

# 3D Schematics of Modelica Models and Gamification

Hilding Elmqvist<sup>1</sup>, Alexander D. Baldwin<sup>1,2</sup>, Simon Dahlberg<sup>2</sup>

<sup>1</sup>Dassault Systèmes, Lund, Sweden, Hilding.Elmqvist@3ds.com

<sup>2</sup>Malmö University, Malmö, Sweden, {alexander.d.baldwin, simondahlberg89}@gmail.com

## Abstract

Block diagrams have been used for a long time to express data flow in dynamic models, i.e. the input output relations between calculation blocks. SysML diagrams are also used to express other relations such as component hierarchy and inheritance. Modelica uses object diagrams, a generalization of block diagrams since acausal connections are allowed. CAD uses a 3D representation to represent the assembly of a mechanism, i.e. how bodies are coupled with joints. This paper describes a generalization of object diagrams, called *3D Schematics*, to utilize 3D representations of the icons/shapes and unification with assembly diagrams and exploded views.

The ideas have been prototyped in a program called Playmola which is inspired by computer games. The goal is to make a model authoring environment that is much more intuitive and fun than existing ones. The hope is that such a tool would be used to promote science for students already in high-school.

**Keywords:** *Block Diagrams, Object Diagrams, MultiBody Assembly, Exploded View, Gamification*

## 1 Introduction

The look and feel of Modelica tools needs to be modernized. It has roots from the 1990s when compromises due to limitations in rendering speed needed to be made with regards to capabilities. As a consequence, a flat 2D graphics representation was introduced in Modelica.

Watching young children experience, handle and enjoy 3D scenes and actions in Minecraft<sup>1</sup> inspired making a completely different 3D environment for building and experiencing Modelica models. The 2D physics environment Algodoo (Algodoo, 2015) has also served as inspiration.

## 2 Modelica Object Diagrams and 3D Schematics

3D schematics as a unification of visual representations of models will be presented by a series of examples.

### 2.1 MultiBody Modeling

An example of how to model MultiBody systems in Modelica is shown in Figure 1. It is a pendulum called the Furuta pendulum.

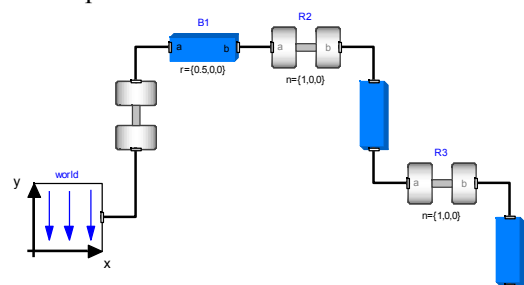


Figure 1. Furuta pendulum model in Dymola

The corresponding 3D schematic of Playmola is shown in Figure 2.

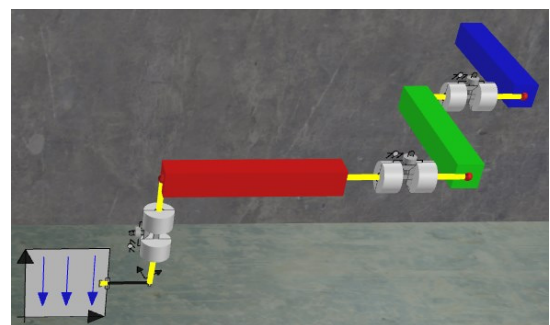


Figure 2. Furuta pendulum model in Playmola

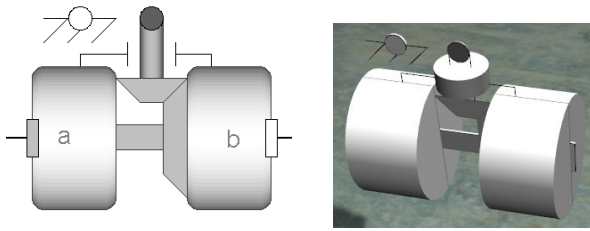
Since it's a 3D view the user can change the camera position to get a different perspective of the Furuta pendulum. It should be noted that the joints are rendered in 3D and are oriented according to the axis of motion. Red spheres represent the connectors.

