Multimodal Bounce

University of Limerick

UNIVERSITY of LIMERICK

OLLSCOIL LUIMNIGH

Erasmus Student: Mario Sánchez García
ID: 17150868

CS4358 - Interactive Multimedia

April 28, 2018

Contents

1 Introduction

Hello, I'm Mario Sánchez García (ID: 17150868), an Erasmus student originally from Spain. I'm a fourth year student who came from the University Carlos III of Madrid, where I completed my first years in the Computer Science degree.

This website is my submission for the Multimodal Bounce assignment within the Interactive Multimedia module (CS4358). The objective of this assignment is to replicate the experiments from Watanabe's paper (Watanabe & Shimojo (2001)) in order to document how including a new factor could affect the results, and how is this factor related to the cross-modal perception.

We call a perception "cross-modal" when it involves the interaction between two or more different sensory modalities, such as sound and light. In Watanabe & Shimojo (2001), the experiments are based in introducing a visual ambiguity, which solution is influenced by audiovisual interaction. A two-dimensional display is presented to the observers, where two visually identical targets are moving across each other so they can be perceived either to bounce off or to stream through each other.

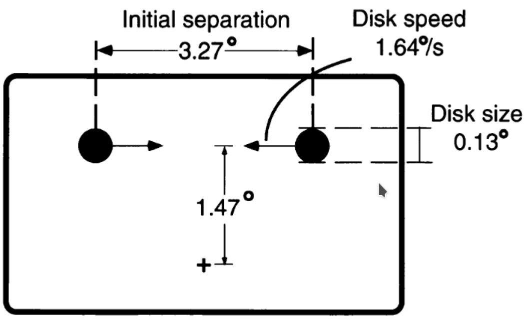

Figure 1: Configuration of the two-dimensional display (Watanabe & Shimojo (2001))

In the article it is explained how despite the ambiguous nature of the visual stimulus, observers usually show a strong bias to see how the targets stream through each other. However, several factors have been reported to increase the relative frequency to percept how the targets bounce. The most remarkable one is to play a brief sound in the moment the targets coincide in the same point (simultaneous sound), suggesting that cross-modal perception is involved.

Using this last factor, Watanabe & Shimojo (2001) prove that:

- The bounce-inducing effect is reduced when the simultaneous sound is preceded and followed by other identical sounds.
- The attenuation of the bounce-inducing effect depends on auditory context and the saliency of the simultaneous sound.
- The auditory context does not alter perceptual properties of the simultaneous sound itself.

In the present assignment, the nature of the sound was changed in order to test how it could change the bounce-induction effect. The results of the experiments show that this effect is attenuated regardless of the moment the sound is played. Nevertheless, if we compare with the experiment that plays the sound when the dots collide (Experiment 2), the attenuation is more pronounced if the sound is played 200ms before (Experiment 3) and less pronounced if the sound is played 200ms after the "colission" (Experiment 4).

2 Description

The objective of this assignment is to replicate the experiments of Watanabe & Shimojo (2001), altering some factors to document the results and explain how are these new factors related to the cross-modal perception. In the original paper it was shown that a brief sound reproduced at the moment the targets collide, increases the relative frequency of perception of a bounce.

In the present assignment we show how the kind of sound played when the targets collide affects the bounce-induction effect, changing this sound from the original ”crack” to a ”whoosh” (See the audio examples below). As the nature of this ”whoosh” is not related to a collision, playing it at the exact moment when the dots collide may not be convenient to test the full potential of this factor. Because of this, a further research has been done about the timing in which the new sound is played.

A total of 4 experiments have been realised in order to see the repercussions of the new sound:

1. The original experiment with the ”crack” sound, but using in this case the same graphical interface as the other three experiments.
2. Same conditions as the first experiment, but changing the collision sound for a ”whoosh”.
3. Same conditions as the second experiment, but the sound is played 200ms before the collision.
4. Same conditions as the second experiment, but the sound is played 200ms after the collision.

