



GTK+ Workshop

Yi-Soo An



안이수 (Yi-Soo An)

안

- Contributor at GNOME
 - Open source software
 - Containerization
-
- Github -- <https://github.com/memnoth>
 - GNOME Gitlab -- <https://gitlab.gnome.org/memnoth>
 - Blog -- <https://yisooan.wordpress.com/>



워크샵 목표

- GTK+ 기본 개념
- 기본 위젯
- 미니 메모장 만들기

Set up



Install Flatpak by script

1. <https://github.com/memnoth/gtkworkshop-script>



Install Flatpak (Ubuntu)

1. `$ sudo add-apt-repository ppa:alexlarsson/flatpak`
2. `$ sudo apt update`
3. `$ sudo apt install flatpak gnome-software-plugin-flatpak`
4. `$ flatpak remote-add --user flathub https://flathub.org/repo/flathub.flatpakrepo`
5. `$ flatpak remote-modify --user --collection-id=org.flathub.Stable flathub`
6. `$ flatpak update`
7. `$ sudo reboot`



Install GNOME Sdk 3.30

1. `$ flatpak install --user flathub org.gnome.Sdk//3.30`



Install GNOME-Builder (IDE)

1. `$ flatpak install --user flathub org.gnome.Builder//stable`

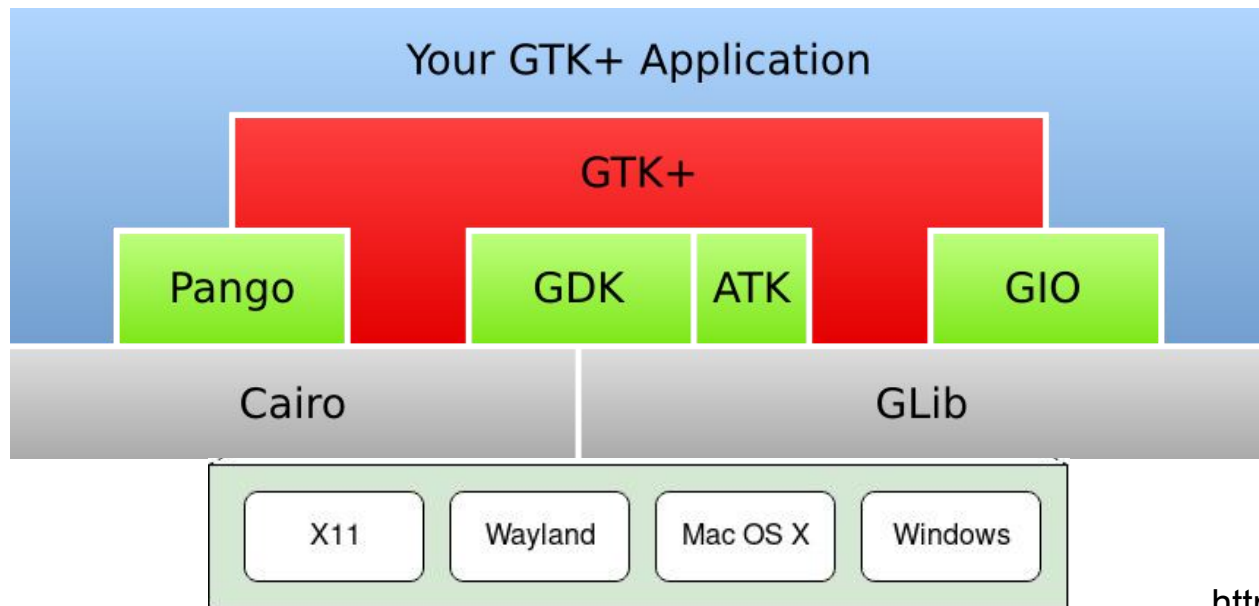


Clone gtkworkshop repo

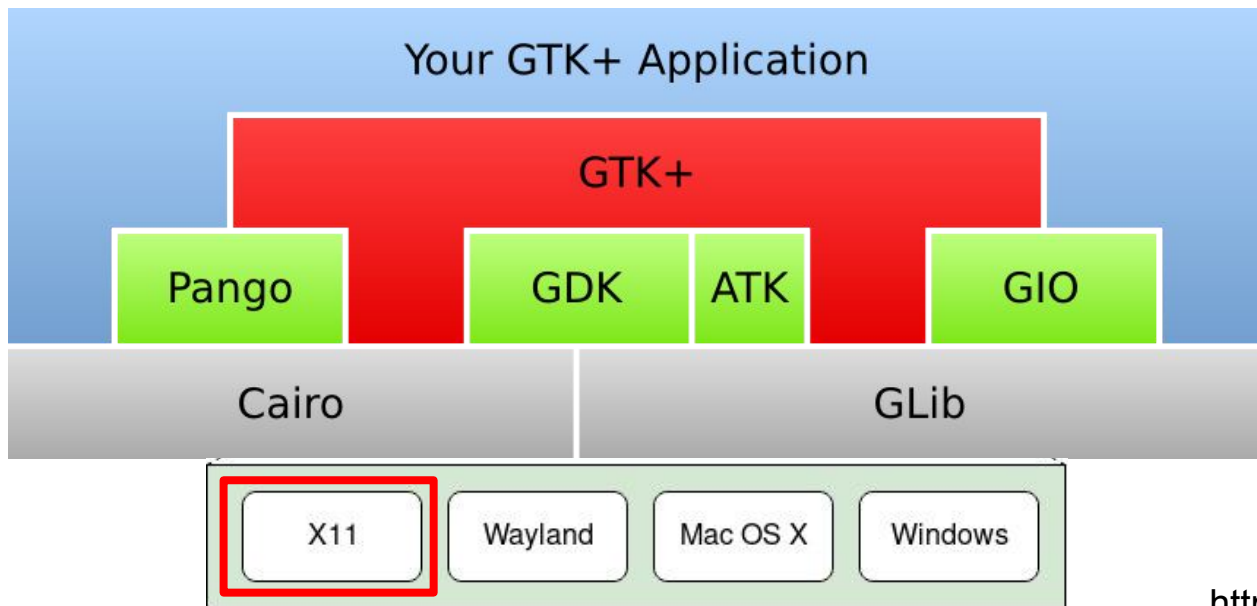
1. <https://github.com/memnoth/gtkworkshop.git>

What and Why GTK+?

GTK+ 어플리케이션 아키텍처



GTK+ 어플리케이션 아키텍처

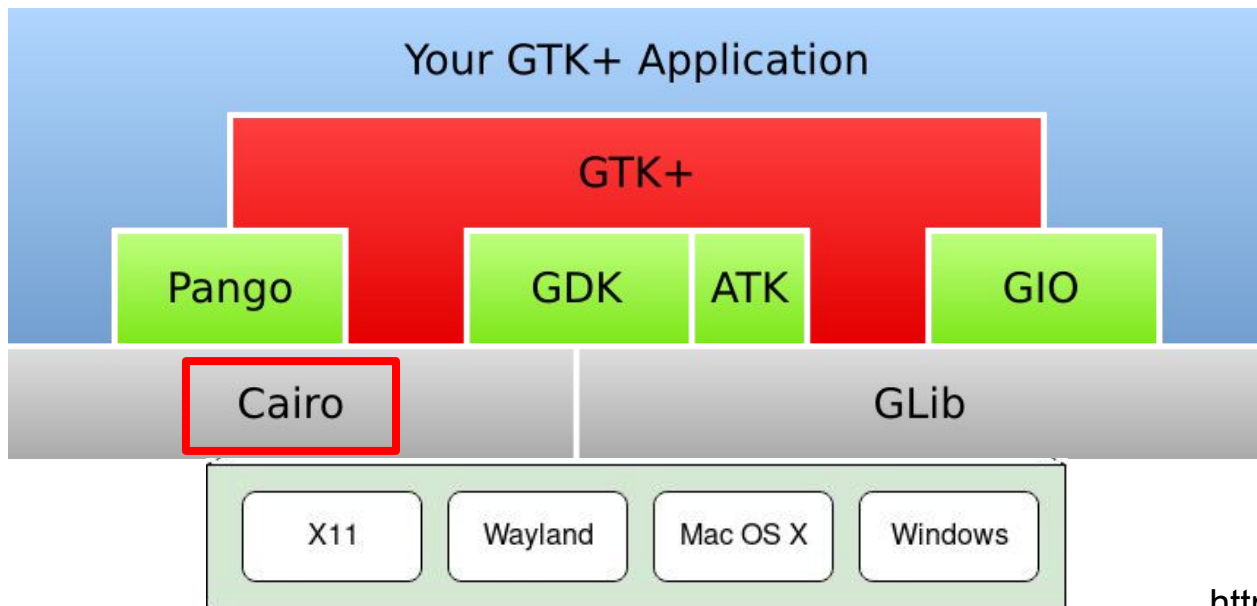




X Window System (X for short)

