**Votary Media Player**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Author** | **Comments** | **Date Modified** |
| 1. |  |  |  |
| 2. |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Content:

Introduction:...........................................................................…....4

Back End Of Media Player:.....................................................….....4

Pipeline diagram.............................................................…....5

Front End Of Media Player:......................................................…....6

CSS..........................................................................……….....7

Working Of Media Player:........................................................…....8

Conclusion:.............................................................................…..10

# Introduction:

Implemented a Media Player Using gstreamer to integrate with GUI. GStreamer takes care of media playback while the GUI toolkit handles user interaction. CSS (Cascading style sheet ) is used to change the color of Gtk+3.0(Graphics tool kit) buttons on Player. For building Gstreamer player application we used individual functionality of elements like source,sink and filters.

This Media player can support a very wide variety of formats, including MP3, Ogg/Vorbis, MPEG-1/2, AVI, Quicktime, vob, webm and more.Player can support Linux,Windows and Mac Os X operating system platforms.

# Back End Of Media Player

**Gstreamer** is a back end of media player which is processing the data. It is a pipeline-based multimedia framework that links together a wide variety of media processing systems to complete complex work flows. The framework is based on plugins that will provide the various codec and other functionality. The plugins can be linked and arranged in a pipeline. This pipeline defines the flow of the data.

GStreamer plug-ins could be classified into:

1. sources: for audio and video (involves protocol plugins)
2. formats: parsers, formaters, muxers, demuxers, metadata, subtitles
3. codecs: encoders and decoders
4. filters: converters, mixers, effects, ...
5. sinks: for audio and video (involves protocol plugins)

**Elements:**

An element is the most important class of objects in GStreamer.An element has one specific function, which can be the reading of data from a file, decoding of this data or outputting this data to your sound card .

**Pads:**

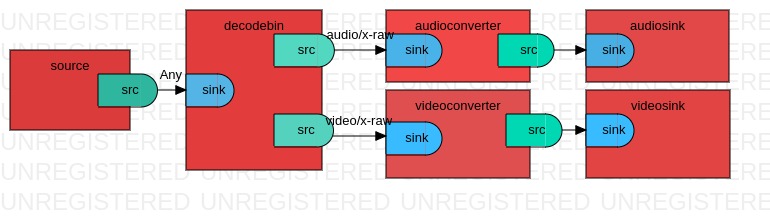
Pads are an element's input and output, where you can connect other elements. element where links may be made with other elements, and through which data can flow to or from those elements.

**Bins :**

A bin is a container for a collection of elements you can mostly control a bin as if it were an element, thereby abstracting away a lot of complexity for your application. You can, for example change state on all elements in a bin by changing the state of that bin itself. Bins also forward bus messages such as error messages, tag messages or EOS messages.

## Pipeline Architecture:

A pipeline is a top-level bin. It provides a bus for the application and manages the synchronization ,to create a pipeline that can do a specific task, for example media playback or capture.



**fig.1:Gstreamer pipeline diagram for player application.**

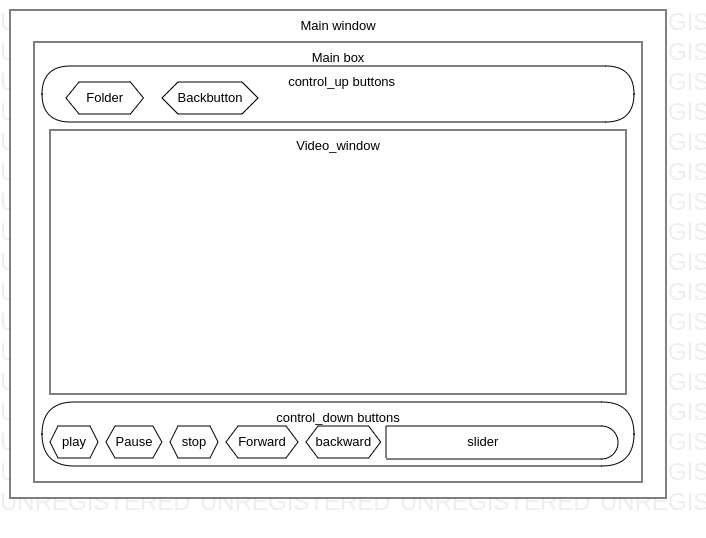
gst-launch-1.0 filesrc location=small.ogv ! decodebin name=dee dee. ! queue ! audioconvert ! audioresample ! autoaudiosink dee. ! queue ! videoconvert ! xvimagesink

The Player application reads the file from disc by using **filesrc** element,it can accept ANY type of caps.The container has both audio and video formats we separate both types and then decode(uncompress) them based on their media type this functionality is performed by using **decodebin** element.