2.1 Method

The group of observers is composed of five people with an age that varies between the 20 and the 23 years. All of them were naive about the purpose that motivated the experiments and have a normal or corrected-to-normal vision and hearing.

Visual stimuli was displayed on the screen (60Hz) of a Toshiba laptop model C50-A-1GU in a room with the light off. The audio stimuli was composed of the two sounds described previously (i.e. "crack" and "whoosh") and the integrated speakers of the laptop were used to play both sounds. A full report of the design of the graphical interface (GUI) used in the experiments can be found in the section 3 of this website.

Besides the controls that compose the GUI, the visual stimuli, displayed on the screen at 100fps, was composed of two white disks of 20 pixels (called "dots" onwards), both placed in the exact centre of the screen at the start of the experiment. Immediately after the onset, one dot starts moving horizontally in a straight line towards the east side of the window, while the other dot does the same towards the west side. The speed of both dots is the same, 5 pixels per frame, and it will be constant during the whole experiment.

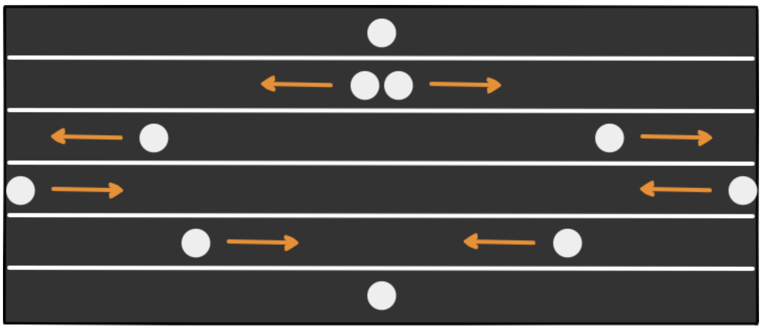

Figure 2: Trajectory of the dots across the screen

Once the dots have reached the border of the window, they will "bounce" on this edge, changing their direction 180 degrees. Moving now towards each other, they will go back to their initial position when they collide in the centre of the screen.

Observers viewed the display from an approximate distance of 1 meter and reported their perception (i.e. streaming or bouncing) either by using the mouse to press the buttons that are shown on the screen or by pressing the keys $\leftarrow$ or $\rightarrow$ of the keyboard, which have the same effect.

3 Design

In this section the design process of this website and of the graphical interface (GUI) used for the experiments is going to be explained in detail.

3.1 Website

The whole website is based in Cafe, one of the several templates provided in open access by W3-Schools. This template was already designed to be responsive, that is, it resizes the contents to make it look good in a wide set of devices. Too small screens make difficult to see the images and read the text clearly, so the minimum width of the window has been set to 700 pixels. This means that, if the window is smaller than 700px, the contents are still accessible, but the user would need to use the horizontal scroll to be able to see them.

Images are horizontally centred and have the option alt configured, so alternative information is provided if a user, for some reason, cannot view it (e.g. slow connection, error in src attribute, user using a screen reader). Anyway, a caption has been also included below each image, explaining what is the image about.

One of the requirements of the website was that it was possible to download the files related to the assignment from it. In order to do that, three links were included in the Home page and again in the top navigation bar, so they will be accessible anywhere on the website. The possibility of downloading the report in PDF has been added in case, despite the measures taken, some of the contents cannot be well visualised.

All pages of this website follow the same pattern:

- Navigation Bar. It is fixed to the top of the page and allows the navigation across the website. The "Report" and "Download" menus are expanded when the user leaves the cursor on the correspondent button, changing the background of the option selected to light grey. The bar stays at the top of the page, losing its "fixed" feature, when the window's width is smaller than 700px, so the horizontal scrolling can be enabled.