- 1984년 MIT의 Jim Gettys, Bob Scheifler에 의해 X Window System 개발
- 1987년 X Version 11 (X11) 릴리즈
- **Linux, Unix-like** 시스템의 Windowing system
- Window라고 불리는 영역을 화면에 생성하는 역할
- Window를 위한 Mouse, Keyboard 이벤트 핸들링

GTK+ 어플리케이션 아키텍처

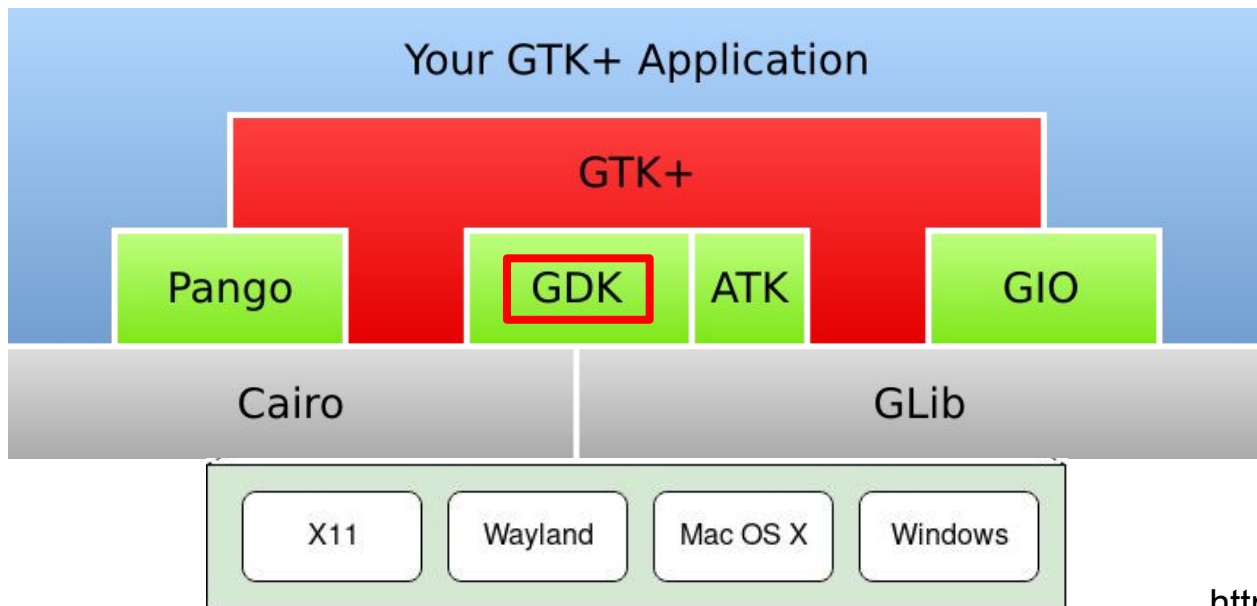




Cairo

- 2D Graphics drawing 라이브러리
- GTK+의 2D Graphics 백엔드로 사용
- LGPL 또는 Mozilla Public 라이선스
- 멀티 플랫폼 라이브러리 (Linux and Unix-like 시스템, Windows, Mac OS X)

