UTA
Department of Computer Science and Engineering

ThereMelo

Senior Design

Ron Nguyen / Sazzadul Islam / Caleb Myatt / Scott Frazier / Farhan Sadiq

Background

There are many reasons why we wanted to create the ThereMelo. This project is a great gateway to show the younger generation about STEM-related projects in a fun way. We want to show there is much more than just standard projects.
Computer vision is integrating itself into our society every passing day. We aim to integrate the use of computer vision into the field of music which also has a great impact on our daily lives. ThereMelo is a way to create music that will make it more accessible to those who may not have the ability as a result of the current state of music creation. All someone needs to have access to is a camera in order to realize their creative musical ideas.
Our aim is also to inspire the younger generation to take up an interest in STEM. We believe it is the job of the older generation to show the capabilities and to inspire others that STEM should be something that is fun and educational. That it can be more than just it seems.

Key Requirements

Accurate Simulation of Theremin : ThereMelo must simulate the properties of a Theremin, such as having continuous pitch control, volume control, and a sine wave sound.

- Sounds are recorded and implemented using FMod.

Gesture Detection and Input Mapping : Our software needs a way to capture the user's hand inputs to get their positionings and convert them into a continuous sound.

- All detection is captured using the Leap Motion Controller 2, and calculations are made within our Unity software.

Color-Coded Note Display : To support individuals in learning the instrument, our software should have a way to visually display what note is being played.

- The simulator incorporates a color-coding system, assigning a distinct color to each note for a quick identification, as well as labeling the note.
- To determine what notes are being played, we used FMod's built in DST to apply a Fast Fourier Transform algorithm to see what dominate frequency is being produced.

Plug and Play, User-Friendly: The software should be easy to install and launch.

- ThereMelo will be a launchable executable, along with the necessary installations for Ultraleap.
- Controller may not be included.