For changing one to another color space audio and video **Converters** elements are used.Decoded audio and video are played by sound cards and display respectively using **sink** elements.This the way our data is processing in the backgound of player using gstreamer.

# Front End Of Media Player:

GTK+ is used as front end of media player .It is a [widget toolkit](http://en.wikipedia.org/wiki/Widget_toolkit). Each user interface created by GTK+ consists of widgets. This is implemented in C using GObject, an object-oriented framework for C. Widgets are organized in a hierarchy. The window widget is the main container. The user interface is then built by adding buttons, drop-down menus, input fields, and other widgets to the window .GTK+ is event-driven. The toolkit listens for events such as a click on a button, and passes the event to your application.

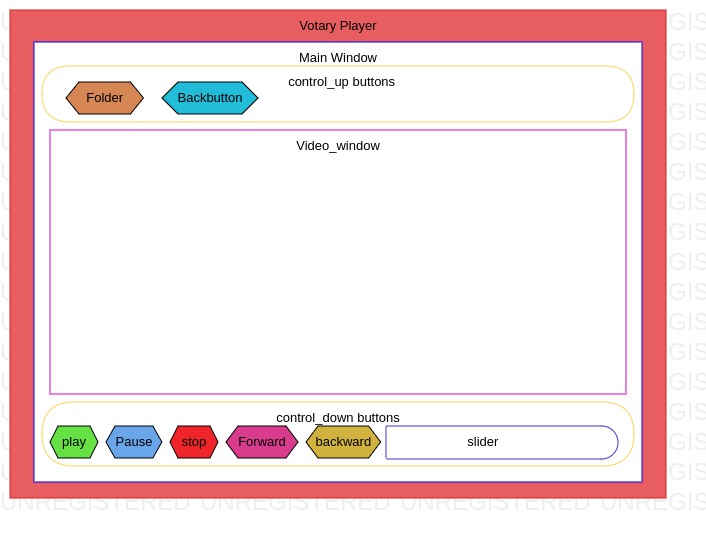


**Fig.2:MediaPlayer with Gtk+buttons**

Created four gtK+box's Control\_Up, Control\_down and main\_hbox and main\_box. In Control\_UP we added file Folder and file buttons.Play ,Pause,Stop,Forward seek ,backward seek and Slider are added in Control\_down box .Video\_window is added in main\_hbox. All this three boxes are added to main\_box. Finally the main\_box is added in Main\_window container.

## Cascading Style sheet(CSS):

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.CSS describes how HTML elements are to be displayed on screen, paper, or in other media .GTK+3.0 buttons ,window,border and background colors are changed using CSS script.



**Fig:3.Media Player Diagram with CSS**

#### Working Of Media Player:

This media player opens a GTK+ window and displays a image. Fetch the media from Files floder button, which media you wanted to play this Media player can support variety of formats.



**Fig.4:Media Player Pop Up Window.**

**Play**:In background of player gstreamer pipeline goes to playing state ,then window and sound card accept and render the data.

**Pause**:The gstreamer pipeline goes to paused state ,then Stops getting chunks from source file to window and display.

**Stop**:The gstreamer pipeline goes to NULL state,then their is no runtime libraries ,resources are not allocated and running time is 0.

**Forward** **Seek**:Seeking 'N' Seconds Forward.

**Backward Seek**:Seeking 'N Seconds Backward.

**Back/previous**:To identify the control or the indicator to stop the present media content and go back to the previous media which is already played.

**Mouse click on window**:Play/Pause a video when you click the left button of the mouse on video window.

**Folder**:Media player provides predefined location for users to select media files from our local disk to play the media content.

**Slider**:A slider to show the current position of the stream, which can be dragged to change the stream position.

# Conclusion

This allows you to build a somewhat complete media player with a proper Graphical User Interface. GUI toolkit creates the physical window that will hold the video..

1.High quality contents can render.

2.Multi operating systems supported media player.

3.Multi formats like ogg, mp4, webm, 3gp, mts... can render.