



Clipboard Pickling APIs

Anupam Snigdha
snianu@microsoft.com

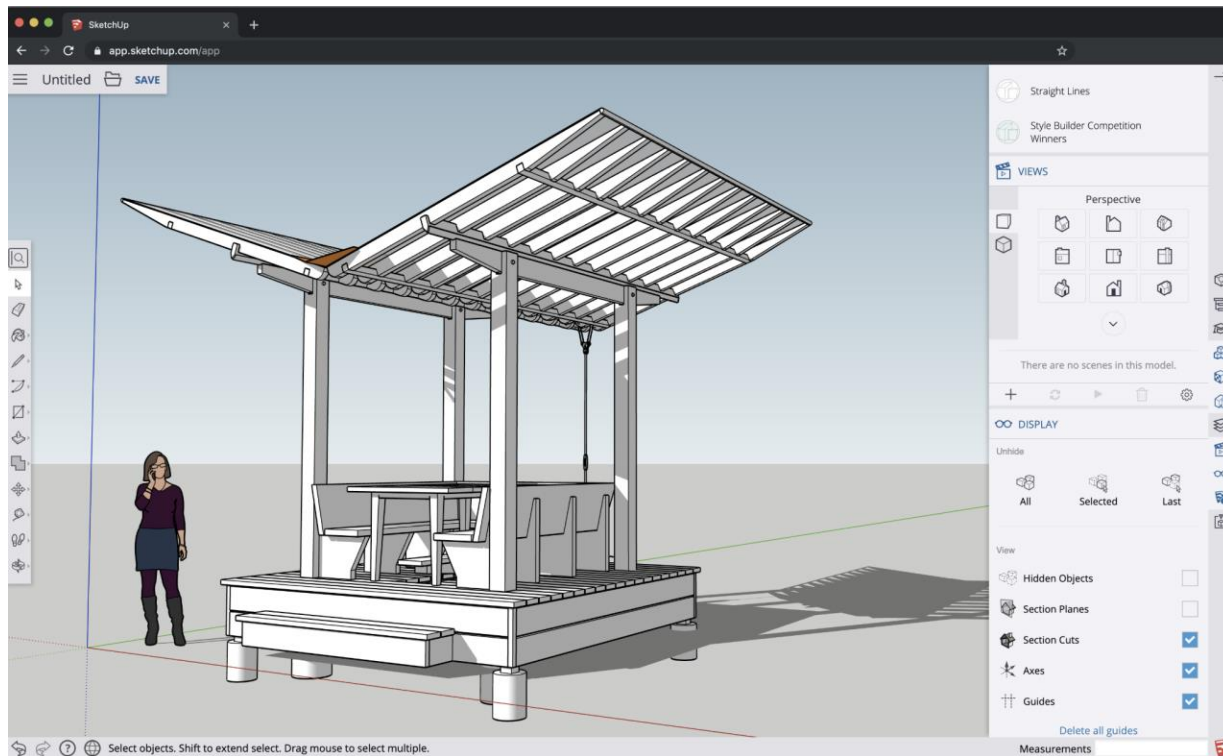
Agenda

- ▶ Introduction
- ▶ Demo
- ▶ Pickling Design
- ▶ Questions

Problems

- ▶ Proprietary formats – Native apps want to interact with sites to support formats that are exposed to all apps on the platform.
- ▶ Websites want to read proprietary formats provided by the native apps.
- ▶ E.g., Excel native apps provide “XML Spreadsheet” format that preserves the semantics of table cell and formulas. Excel online would like to use this format to provide much higher fidelity paste content.

Custom format examples



CAD design web apps(SketchUp)

