### AMP(1) - Lab12 – Scene Graphs

# Content

# Learning objectives

## Exam objectives

By the end of this lab you should be able to (pen and paper):

* Design the object tree of a scene graph rooted in 2D World Space, with all of its subsequent (parent-child) local space nodes
* Construct the embeddings which tie up the total scene graph
* Design the object tree of a bone structure rooted in 2D World Space, with all of its subsequent (parent-child) bone space nodes
* Construct the embedding matrices between bones which tie up the structure
* Design the object tree of a solar system rooted a central star, with all of its subsequent (parent-child) planet space nodes
* Construct the embedding transformation matrices between planets

We advise you to **make your own summary of topics** which are new to you.

## Supportive objectives

### Self-support by GeoGebra

More specifically related to the above you should in GeoGebra:

* Construct the scene graph in GeoGebra
* Visualize the scene graph in GeoGebra
* Organize angular sliders in GeoGebra to demonstrate the scene graph

# Exercises

Dependent of the lab session you may work individually or teamed (organized by the lab attendant). In either case make sure that throughout the course of this lab, you backup sufficiently your solution file on your local machine as

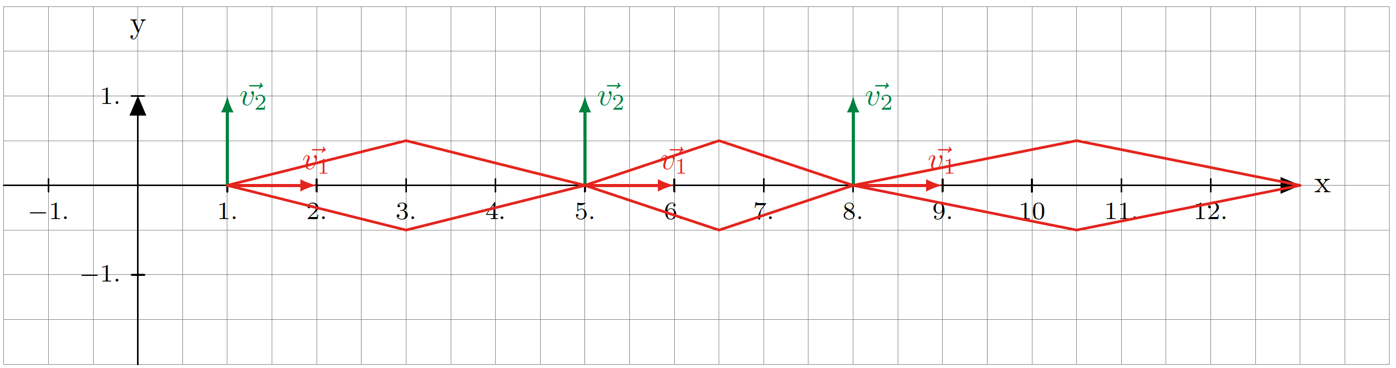
**1DAExx-0y-name1**(+name2+name3).GGB given **xx**=groupcode, **0y**=labindex

If not already on your machine, get **GeoGebra Classic 5.0 or 6.0** via <https://www.geogebra.org/download>

## Contextual practice

### Bone structure

Below you see a representation of three bones B1, B2, B3 that model a robot arm. These bones B1, B2, B3 are respectively 4, 3 and 5 long. The first bone’s position is in point (1,0). In this way the (x,y)-World Space contains three embedded Local Spaces:



We model the bones itself by scaling the blueprint diamond B0 accordingly to their required size:

B0 =

Exercise 1: Design the (parent to child) **object tree** for this robot arm

Exercise 2: Determine the **embedding tranformations E** that link each successive Local Space of the scenegraph. Firstly, do this pen and paper for

B1 constantly 20° inclined (instead of horizontally in the World Space), with

B2 variably ° inclined within its B1-Local Space, and finally

B3 variably ° inclined within its B2-Local Space.

– Scaling has to be applied after the Embedding transformation.

– **Translation matrices**

T1 = First Translation is 1 in the X coord (to reach the first point of B1).

T2 = For B2, there is a translation of 4 in the X to reach the first point of B2 (within the local space of B1).

– E1 = T1 \* R1 → Model1 = E1 \* model0.

Bone1 = E1 \* Scale1 (Applymatrix(E1\*Scale1, Bone0).