The 3D representation of the revolute joint has been automatically derived from the Modelica annotation of the icon which contains:

```
Rectangle(  
  extent={{-100,-60},{-30,60}},  
  lineColor={64,64,64},  
  fillPattern=FillPattern.HorizontalCylinder,  
  fillColor={255,255,255},  
  radius=10),
```

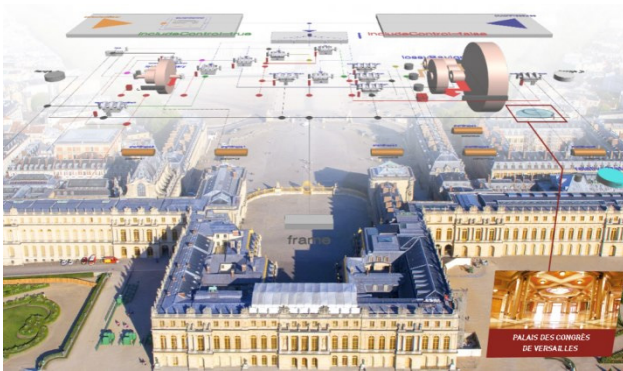
<sup>1</sup> <https://minecraft.net/>

Since the fillPattern is set to HorizontalCylinder, a cylinder is generated for the 3D representation. The standard 2D and the “2.5D” representations are shown in Figure 3.



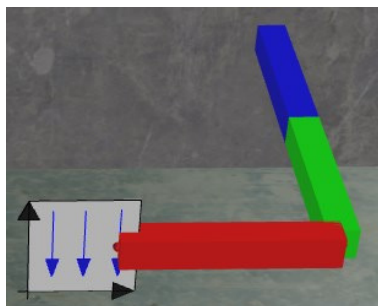
**Figure 3.** Standard 2D and new “2.5D” visualization of the revolute joint

The image for the poster of the Modelica Conference 2015 in Versailles, Figure 4, was generated based on a gearbox model using this “2.5D” technique.



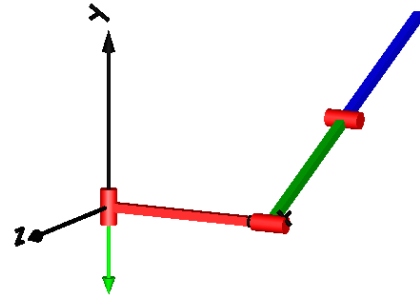
**Figure 4.** Modelica Conference poster using “2.5D” visualization of a gearbox

The view of the Furuta pendulum in Figure 2 can be seen as an exploded view of a 3D assembly drawing. The unexploded view is shown in Figure 5. The joints and connections are then not rendered.



**Figure 5.** Un-exploded view of Furuta pendulum

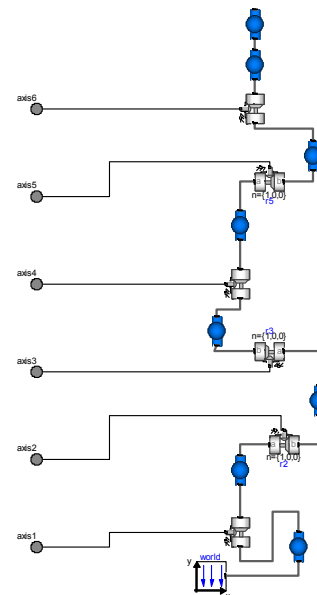
The above view is also the animation view for Playmola. It corresponds to the special animation view of Dymola shown in Figure 6.



**Figure 6.** Animation of Furuta pendulum model in Dymola

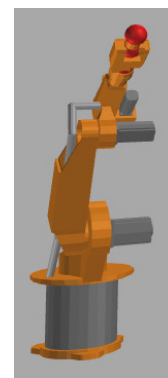
## 2.2 CAD shapes – Robot Model

The mechanics part of the model: Modelica.Mechanics.MultiBody.Examples.Systems.RobotR3.fullRobot is shown in Figure 7.



**Figure 7.** Robot model in Dymola

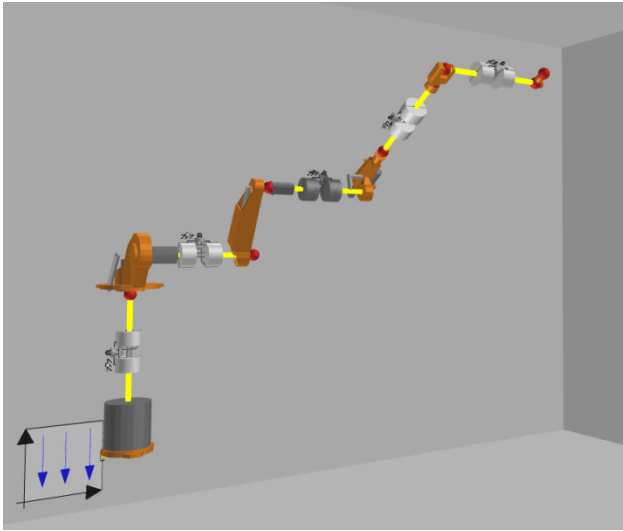
The corresponding non-exploded Playmola model is shown in Figure 8.



