Contents

1 Basic Usage

2 Menus
2.1 Scene
2.2 Basic Settings
2.3 View Settings
2.4 Displayed Info
2.5 List Projection
2.6 Export Textures
2.7 Camera Record Settings
2.8 Instant Stream
2.9 Animation
2.10 Textures

3 Importing your own models
3.1 Importing your own models
3.2 Troubleshooting

4 Streaming eyetracking data

1 Basic Usage

When starting the Attention Visualizer you will have two options. You can set the size of the attention textures used and also the screens resolution. Once you made your selection, the windows size will switch to the size of your selected resolution. While you can still resize the window, the resolution will stay the same and cannot be changed unless you restart the program.

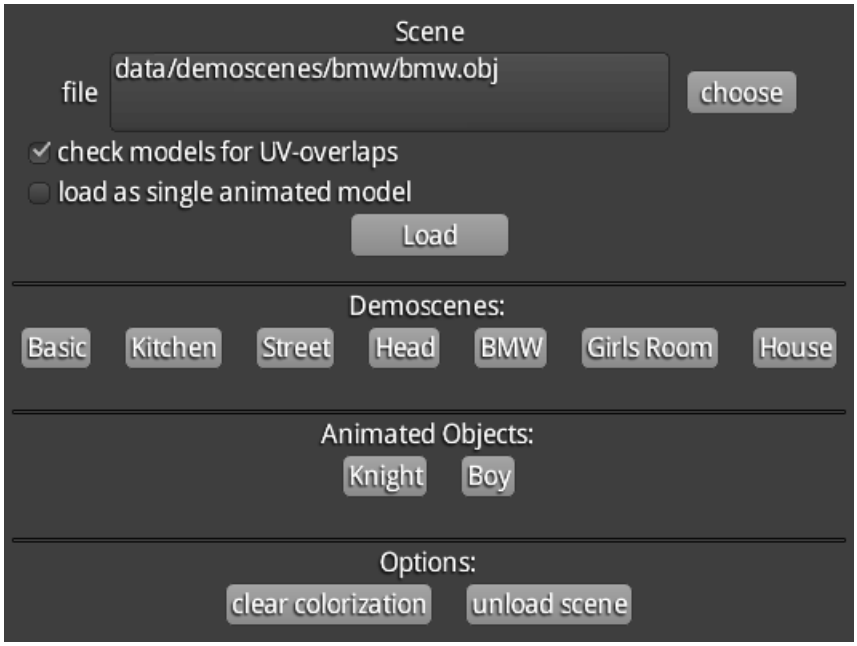
After that you will get into to main program. On the upper right corner you will have visuals that show you which options are currently active. All these options are buttons and can be triggered by clicking, however some also have macro-keys applied.

1. If you are currently projecting. Trigger with 'E'.
2. If you are currently recording your camera movements. Trigger with 'R'.
3. Click to display the next animation if an animated model is loaded. Trigger with 'N'.
4. Pausing and restarting animations. Trigger with 'P'.
5. If streaming of eye data is active.
6. If the restful server for streaming is started.
7. Open and close the menu sidebar.