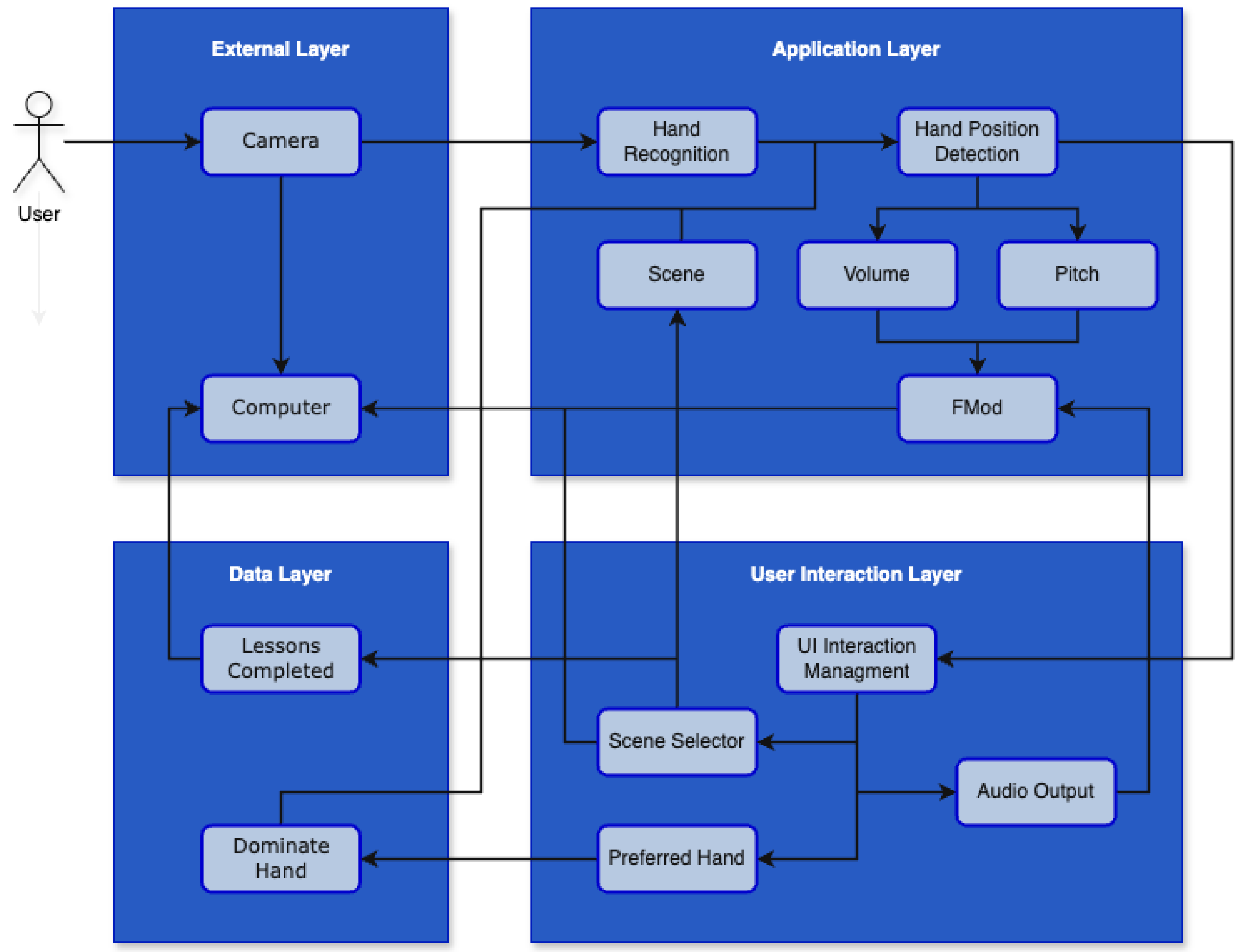
Architectural Design

For the ThereMelo, we've split the structure up into four layers. This would be the External, Application, User Interaction, and Data layers. Each layer has an input and an output with every layer.

The External Layer contains the hardware required for ThereMelo. The Camera, which should be the Leap Motion Controller 2, is connected to the user's preferred computer. Information from the camera is relayed to the computer, and the hand detector from Ultraleap.

The Application Layer is the brain of ThereMelo, containing all the calculations needed to perform the simulation. Both Hand Recognition and Hand Position Detection is done by Ultraleap's Gemini, which sends the information to ThereMelo in Unity. From there, we calculate the volume and pitch parameters with a simple magnitude calculation, inversing it for the pitch. Each parameter is sent onto FMod, which changes the audio source's properties.

The Data Layer just stores the user's preferences on the overall volume of the ThereMelo as well to contain the lessons' they accomplished. As interactive lessons are not fully implemented yet, we'll provide documentations on how to play and learn the instruments. The interactive lessons shall be implemented in a later update.



Implementation Details and Test Plan

Implementations Details :

ThereMelo leverages Unity as its core engine, with FMod integrated as the audio engine. FMod provides a Digital Audio Workstation (DAW), allowing us to implement continuous instruments, apply equalization, audio limiter, and audio parameters. For version control, we opted to use GitHub instead of Plastic SCM, allowing us to implement features within separate branches. Our initial plans were to research on a control scheme that both simulates the theremin and ensures user-friendly operation. After that was implemented, we moved onto recording sounds and implementing a suitable user interface. Along that, we added accessibility onto the software, allowing users to change how they play the ThereMelo