GTK+ 어플리케이션 아키텍처



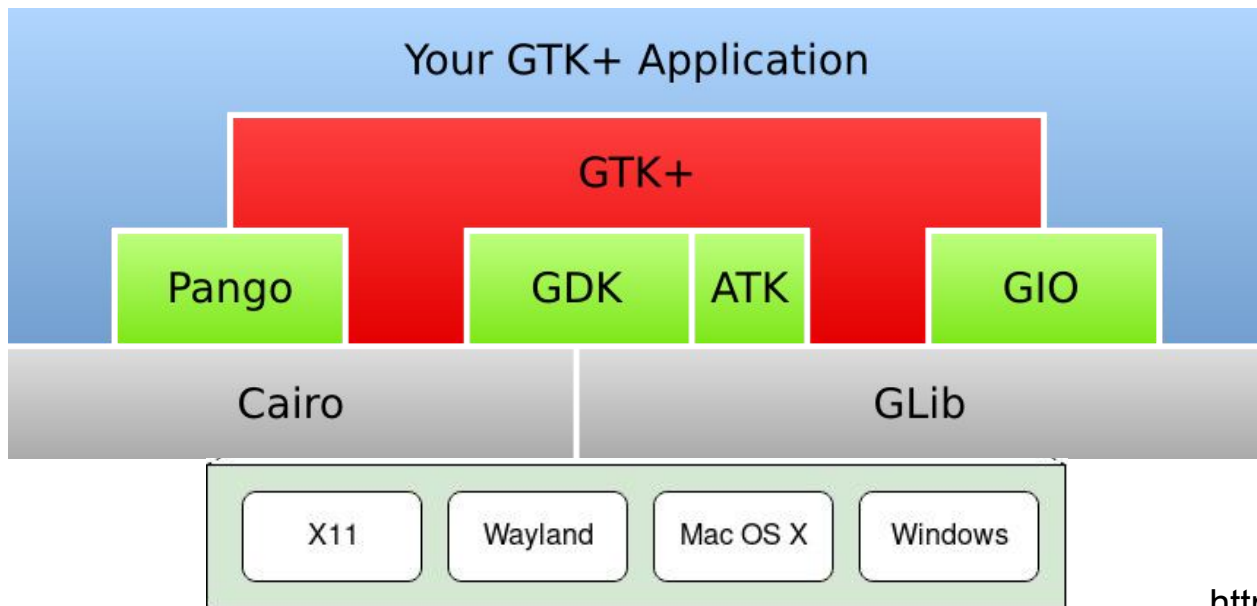


GDK

- GIMP Drawing Kit의 약자
- GTK+를 특정 Windowing system에서 분리 (Platform Abstraction Layer)
- Window를 위한 Mouse, Keyboard 이벤트 핸들링
- LGPL 라이선스
- 멀티 플랫폼 라이브러리 (Linux and Unix-like 시스템, Windows, Mac OS X)