2 Menus

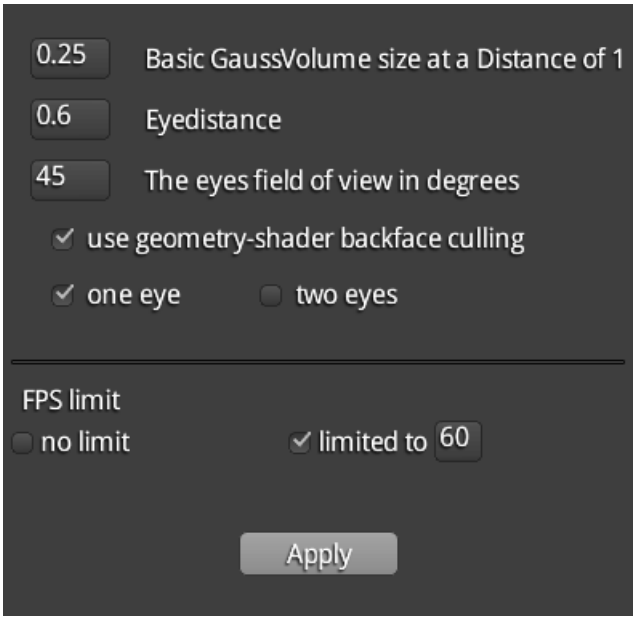
2.1 Scene

In this menu you can load models into the scene. By clicking on the "Demoscenes" buttons you can set the options to load a specific prepared scene. Similarly the "Animated Objects" buttons set the loading options to specific animated models. If you check the "check models for UV-overlaps" box, the program will test alle loaded models for possible uv-overlaps after loading. You can load as many objects as you want, however there is not checking for dedicated graphics space therefore the program will crash if you have loaded too much. If you need to load many models you may be forced to reduce the size of the attention textures (see Textures). By checking the "load as animation" box, the program will link skeleton and mesh data to load a single animated model. On the Options you can clear all colorizations in the scene or unload the scene.



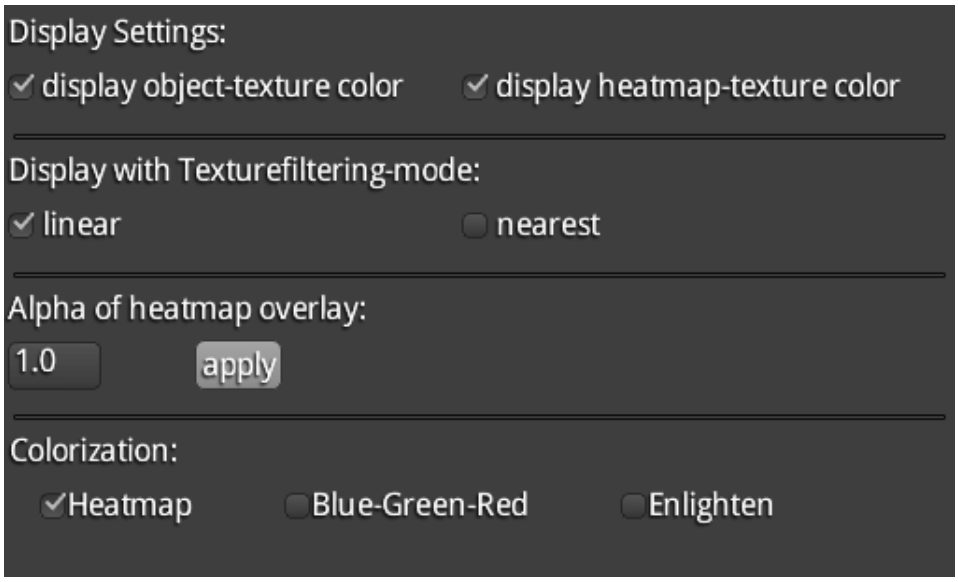
2.2 Basic Settings

Here you can control the basic projection settings. The size of the gauss volume at a unit distance of 1. The distance between eyes. The eyes field of view. You can turn off geometry-shader backface culling for testing purposes. And you can switch between the monocular and binocular mode. Also you can set the FPS limit which is equal to the PPS limit (projections per second). Keep in mind, that unlimited FPS should only be used with streaming or list projection modes. If you are simply using 'E' to project into the scene unlimited FPS will result in more projections on areas where the computations are done faster, resulting in an scene-based unequal PPS rate.



2.3 View Settings

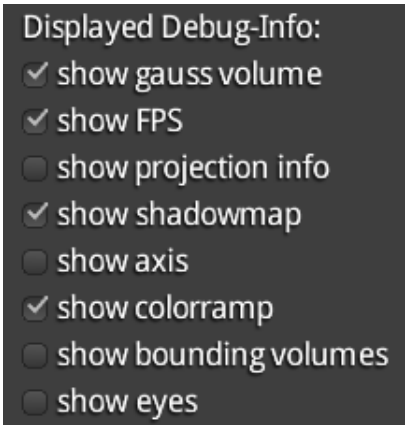
Here you can change how the scene is displayed. You decide if object-textures or heatmap-textures are displayed or not. You can change filtering modes of the displayed heatmap-texture. You can set a reduced alpha for the heatmap-texture. You can switch the colorramp mode.