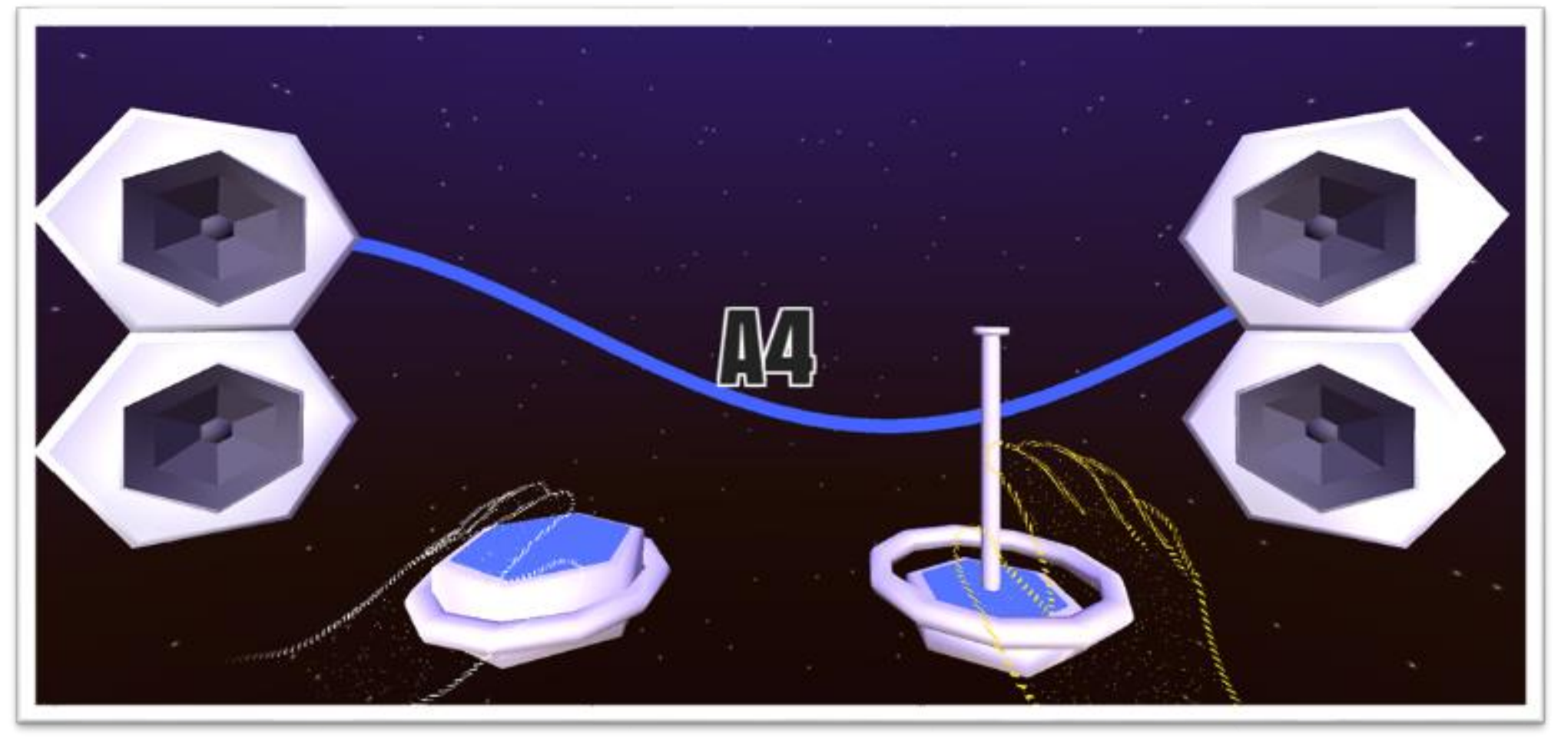


Figure 1. Playing ThereMelo

Test Plan :

Throughout the development phase, we tested our application using Visual Studio Code's integrated debugger, being hooked onto our Unity project. With each key requirements being implemented, we tested to ensure it was working properly. We also ran our applications on various platforms to ensure compatibility. This includes Windows machines, Intel-based Mac, and Silicon-based Mac. Testing on Linux environments is pending.