**Figure 8.** Un-exploded view of Robot model in Playmola

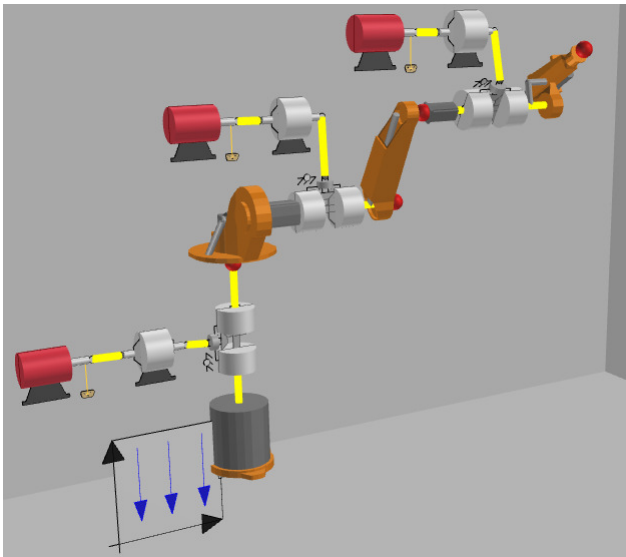
This view uses the actual shapes of the bodies instead of icons and is constructed in the same way as the assembly is made.

Figure 9 shows the exploded view.



**Figure 9.** Exploded view of Robot model in Playmola

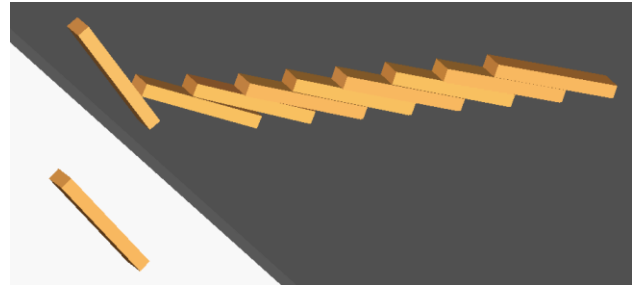
This is a very intuitive visual representation of the robot in which the degrees of freedom are clearly shown. In this representation, additional modeling elements can be introduced such as electrical motors and gearboxes as shown in Figure 10 of the partial robot model (the 3D representations of the motors and gearboxes were automatically generated):



**Figure 10.** Exploded view of Robot model with motors and gearboxes in Playmola

## 2.3 Bodies in Contact

Many real life situations involve bodies in contact or colliding bodies. Playmola utilizes the new functionality available in Dymola for contact handling (Elmqvist, et al., 2015). It is therefore possible to put some boxes on a table (big box), set initial rotation speed on one of the boxes, and experience the domino effect as shown in Figure 11.



**Figure 11.** Domino bricks model in Playmola

## 3 Gamification

### 3.1 Background

Science education has been slow to adopt the use of interactive simulations. (Wieman and Perkins, 2006) are vocal proponents of using interactive simulations as an educational tool in science and argue that traditional forms of education fail to provide students with an understanding of science by suppressing their interest in the subject and failing to engage them. Wieman and Perkins argue that interactive simulations are an effective complement to traditional media in science education – an idea that is supported by research into the use of simulation tools to engage college students (Podolefsky, 2010). According to Wieman and Perkins, the most important features of an engaging interactive simulation tool are:

- Highly interactive animation
- An appealing environment and sophisticated graphics
- Simple and intuitive controls
- Connections to real-life objects (Wieman and Perkins, 2006 p. 291)

Research shows that an effective way of promoting user-engagement and interest in learning environments is the incorporation of elements from digital games (Sabourin and Lester, 2014). The use of game-elements in other contexts, usually referred to as gamification, has been an increasingly popular research topic in recent years, with many studies applying its principles in an educational context (Hamari et al., 2014), (Seaborn and Fels, 2015). The stated motivation for the use of (digital) gamification is often to increase user

engagement and, consequently, user retention in software systems (Deterding et al., 2011).

Most gamification today is implemented using extrinsically motivating elements such as points, achievements and leaderboards - encouraging users to compete for rewards and status. This kind of gamification has been criticised for not being representative of what makes games fun and has been referred to as “pointsification” (Robertson, 2010). Numerous attempts have, however, been made to construct models or frameworks for gamification which focus on intrinsic motivation and fun.

In addition to design-elements from games, game-related technologies such as game engines and 3D-engines can be useful in non-game contexts and have successfully been used in the development of simulation tools on numerous occasions (Bijl et al., 2011). Game engines often boast advanced graphical features with realistic lighting and shadows and support for importing complex animated and textured 3D models, which can be used to provide the appealing environment and sophisticated graphics Wieman and Perkins consider important. Depending on the type of simulation tool to be created, other common features of game engines such as built-in physics engines and artificial intelligence systems can also be useful. These types of features are often provided in a development environment which supports rapid production, commonly with little need for programming.

