Creative use of Technology in Elementary Education in Music Education

Birgit Maas
BSc Project Report

Creative Technology
Faculty EEMCS

Supervisor:
Dr. Ir. D. Reidsma

August 9th, 2018
Acknowledgements

I would like to express my great appreciation to Dennis Reidsma for his valuable feedback on all concepts, his admirable patience during the programming explanation and providing the guidance I needed. I would also like to offer my special thanks to Benno Spieker, because his knowledge about the subject was of great help, his enthusiastic encouragement to make the most of it and his honest and valuable feedback on all ideas. I would like to acknowledge the support of Hinke Bosch, Justin Krooneman, Laura Beunk and Gijs Bolhaar as well, for their patience when explaining and giving me the insight I needed to program the final concept. Lastly, I would like to thank my family, friends and housemates for social distractions and words of optimism.
## Abstract

1. Introduction ........................................................................................................... 2

2. State of the art........................................................................................................ 3
   2.1 Technology in elementary education ................................................................. 3
   2.2 Technology and music ....................................................................................... 4
   2.3 Acceptance of technology ............................................................................... 5
   2.4 Zoltán Kodály .................................................................................................... 5
   2.5 Expert interview ............................................................................................... 6
   2.6 Related work ...................................................................................................... 7
      2.6.1 Sketch-a-Song ............................................................................................ 7
      2.6.2 Song maker .............................................................................................. 8

3. Ideation and concept design .................................................................................. 10
   3.1 Ideation process ............................................................................................... 10
   3.2 First concept ...................................................................................................... 12
   3.3 Second concept ............................................................................................... 15
   3.4 Third concept .................................................................................................... 17
   3.5 Use cases and paper prototype ........................................................................ 20
      3.5.1 Use cases ................................................................................................... 21
      3.5.2 Paper prototype ........................................................................................ 22

4. Implementation requirements and design ............................................................ 24
   4.1 MoSCoW .......................................................................................................... 24
   4.2 Final concept ..................................................................................................... 25
      4.2.1 Visual ......................................................................................................... 25
      4.2.2 Software ................................................................................................... 25
   4.3 Design ................................................................................................................ 26

5. Validation and conclusion .................................................................................... 27
   5.1 Requirements ..................................................................................................... 27
   5.2 Test .................................................................................................................... 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1 Method</td>
<td>27</td>
</tr>
<tr>
<td>5.2.2 Results</td>
<td>28</td>
</tr>
<tr>
<td>5.3 Expert feedback</td>
<td>29</td>
</tr>
<tr>
<td>5.4 Conclusion</td>
<td>30</td>
</tr>
<tr>
<td>6. Discussion and future work</td>
<td>32</td>
</tr>
<tr>
<td>6.1 Discussion</td>
<td>32</td>
</tr>
<tr>
<td>6.2 Future work</td>
<td>32</td>
</tr>
<tr>
<td>References</td>
<td>xxxiv</td>
</tr>
<tr>
<td>Appendix A - Code</td>
<td>xxxvi</td>
</tr>
<tr>
<td>Code main class</td>
<td>xxxvi</td>
</tr>
<tr>
<td>Code of class VisualObject</td>
<td>xlix</td>
</tr>
<tr>
<td>Code of soundCollection</td>
<td>lii</td>
</tr>
<tr>
<td>Code of subclass Drawing</td>
<td>lvii</td>
</tr>
<tr>
<td>Code of sub class Egg</td>
<td>liix</td>
</tr>
<tr>
<td>Appendix B – Consent form</td>
<td>lxxi</td>
</tr>
<tr>
<td>Appendix C – Exercises for test</td>
<td>lxxiii</td>
</tr>
<tr>
<td>Appendix D – Questionnaire</td>
<td>lxxix</td>
</tr>
<tr>
<td>Appendix E – Responses on questionnaire</td>
<td>lxxxii</td>
</tr>
</tbody>
</table>
Abstract

The goal of this thesis was to implement technology in a useful way in music lessons in elementary education. Firstly, background research and an interview with an expert was done to gain information about the current situation and what would be needed to improve that situation. Then a lot of ideas were generated and improved with feedback of the expert. After a few iterations the final idea was there, the interaction was described in use cases, visualized with a paper prototype and realized with Processing 3. This prototype was tested with fellow students and feedback from them and the expert could be concluded that the product is functional and clear, but not ready yet to be implemented in a class yet.
1. Introduction

Over the last decades, the role of technology shifted from being a tool to make survival easier to a communication and entertainment platform [1]. This change of the role of technology has impact on education as well. All education is based on providing information and doing exercises to develop a certain skill. From doing exercises skills can be improved and further developed. In elementary education most teachers are general teachers, this means that they are teaching most or all subjects in a specific grade. It can lead to insecurity when they provide exercises for a subject that they are not fluent or confident in. Music is taught differently on most schools, but with the technology that is available nowadays, it can be implemented in multiple ways to fit and support the education currently given at any school.

Music education doesn’t necessarily have to involve traditional music instruments, but children can learn about music with, for example, a game [2]. Existing technology can be used in many ways to enhance music education, it depends on what the goal of the education is on how technology could best be implemented. Goals could be; children need to gain experience with making music, children need to get familiar with music theory, children need to gain the insight that music is everywhere and cannot only be produced by instruments.

It is important that children get good education and thus guidance to learn new skills. Since at elementary schools there are mostly general teachers, the implementation of technology, that can help those teachers with gaining confidence in a subject they do not dominate that well, could be useful. To start the research a main and sub question are presented and to see where the possibilities and pitfalls of this subject lie background research was done to provide answers to those questions.

Main question: Is the product that will be designed a useful addition to music education in elementary education?
Sub question: What do children need to learn from music education and what musical skill should they develop?

Those questions will be answered by the results and findings in the literature, and the evaluation of the final product. After doing the background research, the process continued with the ideation phase in which a lot of ideas about how to realize a useful product were tried and investigated. After a few iterations the final idea was ready to be realized and tested. In the final part of this report a summary about the findings of the test will be discussed, the research questions will be answered and recommendations for future research are presented.
2. State of the art

In this chapter background information about the project will be discussed. The background information was gathered by doing literature research, searching related work and doing an expert interview. This information helps to gain insight in how to develop a system that could be used during elementary music education and how it would add value to the current music lessons.

It is important to know what already has been done with technology in combination with education as well as music. To develop a system that is a useful addition to the courses, it is important that the system is accepted by the children and the teacher, also the values of the teachers are examined and based on the expert interview the Kodály method will be discussed. And finally, for the related work two music composing applications are reviewed.

2.1 Technology in elementary education

The implementation of technology in education has been done on multiple levels. Most of the technologies serve different purposes and can thus be very well used in education since technology is so versatile. Technology can be used to model difficult real-life problems to reduce costs, guarantee safety or sometimes it can be even more accurate than a physical setup [3]. Another example of the implementation of technology in education is a social assistive robot (SAR) as a teaching assistant in a kindergarten [4]. Assistance from those robots comes with a social rather than physical interaction. The advantage of SAR over computer games is that they will not result in isolation or less creativity of children.

This introduction about technology in education in general will be brief since the scope of this report will be about elementary education specifically. Technology can provide extra possibilities to teach a subject with the help of for instance digital game-based learning (DGBL), mobile gamification learning system (MGLS), software for music learning in early childhood education (SAMI) or graphical interface-based automated music composition (GBAMC). The first technology is DGBL, it has been proven to increase the learning effectiveness and experience with the system is not required [5]. This makes a DGBL system a useful tool to use in the classroom to add value to courses. In addition to those advantages, the second technology, a MGLS has been proven to have positive results on learning motivation and learning achievement [6]. The outcome of the tests with the third technology is quite similar. SAMI has a positive effect on motivation and interest, it enables creativity and the technology makes autonomous learning easier [7]. Also graphical interface-based automated music composition (GBAMC) leads to high levels of learning satisfaction and a positive learning attitude [8].
The results of implementing technology-based games are increasing learning effectiveness, improving motivation and improving engagement. Seeing that they are already used in elementary education and show positive results on reaching their purposes, it can be said that technology-based games are currently a good way to implement technology in elementary education.

2.2 Technology and music

Visualisations and modern installations contribute to possibilities to improve music education for individuals or groups. Knight, Bouliot and Cooperstock [9] developed and tested a system that is able to provide feedback on articulation and dynamics of musicians. The feedback on both articulation and dynamic was presented in three ways, namely (i) audio, (ii) visual and (iii) audio-visual. From the experiment was concluded that it is hard to give clear and unambiguous visualisations for providing feedback on articulation, while on the other hand visual feedback was more valued in the case that music students were using the system to get feedback on their dynamic. When providing feedback with the goal to improve the dynamics, audio-visual feedback improved significantly over audio-only feedback. Audio-visual did not significantly differ from visual-only feedback, thus the visual part in audio-visual feedback is dominant.

Technology can be used for individuals to practice their skills and for groups or group lessons to get better results when making music together. Zhou et al [10] discuss the implementation of MOGCLASS in an elementary school class. MOGCLASS is a multimodal collaborative music environment that improves the musical experience of children by taking away obstacles during music lessons and it provides an easier way to manage the lessons for teachers. According to Zhou et al, there are four reasons to implement this system in a class, (i) limited range of instruments, (ii) lack of time to gain basic skills with instruments, (iii) ineffective practicing due to sound pollution and (iv) management of young children. The system uses mobile devices for the children and teacher. The teacher is completely in control of who is playing together, which students can hear each other and what their tasks are. The mobile device the children use is omnipotent and easy to use, also are they able to play with headphones so that everyone can practice simultaneously without disturbing others.

Technology in music education is already used but needs improvement or further research in certain fields like the providing feedback with the help of visualisations. Other systems are well developed like the MOGCLASS and can help the teacher with class management and the children with motivation and interest in collaborating in learning music.
2.3 Acceptance of technology

Different models can be used to investigate the acceptance of technology. In this section, two of those models will be discussed. First, Davis [11] states that fundamental determinants of user acceptance are perceived usefulness and perceived ease of use. Perceived usefulness was investigated for a correlation with self-reported current usage and self-predicted future usage. Perceived ease of use was investigated for a correlation with current usage and future usage. Both times, usefulness had a greater correlation with usage behaviour than ease of use.

Second, the five-factor model (FFM) is used to take personal traits into account. Different characteristics were taken as guideline to analyse the effect of personal traits on acceptance of technology [12]. Conscientiousness has a direct, positive relation with perceived and actual system use. While neuroticism and extraversion both have a negative relation with actual system use, while neuroticism has a direct negative relation with perceived system use as well. A clear statement gathered from testing with the Unified Theory of Acceptance and Use of Technology (UTAUT) is: ‘Behavioural intention fully mediates the relationship between performance expectancy, effort expectancy and social influence and IT usage (actual and perceived)’. The different models indicate respectively that usefulness is more important than ease of use and that the teacher should not be worried or afraid to use the system.

2.4 Zoltán Kodály

Based on the expert interview with Benno Spieker (next section (2.6)), the method that Zoltán Kodály developed in the mid-20th century will be elaborated on. This method teaches children to sing in a relative system (with do-re-mi-fa-so(l)-la-si(ti)-do, instead of the letters of the alphabet) and with intervals. This method was created to make sure music education has a good base that involves quality and fun of making music. Singing is very important and to make this method work, teachers need good musicianship skills to make this method useful [13]. Music skills are not only taught through singing, but also movement and games are used. Children are among other things taught to read and write music [14].

Results of implementing the Kodály method in education are not omnipresent, but the outcomes of the research that has been done, sound promising. The aim of music education around the 1980’s seems to include musical attitudes and performance and creativity and understanding. Reasons to implement Kodaly’s method in education were to underline variety in activities, systematic progression and interaction between creativity, musical experiences and reproduction. This would result in better pupil achievements than teaching in the traditional way. Also, general teachers (without specific music education) would benefit from a simple and concrete program and from the research could be concluded that they experienced the method
positively [15]. Next to the teachers, the children benefited from this method as well as the children in the experimental group (the group in which the Kodály method was implemented) showed higher scores of musical achievements than the children in the control group [15]. Another conclusion that can be drawn from implementing this method in elementary education is that it is important to gain a great base of musical skills in the elementary education [16]. This means that a lot of time needs to be put in getting to know and learning how to work with this method. Starting to use this method can be difficult for teachers as well as for children, but if the base of the musicality of the children is not good, it will lead to problems later in their musical education, they might find it hard or unworkable to continue with this method since it keeps building on previous knowledge.

Based on those research reports can be said that implementing a new method in elementary education based on Kodály is too difficult, since teachers need to have good musical skills or training to develop them. However, it would be useful to educate teachers to be about this method, so that it can be implemented in education more easily, since it does have a positive effect on musical achievement of children.

2.5 Expert interview

This interview was conducted to get more insight in music education in elementary schools and what should be considered when a system/ an application is designed to be used in education. The expert is Benno Spieker, a teacher at ArtEZ, conservatorium Enschede. He teaches students who will become music teachers at primary or secondary education. After this introduction the, for this project, most valuable comments are written down.

Every class in elementary schools gets music education, in the first four grades the children mainly sing, dance and sometimes make music, also listening to music is to teach children about contradictions in music, for example high-low(pitch), loud-soft(volume) etc. The education is quite often connected to fairy tales or stories or situations the children can easily relate to. Every lesson is about a different musical attribute therefore each lesson has its own goal. Those goals can be found on the website of the SLO, an association for the development of curriculums for education (primary, secondary and special education). During those lessons children should be able to work individually as well as being able to work or perform together with the whole class, since this would be good for the bond within the class.