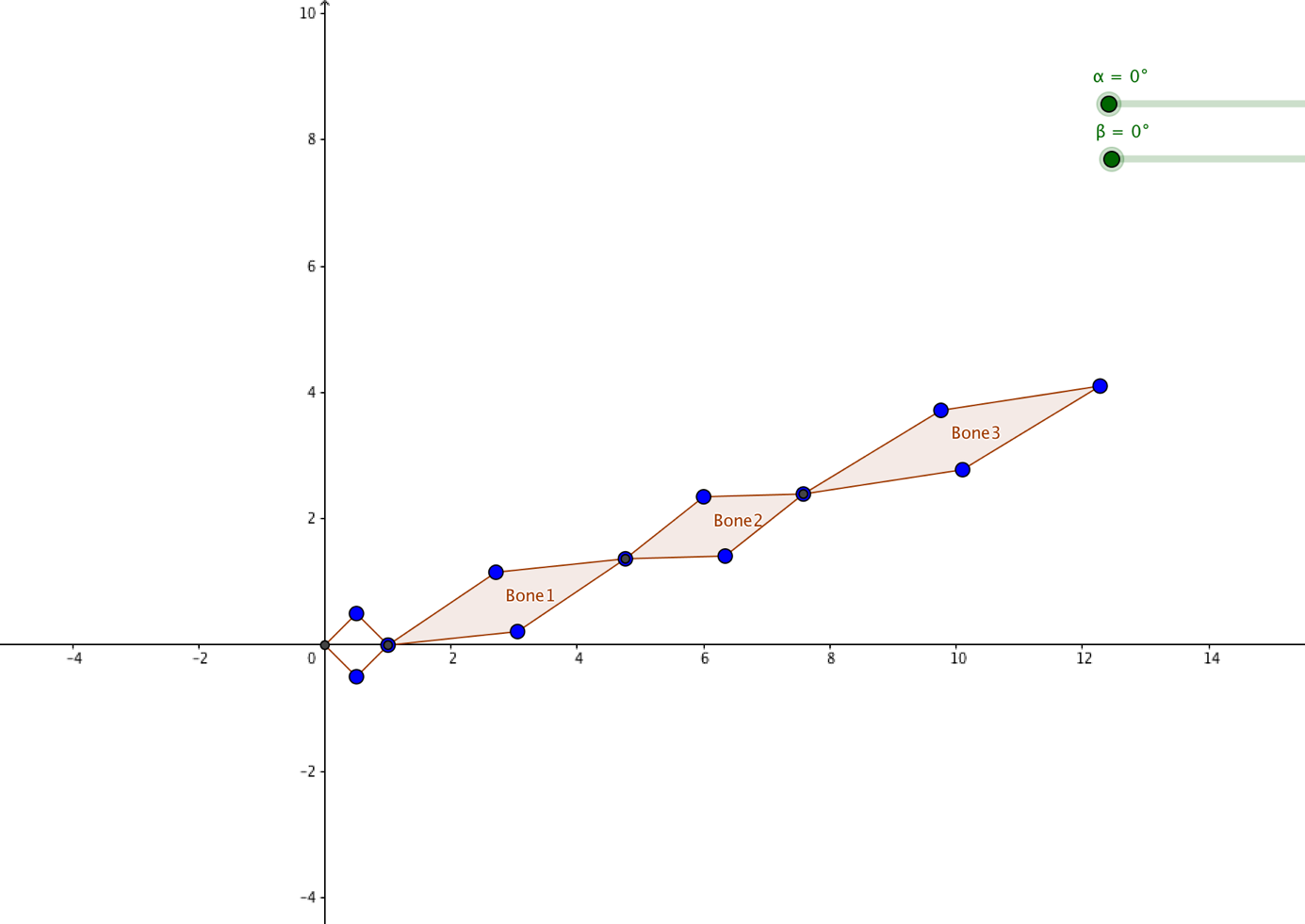
– For the rotation matrices, you can create an alpha slider and apply it to the rotation matrices like (cos alpha) so it will change as you change the slider.

