Introduction

The aim of this project is to develop a collection of
Inspiration is taken from the VizRT suite of applications.
Currently this collection consists of Chroma Viz, Hub, Engine
Chroma Viz, Hub and Artist are built in Golang and are
Chroma Engine is built in C and is contained in the [Chroma
<video width="720" controls>

```
    <source src="https://github.com/jchilds0/chroma-viz/raw/main/data/demo.m
```

</video>

Chroma Viz

Chroma Viz manages templates at a high level, and issues
On startup, Chroma Viz requests the templates from Chroma Hub.
Chroma Hub collects the template IDs of all templates in the
<img src="/assets/chroma-graphics/templates.png"
Pages can be created from templates, by double clicking on the
Chroma Viz will then send a request to Chroma Hub for this
Chroma Hub implements a REST API for accessing assets such as
Pages form shows, which can be saved to/loaded from disk for
<img src="/assets/chroma-graphics/show.png" alt="Templates">
Chroma Viz can connect to any number of Chroma Engine
Each connection is either a Engine or Preview connection. When
Chroma Engine provides a C library which can create a
Chroma Viz sends pages to layer 0 of the preview window, so
The actions at the top of the editor panel, $ exttt{Take On,
$ exttt{Take Off} $, send pages to connected Chroma Engine

Take On animates from Keyframe 1 to Keyframe 2.

Continue runs from the current Keyframe to the next Keyframe.

Take Off runs from the second last Keyframe to the last

Chroma Viz encodes the attributes of a page and sends it to
Chroma Engine parses the message and updates the templates
<img src="/assets/chroma-graphics/chroma-viz.png" alt="Chroma

Chroma Engine

Chroma Engine renders graphics requests from Chroma Viz. At its
We compile both a library, which is used by Chroma Viz to
On startup, Chroma Engine connects to Chroma Hub and requests
This is done so Chroma Engine can build its own database
This has the added cost of needing to allocate resources for
A middle ground between these two options would be allowing the
Chroma Engine renders graphics using OpenGL. To render a
A simple name-attribute format, name=attr#, is sufficient for our
The string contains a header with the format version, layer,

Then each geometry, specified by an integer, followed by a list
As we parse the string, we set the values for each geometry of
Chroma Engine features geometry maskingEach geometry has a
To achieve this we utilise OpenGL stencil buffers, keeping a
GTK restrictions mean we can only have 8 stencil buffers,
Image assets are also contained in Chroma Hub, so before we can
Chroma Hub first send 4 bytes with the length of the image,
Then we store this data for later use, as well as decoding the
In later render calls, if the image id matches the currently
After receiving the graphics request and any image assets,
This includes the absolute position calculationsThe keyframe
For smooth animations, we use a bezier curve to control the
Finally each geometry in the page is rendered to the screen
<img src="/assets/chroma-graphics/chroma-engine.png"

Chroma Hub

The purpose of Chroma Hub is to synchronize the graphic
Chroma Hub also stores any assets needed by the templates such
Chroma Hub wraps a SQL database, which currently needs to be
hub/chroma_hub.sql.Internally we use theencoding/jsonGolang
Chroma Hub implements a REST API for updating/retrieving
Chroma Artist is currently the only application which makes
Chroma Artist/Viz and Engine using GET requests to retrieve

Chroma Artist

Chroma Artist provides a UI for designing templates which can
The key difference between Chroma Viz and Chroma Artist is
In the discussion of Chroma Engine, we omitted the discussion
To enable easier manipulation of graphics, each geometry has a
This gives the geometries a tree structure, and position of a
Chroma Artist gives an easy interface to specify this tree
An example of this functionality is a simple lower frame super,
The designer could set the default position, width and height
Since the text geometries and circle are parented to the
In Chroma Artist, the following image shows an example of this
![keyframe](/assets/chroma-graphics/keyframeartist.png)
Currently the rectangle is static and the width of the
Keyframing allows us to animate the graphic and have the width
The keyframing system is based on a directed acyclic graphWe
This will be extended to construct keyframesConsider the
<img src="/assets/chroma-graphics/keyframeimg.svg"
<img src="/assets/chroma-graphics/keyframeimgdark.svg"
The x position of geometry 4 depends on the x position of
We indicate this dependency with an arrow from the x position
Continuing for all geometries we form the following graph,
<img src="/assets/chroma-graphics/keyframestart.svg"
<img src="/assets/chroma-graphics/keyframestart dark.svg"
At each node of this graph, we want to compute the value of the
To do this we store a function at each node, in the case of
But we need to evaluate each node only once each child node has
This problem is known as topological sortingThis excellent
kiXJYgjSUAU   We can make the rectangle width dynamic by adding
<img src="/assets/chroma-graphics/keyframeexpand.svg"

<img src="/assets/chroma-graphics/keyframeexpanddark.svg"
The keyframing process in Chroma Artist consists of two
Frames have an index 1, 2, ... up to some fixed number $ n $,
For each frame, we create the table of attributes for each
A Keyframe is a way to add dependencies and evaluation
When animated, geometry attributes are interpolated linearly
Current types of keyframes are

Set Frame: Set the value of an attribute of a geometry in a

User Frame: Similar to set frame, except the value from the

Bind Frame: Use a value computed in a keyframe. This adds an

Additionally we can set a keyframe to be an expand keyframe,
This is what we used above, we add an edge to the upper x
The only restrictions on keyframes is they cannot create cycles
Doing so makes evaluating them ambiguous, so Chroma Engine
Putting these together we can create the following graphic.
<video width="720" controls>

    <source src="https://github.com/jchilds0/chroma-viz/raw/main/data/artist

</video><link rel="stylesheet"