Since teaching with technology is not being taught to teachers to be, technology is currently not fully used in music education. The added value of technology should therefore be known to the teachers, because it offers a lot of possibilities in education. Teachers should see the added value (usefulness) and it should be easy to use (in class) for them. A product that
would be used in a class should be able to work with input and based on that input, give feedback for children as well as for teachers.

For the development of an application preferably a lot of people are studied, namely teachers (or people that will use the application) experts, program directors. When all parties are considered all their wishes can be used as input for the development. Another big factor that will influence the outcome of the application is the level of knowledge about music the teacher has. Quite often at elementary schools, teacher come from the PABO (general education to become an elementary school teacher) and did not have specific education in music. Therefore, those teachers can sometimes be uncertain about their skill level and become insecure about their teaching ability.

Next to the knowledge level of teachers also the ability of children and their knowledge is it important to consider when developing a product. Some people learn faster or have already a different level of knowledge to begin with, due to this fact, the application should be adaptable to how well the children perform their assignments as well as the theme, depending on the age of the child using the application.

Requirements for the application from a musical point of view would be that a traditional notation of music, so reading from left to right or presenting tones in a score, is not necessary. Children can use, for example, a drawing and elements with a certain color stand for a certain motive in music. If a certain motive occurs three times, then three elements of the same kind should be coloured. However, this traditional way of teaching music is not necessary to learn about music, it is highly expected that children do learn to read tones. This could be another requirement for the application, being able to work with traditional as well as graphical notation.

Next to the traditional and graphical notation, there are other methods to teach music, for example the method Zoltán Kodály developed. This method teaches children to sing in a relative system and with intervals. If implementing this method is achievable could be researched in literature research.

2.6 Related work

Musical applications that have been developed and could be an inspiration and example for the product that will be developed during this project. Both are music composing applications, they make it possible to create your own song quite intuitively and easily.

2.6.1 Sketch-a-Song

This application is created to make it easy and fun to compose a song. By choosing between eight different background images, a chord progression is chosen. This makes it easier to create a song that sounds good. Different instruments have different colors which can be used
to color a background drawing and thus creating a song, see Figure 1 for an example of such a drawing. It is also possible to go to the demonstration function, ‘Grooves’. Multiple grooves can be chosen from, for example, a fanfare or a woodwind quartet etc. to give the children an idea how such a group would sound and how it looks when you create something like that.

![Interface of sketch-a-song](image)

**Figure 1: Interface of sketch-a-song**

### 2.6.2 Song maker

Chrome Music Lab has multiple programs with which you can learn about music in a fun, hand-on way. Song maker is one of those programs and it is meant to let you create a song very easily. You can choose between five different type of instruments and four different types for the rhythm to compose your song. The interface of the Song maker is shown in Figure 2.
2.7 Summary of background research

Based on the literature research can be said that implementing technology-based games are increasing learning effectiveness, improving motivation and improving engagement, it can also help the teacher with class management and create more opportunities for music education, but it is important that teachers see the added value of the product to accept it. The implementation of the Kodály method is not feasible for this project, since teachers need to have good musical skills to implement this method in the class.

In the expert interview was said that children are educated in contradictions, singing, dancing and composing and each lesson has its own goal and/ or theme. It is important to ask input from a lot of people that will be involved with the product in some way, but especially teachers and children since they will actually use the product. And for the product it is not necessary to work with traditional music instruments and notation only.

The related work from section 2.6 has the convenience that the songs you create will always be in tune. Also, placing the blocks is intuitive, the higher the blocks, the higher the tone sounds. A specific advantage of sketch-a-song is that you can use multiple instruments at the same time. A drawback of both programs is that they work with a timeline from left to right. For children it can be valuable to let go of the traditional notation of music to explore with less boundaries and more creativity.
3. Ideation and concept design

In this chapter the path from requirements based on background research leading to the ideation phase and concept development will be discussed. This path starts with a visit at the conservatory of Enschede, then a concept development with its feedback and iterations will be discussed, and finally use case scenarios and a paper prototype are presented to focus on the interaction with the final product.

3.1 Ideation process

Students of the study ODM (opleiding docent muziek, study for students that want to become a music teacher) at ArtEZ, conservatory of Enschede, investigated technology that could be useful for teaching music in elementary and high schools. They learned about those tools, what their intention originally was and how they could make useful assignments with that tool. One concept that stood out to me came from Nienke (third year ODM student) and was about creating sound stories with the use of an Ableton Push. The Ableton Push is a tool that can be used to create a song from scratch by making your own melody and beat. The different sounds can be assigned to different buttons and when a beat is playing, you can easily play a melody with the buttons, see Figure 3 for the interface of the Ableton Push. The concept of sound stories as Nienke mentioned was to add pictures to the buttons of the Push, the pictures would tell a situation or short story and then the sound that was added to the button on which the picture was placed and would support or amplify the meaning of that picture as can be seen in Figure 4. With the placement of those pictures and accompanying sounds a story would be created.

This idea was meant for children in the third or fourth grade of elementary school. To recreate such an idea, firstly more information was needed about what children must learn in their music education. The SLO, Dutch association for guidelines in education, states a few requirements and goals of lessons that children need to learn. In the case of music education in the third and fourth grade learning about contradictions (high-low, loud-soft, fast-slow, long-short) in music and composing are the two goals that will be focused on in this project.

Subsequently the concept development phase started, different goals could be reached or taught through different assignments. For those assignments it is also important that they do not interfere with the creativity of the teachers. It is not necessary that complete plans for lessons are sketched out, the product should be simple enough that with a bit of explanation, the teachers have the possibility to create a lesson according to their own ideas. Although, it could be helpful to create examples of lessons to help the teachers start using the product,
since they can see the possibilities with the product. The content of the assignments that can be used in the lessons is listed below.

1. When pictures with sounds are given children must come up with a story that fits the pictures and sounds.
2. When a story is given children must complete the assignment by collecting their own sounds and corresponding drawings to support the story.
3. Think of their own story (with or without a starting image) and combine that with their own sounds and pictures to complete the assignment.
4. Contradictions in music can be made or played with technology or traditional instruments or contradictions can be recognized in a short piece of music that is performed.
5. Children can learn about the timbre (tone color) of instruments by coupling color or the intensity of the color to different sounds.
6. Sounds that are given to support a story can be recreated with traditional instruments.

So, the build-up of the assignments is important, children learn to give meaning (sound) to a drawing (situation) or the other way around before they must create a whole story by themselves. The assignments as described above fit the goals of the SLO, they all come back in the concept and its iterations that will be discussed in the following part.
3.2 First concept

The first concept has three main assignments, starting with six small assignments to introduce the kind of exercises children will encounter, then is moved on to the contradictions (high-low, loud-soft, fast-slow) and finally sound and emotion are combined. In this concept the focus lies on what sub assignments are needed to get to combining sound and emotion to give it meaning.

For the first six sub assignments either (i) a situation, (ii) an emotion, (iii) an animal and (iv) multiple images is given and with guiding questions it will lead to finding a fitting sound. For (v) a sound you will create a story and with (vi) a color you can end up with a fitting sound and story. These exercises are shown in Figure 5-10. With these sub assignments, children are guided to let go of the traditional way to present music and learn how sound can be meaningful in different situations.
Hereafter it is time to start with the second main assignment, the contradictions. Those exercises are based on a piece of music that is played and afterwards children should indicate if that piece was played with mainly high or low tones by clicking on the right apple in the case of Figure 11. This concept works the same for loudness and tempo as shown in Figure 12 and Figure 13 respectively.
For the last main assignment children are not guided with in-between steps but asked directly to combine a description with a certain sound or the other way around as can be seen in Figure 14 and Figure 15.

Talking these assignments through with Benno led to the insight that especially the second assignment about contradictions needed more attention, since it now is quite vague and difficult to use for teaching. Also, creating your own story as final assignment should be worked out, since that could be a valuable step.

Next to this, for the second concept not only the kind of assignments but also the content of them became important. It would be convenient to connect the assignments to concepts or situations the children know about. Some examples, Christmas, Easter or holidays. Different elements can be created for each theme and therefore make the
assignments more suiting to the current situation of children. Those elements (respectively balls for in a tree, eggs around a house or ice creams on the beach) could be used to teach the contradictions by increasing them in size (loudness), moving them up and down (pitch), changing them in color (timbre). Creating bigger elements, for example a Christmas tree with multiple balls (of different sizes, heights and colors) would be a small instrument on its own. If some of those bigger elements are combined in one big picture and story is told about the elements, you are creating your own sound story.

For this final version also, background music could be added to the paths between images, and it should be possible to show the final creation by itself or without sound if it is performed live in or with the class. When children in the class are divided in different groups (e.g. clapping or knocking on the table) a story can be performed with the whole class.

Lastly, it would be helpful for the final assignment if in sub assignments differences in behaviour of, for example, animals are shown with differences in sound. In this way children get concrete examples about the different sounds one instrument can make and learn how to give those sounds meaning, which can eventually be used in the final assignment.

3.3 Second concept

So, for the actual assignments, four different types were developed with more guidance in these than in the previous concept. You start with either color, emotion, animal/character or instrument and by answering questions you give meaning to the color, emotion, animal/character or instrument. An example is given for color and is shown in Figure 16. This is a guided way to give meaning to color, emotion, etc. so this could be used in the final assignment.

Then assignments about pitch and tempo would be made with the help of images as can be seen in Figure 17. For the pitch assignment for every played note a different element in a drawing lights up and for testing children must pick the right element when their teachers play a note. For the tempo assignment two short pieces of music are played and the order in which they are played (first slow, then fast or vice versa) in the same order the images need to be pressed.

For the third assignment an animal or a character would be worked out, see Figure 18. With guiding questions, children will implement what they have learned about pitch and sound, for example, coming up with differences in behaviour and how to let that hear in music.

For the final assignment different drawings (created in previous assignments or new ones) in a specific order (thus a path needs to be drawn) would tell a story and sound would support that as can be seen in Figure 19.
Also, adding sound to the paths when considering the build-up and storyline will help with adding the ambiance of the story. It should be able to easily take the sounds from a database pre-filled and/or that can be filled by the children. To perform this in the class it should be possible to mute the sound so that the class or a group can collaborate on a story that has been made.

During this phase in ideation also the idea to use an interactive table came up. In this case the final program would consist of two parts, a story part and a sound part. In the story part it would be easy for children to create their story with drawings, while in the sound part, sounds can be selected, loaded and chosen to combine with an element in the story space. The advantage of an interactive table is the fact that interaction on the story part would be quite easy. Next to the interactive table it would be convenient to be able to show the result on a digital board as well, especially when it needs to be performed in or with the class.

Talking about this second concept with Benno, the conclusion was that all the subquestions in the first assignment are not necessary. Teachers should have the opportunity to
teach children the broad concept of music without all those guided questions to leave more room for creativity. Also, the assignment about the contradictions should be worked out better. Furthermore, the step between the last two assignments is quite big, so an intermediate step was added in the third concept.

3.4 Third concept

With this advice in mind the development of the third concept started. The sub assignments again would be stepping stones to the final assignment, but with less guidance. Starting with an image of an animal/character/emotion and a few sound files to let children pick the instrument they think fits the image the best.

Then a piece of music is played, and children must guess the instrument and emotion. This will be done for a color as well, which emotion, character, instrument fits best in their opinion for that specific color. Those exercises are shown in Figure 20. To practise with instruments an emotion is shown and played with a certain instrument, but now also a different emotion must be played with the same instrument.

The contradictions are part of the sub assignments as well, still in the same way as can be seen in Figure 21. Since this assignment needed more attention a few examples of how this concept can be taught are given.

1) In the assignment that teaches about pitch, the decoration in the Christmas tree corresponds to different tones (see Figure 21). First the highest ball lights up and a high tone sounds, then the lowest ball lights up and a low tone is played. After this ‘show’ round the children can practise listening to pitch and hear the difference. To make it more complicated different number of balls or smaller distances between the notes could be added.

2) A tree with two balls of different colors correspond to different timbres. First one ball with a certain color lights up and a certain tone sounds, then the second ball with a different color lights up and a different tone is played. After this, the children can practise to connect colors to certain sounds. Adding more balls with other colors or different intensities of colors can make this assignment more difficult.

3) The tree with two balls of different sizes correspond to difference in volume. First the big ball lights up and a loud tone is played. Then the small ball lights up and a softer tone is played. Children can now practise to listen carefully to the volume of a tone that is played. Making the difference between soft and loud smaller or add more balls will make this assignment more challenging.
4) Children will have to record a high and a low sound, then load it into the program and connect the high sound to the highest ball and the low sound to the lowest ball. This will be checked by the program.

5) Children will have to record different kind of sounds to connect them to different colors. Since this is quite subjective, it will be hard to let it be checked by the program so checking done by the teacher would be necessary.

6) Children pick an instrument, make a soft and a loud noise on it and record both. The recordings can be loaded into the program and connected to a big and a small ball. This will be checked by the program.

7) Children can record sounds and connect them to balls that they have created themselves so that they match the sounds they recorded. Now their tree is a little instrument that they can present or save for use in a later assignment.

Next to practicing pitch, volume and timbre, tempo is important too. The following exercises are examples of how this concept could be taught.

1) A few tones are played fast after each other, the fast animal lights up. A few tones are played slowly after each other, the slow animal lights up. After this the children can practise this by pressing the right animal after they heard the few tones.

2) First a few tones are play fast after each other, then a few tones are played slowly after each other. The child must press first the image of the fast animal and then the image of the slow animal.

