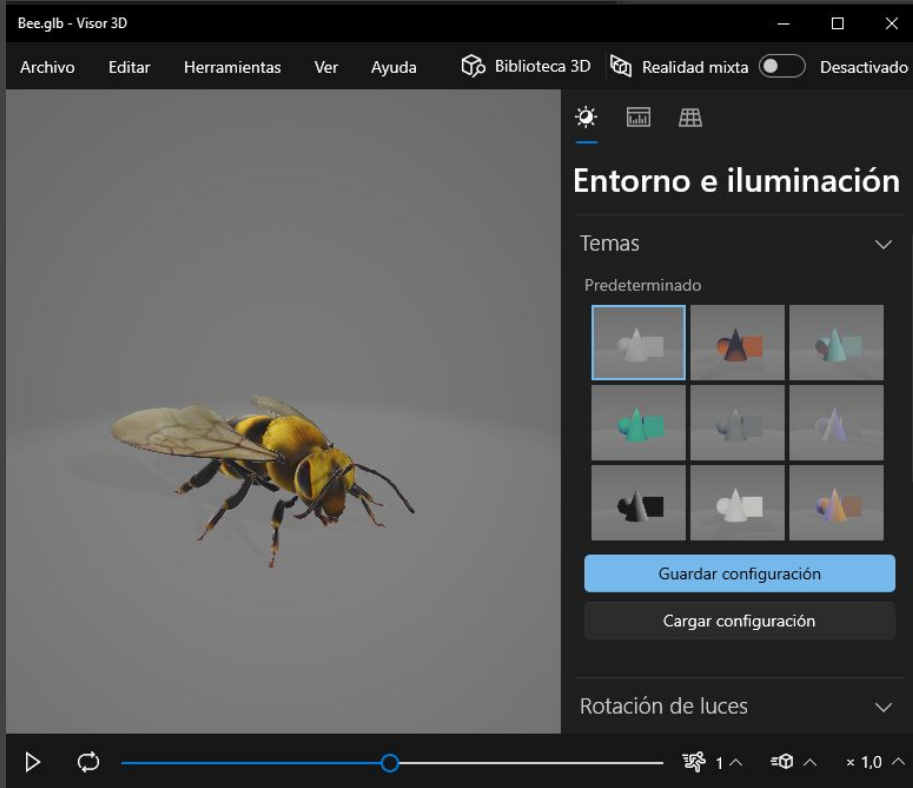




2nd Workshop

Object component. 3D Builder. Animations





Exported from 3D Builder

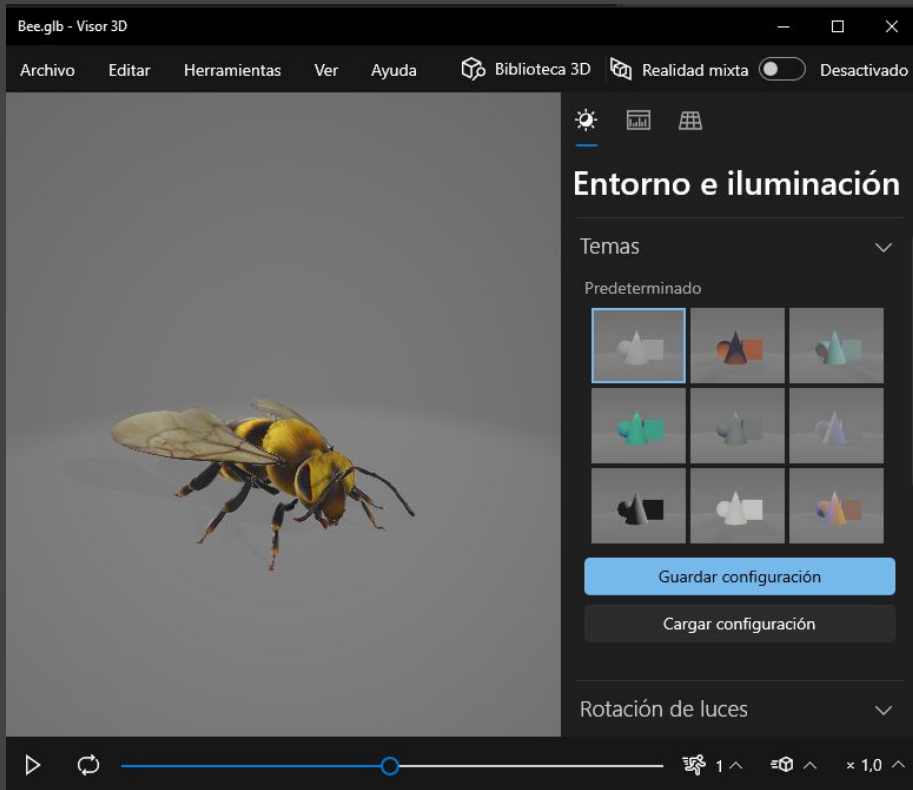
o Object.1

v 2600.481445 2289.817383 8840.607422 255 255 255

v 3052.760742 3579.895996 9028.778320 255 255 255

v 2974.678223 3501.814453 9165.332031 255 255 255

- A model in 3D is just a set of data that represents an object in three dimensions.
- There are different kinds of formats for 3D objects: obj, fbx, gltf, etc.
- Sometimes an object is defined using more than just one file, like mtl files.
- Basically, any 3d object gives information about all the points that define that 3d object, the more detailed the object is modelled, the more points it has.



Exported from 3D Builder
o Object.1

v 2600.481445 2289.817383 8840.607422 255 255 255
v 3052.760742 3579.895996 9028.778320 255 255 255
v 2974.678223 3501.814453 9165.332031 255 255 255

- Obj models are simpler, they have just the points information and color and optionally a mtl file with the information for the textures and materials.
- Gltf (glb or glt) files are more complex, they offer also information about:
 - Hierarchical objects
 - Scene information (light sources, cameras)
 - Skeletal structure and animation
 - More robust materials and shaders
- Other formats are possible, but safest is always obj and mtl.



<a-entity> tag

```
<a-assets>
```

```
<a-asset-item id="tree"
```

```
src="/path/to/tree.gltf">
```

```
</a-asset-item>
```

```
</a-assets>
```

```
<a-entity
```

```
gltf-model="#tree">
```

```
</a-entity>
```

- In order to use gltf objects, they must be included as `<a-assets>` tag.
- They need to have an unique id for identification.
- This id will be used then at the `<a-entity>` tag for loading.
- An `<a-entity>` tag works the same way as regular objects in terms of position, rotation, and scaling.

```
<a-assets>
```

```
<a-asset-item id="tree-obj"
```

```
src="/path/to/tree.obj">
```

```
</a-asset-item>
```

```
<a-asset-item id="tree-mtl"
```

```
src="/path/to/tree.mtl">
```

```
</a-asset-item>
```

```
</a-assets>
```

```
<a-entity
```

```
obj-model="obj: #tree-obj;
```

```
mtl: #tree-mtl">
```

