\_check* *=* *b*

*ui*.canvas:*queue\_draw*()

*end*

*end*

*function* *ui*.canvas:*on\_button\_press\_event*(*ev*)*-- mouse down*

*btnl\_check*(*ev*, 'press', true)

*end*

*function* *ui*.canvas:*on\_button\_release\_event*(*ev*)*-- mouse up*

*btnl\_check*(*ev*, 'release', false)

*end*

*function* *ui*.canvas:*on\_motion\_notify\_event*(*ev*)

    print(*ev*.*x*, *ev*.*y*)

*x* *=* *ev*.*x*

*y* *=* *ev*.*y*

*ui*.canvas:*queue\_draw*()

*end*

*function* *ui*.controller\_red:*on\_value\_changed*()

*r* *=* *ui*.controller\_red:*get\_value*()

*ui*.canvas:*queue\_draw*()

*end*

*function* *ui*.controller\_green:*on\_value\_changed*()

*g* *=* *ui*.controller\_green:*get\_value*()

*ui*.canvas:*queue\_draw*()

*end*

*function* *ui*.controller\_blue:*on\_value\_changed*()

*b* *=* *ui*.controller\_blue:*get\_value*()

*ui*.canvas:*queue\_draw*()

*end*

*function* *ui*.controller\_alpha:*on\_value\_changed*()

*a* *=* *ui*.controller\_alpha:*get\_value*()

*ui*.canvas:*queue\_draw*()

*end*

*function* *ui*.canvas:*on\_draw*(*cr*)

    cr:*set\_source\_rgb*(1, 1, 1, 1)

    cr:*paint*()

    cr:*set\_source\_rgba*(*r* */* 255, *g* */* 255, *b* */* 255, *a* */* 255)

*if* *not*(*btn\_check*) *then*

        cr:*rectangle*(*x* *-* 5, *y* *-* 5, 10, 10)

*else*

        cr:*rectangle*(*x* *-* 10, *y* *-* 10, 20, 20)

*end*

    cr:*fill*()

*end*

*ui*.*wind*.*title* *=* 'lab-09\_Alekseev\_Igor'

*ui*.*wind*.*on\_destroy* *=* *gtk*.*main\_quit*

*ui*.wind:*show\_all*()

*gtk*.*main*()

1. Программа позволяет перемещать квадрат с помощь передвижения курсора мыши по GtkDrawingArea, также можно изменять его цвет и прозрачность с помощью GtkScale, а по нажатию левой кнопки квадрат увеличивается и снова уменьшается, когда она отжимается. При перемещении курсора по GtkDrawingArea в консоль выводятся координаты квадрата, нажатие и отжатие левой кнопки так же выводятся в консоль. Результат:



Рисунок 5 – Запущенное приложение



Рисунок 6 – Перемещение и изменение характеристик квадрата



Рисунок 7 – Нажатие левой кнопки мыши

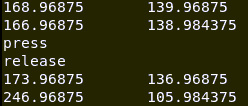
**

Рисунок 8 – Выводимые в консоль данные

1. Файлы программы:

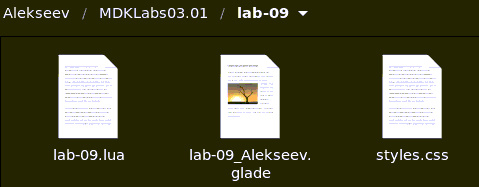


Рисунок 9 – Файлы программы

Ссылка на Github: <https://github.com/Ovroogs/MDKLabs03.01/tree/main/lab-09>