3) Two times a few tones are played, first they sound fast, then they sound slower than the first ones. The children must press the slow animal, since the last piece was slower than the first one.

After these examples will be continued with discussing the remaining assignments. Another image is shown, in this case an elephant. Children can build up these images by choosing an instrument, an emotion they want to express and try that with that instrument. Also, specific actions are asked to be performed. The reason for still having guidance in this assignment, is because it is quite difficult. Instead of an image also a picture with a situation can be the start case and guiding questions will create music for that picture.

The final assignment has not changed either, there are a few drawings and a path and story should be created by the children. Those last three assignments are shown in Figure 22. There is a lot of variation possible for these assignments, so Figure 23 shows a different possibility for the situation assignment and Figure 24 and Figure 25 show alternative possibilities for the final assignment.
Figure 20: Combine/ think of a sound that fits the image/color or feeling that fits a given sound

Figure 21: Think of a sound that fits the emotion, practise with contradictions in music
Talking about this third concept with Benno led to the insight that creativity of the teachers still would be restricted with all those sub assignments. The decision was made that only one interface would be made and that all the assignments should be doable with that interface without a list of required assignments that would restrict the creativity of the teachers.

In the next concept an interface should be designed with which all those sub assignments can be performed, and teachers and children can work with without a big learning curve.

### 3.5 Use cases and paper prototype

After those iterations the idea of the final concept became more and more clear, therefore use cases and a paper prototype design was used to get a better idea of what exactly would happen when using the product and what was needed to realize it.
3.5.1 Use cases

In use cases the behaviour of the system is described when it responses to the input it gets. So, in the next part the interaction with the system for all different kind of assignments is described.

3.5.1.1 Meaning of different colors

Objects can be created in a simple way, when those objects are given a color, a tone is played. All colors have a different instrument attached to them, in this way, children can learn about timbre. Next to this fixed combination of color and instrument it should also be possible for children to make and record their own sounds and connect them to certain colors. This kind of assignment is instructive because, this shows a child that color is not an abstract concept and can help to give meaning to sound. The added value of technology in this case, is that the use of sound as well as saving the assignment (for later use) is easier than when done on paper.

3.5.1.2 Learn about pitch and volume

Moving an object around in the interface that will be made will influence the pitch. A higher placed object will produce a higher tone than a lower placed object. The same can be done for volume, when an object is made larger, it produces a louder tone than when the object is made smaller. Children have now the possibility to learn about pitch and volume by implementing it in the interface and thus starting to make music. The added value of technology in this case is that when the creation is saved it can be used in the final assignment.

3.5.1.3 Give meaning to a drawing/situation

An image of an elephant and a list of instruments is given. The child must pick an instrument that (s)he thinks fits the image best. They start with playing the sound of a happy elephant with their instrument of choice. Then they must express an angry elephant with the same instrument. Different situations can be described to extend this assignment. This is educative because children learn how to express multiple emotions with the same instrument, it asks for creativity. The added value of technology in this case is that all answers are saved and can therefore be compared easily as well as the possibility to implement this drawing with a certain sound in the final assignment.

3.5.1.4 Create a story (almost) from scratch

A few images are given, the children can come up with a path to connect the images and the story behind it. Sounds will have to be added to the images and paths to complete the sound story. When it is finished, the story can be performed in the class. This is still a bit guided, it
can also be done without any given images, thus the children will have to come up with a story, images and sounds themselves. Those can be picked from a database, or elements from earlier assignments can be reused, or new drawings or recordings are made and used. This is significant because children can be creative with the story. All elements that they could use to make an interesting story have been taught in the sub assignments. The added value of technology in this case is that it is easy to create new or add previous drawings and sounds.

3.5.2 Paper prototype

The interaction of the system was based on those user cases and first realized in a paper prototype. For this paper prototype the application Marvel – Design and build apps was used, the interface is shown in Figure 26. With this application it is possible to make pictures of drawings and order them. This ordering can be done by clicking on a specific spot on the picture and then say which image should be shown next. In this way you can make interactive drawings in quite an easy way. The prototype consists of two parts, the first one was about adding images, drawing a path and adding sound as can be seen in Figure 28 and Figure 27, the second part was about creating elements in the screen and give an idea of the interaction for moving, resizing and changing color as can be seen in Figure 29 and Figure 30. To realize this prototype, drawings were made and to get a better idea of the interaction an app was used to combine those drawings. In the Marvel App, those drawings can be linked by choosing which drawing is presented next based on which spot the current drawing is clicked. The interaction of changing size and color was done with a slide bar, this was reported to be unclear. Scaling would be easier to zoom with your fingers (or mouse wheel) and changing color should just be a menu with different boxes for the colors rather than on a scale.
Figure 26: Interface of the Marvel app

Figure 27: One of the last screens of the first interface

Figure 28: First screen of the first interface

Figure 29: First screen of the second interface

Figure 30: Last screen of the second interface
4. Implementation requirements and design

In this chapter, first the requirements based on the research in the previous two chapters are stated with the MoSCoW-method. Hereafter the visual and software part of the final program are discussed and finally the design is described and shown.

4.1 MoSCoW

Based on the paper prototype, user cases and concept iterations of the previous chapter, a list of the requirements for the final design was made, this is shown in Table 1. This was done with the MoSCoW-method, this method is helpful to get a clear overview of the requirements that must, should, could and won’t be implemented in the system. The elements that must be implemented are the minimum to make the program work. The elements that should be implemented are very much desired, but without them the system will still work. The elements that could be implemented are in case there is time left. The elements that won’t be implemented will not be implemented in this system but might be interesting for any future work.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Must</th>
<th>Should</th>
<th>Could</th>
<th>Won’t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface for teachers</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples for teachers</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record own sounds (on/off, decided by teacher)</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Adding new objects* to canvas</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Adding own drawings to database</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Adding own recordings to database</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Change size, location, color of objects*</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Play sound of object* when something is changed</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Sound can be assigned to object* and path</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Assignments are presented on screen</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Recording of own music</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Database of drawings</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Database of sounds</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create paths between images</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Path can consist of multiple lines (a line and color for each instrument)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Build mode</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Present mode (with or without sound)</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

24
4.2 Final concept

For the actual concept that was realized all the sub assignments were left out and the assignment where you can create your own story including sound and images was focussed on.

4.2.1 Visual

The program would visually consist of two parts, a drawing canvas and a menu area. On the drawing canvas it is possible to place objects where sound of instruments can be assigned to. In the menu area is the possibility to pick a drawing to place on the canvas and to select sounds to combine with either an object or drawing.

There are two different categories of sound in the menu area namely color and sound. Each color can be combined with an object and stands for a different instrument, each line on the canvas stands for a different pitch and the volume is influenced by the size of the object. The sounds that can be assigned to the drawings are action like sounds as walking, knocking, blowing of the wind.

When a story of drawings and objects is made on the canvas area and the sounds are added, thus all elements are there to tell your story it is possible to continue to the play mode. First numbers should be given to all elements that are part of the story and when ‘done’ is pressed all elements will be played when a key is pressed. This is convenient, because when the story is being told you can very accurately play the right sound at the right moment.

4.2.2 Software

The interface was realized in Processing 3, which is a convenient tool for programming a visual environment. The complete program can be found in appendix A, but the structure of the program will be briefly discussed here. In the main program the whole interface is drawn, images are loaded, the mouse functions are declared and lastly booleans to check for overlap are written down. Then there is a class for elements in general communicating location, size and sound file to the subclasses. There is a sub class for the objects that can be placed in the drawing canvas and a sub class for the drawings. Both have specific information about how
they should be drawn, coloured and played. The elements can be played because they have either a color or overlay image which is assigned to a specific sound (file). The last class that has been made is for the sound files specifically, here all sounds are declared, and their location specified. This has been done to make sure all those files were only called once instead of every time a new element was created.

4.3 Design

The design of the final program is shown in Figure 31 and Figure 32 and has the drawing canvas on the left and the menu area on the right. The program is designed in such a way, that visual feedback is given when either an object or a drawing is selected. In the first case, the menu tab ‘Kleur’ is black in the second case the menu tab ‘Geluid’ is coloured black, this is done because sound files that are indicated with a color can only be assigned to an object (egg in Figure 32) and sound files that are indicated with actions can only be assigned to drawings.

![Figure 31: Starting interface of system](image1)

![Figure 32: Presenting interface of system](image2)
5. Validation and conclusion

In this chapter the results of the evaluation of the program will be discussed. This evaluation will start with looking back on the requirements that were stated in chapter four to check which were met. Also, a test with students was done to get opinions about the clarity and functionality of the program, so the method and results of this test are discussed. Hereafter a discussion with expert Benno Spieker was done and finally a conclusion about the program could be drawn.

5.1 Requirements

The requirements as stated at the beginning of chapter four will be discussed here to see which are met and which are not.

All but one of the must requirements are met, unfortunately there is no possibility to let background music play or add sound to the path. The main reason for this is that instead of creating a path between elements, the elements are numbered and played in that specific order. Only one of the should requirements is met, namely when an object or a drawing is changed/adapted, it will play its new sound so that you get direct feedback on how an element has changed. For the could requirements two are sort of met, there are multiple examples for teachers, not very extensive so the examples could function as inspiration rather than examples and further on in the program explanation on how to continue numbering and presenting is written down. The requirements that are not met will be discussed in future work.

5.2 Test

When enough of the requirements were met, so the program was functioning and could be tested, students were asked to participate in an experiment to evaluate the program. In this part is explained how and why the program was tested as it was, and the results of the testing phase are discussed.

5.2.1 Method

To test the clarity and functionality of the program eight students were asked to perform a series of exercises and they had to fill out a short questionnaire afterwards. Before the execution of the exercises they were told very shortly about the project. Hereafter they read and signed a consent form because they would be filmed when they were doing the exercises. To keep the knowledge the participants had beforehand as constant as possible, a short-written introduction about the project and program was given with the comment that they could
ask questions at any time during the experiment. When they were done reading and ready to start, the webcam and screen recorder were started, and the first assignment was given.

The first assignment was meant as an easy starter to get to know the program. The second assignment had two versions, so half of the group had to support a given story with sounds, while the other half had to come up with a story based on given elements and sounds. For the last assignment the participants were asked to think of a story and support that with elements with sound. When they were finished with the exercises a questionnaire with closed and open question had to be filled out to share their opinion and experiences with the program.

Testing in this way was done, because it was important to know how difficult the program was to use with the little given information beforehand. The written explanation and standard questionnaire were designed to keep all input about the program and evaluation the same. The content of the exercises was intended to build up to being able to use the program independently. The complete form containing informed consent form, introduction and exercises is presented in appendix B and C respectively, the questionnaire is presented in appendix D.

5.2.2 Results

During the experiment the participants were not only taped, but watched as well, the difficulties that were noticed or directly told are summarized in Table 2.

<table>
<thead>
<tr>
<th>Action/ behaviour</th>
<th>Number of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressed play too early</td>
<td>3</td>
</tr>
<tr>
<td>Explanation back</td>
<td>2</td>
</tr>
<tr>
<td>Explanation done</td>
<td>2</td>
</tr>
<tr>
<td>Explanation play</td>
<td>3</td>
</tr>
<tr>
<td>Explanation change size</td>
<td>5</td>
</tr>
<tr>
<td>Explanation image – sound</td>
<td>3</td>
</tr>
<tr>
<td>Explanation object – sound</td>
<td>2</td>
</tr>
<tr>
<td>Explanation how to create an element</td>
<td>2</td>
</tr>
<tr>
<td>Explanation deleting elements</td>
<td>3</td>
</tr>
<tr>
<td>Participant did not tell story with sounds</td>
<td>3</td>
</tr>
<tr>
<td>Participant did make a horizontal time line</td>
<td>5</td>
</tr>
<tr>
<td>Participant did not make elements different size (last assignment)</td>
<td>6</td>
</tr>
<tr>
<td>Participant missed back option</td>
<td>1</td>
</tr>
<tr>
<td>Participant missed replay option</td>
<td>3</td>
</tr>
<tr>
<td>Participant missed selecting multiple elements</td>
<td>1</td>
</tr>
<tr>
<td>Participant missed feedback when clicking buttons</td>
<td>1</td>
</tr>
<tr>
<td>Participant missed option to save creation</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Number of participants who had difficulties with certain actions

In this table can be seen that changing size needed explicit explanation and was not implemented in the final assignment, though that was what was hoped for. Also, the drawing canvas should act as a blank page where drawings can be placed randomly, but most participants made a story with a horizontal timeline anyways. In Figure 33 and Figure 34 can two results of the final assignment be seen.

![Figure 33: Creation that was made by one of the test participants](image)

![Figure 34: Creation that was made by one of the test participants](image)

Moving on to the questionnaire to analyse the results of the participants' opinions about the program a few general remarks can be made. All but two participants found the first assignment (creating a bird) the most difficult. And everybody thought that the program was clearer after the last assignment was made. From the comments that were given to the open questions ‘How did you experience working with the program?’ and ‘Do you have any recommendations, advice or is something missing?’ the opinions differ. Some say the program is clear (after more assignments) and intuitive, others say it quite difficult and unclear.

General remarks to improve the system are adding a delete, replay, redo option, also playing sound automatically, making it easier to connect sound to elements and making more consistent and clear buttons were suggestions that stood out the most. The complete table with questionnaire responses can be found in appendix E.

5.3 Expert feedback

This feedback session was done with Benno Spieker.

