Contents

Running a simple pipeline

In this section, we demonstrate how to run the full pipeline for the Civilization postgame generation.

Building the namespace

Let's build the namespace from the `src` directory in the root of this project/repository again, so we keep this directory self-contained:

```
> viash ns build \
+   -n civ6_save_renderer \
+   -s ../src \
+   -p docker
Exporting ../src/civ6_save_renderer/combine_plots/ (civ6_save_renderer) =docker=> target/docker/civ
Exporting ../src/civ6_save_renderer/convert_plot/ (civ6_save_renderer) =docker=> target/docker/civ6
Exporting ../src/civ6_save_renderer/parse_header/ (civ6_save_renderer) =docker=> target/docker/civ6
Exporting ../src/civ6_save_renderer/parse_map/ (civ6_save_renderer) =docker=> target/docker/civ6_sa
Exporting ../src/civ6_save_renderer/plot_map/ (civ6_save_renderer) =docker=> target/docker/civ6_sav
```

The result is stored under `target/docker` because we chose to only build the `docker` platform executables.

We have to run the *setup* for the containers that are not just available on Docker Hub. This can be done in one go by using the following CLI:

```
cd ..
viash ns build \
  -n civ6_save_renderer \
  -s ../src \
  -p docker
```

Or, we can run them one-by-one.

A pipeline script

A small dataset with only a few steps from a game are stored under `../data/`. We will use that as a source for the pipeline. Furthermore, we'll *eat our own dogfoot* and create a viash component for the pipeline code itself:

src/civ6_save_renderer/pipeline/config.vsh.yaml:

```yaml
functionality:
  name: pipeline
  arguments:
    - name: "--input"
      alternatives: [ "-i" ]
      type: file
      description: "Input directory with savegames"
      required: true
    - name: "--temp"
      alternatives: [ "-t" ]
      type: file
      description: "Temporary directory"
      direction: output
      default: "temp"
    - name: "--output"
      alternatives: [ "-o" ]
      type: file
      description: "Output video filename"
      direction: output
      required: true
    - name: "--sed"
      type: string
      description: "Path to the GNU version of sed"
      default: 'which gsed || echo -n "/usr/bin/sed"'
    - name: "--component_dir"
      type: file
      description: "Path to the namespace target dir"
      default: "./target"
  resources:
    - type: bash_script
      path: script.sh
platforms:
  - type: native
```

With the following script:

src/civ6_save_renderer/pipeline/script.sh:

```bash
#!/bin/bash

input_dir="$par_input"
temp_dir="$par_temp"
output="$par_output"

# $component_dir is target/ by default
BIN="$par_component_dir/docker/civ6_save_renderer"
# $SED resolves to GNU sed
SED=`eval "$par_sed"`
```

```
mkdir -p "$temp_dir"

function msg {
  echo ">>>>>>> $1"
}

for save_file in $input_dir/*.Civ6Save; do
  file_basename=$(basename $save_file)
  yaml_file="$temp_dir/${file_basename/Civ6Save/yaml}"
  tsv_file="$temp_dir/${file_basename/Civ6Save/tsv}"
  pdf_file="$temp_dir/${file_basename/Civ6Save/pdf}"
  png_file="$temp_dir/${file_basename/Civ6Save/png}"

  if [ ! -f "$yaml_file" ]; then
    msg "parse header '$save_file'"
    $BIN/parse_header/parse_header -i "$save_file" -o "$yaml_file"
  fi

  if [ ! -f "$tsv_file" ]; then
    msg "parse map '$save_file'"
    $BIN/parse_map/parse_map -i "$save_file" -o "$tsv_file"
  fi

  if [ ! -f "$pdf_file" ]; then
    msg "plot map '$save_file'"
    $BIN/plot_map/plot_map -y "$yaml_file" -t "$tsv_file" -o "$pdf_file"
  fi

  if [ ! -f "$png_file" ]; then
    msg "convert plot '$save_file'"
    $BIN/convert_plot/convert_plot -i "$pdf_file" -o "$png_file"
  fi
done

png_inputs=`find "$temp_dir" -name "*.png" | $SED "s#.*#&:#" | tr -d '\n' | $SED 's#:$#\n#'`

if [ ! -f "$output" ]; then
  msg "combine plots"
  $BIN/combine_plots/combine_plots -i "$png_inputs" -o "$output" --framerate 1
fi

msg "DONE"
```

A few remarks are in place here:

TODO

Build the `pipeline` component:

```
> viash ns build -p native
Exporting src/civ6_save_renderer/pipeline/ (civ6_save_renderer) =native=> target/native/civ6_save_r
```

Running the pipeline

Make sure you have run the required `--setup`'s. Then:

```
> target/native/civ6_save_renderer/pipeline/pipeline \
+   -i ../data \
+   -o output.webm
>>>>>>> parse header '../data/AutoSave_0158.Civ6Save'
>>>>>>> parse map '../data/AutoSave_0158.Civ6Save'
(node:9) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues
(Use `node --trace-deprecation ...` to show where the warning was created)
>>>>>>> plot map '../data/AutoSave_0158.Civ6Save'
── Attaching packages ─────────────────────────────────── tidyverse 1.3.0 ──
✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>>>>>>> convert plot '../data/AutoSave_0158.Civ6Save'
>>>>>>> parse header '../data/AutoSave_0159.Civ6Save'
>>>>>>> parse map '../data/AutoSave_0159.Civ6Save'
(node:9) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues
(Use `node --trace-deprecation ...` to show where the warning was created)
>>>>>>> plot map '../data/AutoSave_0159.Civ6Save'
── Attaching packages ─────────────────────────────────── tidyverse 1.3.0 ──
✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>>>>>>> convert plot '../data/AutoSave_0159.Civ6Save'
>>>>>>> parse header '../data/AutoSave_0160.Civ6Save'
>>>>>>> parse map '../data/AutoSave_0160.Civ6Save'
(node:9) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues
(Use `node --trace-deprecation ...` to show where the warning was created)
>>>>>>> plot map '../data/AutoSave_0160.Civ6Save'
── Attaching packages ─────────────────────────────────── tidyverse 1.3.0 ──
✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>>>>>>> convert plot '../data/AutoSave_0160.Civ6Save'
>>>>>>> parse header '../data/AutoSave_0161.Civ6Save'
```

```
>>>>>>> parse map '../data/AutoSave_0161.Civ6Save'
(node:9) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues
(Use `node --trace-deprecation ...` to show where the warning was created)
>>>>>>> plot map '../data/AutoSave_0161.Civ6Save'
── Attaching packages ─────────────────────────────────── tidyverse 1.3.0 ──
✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>>>>>>> convert plot '../data/AutoSave_0161.Civ6Save'
>>>>>>> parse header '../data/AutoSave_0162.Civ6Save'
>>>>>>> parse map '../data/AutoSave_0162.Civ6Save'
(node:9) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues
(Use `node --trace-deprecation ...` to show where the warning was created)
>>>>>>> plot map '../data/AutoSave_0162.Civ6Save'
── Attaching packages ─────────────────────────────────── tidyverse 1.3.0 ──
✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>>>>>>> convert plot '../data/AutoSave_0162.Civ6Save'
>>>>>>> DONE
```