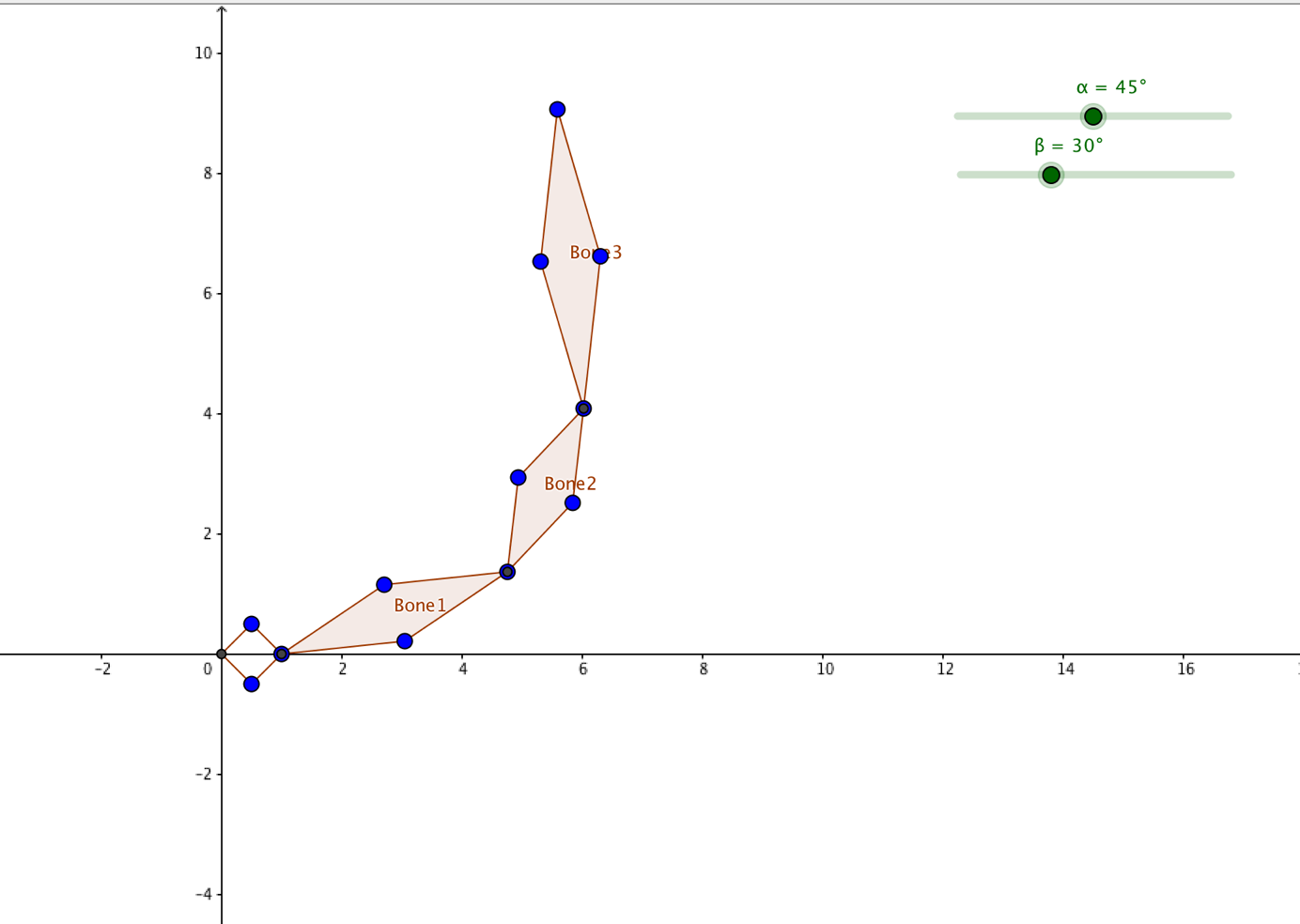
**Hint:** given diamond B0 and the above pictured representation of the robot arm, retrieve the respective scale transformations creating the bones B1, B2, B3. Moreover, sufficiently allow each bone to take its required space via an appropriate translation for that.

Firstly - given the previous hint - we tackle the required scale transformations:

Secondly, we determine all subsequent Local Space - linking transformations:

Exercise 3: Implement all the previous in GeoGebra, realizing a **GeoGebra-simulation** of the robot arm, by means of the two (between 0° and 90° clipped) angular sliders (for respectively  and )**.**