There were actions possible in the program that were much appreciated, like the random placement of drawings, playing multiple elements at the same time, scrolling that changes the
size of objects (is intuitive), actively clicking when a sound should be played is good interaction and the not left to right notation (timeline).

But also points of improvement were talked about. Starting with the interactions and actions that are lacking but are necessary before implementation in education. Lacking interactions are:

- a replay option,
- selecting multiple elements and assigning the same sound to them or moving the selection around,
- an option to play the creation automatically,
- option to play while working on assignment to know how the creation sounds so far,
- feedback on which element is currently played,
- option to give elements multiple numbers so they can be played multiple times,
- colouring images digitally (as in sketch-a-song),
- an option to combine multiple creations of different children,
- possibility to actively stop a sound,
- make groups of a few elements and
- even have the possibility to give each group its own tempi.

Next to these missing interactions, the content can be improved by:

- finding better wind and yell samples and
- the colors of the objects should be brighter than the current pastel colors.

Also, the creativity of adding own drawings and music is missing and therefore the program is less valuable and innovative, a background image is missing as well, this could help with the background music/ ambiance of the story. Lastly the note lines are not necessary, children will figure out high and low when interacting with the program.

5.4 Conclusion

Answers to research questions. Main question: Is the product that will be designed a useful addition to music education in elementary education?

Sub question: What do children need to learn from music education and what musical skill should they develop?

As all participants indicated, the program was sufficient to do the given assignments with, but also quite some recommendations to improve the program came up while they were interacting with the program as well as later in the questionnaire. Given these quite basic and very reasonable suggestions the program is not ready for use in elementary education.

Next to this conclusion based on the test, the scope of the expert review wasn’t much different. The conclusion from this interview was that there are good elements, but many more
that need to be improved or implemented. So shortly, the product was sufficient to test with fellow students, but is not ready to be implemented in elementary education yet.
6. Discussion and future work

In this last chapter the results of this research will be discussed, and suggestions will be given for when this research is continued.

6.1 Discussion

Implementing technology in elementary education is an opportunity that is presented with the technology that is currently available, this project was not intended to solve a specific problem, but rather investigate the possibilities.

For this specific project the evaluation was done with students (age 19-25) which is not the perfect group to test clarity and functionality of the program with, since the program was designed for elementary school children (age 7-9).

6.2 Future work

In any case of continuation with this project, the creativity and freedom of teachers and children (direct users of the program) must be considered. Teachers should get the feeling that they can give any assignment and children should be able to work with the system individually, both without constraint of the technology. When children can work individually, the teachers do not need to help or check on the children all the time so that children could be more creative.

Currently the images and sounds are hard coded in the program, this means that the images and sounds that can be used for the exercises are fixed. This should be changed so that children have more choice in drawings and sound, can create more unique sound stories and most important, that children can collect their own sounds and make their own drawings and use that to make their exercises.

Also, implementing the requirements from chapter four to improve functionality should be considered, and the interaction possibilities that were missing as discussed in the previous chapter during the expert interview with Benno must be dealt with.

Next to those big additions a few more specific actions should be implemented specially to make the product better to use in multiple grades rather than the third and fourth only. Expanding its usefulness can be done by adding on/off switches at the teachers’ application for certain actions and changing the interface of the program. For example, going through the classroom or school to let children collect their own sounds and load them into the database, could be too difficult for children in a lower grade. Therefore, would it be better to let the teacher decide what children can do and see. Also, changing of theme should be an option, since children of the age of 7 like different things than children that are a few years older. So, based
on the skill level of the children and their age, should the teacher be able to (dis)able certain functions and set different themes.
References


Appendix A - Code

Code main class

//Birgit Maas - s1542672
//Graduation Project - Creative use of technology in music education on an elementary school
//July 2nd, 2018

//Import library for sound
import processing.sound.*;

//Initialize ArrayList and variables
ArrayList<Egg> eggs = new ArrayList();
ArrayList<Drawing> drawings = new ArrayList();

//Currently selected object
VisualObject curSelObject;
soundCollection scol;

//To keep track of situation in program
boolean tryToCreateEgg = false;
boolean play = false;
boolean finished = false;
boolean back = false;

//Variables of objects
float newX;
float newW;
float newH;
float newY;
float mousePosY;
float mousePosX;
float x;
float y;

//Determine new y on note line
float diff;
float g;
float a;
float b;
int rb;
float d;

//Determine size of menu elements
float one;
float two;
float three;
float four;
float elW;

//Border around draw canvas and menu & distance between note lines
float dist;

//Names of drawings
PImage house, tree, human, dog;
PImage windImg, feetImg, knockImg, yellImg;
//Initialize global variables
//Keep track of which object is currently selected; egg or drawing
int selectedEgg = 0;
int selectedDrawing = 0;
//Keep count of how many objects are created
int count = 0;
//Keep track of how many objects are numbered
int n = 0;
//Keep track of which object with its sound to play
int playStory = 1;

void setup() {
  size(1800, 900);
  //Calculate distance between note lines
  dist = ((height-37.5)/30);
  //Load images of drawings
  house = loadImage("house.png");
  tree = loadImage("tree.jpg");
  human = loadImage("human.jpg");
  dog = loadImage("dog.png");
  //Load overlay images
  windImg = loadImage("wind.png");
  feetImg = loadImage("feet.png");
  knockImg = loadImage("knock.png");
  yellImg = loadImage("yell.png");
  scol = new soundCollection(this);
}

//* Right 1/3 is menu area, left 2/3 is drawing space, border is equal to line distance */
void draw() {
  //==Reset drawing mode
  rectMode(CORNER);
  //==Draw clear Background
  background(200);

  //==Draw egg canvas
  //Background of canvas
  stroke(100);
  fill(245);
  rect(dist, dist, ((width*0.67)-(2*dist)), height-(2*dist));

  //Draw note lines within boundaries of egg canvas
  for (int i=1; i<30; i++) {
    float l = (dist*i)+37.5/2;
    stroke(0);
    line(dist, l, ((width*0.67)-(dist)), l);
  }
  //Always display all objects and numbers, first images, then eggs
  for (int i=0; i<drawings.size(); i++) {
    drawings.get(i).display();
    drawings.get(i).hover();
    drawings.get(i).numb();
  }
  for (int i=0; i<eggs.size(); i++) {
    eggs.get(i).display();
    eggs.get(i).hover();
    eggs.get(i).numb();
  }
}
//==Draw menu area
//Background of menu
stroke(100);
fill(175, 238, 238);
rect(((width*0.67)+dist), dist, ((width*0.33)-(2*dist)), (height-(2*dist)));

//Draw menu elements
one = (width-(width*0.33)) + dist;
three = ((width-dist)-one)-(4*dist);
elW = three/3;
fill(0);

//Image button
rect(one+dist, (2*dist), elW, (2*dist));
if (curSelObject instanceof Egg) {
  fill(0);
} else {
  fill(175);
}

//Colour button
rect(one+elW+(2*dist), (2*dist), elW, (2*dist));
if (curSelObject instanceof Drawing) {
  fill(0);
} else {
  fill(175);
}

//Sound button
rect((one+2*elW+3*dist), (2*dist), elW, (2*dist));

//Button texts
textSize(32);
fill(255);
text("Plaatje", one+dist+10, (3*dist)+10);
text("Kleur", (one+elW+2*dist+10), (3*dist)+10);
text("Geluid", (one+3*elW+3*dist)+10, (3*dist)+10);

//Image menu
imageMode(CORNER);
image(house, one+dist, 5*dist, 35, 35);
image(tree, one+dist, 6*dist+35, 35, 35);
image(human, one+dist, 7*dist+70, 35, 35);
image(dog, one+dist, 8*dist+105, 35, 35);

//Text in image menu
textSize(22);
fill(0);
text("Huis", one+dist+40, 5*dist+25);
text("Boom", one+dist+40, 6*dist+60);
text("Persoon", one+dist+40, 7*dist+95);
text("Hond", one+dist+40, 8*dist+130);

//Colour menu
fill(165, 137, 193);
rect(one+elW+2*dist, 5*dist, 35, 35);
fill(111, 183, 214);
rect(one+elW+2*dist, 6*dist+35, 35, 35);
fill(255, 237, 81);
rect(one+elW+2*dist, 7*dist+70, 35, 35);
fill(249, 140, 182);
rect(one+elW+2*dist, 8*dist+105, 35, 35);

//Text in colour menu
textSize(22);
fill(0);
text("Piano", one+elW+2*dist+40, 5*dist+25);
text("Gitaar", one+elW+2*dist+40, 6*dist+60);
text("Klarinet", one+elW+2*dist+40, 7*dist+95);
text("Xylofoon", one+elW+2*dist+40, 8*dist+130);

//Sound menu
//Text in sound menu
textSize(22);
fill(0);
text("1. Wind", (one+2*elW+3*dist), 5*dist+25);
text("2. Lopen", (one+2*elW+3*dist), 6*dist+60);
text("3. Kloppen", (one+2*elW+3*dist), 7*dist+95);
text("4. Roepen", (one+2*elW+3*dist), 8*dist+130);

//Button PlayMode
fill(0, 100, 0);
rect(one+elW+2*dist, 0.6*height, elW, 2*dist);
textSize(22);
fill(0);
text("Afspelen", (one+elW+2*dist+10), (0.6*height)+30);
if (play) {
  fill(0);
} else {
  fill(175, 238, 238);
}
textSize(22);
text("Klik op de elementen in de volgorde dat je ze wilt laten afspelen.",
(width*0.67)+dist+10, 0.67*height, ((width-dist)-((0.67*width)+2*dist)),
60);

//Button done with creating path, or go one step back in creating the path
if (play) {
  //Display buttons
  fill(0, 100, 0, 200);
  rect(width*0.67+2*dist, 0.8*height, elW, 2*dist);
  rect(width*0.67+3*dist+elW, 0.8*height, elW, 2*dist);
  fill(0);
textSize(22);
text("Klaar", (width*0.67)+2*dist+10, 0.8*height+dist+5);
text("Terug", (width*0.67)+(3*dist+elW)+10, 0.8*height+dist+5);
}

if(finished){
  //Display what to do
  fill(0);
textSize(22);
text("Klik op een toets om een geluid af te spelen. Herhaal dit voor
eieder geluid.", (width*0.67)+dist+10, 0.85*height+dist+5, ((width-dist)-
((0.67*width)+2*dist)),60);
}

xxxix
void mousePressed() {
    //On egg canvas? Call mousePressedOnEggCanvas
    if (mouseX > dist && mouseX<(width*0.67)-dist && mouseY>dist && mouseY<(height-dist)) {
        mousePressedOnCanvas();
    }
    //On menu area? Call mousePressedOnMenuArea
    if (mouseX>((width*0.67)+dist) && mouseX<(width-dist) && mouseY>dist && mouseY<(height-dist)) {
        mousePressedOnMenuArea();
    }
    //neither? probably on border; ignore
}

//Implementation below matches the drawing of the mouse press logic dated June 21
void mousePressedOnCanvas() {
    //If not in play mode:
    if (!play) {
        tryToCreateEgg = false;
        //Check if mouse is pressed on a egg
        if (mouseOnEgg()) {
            //If no egg is selected
            if (!curSelObject.mouseOnEgg()) {
                //No, deselect any other object, select this egg
                selectedDrawing = 0;
                curSelObject.deselect();
                for (int i = 0; i < eggs.size(); i++) {
                    if (eggs.get(i).mouseOnEgg()) {
                        curSelObject = eggs.get(i);
                        curSelObject.select();
                        selectedEgg = i;
                    }
                }
            }
        }
        else if (mouseOnDrawing()) {
            //If mouse is on a drawing, check if that drawing is selected
            if (!curSelObject.mouseOnDrawing()) {
                //No, deselect any other object, select this drawing
                selectedEgg = 0;
                curSelObject.deselect();
                for (int i = 0; i < drawings.size(); i++) {
                    if (drawings.get(i).mouseOnDrawing()) {
                        curSelObject = drawings.get(i);
                        curSelObject.select();
                        selectedDrawing = 1;
                    }
                }
            }
        } else { //do nothing
        }
    } else {
        //Mouse on neither egg nor drawing? Set tryToCreateEgg to true and set newW and newH to standard values
        tryToCreateEgg = true;
        newW = 25;
        newH = 1.5*newW;
        //Deselect current egg/drawing if necessary
        if (curSelObject != null) {
            if (selectedEgg == 1) {
                curSelObject.deselect();
            }
        }
    }
}
selectedEgg = 0;
} else {
    selectedDrawing = 0;
}
curSelObject.deselect();
}

