Содержание

1. Постановка задачи

Практически любой начинающий программист на Python патологически старается разработать программу для перемещения графического элемента (квадрата) по экрану с помощью клавиатуры. Для начала введем в нашу задачу несколько условностей — мы разрабатываем прграмму для перемещения графического элемента с помощью клавиатуры для локальной сети. Разработать программу для перемещения графического элемета с помощью клавиатуры в среде Python с использованием библиотеки GTK.

2. Разработка алгоритма решения задачи.

2.1 Описание Python.

Python (англ.*Python* – питон, произносится ['paɪθ⟨ə⟩n] – *па́йтон*; в русском языке распространено название *пито́н* [1]) – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Питоне организовывается в функции и классы, которые могут объединяться в модули (которые в свою очередь могут быть объединены в пакеты).

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ [2]. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные [3]. Есть реализации интерпретаторов для JVM (с возможностью компиляции), MSIL (с возможностью компиляции), LLVM и

других. Проект PyPy предлагает реализацию Питона на самом Питоне, что уменьшает затраты на изменения языка и постановку экспериментов над новыми возможностями.

Python – активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие офисы.

2.2 Описание GTK

GTK+ (сокращение от GIMP ToolKit) — кроссплатформенная библиотека элементов интерфейса, имеет простой в использовании API, наряду с Qt является одной из двух наиболее популярных на сегодняшний день библиотек для X Window System.

Будучи изначально частью графического редактора GIMP, она развилась в отдельный проект и приобрела заметную популярность. GTK+ — свободное ПО, распространяемое на условиях GNU LGPL, позволяющей создавать как свободное, так и проприетарное программное обеспечение с использованием библиотеки. GTK+ является официальной библиотекой для создания графического интерфейса проекта GNUиальные стандарты, их роль выполняет Cpython.

Одно из наиболее распространенных недоумений состоит в том, что GTK# требует Mono для работы. Это неверно. GTK# будет запускаться на любой .NET-совместимой среде. GTK# регулярно тестируется в MS .NET и Mono фреймворках, но также может быть запущена в любой полностью совместной среде. Это означает, что если Вы пишите приложение на GTK# и хотите запустить его в Windows, Вы можете развернуть проект только с GTK# с использованием среды MS, или развернуть его в среде Mono для Windows.

3.Листинг программы решения задания.

```
    # -*- coding: utf-8 -*-
    import pygtk, gtk, operator,time,string
    pygtk.require('2.0')

    class DrawingAreaExample:
        def __init__(self):
                event_box = gtk.EventBox()
                window = gtk.Window(gtk.WINDOW_TOPLEVEL)
                window.set_title("Пример Drawing Area")
                window.connect("destroy",     lambda     w:
gtk.main_quit())
                window.set_default_size(800,600)
                self.X = 100
                self.Y = 110
                self.area = gtk.DrawingArea()
                self.area.set_size_request(400, 300)
                self.pangolayout                                =
self.area.create_pango_layout("")
                self.sw = gtk.ScrolledWindow()
                self.sw.add_with_viewport(self.area)
                self.table = gtk.Table(2,2)
                self.hruler = gtk.HRuler()
                self.vruler = gtk.VRuler()
                self.hruler.set_range(0, 400, 0, 400)
                self.vruler.set_range(0, 300, 0, 300)
                self.table.attach(self.hruler,  1,  2,  0,  1,
yoptions=0)
                self.table.attach(self.vruler,  0,  1,  1,  2,
xoptions=0)
                self.table.attach(self.sw, 1, 2, 1, 2)
                window.add(self.table)

        self.area.set_events(gtk.gdk.POINTER_MOTION_MASK |

        gtk.gdk.POINTER_MOTION_HINT_MASK )
                #self.area.connect("expose-event",
self.area_expose_cb)
                window.connect('key_press_event',
self.area_expose_cb)
                event_box.add(self.area)
                def motion_notify(ruler, event):
                        return
ruler.emit("motion_notify_event", event)

        self.area.connect_object("motion_notify_event",
```

```
motion_notify,

       self.hruler)

       self.area.connect_object("motion_notify_event",
motion_notify,

       self.vruler)
              self.hadj = self.sw.get_hadjustment()
              self.vadj = self.sw.get_vadjustment()
              def val_cb(adj, ruler, horiz):
                     if horiz:
                            span                              =
self.sw.get_allocation()[3]
                     else:
                            span                              =
self.sw.get_allocation()[2]
                     l,u,p,m = ruler.get_range()
                     v = adj.value
                     ruler.set_range(v, v+span, p, m)
                     while gtk.events_pending():
                            gtk.main_iteration()
              self.hadj.connect('value-changed',   val_cb,
self.hruler, True)
              self.vadj.connect('value-changed',   val_cb,
self.vruler, False)
              def size_allocate_cb(wid, allocation):
                     x, y, w, h = allocation
                     l,u,p,m = self.hruler.get_range()
                     m = max(m, w)
                     self.hruler.set_range(l, l+w, p, m)
                     l,u,p,m = self.vruler.get_range()
                     m = max(m, h)
                     self.vruler.set_range(l, l+h, p, m)
              self.sw.connect('size-allocate',
size_allocate_cb)
              self.area.show()
              self.hruler.show()
              self.vruler.show()
              self.sw.show()
              self.table.show()
              window.show()

       def area_expose_cb(self, area, event):
              self.style = self.area.get_style()
```

```
            self.gc                                          =
self.style.fg_gc[gtk.STATE_NORMAL]
            self.draw_rectangles()
            key = gtk.gdk.keyval_name(event.keyval)
            if key == "Left":
                  self.X = self.X - 1
                  return True
            elif key == "Right":
                  self.X = self.X + 1
                  return True
            elif key == "Up":
                  self.Y = self.Y - 1
                  return True
            elif key == "Down":
                  self.Y = self.Y + 1
                  return True
            return True

      def draw_rectangles(self):
            self.area.window.draw_rectangle(self.gc,
True, self.X + 20, self.Y + 50, 40, 40)
            self.area.window.draw_layout(self.gc,
self.X + 5, self.Y + 80, self.pangolayout)
            return


   def main():
      gtk.main()
      return 0

   if __name__ == "__main__":
      DrawingAreaExample()
      main()
```

4. Результаты выполнения работы.

В результате выполнения курсовой работы был осуществлена разработка программы для перемещения графического элемента (квадрата) по экрану с помощью клавиатуры в среде интерпретации Python. Проверка программы

показала адекватность его работы.

5. Список используемой литературы.

- http://inform-uchebnik.com
- http://leonov-kuchnirenko.ru
- http://www.python.org/2.5/license.html
- http://inform-programm.com