2.4 Displayed Info

Here you can decide which additional information is displayed. Some notes to the displayed info:

- The gauss volume will only be drawn if a valid point of regard exists that would colorize the scene.
- If you are using binocular mode the shadowmap will display both shadowmaps in one picture with the red channel (left eye) and green channel (right eye).
- The displayed attention times on the colorramp depend on your FPS limit settings in the 'Basic Settings' menu. You should not change your FPS limitation in between gaze projections or else displayed times will not be correct.



2.5 List Projection

In this menu you can select a previously recorded camera movement and reproject it to the scene by choosing the file and hitting start projection.

If you are reprojecting a list projection file, you have to consider that the Attention Visualizer will not recreate your settings of when you recorded. Therefore a list projection allows you to reproject the same camera movements with different settings. For example you can change the gauss volumes base size the eye distance or the field of view. A nice test is to record gaze projections in binocular mode and reproject them in monocular mode to highlight their differences. Keep in mind though, that if you have recorded in monocular mode, only camera data for one eye will be available and the Attention Visualizer will force monocular modes for these gaze projections.

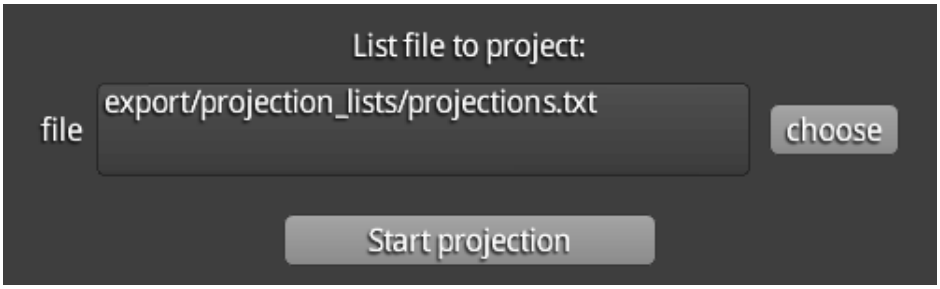
This mode is your option, if you don't want to use streaming for eyetracking data. Simply export your gaze projections in the specified format:
Each gaze projection consists of the head camera, the left eye camera and the right eye camera. For each camera you will have to provide position, direction and up vectors.

Take these vectors and assemble them like this:

```
head.position;head.direction;head.up;leftEye.position;leftEye.direction;leftEye.up;
rightEye.position;rightEye.direction;rightEye.up
```

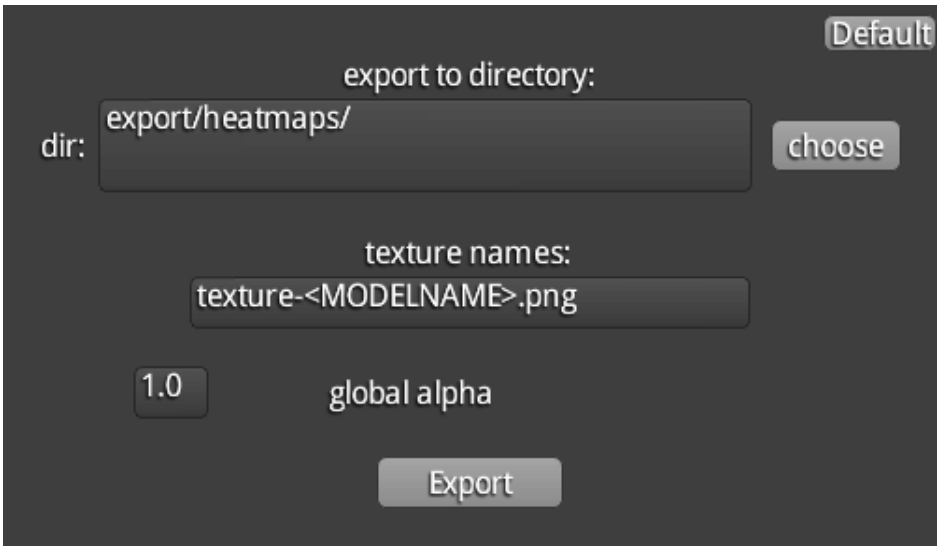
The vectors are separated by a semicolons (';'). While the values of each vector are separated by a commas (','). Also there shouldn't be any spaces or tabs. Single gaze projections are separated by newlines, therefore each line represents one gaze projection.