//Calculate distance between middle of egg and current mousePosition on click
mousePosY = mouseY - newY;
mousePosX = mouseX - newX;

else if (play) {
    //If in play mode:
    //Number the object that was clicked
    if (mouseOnEgg() || mouseOnDrawing()) {
        while (n < count) {
            if (mouseOnEgg()) {
                for (int i = 0; i < eggs.size(); i++) {
                    if (eggs.get(i).mouseOnEgg()) {
                        n = n + 1;
                        eggs.get(i).number = n;
                    }
                }
            }
            if (mouseOnDrawing()) {
                for (int j = 0; j < drawings.size(); j++) {
                    if (drawings.get(j).mouseOnDrawing()) {
                        n = n + 1;
                        drawings.get(j).number = n;
                    }
                }
            }
            break;
        }
    }
}

void mousePressedOnMenuArea() {
    //If not in play mode:
    if (!play) {
        //If mouse is on one of the four drawings
        //Set variables (coordinates and size) for drawing
        if ((mouseX > one + dist && mouseX < one + dist + 35) &&
            (mouseY > 5 * dist && mouseY < 5 * dist + 35) ||
            (mouseX > 6 * dist + 35 && mouseY < 6 * dist + 35 + 35) ||
            (mouseX > 7 * dist + 70 && mouseY < 7 * dist + 70 + 35) ||
            (mouseY > 8 * dist + 105 && mouseY < 8 * dist + 105 + 35)) {
            float x = random(dist, width * 0.67 - dist - 50);
            float y = random(dist, height - dist - 50);
            newX = x;
            newY = y;
            newW = 50;
            newH = 50;
            selectedEgg = 0;
        }
        //If there is no overlap with other drawings and the drawing would be placed within the screen check for menu element
        if (!overlapsWithObjectPlaceDr() && withinScreenPlaceDr(newX, newY)) {
// If mouseOnImage:
  // Create new drawing on canvas, deselect any egg, select this drawing
  if (mouseX>one+dist && mouseX< one+dist+elW && mouseY> 5*dist &&
    mouseY<5*dist+35) {
    // Image is a house
    drawings.add(new Drawing(newX, newY, newW, newH, house, null, this,
      count, scol));
    count++;
    if (selectedDrawing==1 || selectedEgg ==1) {
      curSelObject.deselect();
    }
    curSelObject = drawings.get(drawings.size()-1);
    selectedDrawing = 1;
    curSelObject.select();
  } else if (mouseX>one+dist && mouseX<one+dist+elW && mouseY>6*dist+35 &&
    mouseY<6*dist+70) {
    // Image is a person
    drawings.add(new Drawing(newX, newY, newW, newH, human, null, this,
      count, scol));
    count++;
    if (selectedDrawing==1 || selectedEgg ==1) {
      curSelObject.deselect();
    }
    curSelObject = drawings.get(drawings.size()-1);
    selectedDrawing = 1;
    curSelObject.select();
  } else if (mouseX>one+dist && mouseX<one+dist+elW && mouseY>7*dist+70 &&
    mouseY<7*dist+105) {
    // Image is a tree
    drawings.add(new Drawing(newX, newY, newW, newH, tree, null, this,
      count, scol));
    count++;
    if (selectedDrawing==1 || selectedEgg ==1) {
      curSelObject.deselect();
    }
    curSelObject = drawings.get(drawings.size()-1);
    selectedDrawing = 1;
    curSelObject.select();
  } else if (mouseX>one+dist && mouseX<one+dist+elW && mouseY>8*dist+105 &&
    mouseY<8*dist+140) {
    // Image is a dog
    drawings.add(new Drawing(newX, newY, newW, newH, dog, null, this,
      count, scol));
    count++;
    if (selectedDrawing==1 || selectedEgg ==1) {
      curSelObject.deselect();
    }
    curSelObject = drawings.get(drawings.size()-1);
    selectedDrawing = 1;
    curSelObject.select();
  }
  else {
    // If there is overlap, flash feedback, red block
    blocked();
  }

  // If mouseOnColour:
  // And there is an egg selected, give selected egg the color that is being clicked on
  if (selectedEgg==1) {

  xlii
if (mouseX>one+elW+2*dist && mouseY<one+2*elW+2*dist && mouseY>5*dist+
   //Set color of curSelObject to purple(piano)
   curSelObject.changeColor(color(165, 137, 193));
} else if (mouseX>one+elW+2*dist && mouseY<one+2*elW+2*dist &&
mouseY>5*dist+15 && mouseY<6*dist+70) {
    //Set color of curSelObject to blue(guitar)
    curSelObject.changeColor(color(11, 183, 214));
} else if (mouseX>one+elW+2*dist && mouseY<one+2*elW+2*dist &&
mouseY>5*dist+70 && mouseY<7*dist+105) {
    //Set color of curSelObject to yellow(clarinet)
    curSelObject.changeColor(color(255, 237, 81));
} else if (mouseX>one+elW+2*dist && mouseY<one+2*elW+2*dist &&
mouseY>5*dist+105 && mouseY<8*dist+140) {
    //Set color of curSelObject to pink(xylophone)
    curSelObject.changeColor(color(249, 140, 182));
}
}

//If mouseOnSound:
//And there is a drawing selected, give the selected drawing an overlay image
if (selectedDrawing!==null) {
    if (mouseX>one+2*elW+3*dist && mouseX<one+3*elW+3*dist+elW &&
mouseY>5*dist && mouseY<5*dist+35) {
        //Draw wind image over curSelDrawing
        curSelObject.overlayImage(windImg);
    } else if (mouseX>one+2*elW+3*dist && mouseX<one+3*elW+3*dist+elW &&
mouseY>5*dist+35 && mouseY<6*dist+70) {
        //Draw feet image over curSelDrawing
        curSelObject.overlayImage(feetImg);
    } else if (mouseX>one+2*elW+3*dist && mouseX<one+3*elW+3*dist+elW &&
mouseY>5*dist+70 && mouseY<7*dist+105) {
        //Draw knocking image over curSelDrawing
        curSelObject.overlayImage(knockImg);
    } else if (mouseX>one+2*elW+3*dist && mouseX<one+3*elW+3*dist+elW &&
mouseY>5*dist+105 && mouseY<8*dist+140) {
        //Draw yelling image over curSelDrawing
        curSelObject.overlayImage(yellImg);
    }
}

//Sound can only be combined with drawing
blocked();
println("Kies geluid voor plaatje");
}

//If mousePressed on play button
if (mouseX>one+elW+2*dist && mouseX<one+elW+2*dist+elW &&
mouseY>0.6*height && mouseY<0.6*height+2*dist) {
    //Boolean play is true
    play = true;
    //Set numbers of object to 99(nothing), so that numbering in certain
    //order can overwrite this value
    for (int i=0; i<eggs.size(); i++) {
        eggs.get(i).number = 99;
    }
}
for (int j=0; j<drawings.size(); j++) {
    drawings.get(j).number = 99;
}

//If mousePressed on done button
if (mouseX>((width*0.67)+2*dist) && mouseX<((width*0.67)+(2*dist)+elW) && mouseY>0.8*height && mouseY<((0.8*height)+(2*dist))) {
    //Set boolean finished to true and boolean back to false
    finished = true;
    back = false;
}

//If mousePressed on back button
if (mouseX>(width*0.67)+3*dist+elW && mouseX<(width*0.67)+3*dist+elW+elW && mouseY>0.8*height && mouseY<0.8*height+2*dist) {
    //Set boolean back to true
    back = true;
    //Get last numbered object remove that number and set back to 99(nothing)
    for (int i=0; i<eggs.size(); i++) {
        if (eggs.get(i).number == n) {
            eggs.get(i).number = 99;
        }
    }
    for (int j=0; j<drawings.size(); j++) {
        if (drawings.get(j).number == n) {
            drawings.get(j).number = 99;
        }
    }
    //Decrease value of n so that next object will be numbered correctly
    n--;
}
}

void keyPressed() {
    //If the finished button is pressed, it is possible to play the sounds by key press
    if (finished) {
        //Get the first object and play its sound
        for (int i=0; i<eggs.size(); i++) {
            if (eggs.get(i).number == playStory) {
                eggs.get(i).playSound();
                break;
            }
        }
        for (int j=0; j<drawings.size(); j++) {
            if (drawings.get(j).number == playStory) {
                drawings.get(j).playSound();
                break;
            }
        }
        //Play the next sound
        playStory++;
    }
}

void mouseReleased() {
    //If not in play mode:
    if (!play) {
        //tryToCreateEgg?
        if (tryToCreateEgg) {

xliv
//Set value newX to current mousePosition and calculate newY, so that an egg will be placed on a line
newX = mouseX;
newY = calculateY();
//Yes, check overlap all eggs if new created on this position
if (overlapsWithObjectPlace()) {
    //Yes, do nothing && flash red cross at mouse location && set tryToCreateEgg to false && set curSelObject back to select
    blocked();
    tryToCreateEgg = false;
    selectedEgg = 1;
    curSelObject.select();
} else if (withinScreenPlace()) {
    //No overlap and within screen, create new egg here and select that egg
    eggs.add(new Egg(newX, newY, color(240, 234, 214), newW, newH, 0, this, count, scol));
    count++;
    //Set last created egg to current selected object
    curSelObject = eggs.get(eggs.size() - 1);
    selectedEgg = 1;
    curSelObject.select();
    tryToCreateEgg = false;
} else {
    //Not within screen, do not create egg, flash red block at mouse location, set tryToCreateEgg to false and set curSelObject back to select
    blocked();
    tryToCreateEgg = false;
    selectedEgg = 1;
    curSelObject.select();
}
//No, do nothing
}

void mouseDragged() {
    //If not in play mode:
    if (!play) {
        //tryToCreateEgg?
        if (!tryToCreateEgg) {
            //No, curSelObject?
            if (curSelObject != null) {
                //Set newX to mousePosition minus distance between mousePosition (when clicked) and middle of egg
                if (selectedEgg == 1) {
                    newX = mouseX - mousePosX;
                    newY = calculateY();
                }
                //If there is overlap, draw blocked and save location
                if (overlapsWithObjectMove()) {
                    blocked();
                    newX = curSelObject.xpos;
                    newY = curSelObject ypos;
                }
                //If no overlap and egg moved within screen, update x and y position of curSelObject
                else if (withinScreenMove(newX, newY, newW, newH)) {
                    curSelObject.setLocation(newX, newY);
void mouseWheel(MouseEvent size) {
    // If not in play mode:
    if (!play) {
        // Check if an object is selected
        if (curSelObject != null) {
            // Get wheel information, and calculate new size (with min/max)
            float s = size.getCount();
            // If scrolling up
            if (s < 0) {
                // Egg cannot be larger than 50 in width
                if (newW > 50) {
                    newW = 50;
                } else {
                    // Increase width of egg
                    newW++;
                    // If scrolling down
                }
            } else {
                // Egg cannot be smaller than 15 in width
                if (newW < 15) {
                    newW = 15;
                } else {
                    // Decrease width of egg
                    newW--;
                }
            }
        } else {
            // No object selected, do nothing
        }
    } // Yes tryToCreateEgg, do nothing
}

// selectedDrawing = 1, set newY to mousePosition minus distance
// between mousePosition (when clicked) and middle of drawing
newY = mouseY - mousePosY;
// Yes, overlap with any drawing?
if (overlapsWithObjectMoveDr()) {
    // Yes, flash red @mouseLocation and do not move
    blocked();
    // Save newX and newY to the current position of the selected object
    newX = curSelObject.xpos;
    newY = curSelObject.ypos;
} else if (withinScreenMoveDr(newX, newY, newW, newH)) {
    // No overlap and within screen, update x/y of selectedDrawing
    curSelObject.xpos = newX;
    curSelObject.ypos = newY;
} else {
    // Not within screen, flash red @mouseLocation and do not move
    blocked();
    // Save newX and newY to the current position of the selected object
    newX = curSelObject.xpos;
    newY = curSelObject.ypos;
}

} // No object selected, do nothing
} // Yes tryToCreateEgg, do nothing
}
// Update height of egg to keep right proportion
newH = newW * 1.5;

// Check overlap with new size
if (overlapsWithObjectMove()) {
    // Overlap, flash red @ new size, don't resize
    blocked();
    newW--; 
}
else if (withinScreenMove(newX, newY, newW, newH)) {
    // No overlap and within screen, resize egg
    curSelObject.setResize(newW, newH);
}
else {
    // Not within screen, flash red @ new size, don't resize
    blocked();
    newW--;
}

// No, do nothing
}

// Calculate y value of note lines (which is used to determine the y location of an egg)
float calculateY() {
    float y = 0;
    diff = (height - 37.5) / 30;
    // If egg is placed:
    if (tryToCreateEgg) {
        a = mouseY - (0.5*37.5);
        b = a / diff;
        rb = round(b);
        y = rb * diff + (0.5 * 37.5);
        return y;
    }
    // If egg is moved:
    a = (mouseY - mousePosY) - (0.5 * 37.5);
    b = a / diff;
    rb = round(b);
    y = rb * diff + (0.5 * 37.5);
    return y;
}

// Draw a red block on the mouse position (called if an action is not possible)
void blocked() {
    noStroke();
    fill(255, 0, 0);
    rectMode(CENTER);
    rect(mouseX, mouseY, newW, newH);
}

// Check if mouse is on an object
boolean mouseOnEgg() {
    // Check if mouse is on a egg
    for (int i = 0; i < eggs.size(); i++) {
        if (eggs.get(i).mouseOnEgg()) {
            return true;
        }
    }
    return false;
}

boolean mouseOnDrawing() {

}
//Check if mouse is on a drawing
for (int j=0; j<drawings.size(); j++) {
    if (drawings.get(j).mouseOnDrawing()) {
        return true;
    }
}
return false;

//Check if creating an object leads to overlap with other object
boolean overlapsWithObjectPlace() {
    //Check if egg would be created in conflict with existing eggs
    for (int i = 0; i<eggs.size(); i++) {
        if (eggs.get(i).overlapPlace(newW, newH)) {
            return true;
        }
    }
    return false;
}

boolean overlapsWithObjectPlaceDr() {
    //Check if drawing would be created in conflict with existing drawings
    for (int i = 0; i<drawings.size(); i++) {
        if (drawings.get(i).overlapPlaceDr(newX, newY)) {
            return true;
        }
    }
    return false;
}