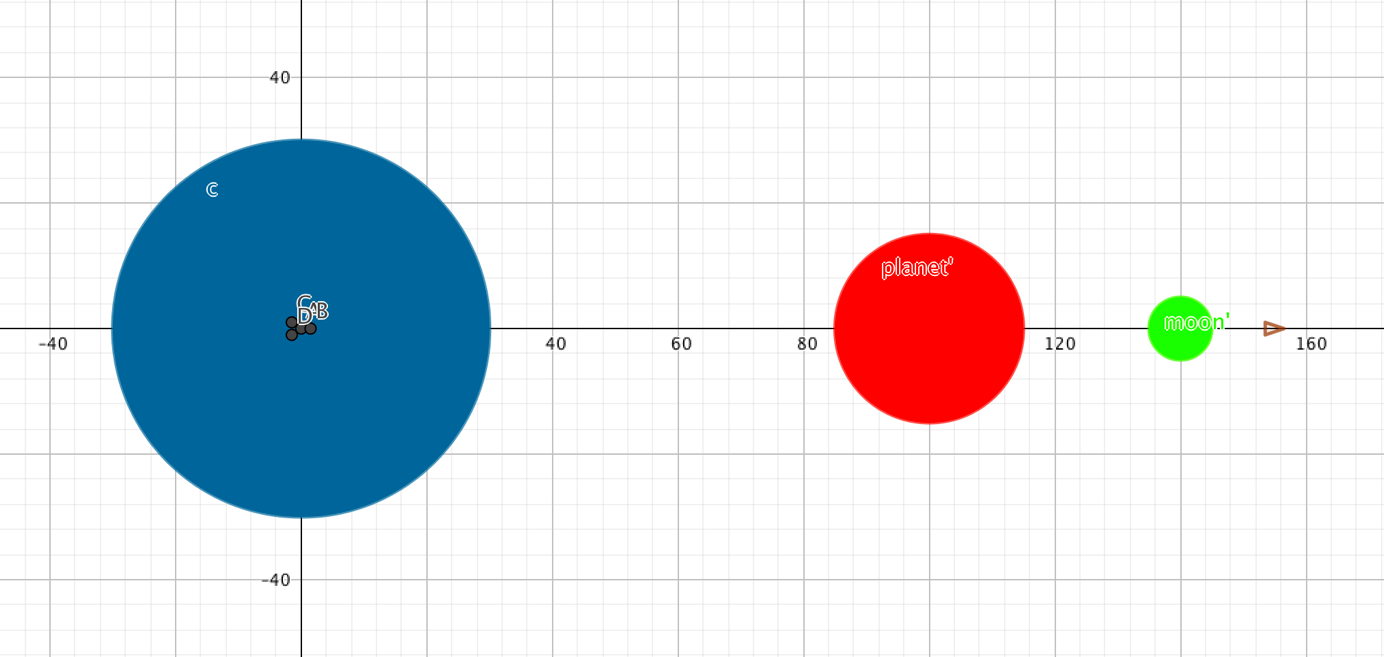




### Solar system

Below you see a representation of a solar system featuring a central star with a radius of 30. A planet with a radius of 15 is circling the star at a distance of 100 between the star center and the planet center. The entire planetary orbit counts 350 days. A moon with radius 5 is circling the planet at a distance of 40 and the lunar orbit counts 50 days. Around the moon an isosceles space craft (with its modeled coordinates below) is circling the moon at a distance of 15 and completes its orbit in only 5 days.

In this way the (x,y)-World Space contains three embedded Local Spaces:



We model the said isosceles space craft by its vertices:

Craft =

Exercise 4: Design the (parent to child) **object tree** for this solar system.

– Initial circle is the unit circle with radius 1. Set up in the geogebra with a filled circle and different color (object properties on the object). From this object we will apply the transformations.

– **Scaling Matrices**

Scaling matrices ScaleS (sun), ScaleP(planet), ScaleM(moon)

We apply ScaleS to circle0 to get the sun scaling.

– **Translation Matrices**

Translation for the planet is 100

Translation for the moon is 40

– **Rotation matrices**

Create a slider for the days it takes the planet to orbit. (from 0 to 350 days)

We have to calculate the angle rotation for the rotation of the planet.

We use the days for this.