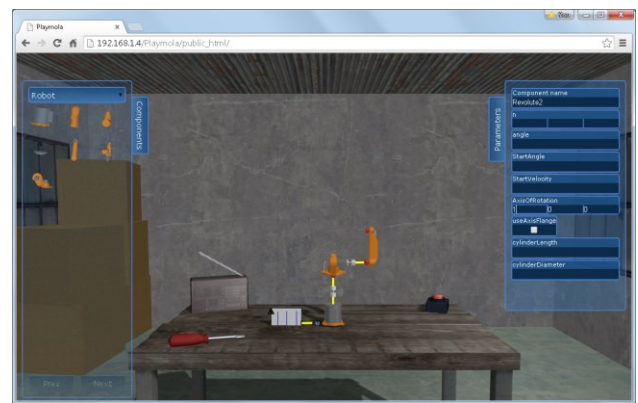
### 3.2 Gamification Frameworks

(Nicholson, 2012) attempts to solve the issue of the negative effects of extrinsic motivators on intrinsic motivation by conceptualising a framework for “meaningful gamification” with a focus on the end-user rather than the organization providing the service, even providing his own definition of meaningful gamification as: “the integration of user-centered game design elements into non-game contexts”. Besides recommending a focus on intrinsic motivators, Nicholson’s framework stresses the importance of context in the use of game elements, referring to the concept of situated motivational affordances (Deterding, 2011), which describes how the motivational effect of a system element depends upon the background of the user and the context in which it is used in the system. Nicholson concludes that meaningful gamification relies on using elements that users with a wide variety of backgrounds can relate to in the right context within a system. In his examples of applications of meaningful gamification, Nicholson suggests removing scoring/rule-based elements and focusing on ‘play’, referring to this as ‘playification’. Deterding et al. refer to this concept as ‘ludification’ and consider gamification a subset of ludification (Deterding et al., 2011 p. 13).

Numerous studies on gamification refer to Malone’s paper (Malone, 1982) on using ideas from games in effective user interface design. While written long before the coining of the term “gamification”, Malone’s ideas still have a lot in common with the aforementioned definitions, particularly Nicholson’s framework for user-centered meaningful gamification. Malone focuses on the user’s enjoyment, defining three principle heuristics for designing enjoyable user interfaces: challenge – the activity should have a clear goal and an uncertain outcome, fantasy – the interface should be “emotionally appealing” and use metaphors that the user can relate to, and curiosity – the interface should provide the right level of informational complexity in order to be novel or surprising while still understandable; it might also use “sensory curiosity”, which refers to the use of audio and visual effects as decorations, to enhance fantasy or as a means of representing aspects of a system.

### 3.3 Use of Gamification in Playmola

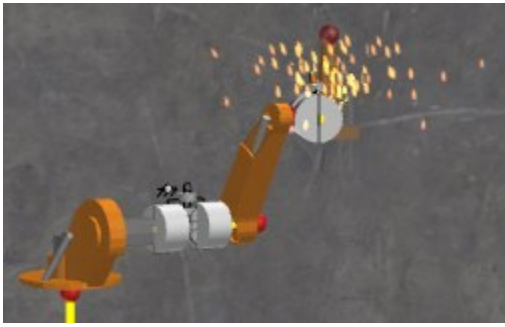
Playmola was constructed with gamification in mind, primarily taking advantage of Malone’s “fantasy” to immerse users in the setting of a workshop where they can experiment and build models in an environment that is both familiar to them (most people have some experience of building things in a garage or shed) and contextually appropriate for the kind of construction being done (mechanical models in the developed prototype), see Figure 12.



**Figure 12.** Garage environment for visual experiments in Playmola

Textured 3D models for the environment (including walls, floor, ceiling, a workbench, a screwdriver and stacked cardboard boxes), particle effects (a welding effect using sparks, see Figure 13) and music and sound effects (specially composed background music and a welding sound effect when components are connected) were all used to invoke this sense of fantasy.





**Figure 13.** Welding a joint in Playmola

Playmola has to a certain extent been inspired by the Kerbal Space Program<sup>2</sup> which allows you to construct and launch a rocket built-in a hangar, see Figure 14.



**Figure 14.** Kerbal Space Program

#### 4 Modeling with Playmola

As demonstrated above, Playmola allows modeling with the predefined 3D objects, Box, Cylinder and Sphere in addition to objects defined by triangular meshes imported from CAD programs. The components are organized in groups, see Figure 15.