//Check if creating an object leads to overlap with other objects
boolean overlapsWithObjectMove() {
    //Check if egg would be moved/resized in conflict with existing eggs
    for (int i = 0; i<eggs.size(); i++) {
        if (eggs.get(i).overlapMove(newX, newY, newW, newH) &&
            !eggs.get(i).equals(curSelObject)) {
            return true;
        }
    }
    return false;
}

boolean overlapsWithObjectMoveDr() {
    //Check if drawing would be moved in conflict with existing drawings
    for (int i = 0; i<drawings.size(); i++) {
        if (drawings.get(i).overlapMoveDr(newX, newY, newW, newH) &&
            !drawings.get(i).equals(curSelObject)) {
            return true;
        }
    }
    return false;
}

//Check if an object is placed outside the screen
boolean withinScreenPlace() {
    //Check if an egg is placed
    if (mouseX-0.5*newW>dist && mouseX+0.5*newW<(width*0.67)-dist && mouseY-
        0.5*newH>dist && mouseY+0.5*newH<height-dist) {
        return true;
    } else {
        return false;
    }
}
boolean withinScreenPlaceDr(float newX, float newY) {
    // Check if a drawing is placed outside the screen
    if (newX > dist && newX + newW < (width * 0.67) - dist && newY > dist &&
        newY + newH < height - dist) {
        return true;
    } else {
        return false;
    }
}

// Check if an object is moved/resized outside the screen
boolean withinScreenMove(float newX, float newY, float newW, float newH) {
    // Check if an egg is moved/resized outside the screen
    if ((newX - 0.5 * newW) > dist && (newX + 0.5 * newW) < (width * 0.67) - dist &&
        newY - 0.5 * newH) >= dist && (newY + 0.5 * newH) <= (height - dist)) {
        return true;
    } else {
        return false;
    }
}

boolean withinScreenMoveDr(float newX, float newY, float newW, float newH) {
    // Check if a drawing is moved outside the screen
    if (newX > dist && (newX + newW) < (width * 0.67) - dist &&
        newY >= dist && (newY + newH) <= (height - dist)) {
        return true;
    } else {
        return false;
    }
}

Code of class VisualObject

// Class for all objects
class VisualObject {
    // Declaring variables for location, size, number and color
    protected float xpos;
    protected float ypos;
    float w;
    float h;
    int number;
    color fillColor;
    color strokeColor;
    boolean select = false;
    // Object is locked on mouseposition when moving around
    float xoff = 0.0;
    float yoff = 0.0;

    // To refer to main program
    PApplet app;
    soundCollection scol;

    // All objects have a position, size, number and need to communicate with main program
VisualObject(float xpos_, float ypos_, float w_, float h_, PApplet p, int num, soundCollection sc) {
    w = w_;
    h = h_;
    xpos = xpos_;
    ypos = ypos_;
    app = p;
    number = num;
    scol = sc;
}

//Display the "image" of this object (in abstract version, just a box)
void display() {
    fill(fillColor);
}

//Display number of object
void numb(){
    if (number != 99) {
        fill(255, 0, 0);
    } else {
        fill(255);
        noStroke();
    }
}

//The hover function will draw a "hover line" (generally a red outline) if mouse is on object.
//Abstract version just sets the drawing tool (red line or no line) depending on whether mouse is over object. Subclass will do the actual drawing
void hover() {
    if (mouseOnEgg() || mouseOnDrawing()) {
        stroke(255, 0, 0);
    } else {
        noStroke();
    }
}

//Set whether the object is selected
void select() {
    select = true;
    //If object is selected, use newX, newY, newW and newH to change values of the currently selected object
    newX = curSelObject.xpos;
    newY = curSelObject.ypos;
    newW = curSelObject.w;
    newH = curSelObject.h;
}

//Set whether the object is deselected
void deselect() {
    select = false;
}

//Pass color through
void changeColor(color c) {
    //Solves problem of resizing after color
    newX = curSelObject.xpos;
    newY = curSelObject.ypos;
    newW = curSelObject.w;
    newH = curSelObject.h;
}
//Pass overlay image
void overlayImage(PImage imageSound) {
}

//Save location of curSelObject
void setLocation(float newX_, float newY_) {
    //Set coord in variable
    curSelObject.xpos = newX_;
    curSelObject.ypos = newY_;
}

//Save resized value of curSelObject
void setResize(float newW_, float newH_) {
    curSelObject.w = newW_;
    newH_ = newW_*1.5;
    curSelObject.h = newH_;
}

//Play sound of object
void playSound() {
}

//Stop playing sound
void stopSound() {
}

//Check if mouse is on an object
boolean mouseOnEgg() {
    //Check if mouse is on a egg
    if (mouseX>(xpos-w/2) && mouseX<(xpos+w/2) && mouseY>(ypos-h/2) && mouseY<(ypos+h/2)) {
        return true;
    } else {
        return false;
    }
}

boolean mouseOnDrawing() {
    //Check if mouse is on a drawing
    if (mouseX>xpos && mouseX<xpos+w & mouseY>ypos & mouseY<ypos+h) {
        return true;
    } else {
        return false;
    }
}

//Check if there is overlap between the present objects and a new object that would be placed on the current mousePosition(egg)/ random position(drawing)
boolean overlapPlace(float newW_, float newH_) {
    //Check if egg would be created in conflict with existing eggs
    if (((mouseX+(0.5*w + 0.5*newW_))>xpos && (mouseX-(0.5*w + 0.5*newW_))<xpos && (mouseY+(0.5*h+0.5*newH_))>ypos && (mouseY-(0.5*h+0.5*newH_))<ypos) {
        return true;
    } else {
        return false;
    }
}
boolean overlapPlaceDr(float x, float y) {
    //Check if drawing would be created on random location in conflict with
    //existing drawings
    if ((x+w)>xpos && (x-w)<xpos && (y+h)>ypos && (y-h)<ypos) {
        return true;
    } else {
        return false;
    }
}

//Check if there is overlap between the present objects and a
//dragged/resized object
boolean overlapMove(float newX, float newY, float newW, float newH) {
    //Check if egg would be moved/resized in conflict with existing eggs
    if (((newX + 0.5*newW) > (xpos-0.5*w) && (newX - 0.5*newW)<(xpos+0.5*w))
        && (newY +0.5*newH) > (ypos-0.5*h) && (newY -0.5*newH)<(ypos+0.5*h)) {
        return true;
    } else {
        return false;
    }
}

boolean overlapMoveDr(float newX_, float newY_, float newW_, float newH_) {
    //Check if drawing would be moved in conflict with existing drawings
    if (((newX_+newW_)>xpos && (newX_-newW_)<xpos && (newX_+newH_)>ypos &&
        (newY_-newH_)<ypos) {
        return true;
    } else {
        return false;
    }
}

Code of soundCollection

class soundCollection {
    //Sound files for eggs: piano, guitar, clarinet and xylophone
    public SoundFile p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13,
            p14, p15, p16, p17, p18, p19, p20, p21, p22, p23, p24, p25, p26,
            p27, p28, p29;
    public SoundFile g1, g2, g3, g4, g5, g6, g7, g8, g9, g10, g11, g12, g13,
            g14, g15, g16, g17, g18, g19, g20, g21, g22, g23, g24, g25, g26,
            g27, g28, g29;
    public SoundFile c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13,
            c14, c15, c16, c17, c18, c19, c20, c21, c22, c23, c24, c25, c26,
            c27, c28, c29;
    public SoundFile v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13,
            v27, v28, v29;
    //Soundfiles for drawings: wind, walking, knocking and yelling
    public SoundFile windSound, walkSound, knockSound, yellSound;

    soundCollection(PApplet app) {
        //Load sound files for eggs
    }
}
p1 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/c6.wav");
p2 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/b5.wav");
p3 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/a5.wav");
p4 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/g5.wav");
p5 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/f5.wav");
p6 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/e5.wav");
p7 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/d5.wav");
p8 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/c5.wav");
p9 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/b4.wav");
p10 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/a4.wav");
p11 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/g4.wav");
p12 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/f4.wav");
p13 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/e4.wav");
p14 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/d4.wav");
p15 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/c4.wav");
p16 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/b3.wav");
p17 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/a3.wav");
p18 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/g3.wav");
p19 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/f3.wav");
p20 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/e3.wav");
p21 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/d3.wav");
p22 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/c3.wav");
p23 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/b2.wav");
p24 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/a2.wav");
p25 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/g2.wav");
p26 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/f2.wav");
p27 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/e2.wav");
p28 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/d2.wav");
p29 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/piano/c2.wav"));
g1 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/c6.wav");
g2 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/b5.wav");
g3 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/a5.wav");
g4 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/g5.wav");
g5 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/f5.wav");
g6 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/e5.wav");
g7 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/d5.wav");
g8 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/c5.wav");
g9 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/b4.wav");
g10 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/a4.wav");
g11 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/g4.wav");
g12 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/f4.wav");
g13 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/e4.wav");
g14 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/d4.wav");
g15 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/c4.wav");
g16 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/b3.wav");
g17 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/a3.wav");
g18 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/g3.wav");
g19 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/f3.wav");
g20 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/e3.wav");
g21 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/d3.wav");
g22 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/c3.wav");
g23 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/b2.wav");
g24 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/a2.wav");
g25 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/g2.wav");
g26 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/e2.wav");
g27 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/d2.wav");
g28 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/guitar/c2.wav");
c1 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/e6.wav");
c2 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/d5.wav");
c3 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/c5.wav")
c4 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/b5.wav")
c5 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/a5.wav")
c6 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/g5.wav")
c7 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/f5.wav")
c8 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/e5.wav")
c9 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/d4.wav")
c10 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/c4.wav")
c11 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/b4.wav")
c12 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/a4.wav")
c13 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/g4.wav")
c14 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/f4.wav")
c15 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/e4.wav")
c16 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/d3.wav")
c17 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/c3.wav")
c18 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/b3.wav")
c19 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/a3.wav")
c20 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/g3.wav")
c21 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/f3.wav")
c22 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/e3.wav")
c23 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/d2.wav")
c24 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/c2.wav")
c25 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/b2.wav")
c26 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/a2.wav")
c27 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/g2.wav")
c28 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/f2.wav")
c29 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/clarinet/e2.wav")
v1 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/c6.wav")
v2 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/b5.wav")
v3 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/a5.wav")
v4 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/g5.wav");
v5 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/f5.wav");
v6 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/e5.wav");
v7 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/d5.wav");
v8 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/c5.wav");
v9 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/b4.wav");
v10 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/a4.wav");
v11 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/g4.wav");
v12 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/f4.wav");
v13 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/e4.wav");
v14 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/d4.wav");
v15 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/c4.wav");
v16 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/b3.wav");
v17 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/a3.wav");
v18 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/g3.wav");
v19 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/f3.wav");
v20 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/e3.wav");
v21 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/d3.wav");
v22 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/c3.wav");
v23 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/b2.wav");
v24 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/a2.wav");
v25 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/g2.wav");
v26 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/f2.wav");
v27 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/e2.wav");
v28 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/d2.wav");
v29 = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/xylophone/c2.wav");

// Load sound files for drawings
windSound = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/wind.wav");
walkSound = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/walk.wav");
knockSound = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/knock.wav");
yellSound = new SoundFile(app, "C:/Users/Gebruiker/Documents/Processing/GP/FinalPath/yell.wav");
Code of subclass Drawing

//Subclass for drawings
class Drawing extends VisualObject {
    //Declaring variables for color and images
    color strokeColor;
    color fillColor;
    PImage image;
    PImage imageSound;
    char over;

    //All drawings have a position, size, images, number and need to communicate
    //with main program
    Drawing(float xpos_, float ypos_, float w_, float h_, PImage img, PImage overlay, PApplet p, int num, soundCollection sc) {
        //Get position, size, number and ability to communicate with main program
        super(xpos_, ypos_, w_, h_, p, num, sc);
        image = img;
        imageSound = overlay;
    }

    //Display drawing
    void display() {
        super.display();
        //How to draw a drawing
        stroke(strokeColor);
        noFill();
        rect(xpos-1, ypos-1, 52, 52);
        imageMode(CORNER);
        image(image, xpos, ypos, 50, 50);
        //Draw overlay image(sound)
        if (imageSound != null) {
            image(imageSound, xpos, ypos, 50, 50);
        }
    }

    //Display number of drawing
    void numb() {
        super.numb();
        if (play) {
            text(number, xpos+20, ypos+30);
        }
    }

    //Draw red hover line around drawing if mouse is over drawing
    void hover() {
        super.hover();
        if (mouseOnDrawing()) {
            noFill();
            rect(xpos, ypos, 50, 50);
        }
    }

    //Set whether the object is selected
void select() {
    super.select();
    //If drawing is selected, draw a green select line around the image
    strokeColor=color(0, 255, 0);
    rect(50, 50, w, h);
}

//Set whether the object is deselected
void deselect() {
    super.deselect();
    //Don't show color and stop playing sound if that was the case
    strokeColor = 175;
    stopSound();
}

//Get resized value of curSelObject (in case of curSelObject == drawing)
void setResize(float w_, float h_) {
    super.setResize(w_, h_);
    //change sound (amplitude)
}

//Set image over drawing for sound
void overlayImage(PImage ove) {
    super.overlayImage(ove);
    imageSound = ove;
    if (ove == windImg) {
        over = 'w';
    } else if (ove == feetImg) {
        over = 'x';
    } else if (ove == knockImg) {
        over = 'y';
    } else if (ove == yellImg) {
        over = 'z';
    }
    stopSound();
    playSound();
}

//Get correct sound to play if function is called
void playSound() {
    super.playSound();
    if (over == 'w') {
        scol.windSound.play();
    } else if (over == 'x') {
        scol.walkSound.play();
    } else if (over == 'y') {
        scol.knockSound.play();
    } else if (over == 'z') {
        scol.yellSound.play();
    }
}