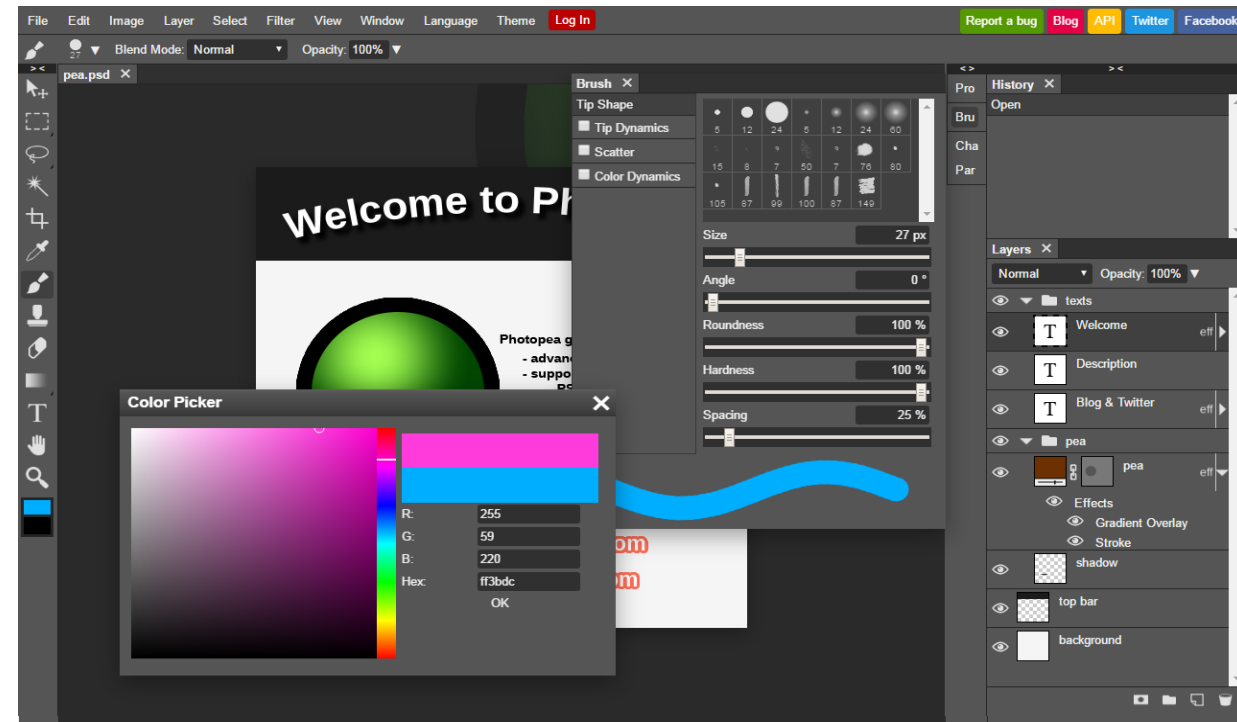


Image editor web apps(Photopea)
Image with layers and metadata

Pickling Design

- ▶ Expose an option to web authors to read custom formats. This is done via an `unsanitized` option in `ClipboardItemOptions` while using async clipboard read/write apis.
- ▶ Web originated formats so no interop with native apps. This is by-design as we don't want to expose unsanitized content to legacy apps that don't receive any updates.
- ▶ Native apps may opt-in to using custom formats.
- ▶ Sanitized standard formats (such as html, png, plain text etc) are used for general use for compat & safety.

Demo

Pickling Write

```
// Pickling write example.
// This format 'text/plain' is recognized by the Clipboard API, so will be
// written as usual.
const text = new Blob(['text'], {type: 'text/plain'});
// This format 'text/custom' is not sanitized by the Clipboard API. It will be
// pickled if the format is specified in the {unsanitized: []} formats list.
const customText = new Blob(
  ['<custom_markup>pickled_text</custom_markup>'], {type: 'text/custom'});

// Clipboard format ordering: Pickled formats will be written before sanitized
// formats by the browser, since they're more "custom" and likely more targeted
// towards this use case.
const clipboardItem = new ClipboardItem({
  'text/plain': text,      /* Sanitized format. */
  'text/custom': customText /* Pickled format. This new format will be accepted
                             and written without rejection, as long as the new
                             unsanitized list contains this format. */
},
{unsanitized: ['text/custom']} /* This new list specifies the pickled format
                                'text/custom'. */
);
navigator.clipboard.write([clipboard_item]);
```

Pickling Read

```
// Pickling read example. ClipboardItems returned by clipboard.read() may
// contain pickled formats.
const clipboardItems = await navigator.clipboard.read(
  {unsanitized: ['text/custom']} /* This new list specifies the pickled format
                                'text/custom' for all read ClipboardItems. */
);
const clipboardItem = clipboardItems[0];

const textBlob = await clipboardItem.getType('text/plain');
// This format reads as a pickled format, only if it is included in the unsanitized
// format list.
const customTextBlob = await clipboardItem.getType('text/custom');
```


OS format mapping

- ▶ Due to limitations in number of formats that can be registered in Windows & X11 clipboards, we propose to support maximum of 100 custom formats. This avoids exhausting the atom pool in the OS.
- ▶ A web custom format map is introduced that contains the mapping of web custom format to MIME types. E.g.
`{"text/html": "Web Custom Format0", "xml/spreadsheet": "Web Custom Format1"}`
- ▶ The actual payload of the MIME types will be written to the clipboard under the web custom formats. For xml/spreadsheet, it will be present in Web Custom Format1.
- ▶ More info specific to platforms are described [here](#).

Questions?

- ▶ Specifically looking for feedback on the shape of the web API & OS format naming convention.
- ▶ File issues in the Editing repo tagged with clipboard-pickle-APIs: <https://github.com/w3c/editing/issues>

More references

- ▶ Explainer: <https://github.com/w3c/editing/blob/gh-pages/docs/clipboard-pickling/explainer.md#pickling-for-async-clipboard-api>
- ▶ Editing TF Minutes: <https://lists.w3.org/Archives/Public/public-editing-tf/2020Oct/0017.html>
- ▶ Chrome Status entry: <https://chromestatus.com/feature/5649558757441536>
- ▶ Design Doc: <https://docs.google.com/document/d/1afc45MQuwxEWgoUeJCO-sOWRSzs31V4JS-kKXJNMTXw/edit?usp=sharing>