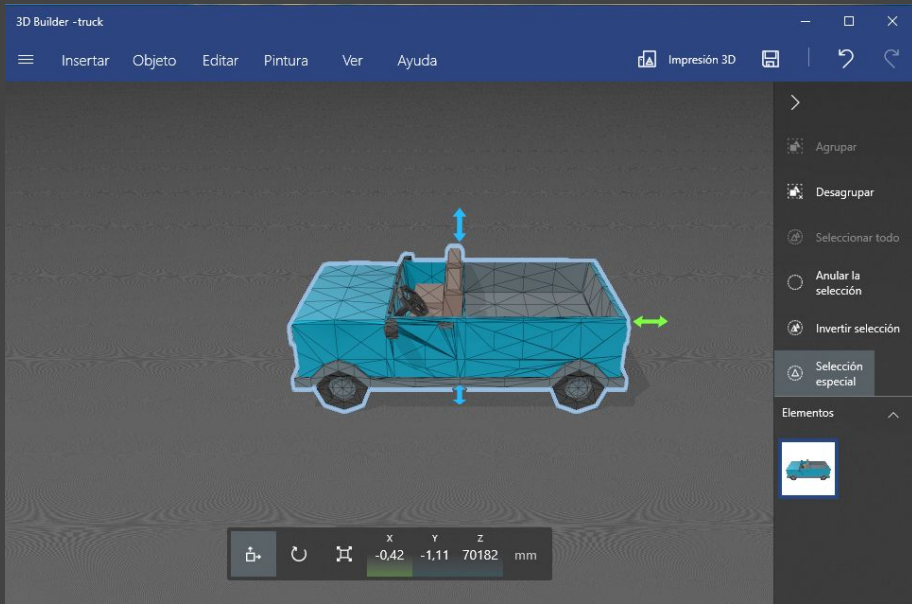
```
</a-entity>
```

- In order to use obj objects, they must be included as <a-assets> tag.
- They need to have an unique id for identification.
- This id will be used then at the <a-entity> tag for loading.
- An <a-entity> tag works the same way as regular objects in terms of position, rotation, and scaling.
- If an object has more than one file, all of them must be included in the <a-assets> tag and their ids must appear on the obj-model entity

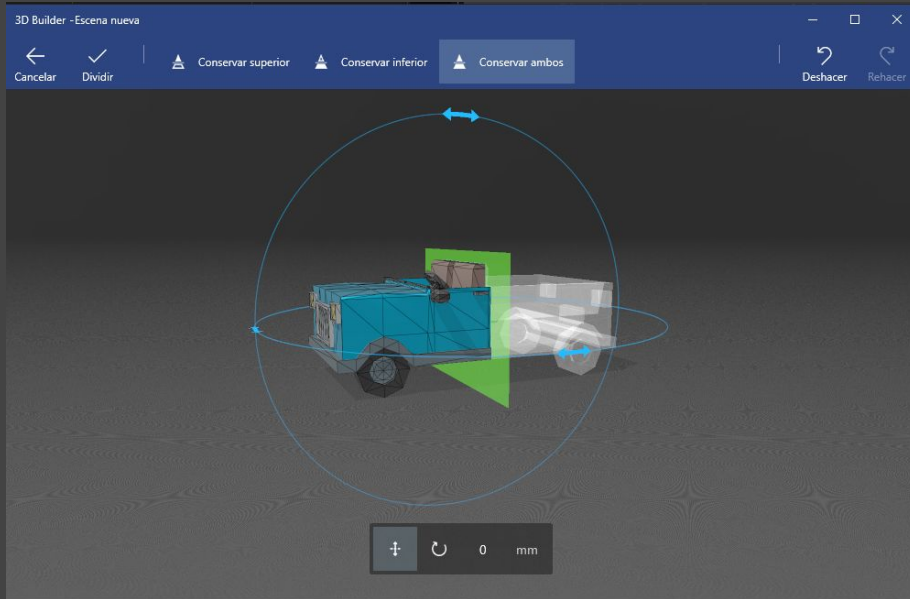


RIT
REALITIES IN
TRANSITION

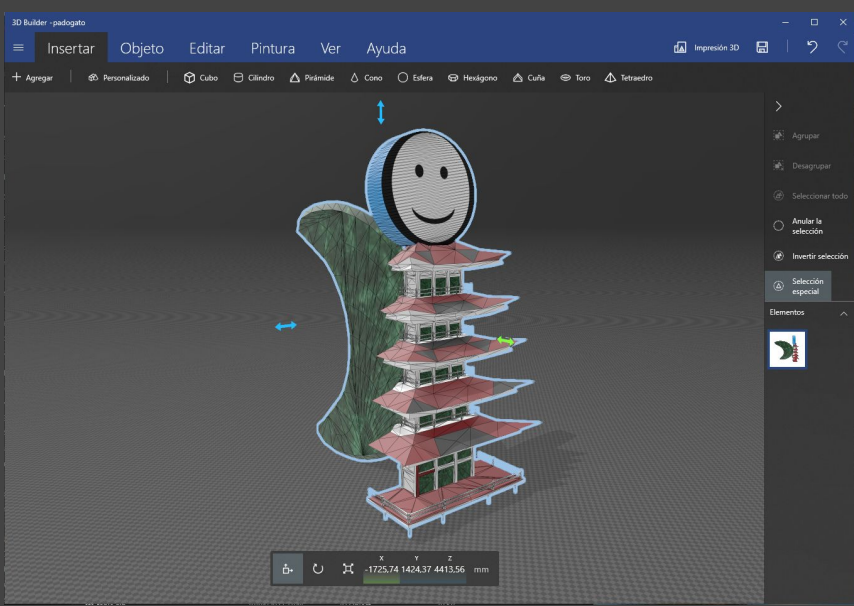
3D Builder



- 3D Builder is an application free for Windows, that edits 3d models.
- It is primarily designed for 3d printing not for modeling but does makes it simpler to use.
- On the top menu we have: Insert, Object, Edit, Paint, View and Help, Print and Save.
- On the down menu we have: position, rotation and scale tools.



