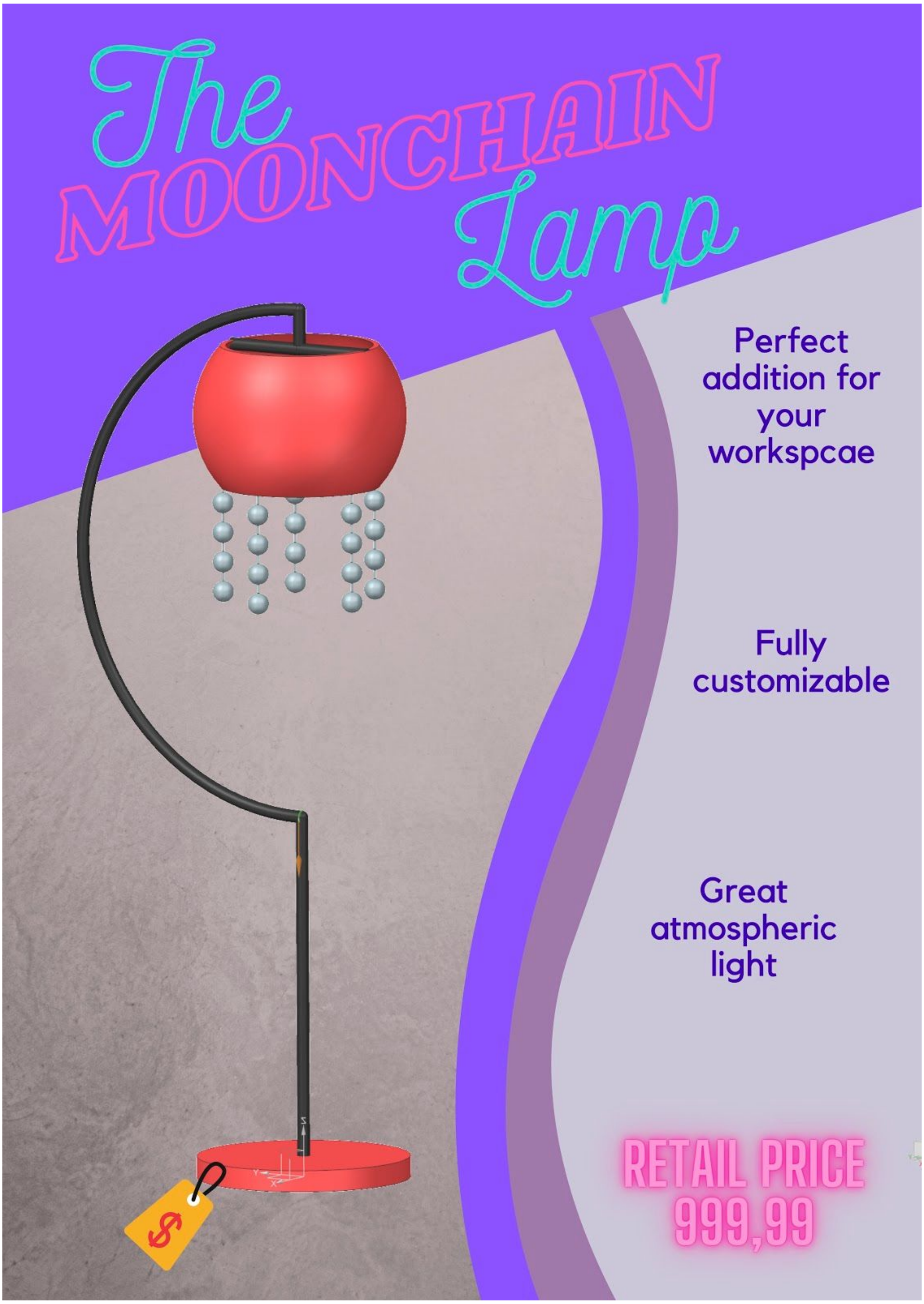
TMM4270 – Eldar Sandanger, Ane Dyveke Blaauw

Assignment 1: Design a table lamp with KF

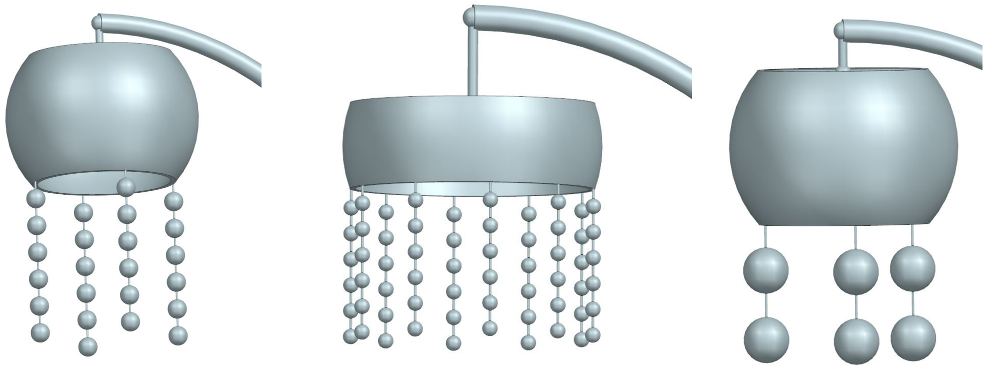


Our product is a flexible design of a table lamp that can be customized by the user in various ways. It consists of a lamp hood with beaded strings hanging around a circular edge, a curved leg that aesthetically surrounds the hood in a half circle and a cylindrical base. The light bulb is placed in the middle of the hood and will, with the help of the beads, create a beautiful atmospheric vibe for your table.