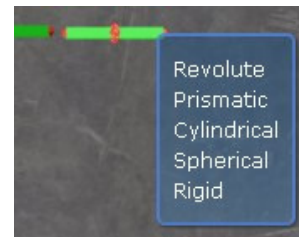


**Figure 15.** Group of Robot parts

<sup>2</sup> <https://kerbalspaceprogram.com/>

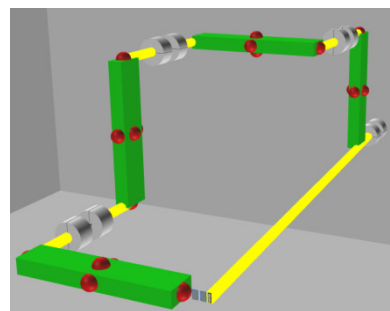
When hovering over the group, the 3D parts are rotated in order to give better perception. It should be noted that the same 3D representation is used in composition mode as in animation mode. In addition, all Modelica library components can be used when modeling. Their visual representation is usually flat and only shown in the exploded view.

When a connector of a part is moved close to a connector of another part, the connectors are highlighted. A dialog is shown (Figure 16) to allow selection of joint type to be inserted.



**Figure 16.** Dialog for selecting joints when connecting parts

When parts and joints are connected in a loop (kinematic loop), special handling in the Dymola solver is required. To simplify for the user, such a situation is automatically detected and currently a strong spring damper is inserted to act as a cut joint and brake the kinematic loop. This cut joint is visualized with a dashed line (Figure 17).



**Figure 17.** Kinematic loop

In order to promote ease of use for younger users, most components used in Playmola are simplified wrappers of components from the Modelica Standard Library, dramatically reducing the number of parameters the user can change to a subset of key parameters that allow the user freedom to experiment without becoming confused by an overload of information.

For example, when a revolute joint is selected a dialog slides in from the right (Figure 18) showing only certain parameters.



**Figure 18.** Simplified dialog for revolute joint

These kinds of constraints are recommended by (Podolefsky et al., 2010) in their discussion of the use of simulation tools in physics education: ‘constraining what students can do reduces cognitive demands and frees up resources for sense making and development of an expert-like mental framework.’ In addition to these simplified components, other components can be loaded as desired.

## 5 User Experiences

### 5.1 First Grader

Playmola was presented to a 7-year old boy. He is interested in machines and had previous experiences with simpler 2D educational games. The first exercise was to construct a double pendulum, i.e. introducing the concepts of setting the size of a box and coupling the parts by means of revolute joints. This was carried out without problems especially since changing a parameter such as length is immediately visualized.

The next exercise was to construct a robot using the set of bodies shown in Figure 15. Axis of motion then becomes very important. The concept of direction was introduced by showing 3 fingers pointing right, up and towards him and corresponding to the 3 input boxes in the dialog. It was explained that 0,0,1 meant choosing the axis towards him and 0,1,0 meant up. It was also explained how to move the parts by setting the StartAngle by a short introduction to the concept of degrees for angles. After that he could build the robot as shown in Figure 9.

### 5.2 Bachelor Students

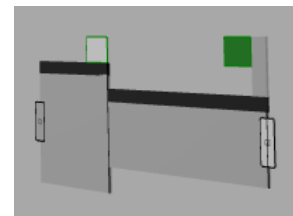
In a small-scale study, eight users were about to recognize Playmola’s theme and felt that it was enjoyable and suitable for the application, using words like ‘charming’ and ‘great’. One user described how the environment actively enhanced the user experience: ‘You come into a calm environment with calming music which isn’t distracting. It makes it easier to concentrate on what you actually want to do.’

The use of a single, unified 3D view for construction and animation of models - instead of the disconnected 2D view for construction of models and 3D view for animation of simulation results seen in Dymola -

appears useful from the perspective of helping new users understand the application. All users agreed that the 3D environment made it easier for them to visualize the models, one pointing out that the unified view made it easy to tweak parameters in a model and get direct visual feedback reflecting the changes. Observations also showed that the ability to view the complete model in 3D allowed users to quickly discover the source of any issues.

## 6 Proposed Extensions to Modelica

As demonstrated above, Rectangle with fillPattern=HorizontalCylinder can be interpreted as a Cylinder enabling the “2.5D” representation. Other graphical primitives such as Ellipse, Rectangle and Polygon just get a fixed small extrusion in the z-direction. This means that, for example, the icon of the Prismatic joint does not look right, see Figure 19.



**Figure 19.** Standard prismatic joint rendered in Playmola

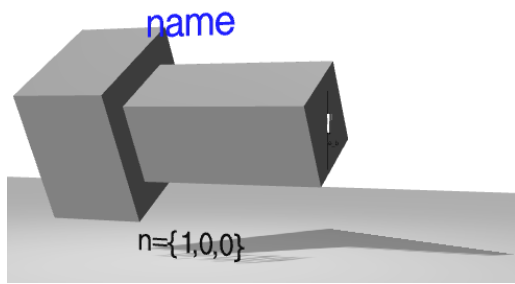
We propose that general 3D primitives are introduced in the graphics annotations of Modelica. In particular, a triangular mesh representation gives a very flexible way of defining the component shapes.