As an example you can record some gaze projections and view the exported file.



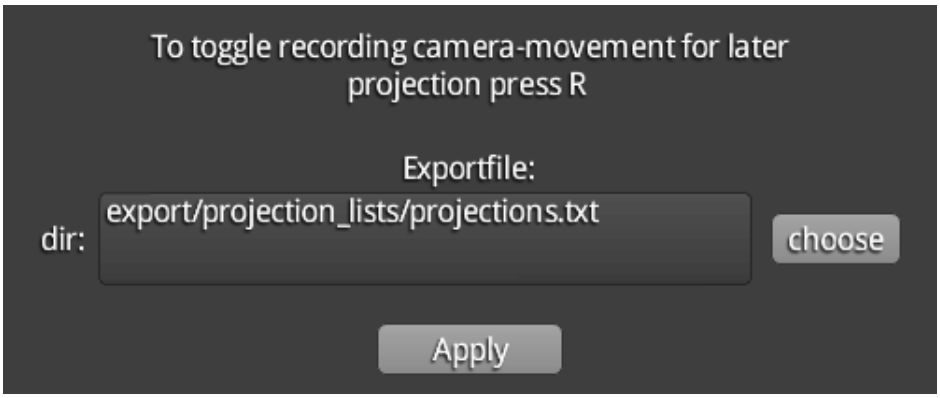
2.6 Export Textures

Here you can export the colorized object-heatmaps. Choose a folder for export, enter your naming convention for the files where '<MODELNAME>' is replaced by the models name and adjust the alpha of the exported texture. The exported textures will always be unfiltered, and uncompressed. The colorization will be equal to the colorramp selection you made in the 'View Settings' menu.



