**Personal Report**

**-Cubes-**

**Philipp Grüger INF4**

**Group Members: Lars Kalkuhl, Patrick Beinlich, Patrick Engesser, Florian Bertele**

**What did i do**

1. **Menu**  
   One of my tasks within the Game-Programming-Project was to implement the whole options menu. Inside this menu, there are more submenus. These menus are, one the one hand, the visuals part. And on the other hand the sound part. Inside the visuals menu, you are able to customize the screen size. There are three offered alternatives. Small, medium and big. To scale the window size i had to pass over the Scene, Stage and the BorderPane. There i had to build up a menu with the similar look like the titlescreen to keep it in the same appearance. Within the visuals menu i implemented three RadioButtons with the window sizes and a Button to get back to the previous menu. To provide the function of these RadioButtons by clicking on them, it was necessary to put all three RadioButton into a ToggleGroup. After setting them into a ToggleGroup with, e.g. smallSize.setToggleGroup(toggle), the Buttons needed their function. In order to get the instant results of the Buttons by clicking them, i had to write a Listener which is responsible for the immediate change of the Window size. Inside the sound menu, you are able to start or stop the titlemusic, again done with a toggleProperty Listener. Since the titlemusic has to run “forever” i wrote a musicLoop method to provide exactly that. Of course by selecting the “Off”-RadioButton the titlemusic will stop. Additional to controlling the titlemusic, you are able to control the whole sound with a VolumeSlider. This slider is responsible for the whole sound, even the sound ingame. Java offers a Slider-Object. You need a FloatControl object, which provides control over a range of floating-point values. Float controls are often represented in graphical user interfaces by continuously adjustable objects such as sliders. The ingame sound is produced by the Object Clip which is going to be explained in the next Point “Ingame Sound”.   
   The slider has to be in a valueProperty method with a listener to notice immediately any changes. After “listened” to the slider the sound has to be set directly. So the   
   sound ingame is from now on controllable.
2. **Ingame Sound**The ingame sound is produced by the Object Clip, which is already implemented in Java itself. Every Clip is run in a thread to provide a good procedure. A AudioStream reads the Path of the sound-file, which is saved in a folder within the Game Project. The class Audio is the container for the whole sound for the game.   
     
   The ingame sound includes almost every sound. For example, if the avatar is walking on stone, grass, in a waterpit or if the avatars is swimming. Also sounds like swaying a sword and killing a zombie is implemented.