The paper (Elmqvist, et al., 2015) presents details about modeling with triangular meshes. A popular representation is an array of 3D vertices and an array of triangles defined as 3 indices into the vertices array. A set of functions are defined for building triangular meshes of Box, Cylinder, Sphere and other elementary shapes. Transformation functions for translating, rotating and scaling of shapes are defined. In addition, functions from extrusion of shapes from polygons are available. Finally the Constructive Solid Geometry (CSG) operations, union, difference and intersection are defined.

By replacing the four rectangles of the Prismatic joint icon graphics by the following annotation:

```
TriangleMesh(mesh=
  rotate(
    union(
      translate(
        Box(size={200, 60, 60}),
        r={-100, -30, 30}),
      translate(Box(size={70, 100, 100}), r={-100, -0, 50})),
    angles={3.14159265/6, 0, 0}),
  triangleColor={175, 175, 175})
```

the visual representation of Figure 20 of the Prismatic joint is achieved.



**Figure 20.** Prismatic joint using triangular mesh

The calls to the function `Box` return 12 triangles each. These are translated and combined by the CSG union operator to a new triangular mesh representing the union. Finally this solid is rotated. A default argument might be introduced, in addition to the mesh argument, with a 2D standard representation for those tools that does not support 3D primitives.

In addition to this triangular mesh primitive extending the visual representation to 3D, also the coordinate system need to be extended, for example:

```
coordinateSystem(
  extent3D={{-100,-100,-100},{100,100,100}})
```

Furthermore, the Placement annotations of components need to be extended to 3D positioning, sizing and orientation, for example (with `qx`, `qy` and `qz` representing the direction of the local `x`, `y` and `z` axis):

```
Placement(transformation3D(
  position={-50,50,10},
  size={10,10,10},
  rotation={qx, qy, qz}))
```

The position and rotation of connectors in 3D is especially important to enable assembly operations where the connected components immediately rotate to be correctly assembled.

## 7 Playmola Architecture

Playmola is designed to run in a web browser and it communicates with Dymola using remote JavaScript calls encoded in JSON format. Dymola 2016 version (Dassault Systèmes, 2015) has support for remote Modelica function calls performed in Java, JavaScript or Python. This API has been extended by certain new functions.

Three.js (Dirksen, 2013), (three.js, 2015), a popular JavaScript 3D rendering API based on WebGL, was chosen for Playmola's development since it provides the required graphical functionality. Dymola was enhanced to include functions for exporting 3D models in a format compatible with three.js. The built-in VRML loader was extended with extra functionality to be able to load 3D models in VRML format.

Playmola imports data from Dymola by the means of a client/server relationship. This architecture is used in order to load component models defined in Dymola into Playmola, to send the parameterized component instances back to Dymola in the form of generated Modelica code, for performing simulation and to query frame data for the result animation.

jQuery Mobile (jQuery Foundation, 2015) was used for GUI elements and input handling, which, together with the cross-platform nature of HTML5, allows Playmola to be used on multiple devices. The prototype was developed on browsers running on a PC, but also runs on iPad (Figure 21) and other touch-based devices.

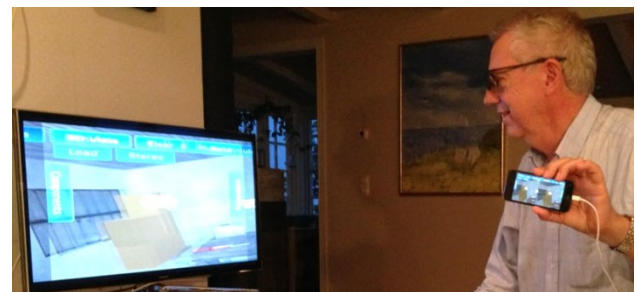


**Figure 21.** Playmola on the iPad

## 8 Stereo Viewing and Virtual Reality

In order to become more interesting and allow a 3D experience, Playmola can be run in stereo mode for 3D viewing. The rendering package used, Three.js, supports side-by-side rendering of the scene.

Figure 22 shows running Playmola on an iPhone, transferring this stereo image to a 3D TV using Apple TV. The TV is set to stereo side-by-side mode and active 3D glasses are used for viewing the 3D content.



**Figure 22.** Stereo image on iPhone and 3D TV

The stereo side-by-side rendering is also what is needed for virtual reality gear such as Oculus Rift or the low cost Google Cardboard. Google Cardboard



mounts a smart phone in front of your eyes. Head motion is tracked and controls the viewing. It is possible to make compositions by looking at an object, pressing the button on the Cardboard (Google Cardboard V2 is needed for iPhone), looking in the direction of the destination and releasing the button, Figure 23. The white circle in the middle of the view represents the focus point.



**Figure 23.** Google Cardboard and side-by-side projection

We have tested the Leap Motion controller that tracks your fingers. However, our initial assessment is that the resolution is not sufficient for gripping objects when authoring models.