- *Insert*: Allows to import more than one object.
- *Object*: Allows us to copy, mirror or eliminate an object.
- *Edit*: Allows us to split an object into two different objects and to combine them.
- *Paint*: Allows us to modify the color and add textures to our objects.
- *Save*: Allows us to export the new edited object into various 3d formats. Best option .glTF



This options will provide us with the ability of combining different models to create new ones.

On the side image we can see a new model created via other models. It is the combination of a moon, a pagoda tower and a cylinder.



That is the idea of recycling the AI in the models. Recycled art is an aesthetic of its own.

Imagine There Is No Garbage by Kushane Chobanyan



RIT
REALITIES IN
TRANSITION

Animation property

```
<a-box
```

```
  position="-1 1.6 -5"
```

```
  animation="
```

```
    property: position;
```

```
    to: 1 8 -10;
```

```
    dur: 2000;
```

```
    loop: true">
```

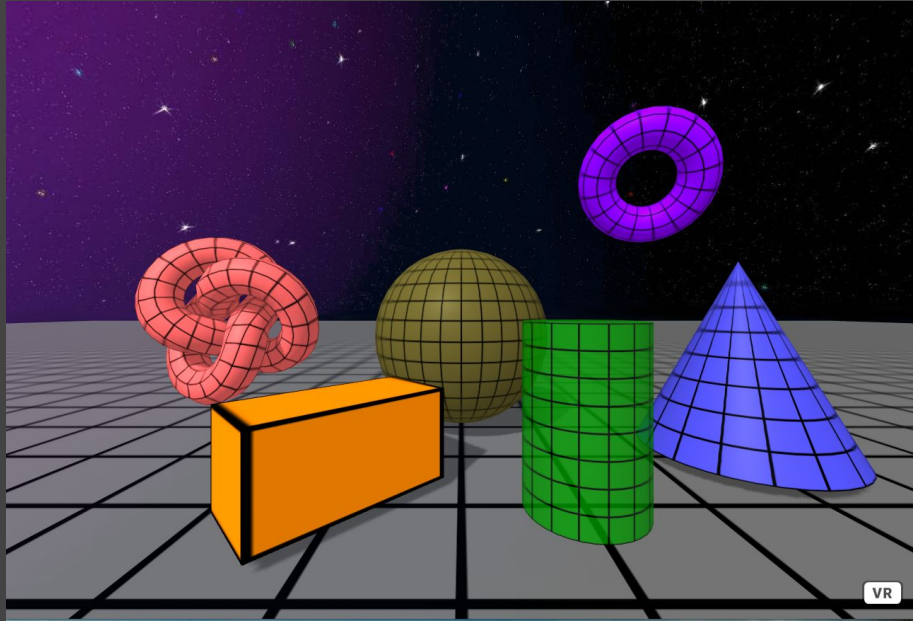
```
</a-box>
```

Animation is a property that can be added to an object or to an entity.

It allows to change values from its properties from an initial point to an end point, giving instructions about how long this transition last.

It can be given to a more than one property and more than one value at a time.

Not to confuse with the animation that an object can have in itself.



Its main features are:

- property: which property to animate, it could be position, rotation but color as well.
- from: initial value.
- to: final value.
- dur: duration in milliseconds.
- loop: how many repetitions, either a number or *true* for infinite.
- delay: how many ms delay before begin the animation.

See

<https://aframe.io/docs/1.4.0/components/animation.html>
for more.