//Stop all sounds whichever sound is currently playing
void stopSound() {
    super.stopSound();
    if (over == 'w' || over == 'x' || over == 'y' || over == 'z') {
        scol.windSound.stop();
        scol.walkSound.stop();
        scol.knockSound.stop();
        scol.yellSound.stop();
    }
}
Code of sub class Egg

//Subclass for eggs
class Egg extends VisualObject {
    //Declaring variables for color and sound
    color c;
    float sound;
    char colour;

    //All eggs have a position, size, color, number, sound and need to
    //communicate with main program
    Egg(float xpos_, float ypos_, color c_, float w_, float h_, float sound_, PApplet p, int num, soundCollection sc) {
        super(xpos_, ypos_, w_, h_, p, num, sc);
        c = c_;
        sound = sound_;
    }

    //Display egg
    void display() {
        super.display();
        //How to draw an egg
        stroke(strokeColor);
        fill(c);
        ellipse(xpos, ypos, w, h);
    }

    //Display number of drawing
    void numb() {
        super.numb();
        if (play) {
            text(number, xpos-5, ypos+5);
        }
    }

    //Draw red hover line around egg if mouse is over egg
    void hover() {
        super.hover();
        if (mouseOnEgg()) {
            noFill();
            ellipse(xpos, ypos, w, h);
        }
    }

    //Set whether the object is selected
    void select() {
        super.select();
        //If egg is selected, draw a green select line around the ellipse
        strokeColor = color(0, 255, 0);
    }

    //Set whether the object is deselected
    void deselect() {
        }
super.deselect(); //Set dark select line around egg
strokeColor = 175;
}

//Calculate y value of note lines
int calculateNote() {
    int note;
    float line = ypos/dist;
    note = round(line)-1;
    return note;
}

//Calculate amplitude value based on size(width) of egg
float amplitudeNote() {
    float amplitude = map(w, 10, 51, 0, 1);
    return amplitude;
}

//Change color of egg
void changeColor(color c_) {
    super.changeColor(c_);
    c = c_;
    if (c_ == color(165, 137, 193)) {
        colour = 'p';
    } else if (c_ == color(111, 183, 214)) {
        colour = 'b';
    } else if (c_ == color(255, 237, 81)) {
        colour = 'y';
    } else if (c_ == color(249, 140, 182)) {
        colour = 'i';
    }
    //Play sound if color has been changed
    playSound();
}

//Set location of egg
void setLocation(float xpos_, float ypos_) {
    super.setLocation(xpos_, ypos_);
    //Play if egg has been moved
    playSound();
}

void setResize(float w_, float h_) {
    super.setResize(w_, h_);
    //Play if egg has been resized
    playSound();
}

//Get correct sound to play if function is called
void playSound() {
    super.playSound();
    //Check y-location, color, size (int note, char colour, float amplitude)
to play correct sound file
    int n = calculateNote();
    float a = amplitudeNote();
    //Color is purple, piano sound
    if (colour == 'p') {
        //Check which note line
        switch(n) {
        |x
case 1:
    //Purple, line 1
    scol.p1.play(1, -1, a);
    break;

case 2:
    //Purple, line 2
    scol.p2.play(1, -1, a);
    break;

case 3:
    //Purple, line 3
    scol.p3.play(1, -1, a);
    break;

case 4:
    //Purple, line 4
    scol.p4.play(1, -1, a);
    break;

case 5:
    //Purple, line 5
    scol.p5.play(1, -1, a);
    break;

case 6:
    //Purple, line 6
    scol.p6.play(1, -1, a);
    break;

case 7:
    //Purple, line 7
    scol.p7.play(1, -1, a);
    break;

case 8:
    //Purple, line 8
    scol.p8.play(1, -1, a);
    break;

case 9:
    //Purple, line 9
    scol.p9.play(1, -1, a);
    break;

case 10:
    //Purple, line 10
    scol.p10.play(1, -1, a);
    break;

case 11:
    //Purple, line 11
    scol.p11.play(1, -1, a);
    break;

case 12:
    //Purple, line 12
    scol.p12.play(1, -1, a);
    break;

case 13:
    //Purple, line 13
    scol.p13.play(1, -1, a);
    break;

case 14:
    //Purple, line 14
    scol.p14.play(1, -1, a);
    break;

case 15:
    //Purple, line 15
    scol.p15.play(1, -1, a);
    break;

case 16:
//Purple, line 16
scol.p16.play(1, -1, a);
break;
case 17:
//Purple, line 17
scol.p17.play(1, -1, a);
break;
case 18:
//Purple, line 18
scol.p18.play(1, -1, a);
break;
case 19:
//Purple, line 19
scol.p19.play(1, -1, a);
break;
case 20:
//Purple, line 20
scol.p20.play(1, -1, a);
break;
case 21:
//Purple, line 21
scol.p21.play(1, -1, a);
break;
case 22:
//Purple, line 22
scol.p22.play(1, -1, a);
break;
case 23:
//Purple, line 23
scol.p23.play(1, -1, a);
break;
case 24:
//Purple, line 24
scol.p24.play(1, -1, a);
break;
case 25:
//Purple, line 25
scol.p25.play(1, -1, a);
break;
case 26:
//Purple, line 26
scol.p26.play(1, -1, a);
break;
case 27:
//Purple, line 27
scol.p27.play(1, -1, a);
break;
case 28:
//Purple, line 28
scol.p28.play(1, -1, a);
break;
case 29:
//Purple, line 29
scol.p29.play(1, -1, a);
break;
}
else if (colour == 'b') {
//Color is blue, guitar sound
//Check which note line
switch(n) {
case 1:
//Blue, line 1
scol.g1.play(1, -1, a);
break;
case 2:
  //Blue, line 2
scol.g2.play(1, -1, a);
break;
case 3:
  //Blue, line 3
scol.g3.play(1, -1, a);
break;
case 4:
  //Blue, line 4
scol.g4.play(1, -1, a);
break;
case 5:
  //Blue, line 5
scol.g5.play(1, -1, a);
break;
case 6:
  //Blue, line 6
scol.g6.play(1, -1, a);
break;
case 7:
  //Blue, line 7
scol.g7.play(1, -1, a);
break;
case 8:
  //Blue, line 8
scol.g8.play(1, -1, a);
break;
case 9:
  //Blue, line 9
scol.g9.play(1, -1, a);
break;
case 10:
  //Blue, line 10
scol.g10.play(1, -1, a);
break;
case 11:
  //Blue, line 11
scol.g11.play(1, -1, a);
break;
case 12:
  //Blue, line 12
scol.g12.play(1, -1, a);
break;
case 13:
  //Blue, line 13
scol.g13.play(1, -1, a);
break;
case 14:
  //Blue, line 14
scol.g14.play(1, -1, a);
break;
case 15:
  //Blue, line 15
scol.g15.play(1, -1, a);
break;
case 16:
  //Blue, line 16
scol.g16.play(1, -1, a);
break;
case 17:
    //Blue, line 17
scol.g17.play(1, -1, a);
break;
case 18:
    //Blue, line 18
scol.g18.play(1, -1, a);
break;
case 19:
    //Blue, line 19
scol.g19.play(1, -1, a);
break;
case 20:
    //Blue, line 20
scol.g20.play(1, -1, a);
break;
case 21:
    //Blue, line 21
scol.g21.play(1, -1, a);
break;
case 22:
    //Blue, line 22
scol.g22.play(1, -1, a);
break;
case 23:
    //Blue, line 23
scol.g23.play(1, -1, a);
break;
case 24:
    //Blue, line 24
scol.g24.play(1, -1, a);
break;
case 25:
    //Blue, line 25
scol.g25.play(1, -1, a);
break;
case 26:
    //Blue, line 26
scol.g26.play(1, -1, a);
break;
case 27:
    //Blue, line 27
scol.g27.play(1, -1, a);
break;
case 28:
    //Blue, line 28
scol.g28.play(1, -1, a);
break;
case 29:
    //Blue, line 29
scol.g29.play(1, -1, a);
break;
}

} else if (colour == 'y') {
    //Color is yellow, clarinet sound
    //Check which note line
switch(n) {
    case 1:
        //Yellow, line 1
scol.c1.play(1, -1, a);
break;
case 2:
  //Yellow, line 2
scol.c2.play(1, -1, a);
break;
case 3:
  //Yellow, line 3
scol.c3.play(1, -1, a);
break;
case 4:
  //Yellow, line 4
scol.c4.play(1, -1, a);
break;
case 5:
  //Yellow, line 5
scol.c5.play(1, -1, a);
break;
case 6:
  //Yellow, line 6
scol.c6.play(1, -1, a);
break;
case 7:
  //Yellow, line 7
scol.c7.play(1, -1, a);
break;
case 8:
  //Yellow, line 8
scol.c8.play(1, -1, a);
break;
case 9:
  //Yellow, line 9
scol.c9.play(1, -1, a);
break;
case 10:
  //Yellow, line 10
scol.c10.play(1, -1, a);
break;
case 11:
  //Yellow, line 11
scol.c11.play(1, -1, a);
break;
case 12:
  //Yellow, line 12
scol.c12.play(1, -1, a);
break;
case 13:
  //Yellow, line 13
scol.c13.play(1, -1, a);
break;
case 14:
  //Yellow, line 14
scol.c14.play(1, -1, a);
break;
case 15:
  //Yellow, line 15
scol.c15.play(1, -1, a);
break;
case 16:
  //Yellow, line 16
scol.c16.play(1, -1, a);
break;
case 17:
    //Yellow, line 17
    scol.c17.play(1, -1, a);
    break;
case 18:
    //Yellow, line 18
    scol.c18.play(1, -1, a);
    break;
case 19:
    //Yellow, line 19
    scol.c19.play(1, -1, a);
    break;
case 20:
    //Yellow, line 20
    scol.c20.play(1, -1, a);
    break;
case 21:
    //Yellow, line 21
    scol.c21.play(1, -1, a);
    break;
case 22:
    //Yellow, line 22
    scol.c22.play(1, -1, a);
    break;
case 23:
    //Yellow, line 23
    scol.c23.play(1, -1, a);
    break;
case 24:
    //Yellow, line 24
    scol.c24.play(1, -1, a);
    break;
case 25:
    //Yellow, line 25
    scol.c25.play(1, -1, a);
    break;
case 26:
    //Yellow, line 26
    scol.c26.play(1, -1, a);
    break;
case 27:
    //Yellow, line 27
    scol.c27.play(1, -1, a);
    break;
case 28:
    //Yellow, line 28
    scol.c28.play(1, -1, a);
    break;
case 29:
    //Yellow, line 29
    scol.c29.play(1, -1, a);
    break;
}
break;
case 2:
    //Pink, line 2
    scol. v2.play(1, -1, a);
    break;
case 3:
    //Pink, line 3
    scol. v3.play(1, -1, a);
    break;
case 4:
    //Pink, line 4
    scol. v4.play(1, -1, a);
    break;
case 5:
    //Pink, line 5
    scol. v5.play(1, -1, a);
    break;
case 6:
    //Pink, line 6
    scol. v6.play(1, -1, a);
    break;
case 7:
    //Pink, line 7
    scol. v7.play(1, -1, a);
    break;
case 8:
    //Pink, line 8
    scol. v8.play(1, -1, a);
    break;
case 9:
    //Pink, line 9
    scol. v9.play(1, -1, a);
    break;
case 10:
    //Pink, line 10
    scol. v10.play(1, -1, a);
    break;
case 11:
    //Pink, line 11
    scol. v11.play(1, -1, a);
    break;
case 12:
    //Pink, line 12
    scol. v12.play(1, -1, a);
    break;
case 13:
    //Pink, line 13
    scol. v13.play(1, -1, a);
    break;
case 14:
    //Pink, line 14
    scol. v14.play(1, -1, a);
    break;
case 15:
    //Pink, line 15
    scol. v15.play(1, -1, a);
    break;
case 16:
    //Pink, line 16
    scol. v16.play(1, -1, a);
    break;
case 17:
    //Pink, line 17
    scol. v17.play(1, -1, a);
    break;
case 18:
    //Pink, line 18
    scol. v18.play(1, -1, a);
    break;
case 19:
    //Pink, line 19
    scol. v19.play(1, -1, a);
    break;
case 20:
    //Pink, line 20
    scol. v20.play(1, -1, a);
    break;
case 21:
    //Pink, line 21
    scol. v21.play(1, -1, a);
    break;
case 22:
    //Pink, line 22
    scol. v22.play(1, -1, a);
    break;
case 23:
    //Pink, line 23
    scol. v23.play(1, -1, a);
    break;
case 24:
    //Pink, line 24
    scol. v24.play(1, -1, a);
    break;
case 25:
    //Pink, line 25
    scol. v25.play(1, -1, a);
    break;
case 26:
    //Pink, line 26
    scol. v26.play(1, -1, a);
    break;
case 27:
    //Pink, line 27
    scol. v27.play(1, -1, a);
    break;
case 28:
    //Pink, line 28
    scol. v28.play(1, -1, a);
    break;
    //Pink, line 29
    scol. v29.play(1, -1, a);
    break;
    }
}