We believe Cardboard will revolutionize education. It is then important that the teacher and the students can be in the same virtual reality. We are currently investigating technologies to enable sharing the same scene. It means that all clients communicate with a scene server that broadcasts any changes to the scene.

As the next step, we are eager to start testing HoloLens from Microsoft since it supports *augmented reality*. It means that the teacher and the students can be seen in the virtual model and can act on it.

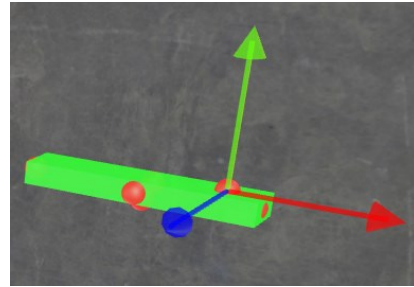
## 9 Future Work

The work is continuing and this section will summarize some of the features in the pipeline.

Traditionally, a mechanism assembly is done by putting joints between frames of bodies. However, it makes sense from an object oriented point of view to associate more information with the bodies. For example, a bar that has a cylindrical hole is prepared for mating with another body with a hole, i.e. it is prepared to have one rotational and optionally one translational degree of freedom. By associating this information with the body, the tool can automatically insert either a Revolute or a Cylindrical joint when connected to another body. Standardized base classes with predefined properties for such *smart connectors* will be used to store such information about degrees of freedom.

In order to better support direct manipulation and virtual reality authoring without keyboard, we are

working on widgets for positioning, rotation, scaling and setting DOFs. Figure 24 shows the positioning tool which is inspired by such a tool in Unity.

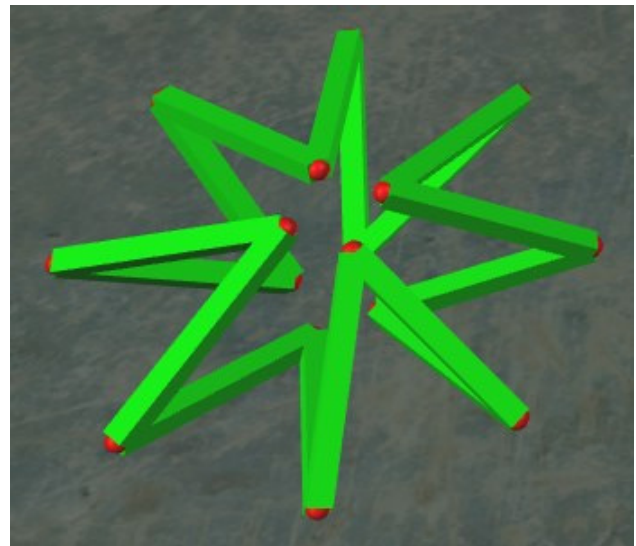


**Figure 24.** Positioning tool

Orientation in 3D is complex to comprehend for students. However, it is important when building and initializing complex mechanisms such as a Stewart platform. Therefore support should be given in Playmola.

The Turtle graphics methodology is a good way of learning 2D computer graphics. It is based on two commands, *forward* a certain distance and *rotate* a certain angle plus *repetition*. For *3D Turtle graphics*, two rotation commands are needed such as Roll and Turn (Verhoeff, 2009). Roll is rotation around the x-axis and Turn around the z-axis.

By using 16 bars and connectors with Roll( $\pm 90^\circ$ ) and Turn( $130.06^\circ$ ), it is possible to build a regular 3D polygon as shown in Figure 25 (Verhoeff, 2009).



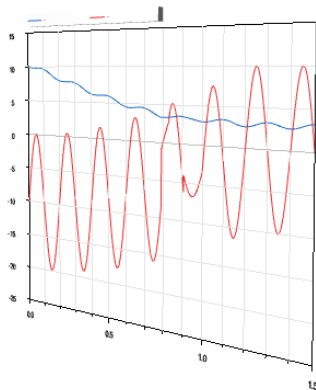
**Figure 25.** Regular 3D polygon

In order to build such regular structures we plan a Repeat Paste command which pastes a certain number of levels and makes the connections.

Playmola will also allow the inclusion of *plots* in the environment, i.e. on the walls, on the computer or on



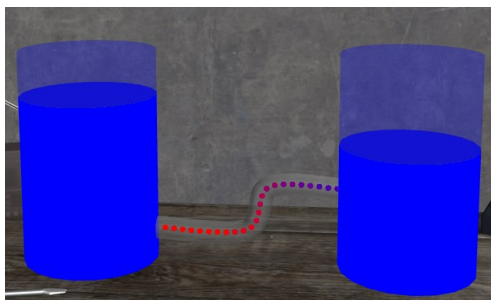
the desk. The plots are regular 3D object although flat, i.e. they have position, orientation and scale 3D transformation as shown in Figure 26.