- Header. It contains the name of the student on the left, the name of the assignment in the center and the student's ID on the right. Depending of the width and height of the window, a different part of the image will be shown, filling always the width of the window.
- Text Body. The text is centred horizontally and the alignment is set as justified.
- Footer. It allows the user to go back to the top of the page by clicking on the button "To the top". Besides, it contains icons that links to the personal Twitter and LinkedIn accounts of the student and the Github repository used for the assignment. When the cursor is on it, the background is changed to black.

The report could have been included as one big page, but in order to allow the reader to quickly select what to read and improve the usability in general, it has been divided in different pages.

- Home. Here we find the background of the author, the explanation of the original experiment of Watanabe et al. and a brief description of the approach followed in this assignment. Here we also find the links, placed into icons, to download the different files of the assignment. When we place the cursor on one of this icons, it becomes bigger, its colour becomes more saturated and the text below is underlined. The user needs to have Javascript activated in the browser for the latter, but the rest of features would work as usual.
- Description. Full description of the approach followed for the experiments. The two sounds used are located inside audio tags. They try to load the sound from the first source, which is the sound file. If the browser does not support HTML audio tags, they will load the message instead, reporting the situation to the user.

```
<audio controls>
  <source src="sounds/clickon.wav" type="audio/wav">
  Your browser does not support the audio element.
</audio>
```

Figure 3: Code used to include the audio in the Description

- Design. Current website where the design of the website and the GUI is explained.
- Evaluation. The results of the experiments detailed in the Description section are explained and discussed in this section.
- Summary. Brief summary of all the information of the report.

3.2 Graphical User Interface

The design of the GUI started analysing the controls that would be necessary to add in order to perform the altered experiment. These controls are, first, being able to select one (and only one) of the sounds; and second, being able to select the latency of the sound among three possible values: 200ms before the collision, at the moment of the collision or 200ms after. Some other controls were added besides these two:

- Play/Pause. Stops the animation of the dots, so it is easier to explain how the GUI works to the observers.
- Result Input. Allow the observers to use the mouse instead of the keyboard (keys $\leftarrow$ and $\rightarrow$) to report their perception.
- Sound on/off. Same as the Play/Pause control, make it easier the explanation without the collision sound in the background.

Once we know what the GUI should include, the next step is to create a mockup. The image below shows the mockup used in the design process of our GUI, created using the online tool Balsamiq Cloud.