See more <https://stackoverflow.com/a/28454674>

GTK+ 어플리케이션 아키텍처

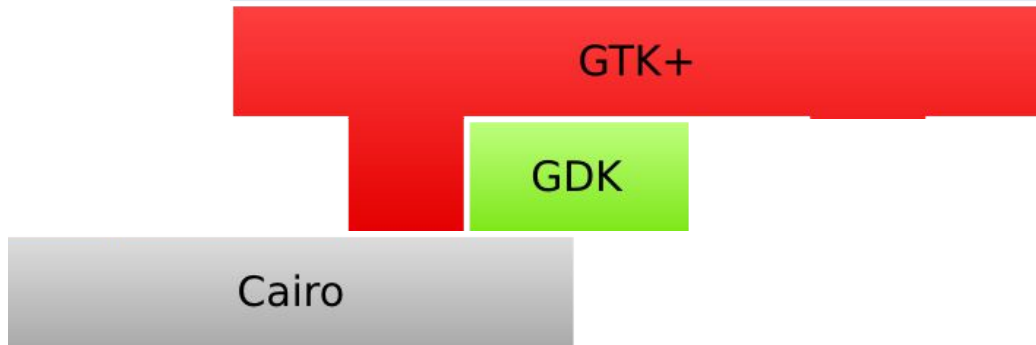


GTK+ 어플리케이션 아키텍처





GTK+ 어플리케이션 아키텍처

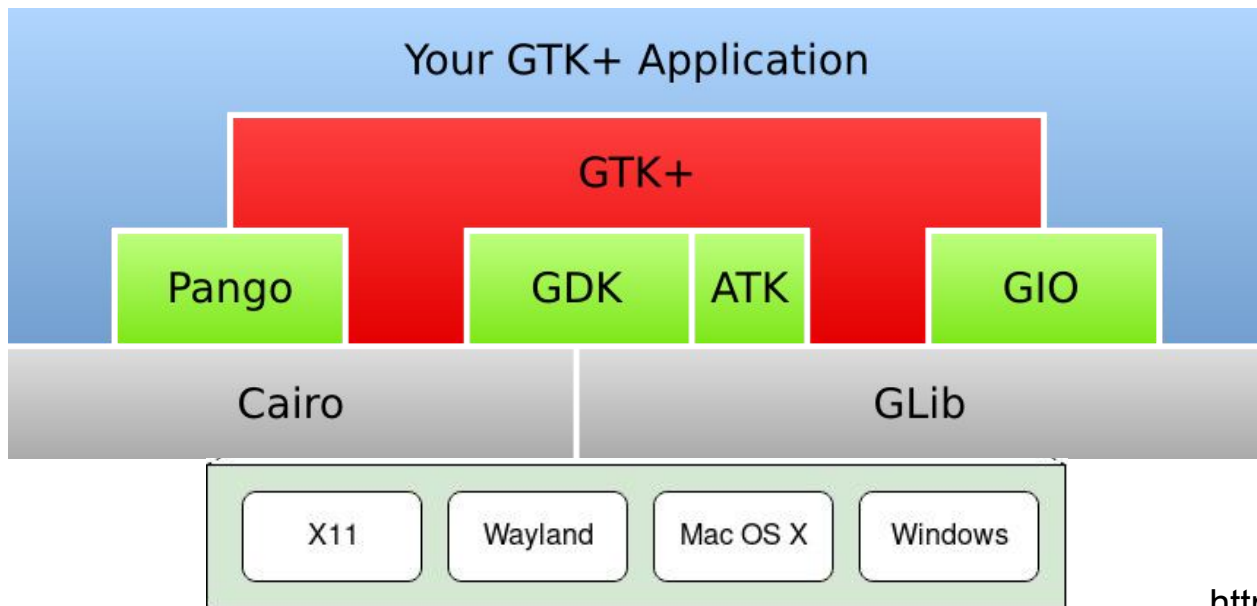




GTK+

- GIMP Tool Kit의 약자
- GIMP는 GNU Image Manipulation Program의 약자
- GIMP 개발에 사용된 GUI 라이브러리
- 1997년 Peter Mattis, Spencer Kimball, Josh MacDonald에 의해 개발
- LGPL 라이선스
- 멀티 플랫폼 라이브러리 (Linux and Unix-like 시스템, Windows, Mac OS X)