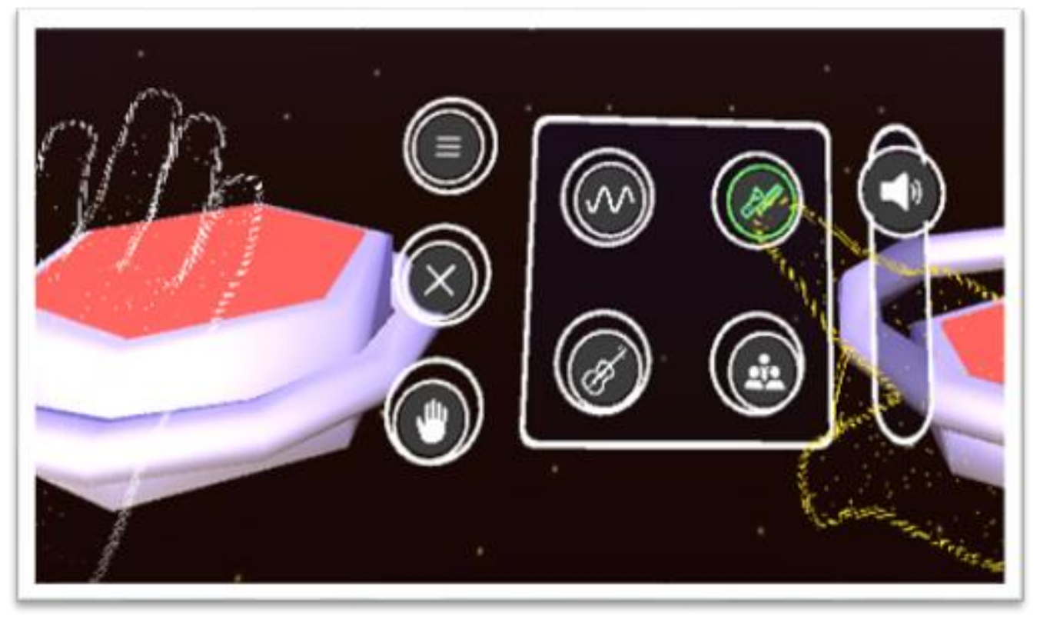


Figure 2. Selecting Trombone

Conclusions and Future Work

- ThereMelo offers the user with an accurate simulation of the theremin and other continuous-tone instruments. Our software enhances learning by displaying what note is being played, highlighted with a specific color and labeled with the note name. These implementations helps the user in developing muscle memory on which note they are playing.
- All but one planned implementations were accomplished. Interactable lessons can be implemented in the future. Future works may include implementing compatibility for Virtual Reality users, more instruments, and more lessons