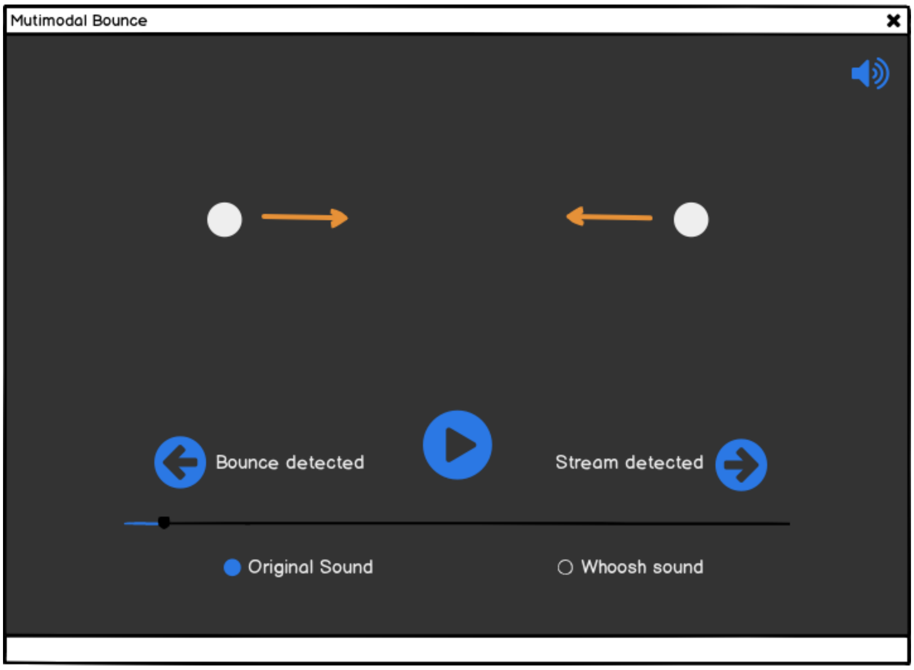

Figure 4: Code used to include the audio in the Description

The Play/Pause button is placed in the centre, taking the usual structure of a music or video player. The Result Input buttons, located on the sides, have attached a label that indicates their function. The latency is controlled with a slider, located just below these buttons. It will have a total of three values, being the latency zero (i.e. sound played when dots collide) the default.

The sound used in the experiment is set using radio buttons, which means that when one of them is selected, the other one is automatically unselected. As we did with the Result Input buttons, they also have attached a label indicating the sound enabled by each one. Finally, the Sound on/off button is located on the east top corner of the window.

Using the given code and with the help of the mockup, we can finally start developing the GUI. The language used in the implementation is Processing, an open-source programming language built on Java that uses a simplified syntax. The library ControlP5, written by Andreas Schlegel, provide a wide set of controls that we have used to build our GUI.

The library provide the functionality of the controls, but the default layout is very basic. To improve it we have used icons and images downloaded from Flaticon, loading them when the buttons are created. These buttons, however, do not have any on hover properties that alert the user that they are actually buttons. In order to create this "on hover", we load the same button image, but with a lighter background.

The function hover_toggle_buttons() is in charge of this feature. Every iteration of the draw loop it checks the property isMouseOver of all the buttons. If it has the mouse over on the current iteration, it loads the "light" image; while if it had the mouse over in the previous iteration, it loads the "dark" image again.

```
void hover_toggle_buttons(){
  // widgets with hover
  String ctrlls [] = {"soundonoff", "stop_animation", "bounce", "pass"};

  for (int i = 0; i < ctrlls.length; i++){
    // if mouse over widget
    if (cp5.isMouseOver(cp5.getController(ctrlls[i]))){
      cp5.getController(ctrlls[i]).setImages(bt_images_onhover[i]);
      mouseover_ctrller[i] = true;
    }
    // if mouse out widget
    else{
      if (mouseover_ctrller[i]){
        mouseover_ctrller[i] = false;
        cp5.getController(ctrlls[i]).setImages(bt_images_offhover[i]);
      }
```

Figure 5: Code used to apply on hover features to the buttons

Besides when the mouse is over them, the buttons that report the perception of the experiment also change to a lighter color when the related keys (← and →) are pressed. The functions keyPressed, to change to the light color, and keyReleased, to change back to the usual color, were used to perform this feature.

In the case of the Play/Pause and the Sound on/off, the widget ToogleButton was used instead of the usual buttons, as they must keep the state after the user release the mouse click. The second state of the buttons (i.e. pause and sound off) is identified with different icons. They are automatically loaded by the library because when we created the buttons, we passed both "state" images to the method.

As said before, to change the sound we finally used Radio Buttons, which only allow to activate one of the options at the same time. However, unlike the rest of buttons, they do not change the value of the associated variable when the user change the state of the button. Instead, we need to use an event listener function controlEvent, which is called only when the state of the button changes, and get the current value of the radio button to update the variable that controls which sound is played.