GTK+ 어플리케이션 아키텍처

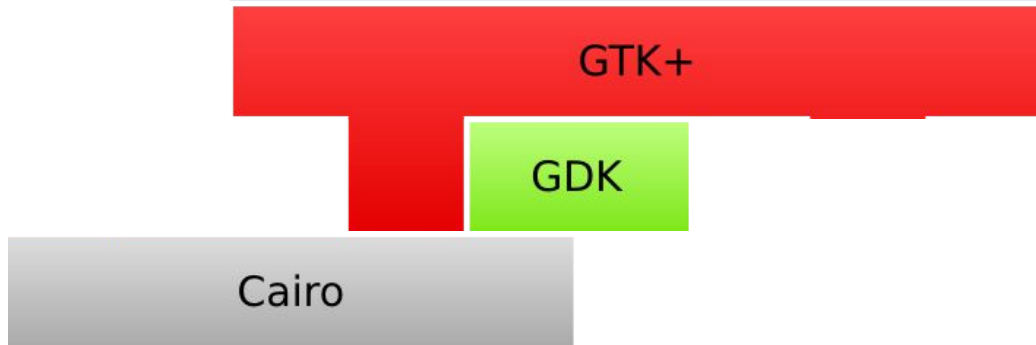


GTK+ 어플리케이션 아키텍처





GTK+ 어플리케이션 아키텍처





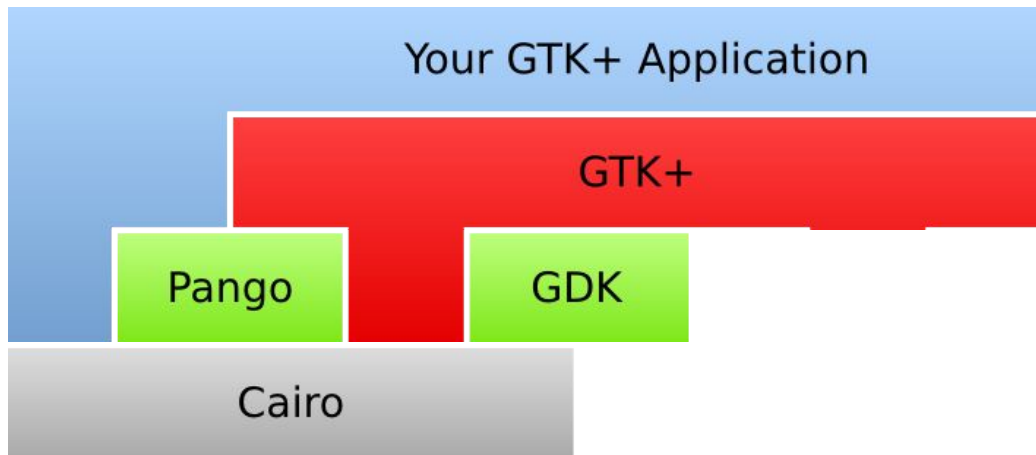
GTK+ 어플리케이션 아키텍처



Your GTK+ Application


GTK+

GTK+ 어플리케이션 아키텍처





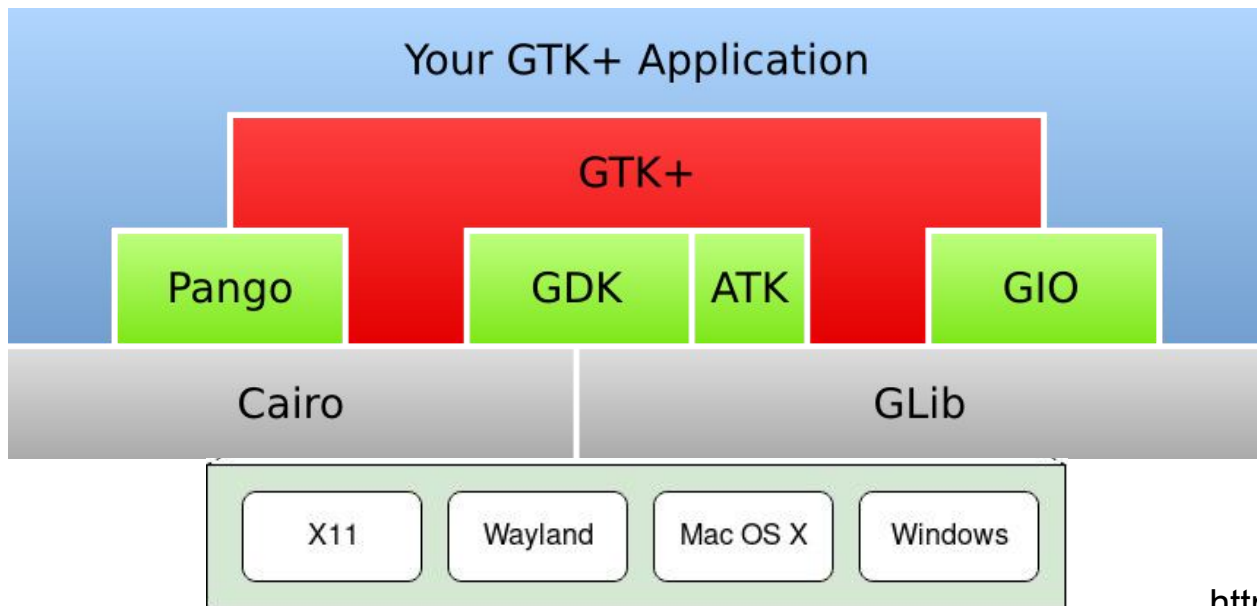
GTK+ 어플리케이션 아키텍처



Your GTK+ Application

GTK+

GTK+ 어플리케이션 아키텍처





GTK+ 어플리케이션 아키텍처



Your GTK+ Application

GTK+



왜 GTK+?

- 개발자가 2D Graphics, Windowing system API 사용 X
- GTK+는 멀티 플랫폼 라이브러리!
- OS에 관계없이 동일한 출력 결과

Basic Principles



Event-Driven Programming

- 프로그램의 흐름이 Events에 의해 결정
- 대부분의 GUI 프로그래밍은 Event-driven programming
- 마우스, 키보드, Pipes, Network sockets, IPC Messages, Processes/Threads 등등
- 이벤트를 받는 Main loop 존재