**Figure 26.** Plot as 3D object

3D is not only important for mechanical systems. We have started to experiment on how to handle *fluid systems*. Interestingly, some of the data associated with a mechanical connector, i.e. position and orientation are also needed for fluid components such as tanks and reactors. For pipes, also the routing is important.

Regarding animation of fluid flows and their properties and electrical flows, we will use the particle simulations available in packages such as three.js. The speed of particles corresponds to the flow rate and color can be used for temperature as shown in Figure 27. The level in a vessel can conveniently be handled by the shaders of the GPU by making the vessel transparent and filling the vessel up to a certain level when rendering the fluid.



**Figure 27.** Flow and level animation.

## 10 Conclusions

A completely different and more modern 3D representation of Modelica models is proposed. It will help professionals to comprehend complex models and is more attractive, intuitive and fun for new-comers.

These ideas have been prototyped in a web application which uses Dymola in server mode for creating 3D representations and for performing simulations.

The support for low cost virtual reality gear such as Google Cardboard will enable faster adaption of this kind of modeling and simulation techniques in education.

## Acknowledgements

This work has partly been performed as a bachelor degree project at Malmö University. The first author served as an industrial advisor and Olle Lindeberg as the formal supervisor.

The authors want to thank Carl Fredrik Abelson for extending Dymola with needed remote JavaScript calls for various functionalities and for exporting WebGL code for 3D Modelica representations.

The authors also want to thank Martin Malmheden, who has been heavily involved in the discussions on how to continue the work after the bachelor degree project, for his contributions.

## References

- Algodoo (2015): <http://www.algodoo.com/>
- Bijl, Jonatan L. and Boer, Csaba A. (2011): Advanced 3D Visualization for Simulation Using Game Technology. In: Proceedings of the Winter Simulation Conference. WSC'11. Phoenix, Arizona: Winter Simulation Conference, 2011, pp. 2815–2826.
- Dassault Systèmes (2015): Dymola 2016. <http://www.Dymola.com>
- Deterding, Sebastian. (2011) Situated motivational affordances of game elements: A conceptual model. In: CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
- Deterding, Sebastian et al. (2011): From Game Design Elements to Gamefulness: Defining “Gamification”. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. MindTrek '11. New York, NY, USA: ACM, 2011, pp. 9–15
- Dirksen, Jos (2013): Learning Three.js: The JavaScript 3D Library for WebGL. Packt Publishing (October 2013)
- Elmqvist H., Goteman A., Roxling V., Ghandriz T. (2015): Generic Modelica Framework for MultiBody Contacts and Discrete Element Method. Proceedings 11th International Modelica Conference, Versailles, September 21-23, 2015.
- Hamari, J., Koivisto, J., and Sarsa, H. (2014): Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In: System Sciences (HICSS), 2014 47th Hawaii International Conference on. Jan. 2014, pp. 3025–3034.
- jQuery Foundation (2015): jQuery Mobile <https://jquerymobile.com> (visited on 05/16/2015).
- Malone, Thomas W. (1982): Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games. In: Proceedings of the 1982 Conference on Human Factors in Computing Systems. CHI '82. New York, NY, USA: ACM, 1982, pp. 63–68.
- Nicholson, Scott. (2012): A User-Centered Theoretical Framework for Meaningful Gamification. In: Proceedings

- of Games+Learning+Society 8.0. Madison, WI, USA, 2012.
- Podolefsky, Noah S., Perkins, Katherine K., and Adams, Wendy K. (2010): Factors promoting engaged exploration with computer simulations. In: Phys. Rev. ST Phys. Educ. Res. 6.2 (Oct. 2010), p. 020117
- Robertson, Margaret. (2010): Can't play, won't play. <http://hideandseek.net/2010/10/06/cant-play-wont-play/> (visited on 04/07/2015).
- Sabourin, J.L. and Lester, J.C. (2014): Affect and Engagement in Game-Based Learning Environments. In: Affective Computing, IEEE Transactions on 5.1 (Jan. 2014), pp. 45– 56.
- Seaborn, Katie and Fels, Deborah I. (2015): Gamification in theory and action: A survey. In: International Journal of Human-Computer Studies 74 (2015), pp. 14–31.
- three.js (2015): <http://threejs.org/> (visited on 04/12/2015).
- Verhoeff T., Verhoeff K. (2009): Regular 3D Polygonal Circuits of Constant Torsion. Bridges 2009: Mathematics, Music, Art, Architecture, Culture. <http://archive.bridgesmathart.org/2009/bridges2009-223.pdf>
- Wieman, Carl E. and Perkins, Katherine K. (2006): A powerful tool for teaching science. In: *Nat Phys* 2.5 (May 2006), pp. 290-292.