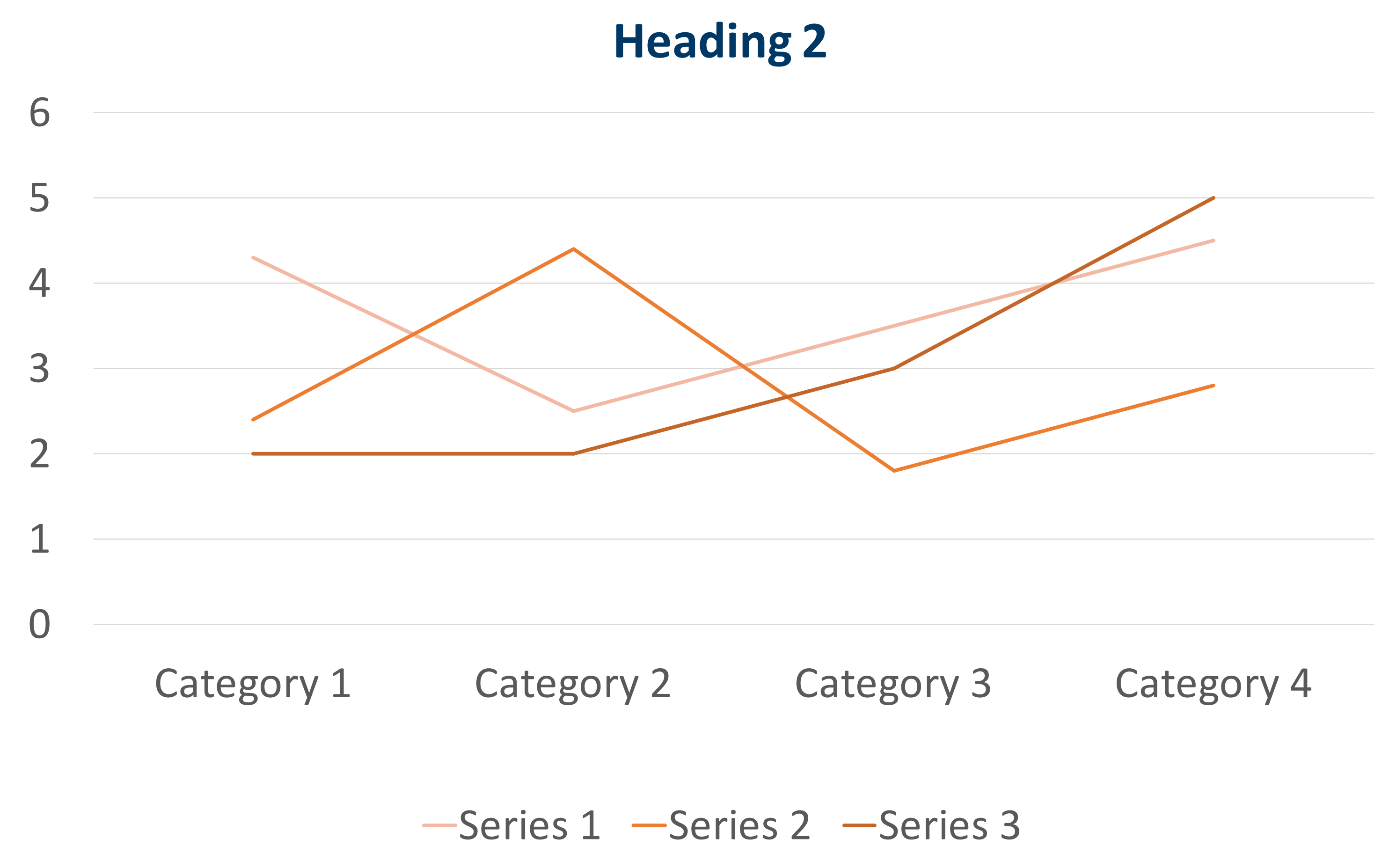
References

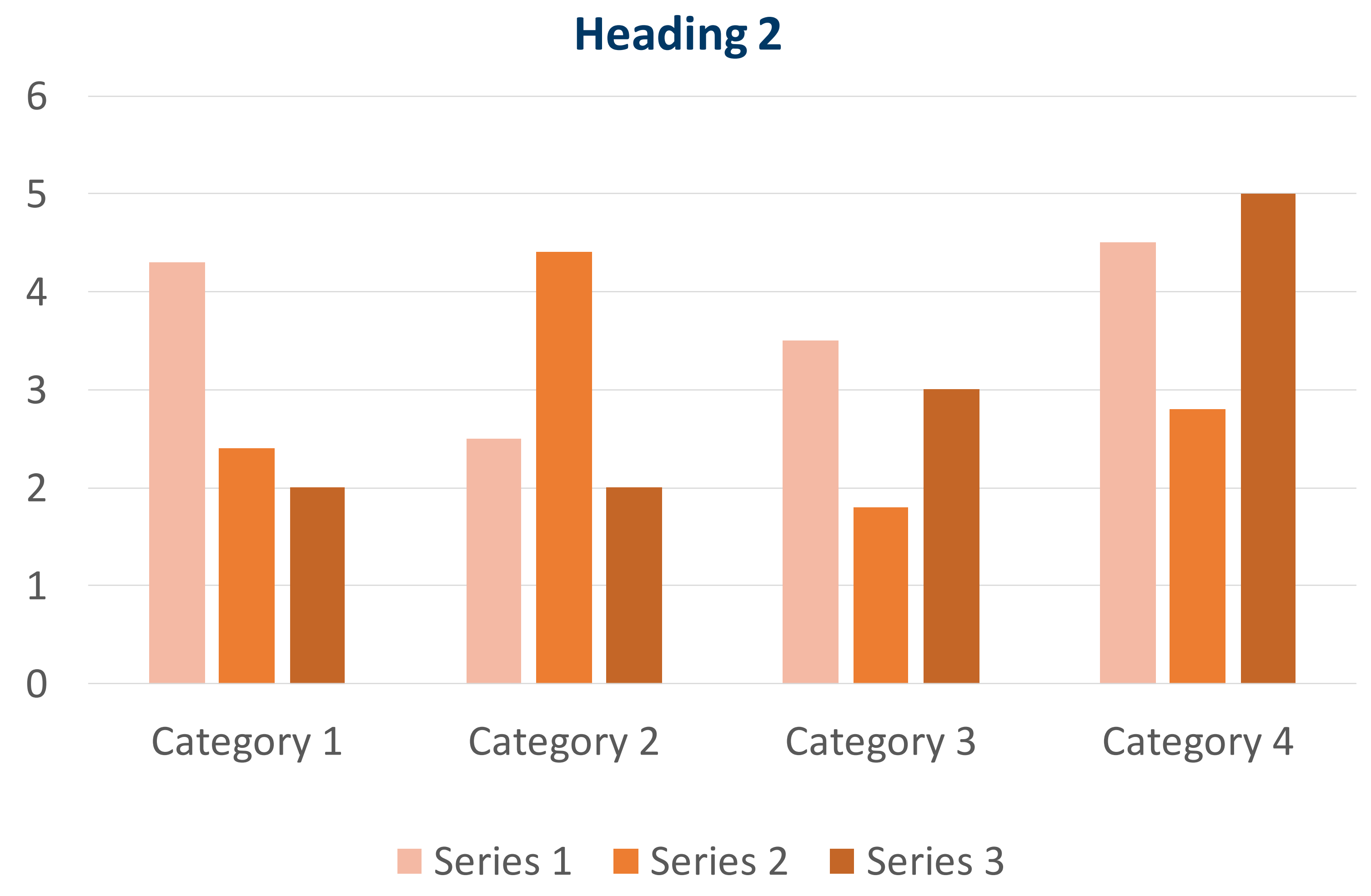
1. David Abbott. Using fmod studio with unity. 2018.
2. Daniel Bachmann, Frank Weichert, and Gerhard Rinkenauer. Evaluation of the leap motion controller as a new contact-free pointing device. Sensors, 15(1):214–233, 2014.
3. Nenad Breslauer, Irena Galí c, Mihael Kukec, and Ivan Samardží c. Leap motion sensor for natural user interface. Tehniˇcki vjesnik, 26(2):560–565, 2019.
4. Jože Guna, Grega Jakus, Matevž Pogaˇcnik, Sašo Tomažiˇc, and Jaka Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors, 14(2):3702–3720, 2014.
5. Robert McCartney, Jie Yuan, and Hans-Peter Bischof. Gesture recognition with the leap motion controller. 2015.
6. Ciarán Robinson. Game Audio with FMOD and Unity. Routledge, 2019.
7. Markus Sukoinen. Audio implementation methods in unity. 2019.
8. Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. Sensors, 13(5):6380–6393, 2013