Applying all this changes, the final version of the GUI is as follows:

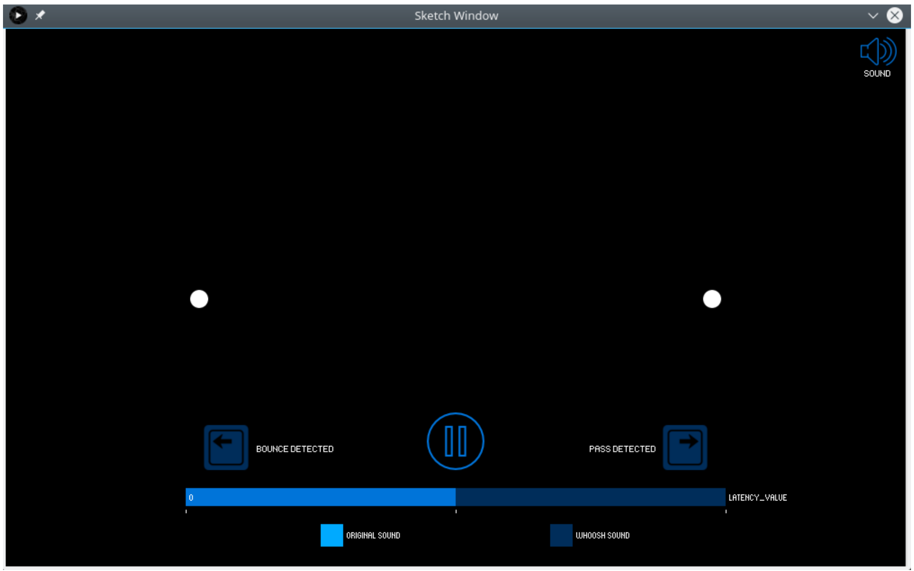

Figure 6: Final design of the GUI

4 Evaluation

4.1 Experiment 1

The results summarized in the bar chart Experiment 1 are consistent with the results in Sekuler et al. (1997), that is, when a brief sound is played at the moment the targets coincide, biases perception toward bouncing. We can say then that the designed GUI does not affect the bounce induction effect. From a total of 50 tests (5 observers and 10 tests per observer), 74% of them were perceived as a bounce.

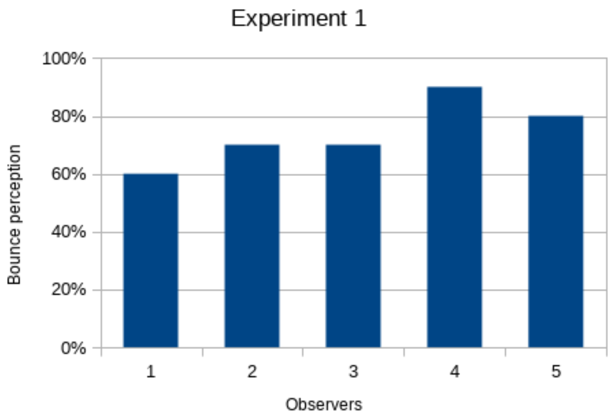

Figure 7: Results of the Experiment 1

4.2 Experiment 2

As we can see in the bar chart Experiment 2, contrary to the previous experiment, none of the observers perceived a bounce in more than 3 of the tests. In fact, only 20% from the total 50 tests were perceived as a bounce.

This pattern of results indicates that the nature of the sound does affect the bounce induction effect. In our case, the chosen sound for this experiment causes an attenuation of the bounce inducing effect of 54%, from 74% to 20% (Experiments 1 and 2).

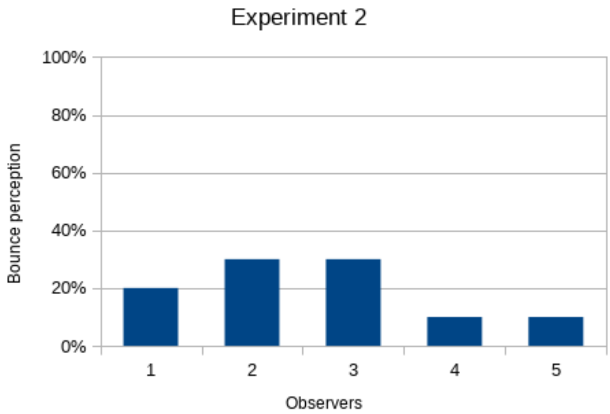

Figure 8: Results of the Experiment 2

4.3 Experiment 3

The bar chart Experiment 3 shows the results for when the "whoosh" sound is played 200ms before the dots coincide in the same point. The mean bounce perception is 18% (for 50 tests), but we can see how one of the observers have perceived many more bounces than the others, incrementing the variability of the data. If we omit the data of the observer 3 in the analysis, the mean is now 8% and the deviation from the mean is reduced to a maximum of 12%.