Every parameter is adjustable, from the base to the hood, due to almost every coordinate and vector parameter being explicitly declared in the beginning of the code and then referenced when used. To make sure every point that makes up the skeleton of the shape is correctly calculated, important calculations are performed by methods. The points are dependent on each other from the base to the hood, such that when a change in the base or leg height occurs, the position of the hoods origin point is correspondingly relocated. This ensures that every dependency correlates perfectly and that almost every parameter can be adjusted according to the customers wish.

Lamp hood

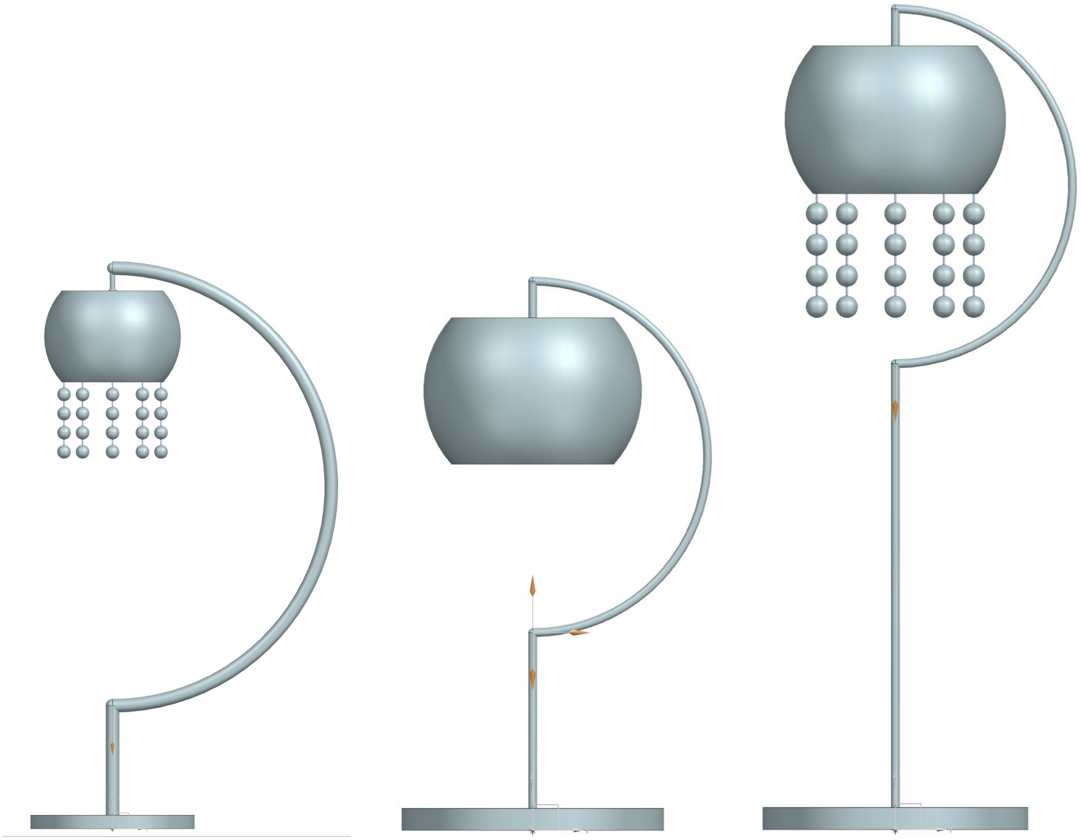
The number of different possibilities to customize the lamp hood are many. For instance, you can change how many beads you want on each string, how big each bead should be, and the spacing between each of them. How many beadstrings you want around the circular edge can also be customized. The lamp hood itself consists of a sphere that is cut off at the top and bottom. The size of the hood can be adjusted both by changing the radius of the sphere, and also by selecting how much you want to be cut off. Here are some samples displaying how different parameters affect the look of the hood:



The bead string origins are generated using a series of methods calculating the angular distribution and corresponding vectors from the center that correlates to a given number of strings. From there, the position of the spheres that make up the beads are calculated based on bead diameter and space between beads that the user supplies.