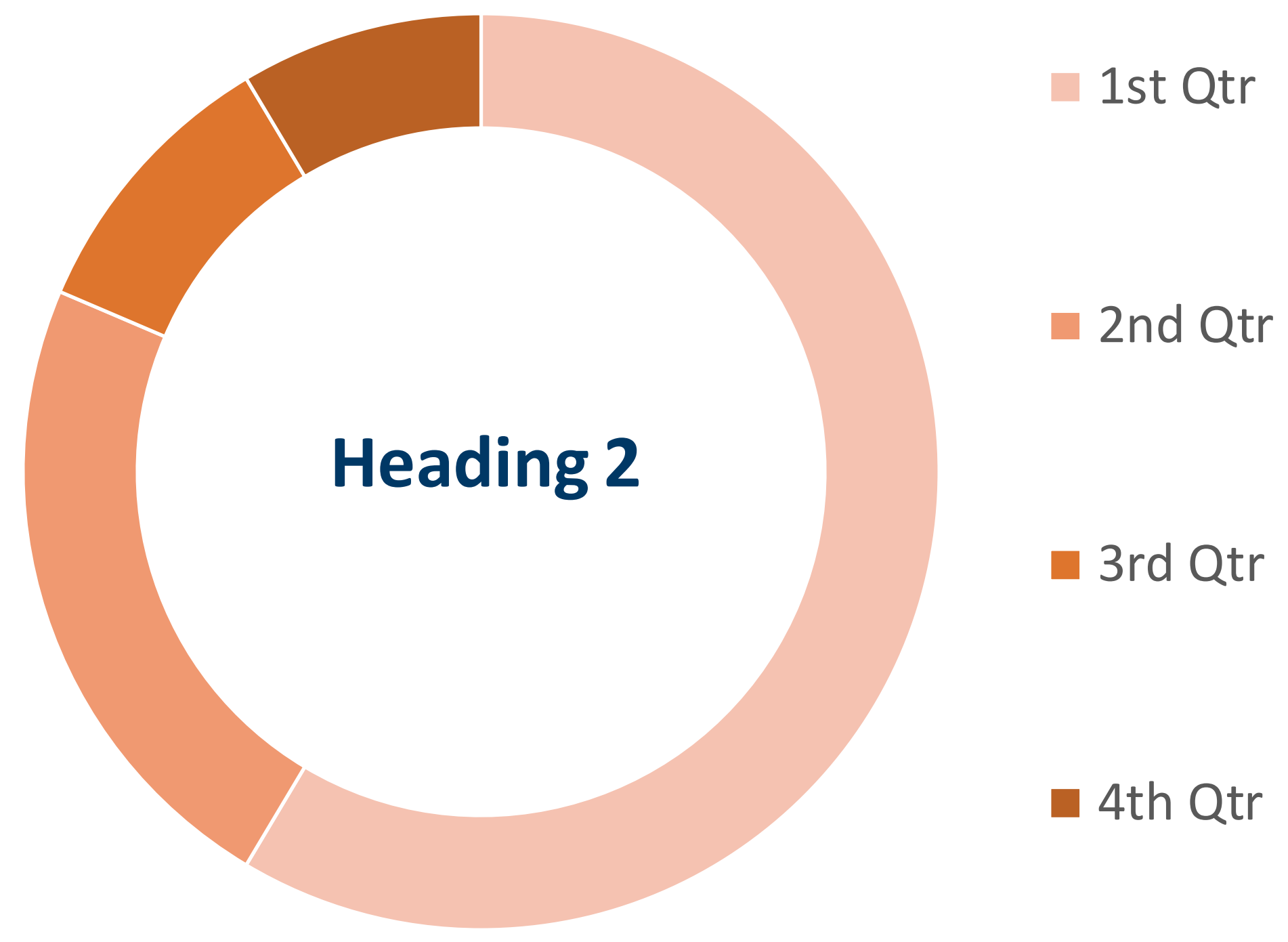
April 2024

Charts

Feel free to add your data to these charts and add them to your poster.


Heading 2
Category 1
Category 2
Category 3
Category 4
Series 1
Series 2
Series 3


Heading 2
Category 1
Category 2
Category 3
Category 4
Series 1
Series 2
Series 3


Heading 2
1st Qtr
2nd Qtr
3rd Qtr
4th Qtr