Despite of including or not the Observer 3 into the analysis, the results show that, as theorized in the Description section, the different nature of the sound make important the moment in which it is played. Comparing the results of Experiments 2 and 3, the attenuation of the bounce induction effect is reduced another 2%/12%, from 20% to 18%/8%.

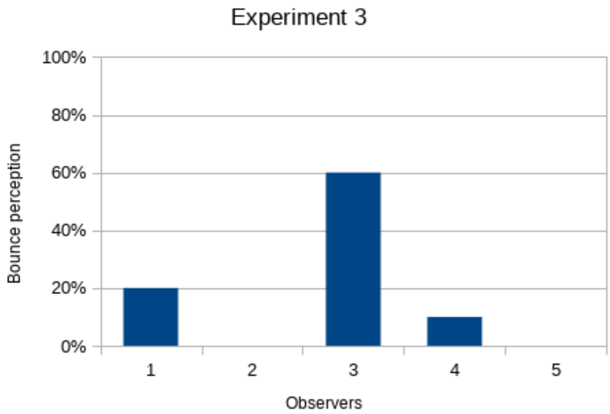

Figure 9: Results of the Experiment 3

4.4 Experiment 4

Contrary to the previous experiment, the bar chart Experiment 4 now shows the results for when the "whoosh" sound is played 200ms after the dots coincide in the same point. The data in this case has an acceptable variability and the mean bounce perception is 38%. It could be expected, as the sound is heard after the "collision", the results should be related to no-sound tests of Watanabe & Shimojo (2001), where the bounce perception was close to 0%.

Nevertheless, these unusual results show how the bounce perception is not close to 0% but even higher than when the sound was played when the dots coincided in the same point (20%, Experiment 2). No explanation has been found for the increment in the bouce perception, so a further analysis is required in order to analyse it.

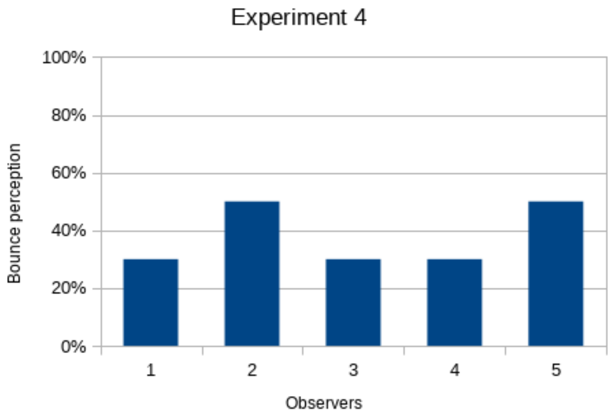

Figure 10: Results of the Experiment 4

However, we theorize that it could be related to the fact that the round of tests was done starting from the Experiment 1 and ending with the Experiment 4, in order. The observers, seeing that the Experiment 4 is clearly different to the previous two experiments (somehow similar, as two of the observers asked what was the difference between experiments 2 and 3), are induced to think that the results of this last Experiment must be different too.

5 Summary

The objective of this assignment is to replicate the experiments from Watanabe's paper (Watanabe & Shimojo (2001)) in order to document how including a new factor could affect the results. The approach chosen was to change the nature of the sound played (from a "crack" to a "whoosh") when the dots coincide in the same point and see how the timing of when this sound is played could alter the bounce-induction effect.

The experiments are displayed in a GUI designed by the student, similar to the used in the original paper, using the library ControlP5 for the language Processing.