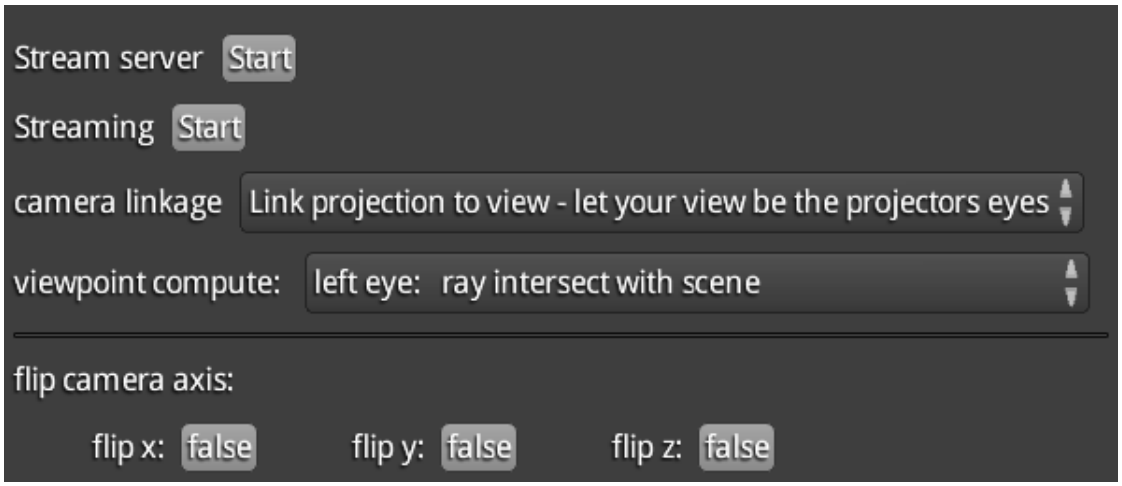
2.7 Camera Record Settings

Here you can change the file to which the program should export recorded camera movements. If the selected file does not exist it will be created. Also multiple recordings into one file will always be appended and not overwritten. When recording camera movement the program will only record valid gaze projections, where attention values are added to the scene.



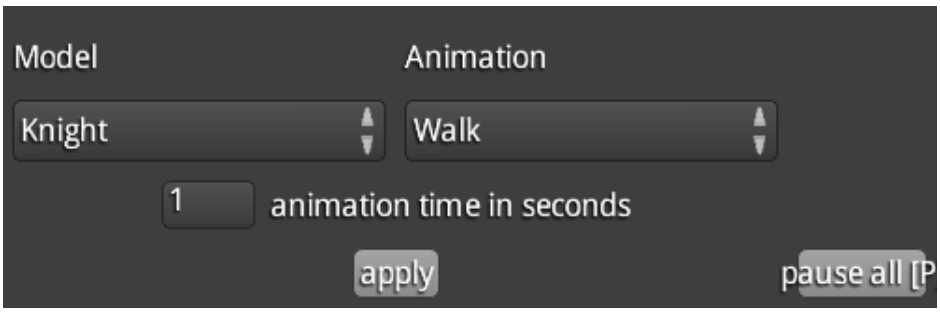
2.8 Instant Stream

This menu allows you to stream eyetracking data and to change camera linkage modes. Also the viewpoint compute setting is very important, not only for streaming, but also for list projections. If you are using the streaming or list projection modes it allows you to choose how the point of regard is computed. Usually it will use the left eye for intersections with the scene in monocular and binocular modes to compute the point of regard. However if you have binocular data available you can also choose the right eye or a closest point between two eye-lines computation.



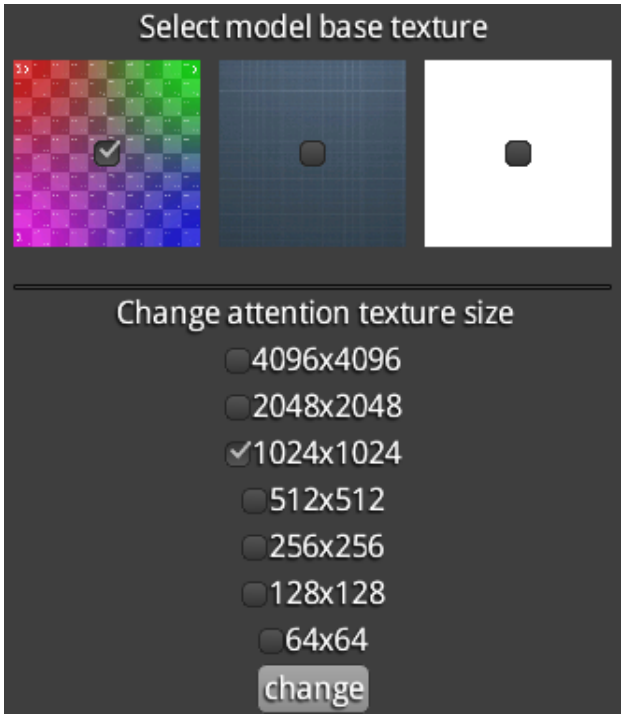
2.9 Animation

If a animated Model is loaded into the scene you can manually change the currently played animation in this menu. Simply choose the model and the animation and hit apply. You can also change the speed of the animation.



2.10 Textures

In this menu you can change the basic texture that is applied to the models. Also you can change the size of the attention-textures. Keep in mind, that changing the texture size will unload the scene.



