Chapter 2

Accessible Audio Toolkit

2.1. Description

Accessible Audio Toolkit is a set of tools designed with the purpouse of helping developers include solutions for hearing disabilities into their games. It is comprised of two, highly configurable, main tools that are easy to implement into any game using Unity Audio Sources. This guide will help you set up the different tools into your environment.

It can be downloaded from this website: `https://narratech.com/tool/accessible-audio-toolkit`

2.2. Step by step guide

2.2.1. Step 1: Set up the manager

AAT is fully configured from an unified script, the AATManager. Just create an empty object and drag-and-drop the AATManager script into it or add it from the *Add component* menu in Unity.

You can now set your preferred audio tags from the AATManager's object interface. Set a number of *Audio Type* objects and give each one of them a *Tag Name* and a *Tag Color*. Make sure to check the transparency (A) attribute of each color since it will be at 0 by default and colors will not be displayed. You will need further configuration if you want to use the *Bubble* or the *Compass* tools, but we will explain it in their own section.

2.2.2. Step 2: Set up the sources

Each object in your game that you want to be tracked by the system will need to have an AATSource script attached to it. You can do this the same way as you did with the manager, just drag-and-drop the script into your object or use Unity's *Add component* menu. Keep in mind that AAT interacts with Unity Audio Sources, so make sure to place the AATSource script in an object that has a Unity Audio Source.

Give each object a tag matching any of the *Tag Names* you configured in the AATManager. The object will be represented in the system by the color associated to that tag.

Now you just need to tell each AATSource which object is their manager. To do this, drag the AATManager object that you just created in the previous step into the *Manager* field of each AATSource.

2.2.3. Step 3: Set up the *Bubble*

If you are not planning on using the *Bubble* tool you can skip this step.

- **Instantiate the *AAT Bubble Canvas* prefab**
  You can find the *AAT Bubble Canvas* prefab in the *Prefabs* folder. Just drag and drop it anywhere on your scene. Set the *Render Camera* property in the inspector to your Main Camera.

  The canvas includes an icosphere mesh that makes the core of the tool. You can change this mesh to any of your liking, but bear in mind that a sphere with a large number of vertices may cause performance issues, and one with too few may look uglier.

- **Instantiate the *AAT Camera* prefab**
  You can find the *AAT Camera* prefab in the *Prefabs* folder. Just drag an drop it into your game's main camera so it ends as a child of your camera. This tool is designed for first person games, so it will most likely be attached to your player object.

  The *Bubble* uses its own camera so it is not distorted by the common perspective projection camera, but you can ignore this and use your own camera. If you use the AATCamera, you will need to add the AAT layer to your main camera's culling mask unless you want to render the *Bubble* twice.

- **Place the sphere and the camera** Make sure you place the sphere object in the correct position of the camera viewport and with an adequate size. This may vary depending on your window resolution and your own preferences. We suggest you place the sphere in the same position as the player or right next to it (which usually means a Z position of about -545 from the player), and then move the AATCamera slightly so the sphere is rendered in your preferred position

- **Configure the manager**
  Go back to the AATManager object from Step 1 and drag-and-drop the *Bubble* object from the *AAT Bubble Canvas* into the *Sphere* field of the AATManager.

  You can now configure the different behaviour parameters of the tool, each one will have a tooltip on its purpouse. When you are done, make sure to click the *Enable Sphere* checkbox and you are ready to go.

2.2.4. Step 4: Set up the *Compass*

If you are not planning on using the *Compass* tool you can skip this step.

- **Instantiate the *AAT Compass* prefab**
  You can find the *AAT Compass* prefab in the *Prefabs* folder. If your game has an UI Canvas, we suggest you drag-and-drop it inside of it, if not, just create one by right clicking in Unity's *Hierarchy* tab => UI => Canvas and set the *AAT Compass* prefab as his child.

  The prefab includes a default black background and a default compass image with the N-W-S-E directions. You can change these to any of your liking.

- **Configure the manager**
  The manager needs pointers to many of the compass elements. You can set them as the following:

  - **Main camera:** Drag and drop your main camera object here.
  - **Player:** Drag and drop your player object here.
  - **Icon:** This is the image that will represent each sound in the compass. You can find a default one under the name *icon* in the *Images* folder but you can use any of your liking. Just remember it will be recolored to match each source's associated tag's color.
  - **Arrow:** The icon that represents objects that are above or below the players viewport. You can find a default one under the name *arrow* in the *Images* folder but you can use any of your liking.
  - **Compass marker:** The prefab that envelopes both the previous icon and the arrow. You can find it in the *Prefabs* folder under the name *AAT Compass Marker*.

- **Compass Image:** The object that contains the foreground image of the compass, which by default is the one with the N-W-S-E directions. Just drag and drop it from inside the AATCompass object you prevoiusly instantiated.

You can now configure the different behaviour parameters of the tool, each one will have a tooltip on its purpouse. When you are done, make sure to click the *Enable Compass* checkbox and you are ready to go.