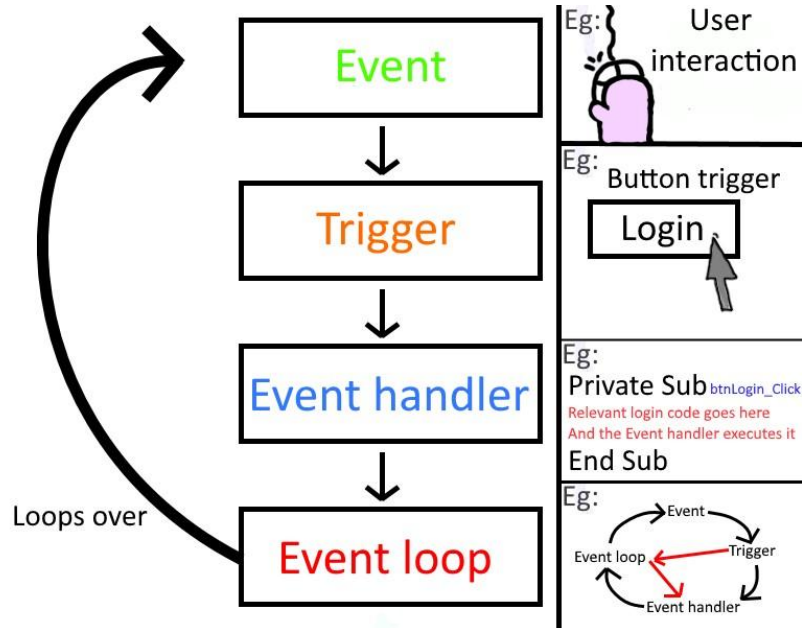
What does Main loop do?

The main event loop manages all the available sources of events for GLib and GTK+ applications. These events can come from any number of different types of sources such as file descriptors (plain files, pipes or sockets) and timeouts.

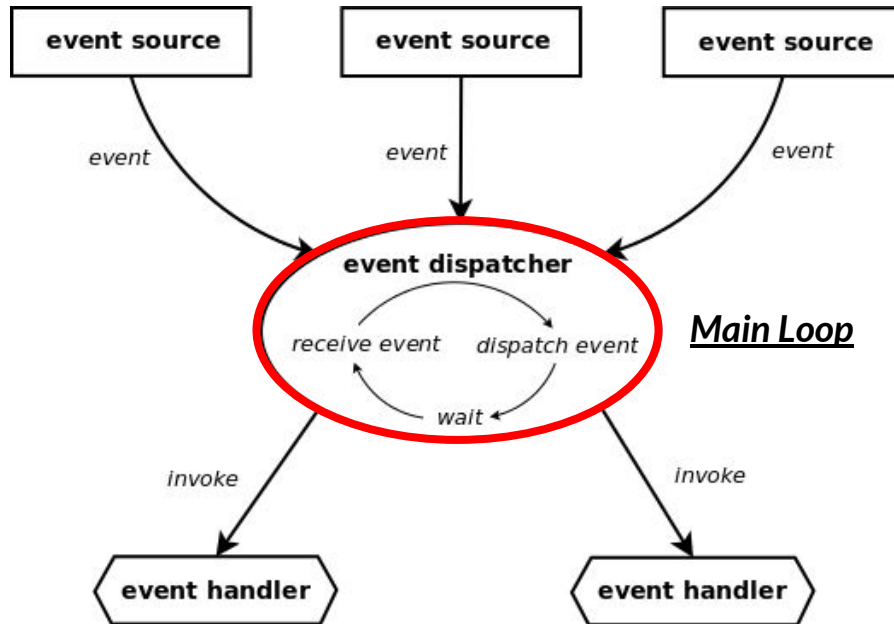
<https://developer.gnome.org/glib/stable/glib-The-Main-Event-Loop.html>

1. Events와 Event Handlers 등록
2. 등록된 모든 Events를 관리
3. 마우스, 키보드, Pipes, Network sockets, IPC Messages, Processes/Threads 등등

What does Main loop do?



What does Main loop do?





Signals

- GUI 프로그래밍의 기반이 되는 메커니즘
- 특정 액션이 발생했다고 알람을 울리는 것
- Signal이 발생하면 항상 **Callback** 함수가 실행
- 하나의 **Signal**에 여러가지 **Callback** 함수가 연결 될 수 있음



Signals and Events

Events

- 프로그램 외부에서 발생한 신호
- 예) Touch, Keyboard, Mouse

Signals

- 프로그램 내부에서 발생한 신호
- 예) Interclass communication

Basic Widgets



Gtk.Box

- 위젯(Widget)을 담을 수 있는 컨테이너 중 하나
- 화면에 보이지 않음
- Horizontal Box (수평) / Vertical Box (수직)
- `Gtk.Box.pack_start()` / `Gtk.Box.pack_end()` 메서드를 이용하여 위젯 추가
- Box 안에 또 다른 Box 를 담을 수 있음

lesson2.py



Gtk.HeaderBar

- Gtk.Box의 Horizontal Box와 비슷
- 위젯을 추가할 수 있음
- Gtk.HeaderBar.pack_start() / Gtk.HeaderBar.pack_end() 메서드를 이용하여 위젯 추가
- Title / Subtitle 설정
- 윈도우 프레임 제어 기능 제공 (최소화 / 최대화 / 닫기)
- GTK+ 어플리케이션 개발에서 일반적으로 사용하는 기능