3 Importing your own models

3.1 Importing your own models

Things you need to do/check/consider:

- Add texture coordinates to every model, or else you won't be able to project attention onto them
- Remove all overlapping texture regions, this includes repeated wrapping UVs.
- Export with normals if possible.
- Exported models should not have meshes with only a single vertex or two vertices.
- Exported models should not have meshes with more than 32767 indices. This is a libgdx limitation. If your model won't show in the program then this might the problem. You can simply separate the model into multiple pieces ('separate by loose parts' in Blender). If you don't know which models have too many indices you may use fbx-conv (see below) to check.

Additionaly if you want to export an animated model you have to deactivate NLA strips (this applies to Blender 2.73+).

Supported filetypes for loading are '.obj' and '.fbx'. If you want to export an animated model you will be forced to export as '.fbx' since '.obj' doesn't support bone information. Fbx files will then need to be converted to the libgdx internal '.g3db' file format. Converting is easy using fbx-conv:

https://github.com/libgdx/fbx-conv

3.2 Troubleshooting

- My model won't show up: It either has too many indices (convert to g3db with fbx-conv in verbose to check), or it has single vertices and lines that do not create faces.
- Model loads but i can't colorize it: Your model is missing texture coordinates.
- The program crashes when loading the model: The loaded scene might have too many models and you go out of memory (reduce texture size before loading and see if it still happens). Or the textures are missing add them to solve this. If you are using '.obj' you can delete the texture references in the '.mtl' file.
- Model loads but texture is wrongly applied to the surface: Happens usually with '.fbx' models. Use '-f' to flip texture coordinates when converting to g3db with fbx-conv and manually flip the models texture vertically.

4 Streaming eyetracking data

You can link the Attention Visualizers camera and eye movements to any eyetracker by using the streaming interface. To do this first of start the Server in the 'Instant Stream' menu. This will start a Restful Server that accepts camera information for a head camera, the left eye camera and the right eye camera in the following json format:

```
{
   "camCamera":{
      "name":"camera",
      "position":{
         "x":0.0,
         "y":0.0,
         "z":10.0
      },
      "direction":{
         "x":-0.0,
         "y":-0.0,
         "z":-1.0
      },
      "up":{
         "x":0.0,
         "y":1.0,
         "z":0.0
      }
```

```
    },
    "camLeftEye":{
        "name":"leftEye",
        "position":{
            "x":0.0,
            "y":0.0,
            "z":10.0
        },
        "direction":{
            "x":-0.0,
            "y":-0.0,
            "z":-1.0
        },
        "up":{
            "x":0.0,
            "y":1.0,
            "z":0.0
        }
    },
    "camRightEye":{
        "name":"rightEye",
        "position":{
            "x":0.0,
            "y":0.0,
            "z":10.0
        },
        "direction":{
            "x":-0.0,
            "y":-0.0,
            "z":-1.0
        },
        "up":{
            "x":0.0,
            "y":1.0,
            "z":0.0
        }
    }
}
```

Any json formating is accepted, but the data has to be sent as plain text. Simply send one package of data for each gaze projection to: 'http://localhost:8080/rest/message/data' where 'localhost' can be replaced with the ip or name of the machine that is running the Attention Visualizer.

Additionally you will have to start the streaming mode in the "Instant Stream" menu. Once you have switched into streaming mode, the incoming camera data will be applied to internal camera.

You can test if you have sent the correct data either by displaying the eyes (while projecting) or by linking the view camera to the projection camera (second option in the 'Instant Stream' menu). You need to at least send the head camera for camera linkage and the left eye camera for one eyed projection. The right eye camera is only used if you activate two eyed projection in the 'Basic Settings' menu. If you do use two eyes, you can change how the point of regard is computed to a closest point of to lines computation. This can be changed under viewpoint computation in the 'Instant Stream' menu. Also you might want to set the FPS to unlimited if you are sending more than 60 projections per second so the Attention Visualizer won't lag behind.