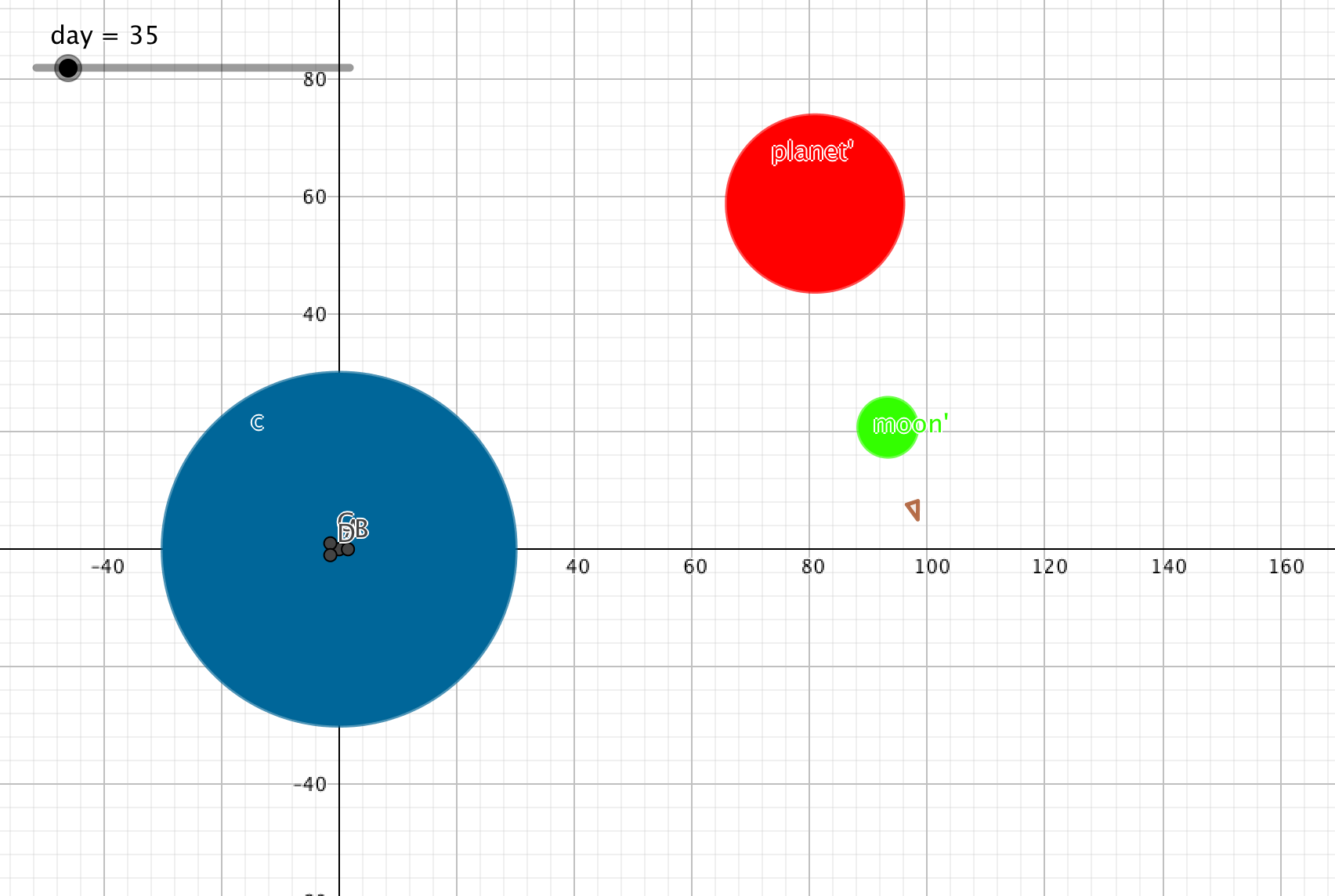
Calculate the emb for the planet taking into account that first rotate and then translate.

After that you applyMatrix calculating the embP \* scaleP and you apply this to the circle0 (this is for the visualitation on Geogebra).

Exercise 5: Determine the **embedding** **tranformations E** that link each successive Local Space of the scenegraph making use of the variable ‘**day**’ where appropriate.

Let us now specify all subsequent Local Space - linking transformations:

Exercise 6: Implement all the previous in GeoGebra, realizing a **GeoGebra-simulation** of the solar system, by means of a (between 0 and 350 clipped) slider for the variable ‘**day**’**.**



Exercise 7 (EXTRA): Extend the previous exercise 6, by making the space craft finally spinning around its centroid C every 4 days**.**

# Referecences

## Basics

### English maths dictionary

[http://www.mathwords.com](http://www.mathwords.com/)

## Demos in art and programming

### Scenegraphs in game engines

<https://www.haroldserrano.com/blog/the-purpose-of-a-scenograph-in-a-game-engine>