//Stop sound
void stopSound() {
    scol.p1.stop();
    scol.p2.stop();
    scol.p3.stop();
}
scol. p4.stop();
scol.p5.stop();
scol.p6.stop();
scol.p7.stop();
scol.p8.stop();
scol.p9.stop();
scol.p10.stop();
scol.p11.stop();
scol.p12.stop();
scol.p13.stop();
scol.p14.stop();
scol.p15.stop();
scol.p16.stop();
scol.p17.stop();
scol.p18.stop();
scol.p19.stop();
scol.p20.stop();
scol.p21.stop();
scol.p22.stop();
scol.p23.stop();
scol.p24.stop();
scol.p25.stop();
scol.p26.stop();
scol.p27.stop();
scol.p28.stop();
scol.p29.stop();
scol.g1.stop();
scol.g2.stop();
scol.g3.stop();
scol.g4.stop();
scol.g5.stop();
scol.g6.stop();
scol.g7.stop();
scol.g8.stop();
scol.g9.stop();
scol.g10.stop();
scol.g11.stop();
scol.g12.stop();
scol.g13.stop();
scol.g14.stop();
scol.g15.stop();
scol.g16.stop();
scol.g17.stop();
scol.g18.stop();
scol.g19.stop();
scol.g20.stop();
scol.g21.stop();
scol.g22.stop();
scol.g23.stop();
scol.g24.stop();
scol.g25.stop();
scol.g26.stop();
scol.g27.stop();
scol.g28.stop();
scol.g29.stop();
scol.c1.stop();
scol.c2.stop();
scol.c3.stop();
scol.c4.stop();
scol.c5.stop();
scol.c6.stop();
scol.c7.stop();
scol.c8.stop();
scol.c9.stop();
scol.c10.stop();
scol.c11.stop();
scol.c12.stop();
scol.c13.stop();
scol.c14.stop();
scol.c15.stop();
scol.c16.stop();
scol.c17.stop();
scol.c18.stop();
scol.c19.stop();
scol.c20.stop();
scol.c21.stop();
scol.c22.stop();
scol.c23.stop();
scol.c24.stop();
scol.c25.stop();
scol.c26.stop();
scol.c27.stop();
scol.c28.stop();
scol.c29.stop();
scol.v1.stop();
scol.v2.stop();
scol.v3.stop();
scol.v4.stop();
scol.v5.stop();
scol.v6.stop();
scol.v7.stop();
scol.v8.stop();
scol.v9.stop();
scol.v10.stop();
scol.v11.stop();
scol.v12.stop();
scol.v13.stop();
scol.v14.stop();
scol.v15.stop();
scol.v16.stop();
scol.v17.stop();
scol.v18.stop();
scol.v19.stop();
scol.v20.stop();
scol.v21.stop();
scol.v22.stop();
scol.v23.stop();
scol.v24.stop();
scol.v25.stop();
scol.v26.stop();
scol.v27.stop();
scol.v28.stop();
scol.v29.stop();
}
Appendix B – Consent form

INFORMED CONSENT FORMULIER

Naam van het onderzoeksproject
Creative Use of Technology for Music Education in Primary Education

Doel van het onderzoek
Het doel van dit onderzoek is de functionaliteit en duidelijkheid van het gemaakte product vast te stellen.

Gang van zaken tijdens het onderzoek
Je neemt deel aan een onderzoek waarbij je opdrachten zal maken. Allereerst krijg je een formulier met uitleg over de context van het product en hoe het werkt. Hierna krijg je drie opdrachten die je achter elkaar uit kunt voeren. Daarvoor zullen de camera en schermopname starten, zodat de test later bekeken kan worden om conclusies te trekken uit de interactie met het product. Deze conclusies zullen worden gebruikt tijdens de presentatie en in het verslag. Na het uitvoeren van deze opdracht, staat er een korte vragenlijst klaar die je dan even in moet vullen.

Vertrouwelijkheid van gegevens
Je privacy is en blijft maximaal beschermd. Er wordt op geen enkele wijze vertrouwelijke informatie of persoonsgegevens van of over je naar buiten gebracht, waardoor iemand je zal kunnen herkennen. In een publicatie of presentaties zullen of anonieme gegevens of pseudoniemen worden gebruikt. De video- en audio-opnamen, formulieren en andere documenten die in het kader van deze studie worden gemaakt of verzameld, worden opgeslagen op de beveiligde (versleutelde) computers van de onderzoeker.

Vrijwilligheid
Deelname aan dit onderzoek is geheel vrijwillig. Je kunt als deelnemer jouw medewerking aan het onderzoek te allen tijde stoppen, of weigeren dat jouw gegevens voor het onderzoek mogen worden gebruikt, zonder opgaaf van redenen.

Dit betekent dat als je voorafgaand aan het onderzoek of achteraf besluit om af te zien van deelname aan dit onderzoek, dat dit op geen enkele wijze gevolgen voor jou zal hebben. In deze gevallen zullen jouw gegevens uit onze bestanden worden verwijderd en vernietigd.

Toestemmings-verklaring
Ik verklaar op een voor mij duidelijke wijze te zijn ingelicht over de aard, methode en doel van het onderzoek. Ik weet dat de gegevens en resultaten van het onderzoek alleen anoniem en vertrouwelijk aan derden bekend gemaakt zullen worden. Mijn vragen zijn naar tevredenheid beantwoord. Ik begrijp dat videomateriaal of bewerking daarvan uitsluitend voor analyse en/of wetenschappelijke presentaties zal worden gebruikt.
Ik stem geheel vrijwillig in met deelname aan dit onderzoek. Ik behoud me daarbij het recht voor om op elk moment zonder opgaaf van redenen mijn deelname aan dit onderzoek te beëindigen.
Naam deelnemer   Handtekening   Datum

Naam Onderzoeker   Handtekening   Datum

UNIVERSITY OF TWENTE.
Appendix C – Exercises for test

Creative Use of Technology for Music Education in Primary Education
Het programma is bedoeld voor kinderen van groep 3 en/of 4 van de basisschool. Kinderen krijgen dan muziekonderwijs waarbij onder andere wordt ingegaan op toonhoogtes en volume van noten. Deze concepten worden vaak bijgebracht in een context die aansluit bij hun belevingswereld. Door deze wat abstracte begrippen te koppelen aan concrete situaties wordt het concept van hard-zacht of snel-langzaam hopelijk beter bijgebracht. Deze belevingswereld kan worden omschreven in een verhaaltje dat kan dienen als uitgangspunt voor de opdracht die moet worden uitgevoerd.
Je zult zo drie verschillende soorten opdrachten krijgen zodat de functionaliteit van het programma voldoende naar voren komt.
Over de functionaliteit van het programma:
- het is mogelijk objecten te maken (door klikken in het linkerdeel), te veranderen van toonhoogte, te veranderen van volume en instrumentgeluid
- het is mogelijk om afbeeldingen in het scherm te krijgen (door klikken op de afbeelding) en ze te voorzien van geluid
Mocht er straks iets onduidelijk zijn, dan kun je altijd vragen stellen.
Opdracht 1
Deze opdracht is bedoeld om een beetje kennis te maken met het programma en de mogelijkheden.

Beeld een vogeltje uit met geluid; laat hem hard en zacht zingen of hoog en laag.
Opdracht 2a
Maak een ondersteuning voor het onderstaande verhaaltje en lees het vervolgens voor met de geluiden die je erbij hebt bedacht.

Er is een verrassing verstopt voor een hond en hij gaat op avontuur om dit te zoeken. Hij vertrekt bij zijn huisje en komt door een bos en vindt vervolgens zijn bot.
Ondersteun dit verhaaltje met plaatjes en geluiden.
Opdracht 2b
Maak een verhaaltje bij de onderstaande situatie en lees als je klaar bent je verhaaltje voor en gebruik daarbij de geluiden die gegeven zijn.

![Diagram met geluiden en plaatjes](image)

Klik op de elementen in de volgorde dat je wilt laten afspelen.

<table>
<thead>
<tr>
<th>Plaatje</th>
<th>Kleur</th>
<th>Geluid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huis</td>
<td>Piano</td>
<td>1. Wind</td>
</tr>
<tr>
<td>Boom</td>
<td>Gitaar</td>
<td>2. Lopen</td>
</tr>
<tr>
<td>Persoon</td>
<td>Klarinet</td>
<td>3. Kloppen</td>
</tr>
<tr>
<td>Hond</td>
<td>Xylofoon4</td>
<td>4. Roepen</td>
</tr>
</tbody>
</table>

Klaar - Terug
Opdracht 3
Bedenk zelf een kort verhaaltje of situatie, ondersteun dit vervolgens met de opties uit het programma en laat daarna je creatie horen.
Heel erg bedankt voor je deelname aan deze test! Op de computer staat nu een korte vragenlijst klaar die je in kunt vullen.
Appendix D – Questionnaire

Evaluatie programma klankverhaal

Dank je wel voor het doorlopen van de test. De vragen hieronder helpen mij inzicht te krijgen over de functionaliteit van het programma en de verbeterpunten. Mocht er een iets onduidelijk zijn, dan kun je het altijd vragen.

* Required

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Heel makkelijk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Heel moeilijk</td>
</tr>
</tbody>
</table>

Hoe makkelijk vond je de eerste opdracht uit te voeren? *

Hoe makkelijk vond je de tweede opdracht uit te voeren? *

Hoe makkelijk vond je de derde opdracht uit te voeren? *

lxxix
Vond je dat je voldoende creatief kon zijn tijdens de laatste opdracht (ondanks de beperkte afbeeldingen en geluiden)? *

- Ja
- Een beetje
- Nee

Hoe duidelijk vond je het programma voordat je ermee aan de slag moest? *

1 2 3 4 5
Heel duidelijk

Hoe duidelijk vond je het programma nadat je de laatste opdracht had uitgevoerd? *

1 2 3 4 5
Heel duidelijk

Vind je het programma geschikt voor de opdrachten die je uit moest voeren? *

- Ja
- Nee
- Other:  

Ixxx
Hoe heb je het werken met het programma ervaren? *
Your answer

Heb je nog suggesties, tips of miste er iets in het programma?
Your answer

Evaluatie programma klankverhaal

* Required

**Persoonlijke informatie**

Wat doe je momenteel? *
- Student - bachelor
- Student - master
- Student - hogeschool
- Werkende
- Other:

Specificeer je (vak)gebied. *
Your answer
<table>
<thead>
<tr>
<th>Heb je nog suggesties, tips of miste er iets in het programma?</th>
<th>Wat doe je momenteel?</th>
<th>Specificeer je (vak)gebied.</th>
</tr>
</thead>
<tbody>
<tr>
<td>makkelijker toevoegen van een geluidje met direct een instrument ipv steeds heen en weer met de muis</td>
<td>Student - bachelor</td>
<td>Electrical Engineering</td>
</tr>
<tr>
<td>Automatisch afspelen, dingen ongedaan kunnen maken, misschien meer consistentie in knoppen (e.g. tekst van geluiden ziet er niet direct uit alsof het geselecteerd kan worden)</td>
<td>Student - bachelor</td>
<td>Creative Technology</td>
</tr>
<tr>
<td>Automatisch afspelen, dingen ongedaan kunnen maken, misschien meer consistentie in knoppen (e.g. tekst van geluiden ziet er niet direct uit alsof het geselecteerd kan worden)</td>
<td>Student - bachelor</td>
<td>Computer Science, Electrical Engineering en Design</td>
</tr>
<tr>
<td>Mogelijkheid tot het verwijderen van icoontjes</td>
<td>Student - master</td>
<td>BME</td>
</tr>
<tr>
<td>ik vond het erg lastig dat je het eitje moest verplaatsen om te horen welk geluid/toon/volume het eitje heeft. Het zou fijn zijn dat als je er op klikt dat je dan hoort welk geluid het heeft. Ook was het lastig dat je geen elementen meer kon verwijderen.</td>
<td>Student - bachelor</td>
<td>Creative Technology</td>
</tr>
<tr>
<td>De optie om fouten te herstellen, en het geluid vaker af te spelen</td>
<td>Student – master</td>
<td>Electrical Engineering</td>
</tr>
<tr>
<td>Ik vraag mij wel een beetje af of kinderen die iets minder creatief aangelegd zijn niet misschien vast zouden lopen op het begin. Maar ik denk dat als ze met uitleg van een juf of meester erbij het wel leuk kunnen vinden en het verder toe kunnen passen</td>
<td>Student - bachelor</td>
<td>Creative Technology ~ New Media</td>
</tr>
<tr>
<td>misschien een tutorial aan het begin? Dat is vaak duidelijker dan een uitleg in woorden.</td>
<td>Student - bachelor</td>
<td>CreaTe</td>
</tr>
<tr>
<td>Vind je het programma geschikt voor de opdrachten die je uit moest voeren?</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Hoe heb je het werken met het programma ervaren?</td>
<td>lastig om veel dingen snel toe te voegen</td>
<td>heel leuk om door toonladders heen te scrollen en te horen hoe ze klinken, grappige geluidjes</td>
</tr>
<tr>
<td>Vond je dat je voldoende creatief kon zijn tijdens de laatste opdracht (ondanks de beperkte afbeeldingen en geluiden)?</td>
<td>Ja</td>
<td>Ja</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Hoe duidelijk vond je het programma voordat je ermee aan de slag moest?</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Hoe duidelijk vond je het programma nadat je de laatste opdracht had uitgevoerd?</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Hoe makkelijk vond je de eerste opdracht uit te voeren?</td>
<td>Hoe makkelijk vond je de tweede opdracht uit te voeren?</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>7/3/2018 22:22:45</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>7/4/2018 9:28:06</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7/4/2018 10:01:06</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>7/4/2018 12:23:32</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>7/4/2018 14:25:25</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>7/4/2018 15:42:32</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7/4/2018 16:05:58</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>