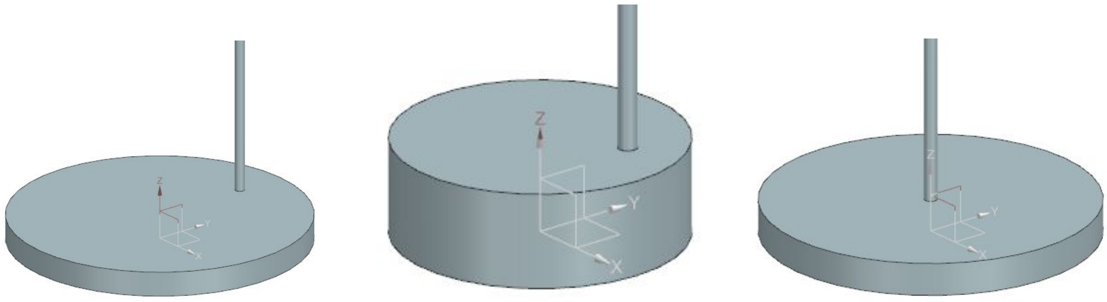
Lamp leg

The lamp leg consists of a cylindrical pole that is split up into a curved part, shaped like a half circle, and a straight part that connects this curve to the base. The conjunction is made from a spherical joint. To change the total height of the table lamp you can adjust the height of the straight part, and the radius of the curved part. The diameter of the pole can also be changed, but to make the design of the lamp smoother we have decided to keep the two parts and the spherical joint the same diameter.



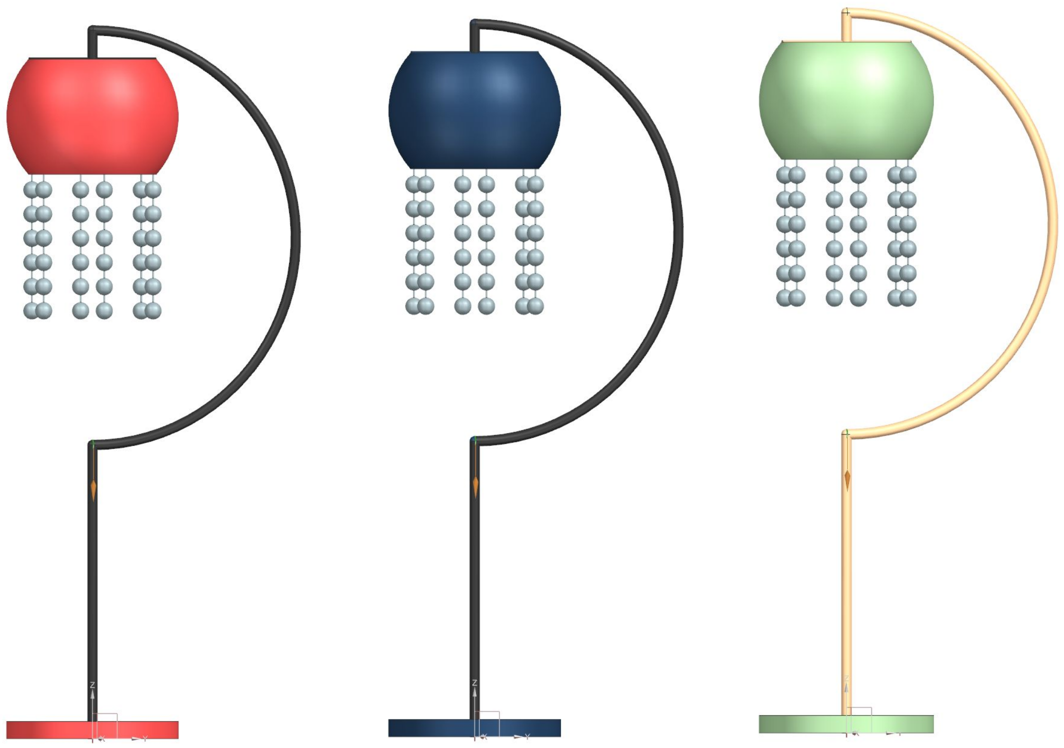
Lamp base

The base takes the shape of a cylinder, and both the height and diameter of this can be adjusted. To spice it up even more, you can also customize where you want the leg to start from on the base.



Changing colors

The colors can also be adjusted to fit the customers wishes. Here are some samples**:**



Further maintenance of the design

Scalability:

The solution uses scaled unit vectors to define the dependencies of each point. Therefore, scaling each part of the design is incredibly simple with a low probability of causing conflict with other parts of the model. This also makes adding new features would also be a relatively simple task. For example, if someone would want to add a figure on top of the lamp, they could use the center of the top conjunction sphere as the reference for a scaled vector pointing to a new point above the lamp.

Naming convention:

Some examples of basic naming convention rules used in the code:

- Methods: thisMethod
- Variables: this_variable
- Child: this_child
- Vector: this_vector
- Point: this_point
- Color: THIS_COLOR

Things that could be improved:

- **Error handling:** During the design process we have down prioritized error handling. There are no exceptions raised if any of the parameters are out of bounds.
- **File organization:** All the code for this lamp is collected in one file. This solution works for now, but if the code were to be further extended it would easily be overly

complex and difficult to follow. Therefore it could be reasonable to split the code into different files. For example, one file could store all the methods, another all the necessary parameters, another on the children etc.

- **Color for loop-generated children:** You can change the color as you like on every part of the lamp, except the beads. This is due to not being able to figure out how to apply colors to loop-generated children, but it's also a feature that could be extended.