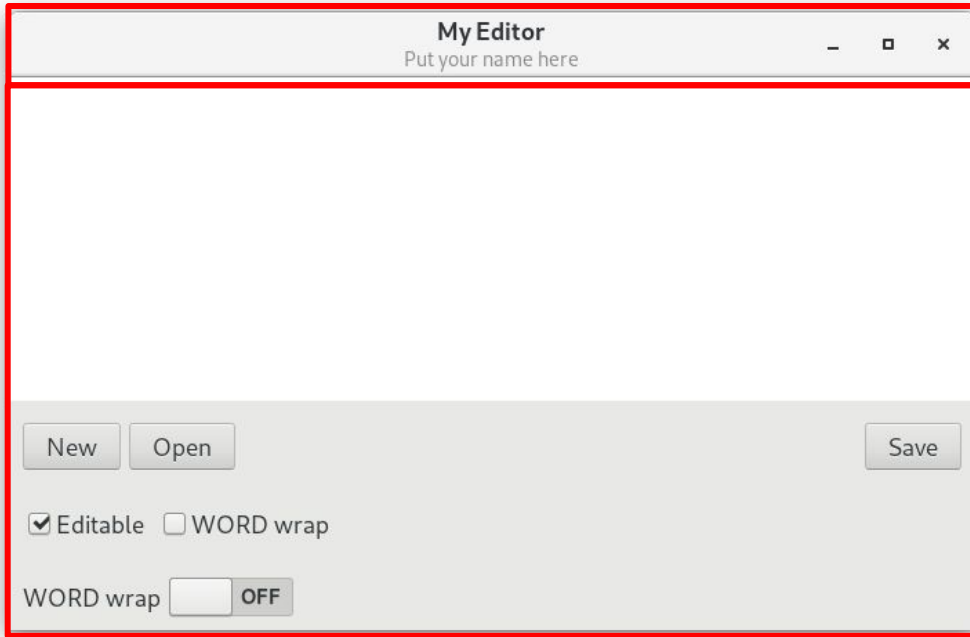


Mini Project (My Editor)



Mini Project (My Editor)

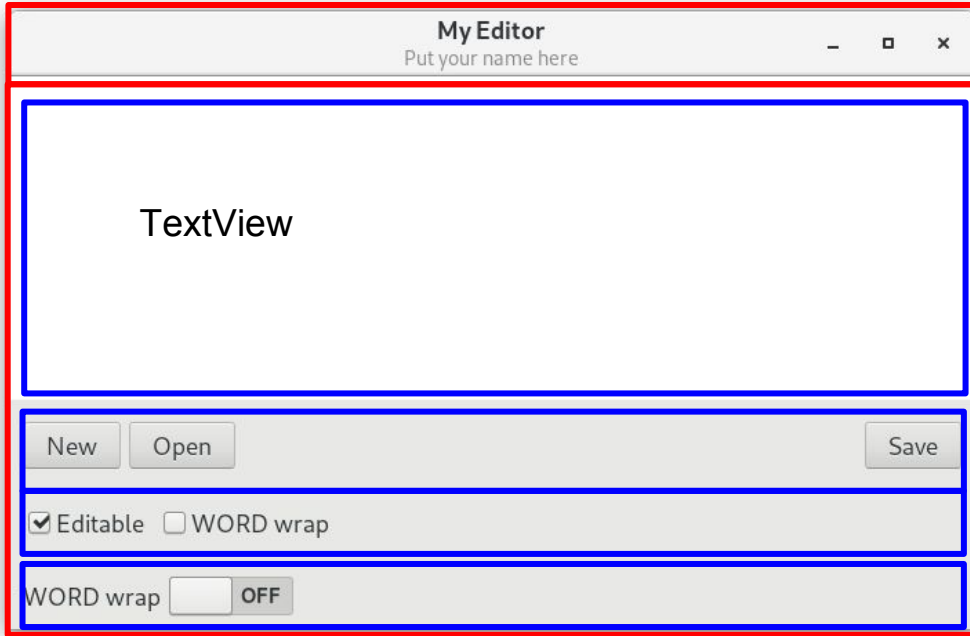
MyHeaderBar.py



MyMainBox.py

Mini Project (My Editor)

MyHeaderBar.py



MyMainBox.py