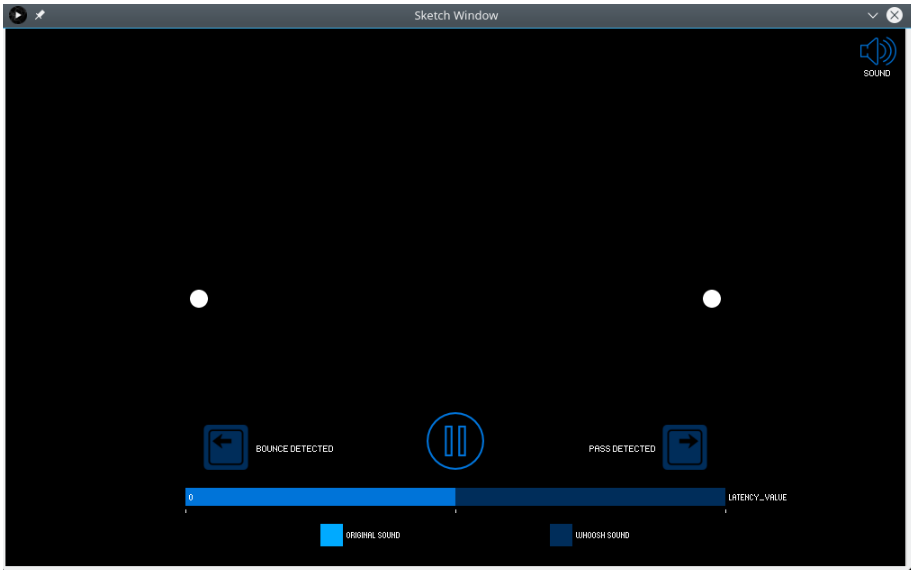

Figure 11: Final design of the GUI

A total of 4 experiments were performed to test our thesis: the original experiment with the "crack" sound, the same experiment now using the "whoosh" sound, and another two experiments with the same sound, but playing it 200ms before and after the "colission", respectively. The results of these experiments are summarized in the following bar chart, where the vertical red lines indicate the maximum values registered.

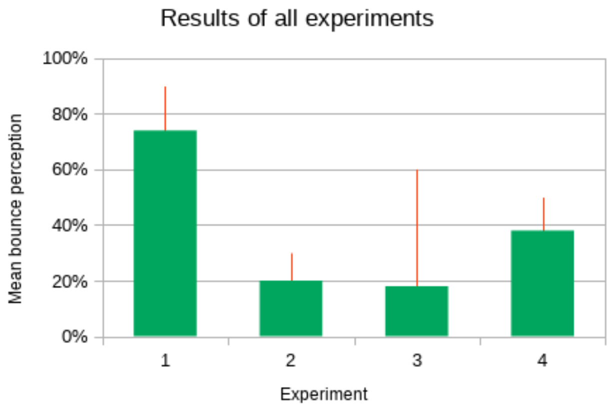

Figure 12: Results of all experiments

The results of the first experiment are consistent with Watanabe & Shimojo (2001), showing that the GUI does not affect the results. The next three experiments prove our thesis, that is, changing the nature of the sound does affect the results, causing an attenuation of the bounce induction effect (Experiment 2). Playing the sound before the "collision" attenuates even more the effect (Experiment 3, omitting extreme value of the observer 3), while playing it after makes the attenuation less effective (Experiment 4).

References

Sekuler, R., Sekuler, A. B. & Lau, R. (1997), 'Sound alters visual motion perception', *Nature* **385**, 308.
URL: *http://dx.doi.org/10.1038/385308a0*

Watanabe, K. & Shimojo, S. (2001), 'When sound affects vision: Effects of auditory grouping on visual motion perception', *Psychological Science* **12**(2), 109–116. PMID: 11340918.
URL: *https://doi.org/10.1111/1467-9280.00319*