RIT
REALITIES IN
TRANSITION

Working Group Time

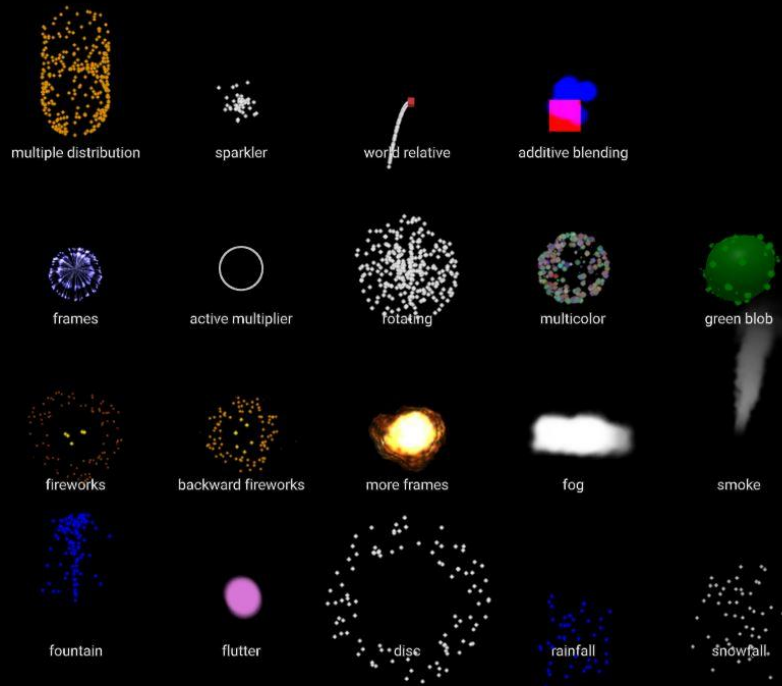
What is it expected from you at the end of the next working group session?

- 1) Continue collecting your models.
- 2) Finish your scene design.
- 3) Begin importing your models into your scene.
- 4) Edit those models that are hard to find with 3D Builder.
- 5) Think of possible animations.

Some extra features

Particles and physics



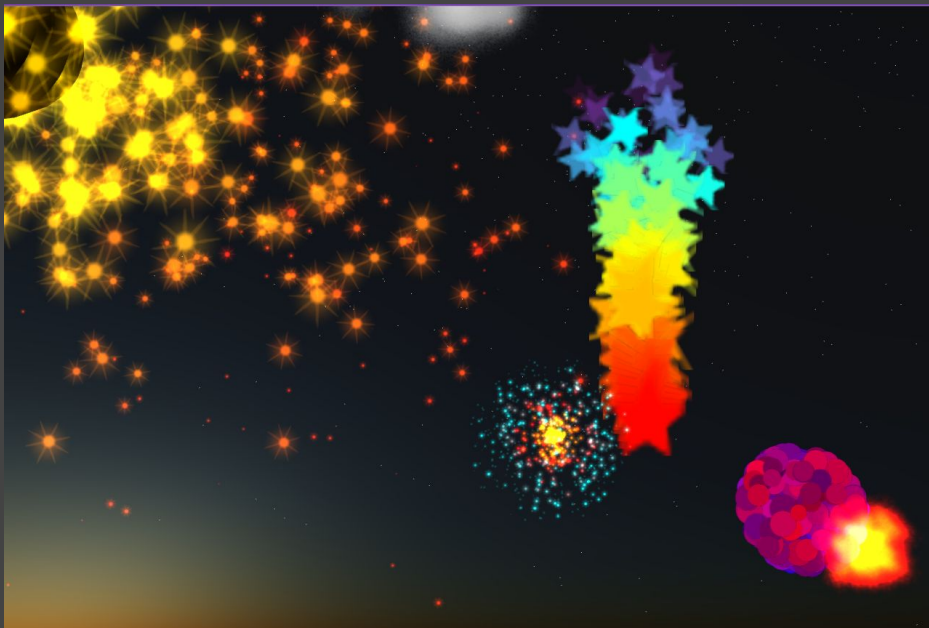


A particle system is a technique in game physics, motion graphics, and computer graphics that uses many minute sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena.

They are used to represent things like fire that otherwise very hard to reproduce with conventional rendering techniques – usually highly chaotic systems.



Particle Component



```
<script
```

```
src="https://unpkg.com/aframe-spe-pa  
rticles-component@^1.0.4/dist/aframe  
-spe-particles-component.min.js"
```

```
</script>
```

As component, in the same way that there is an environment component to be added at the <head> tag while using it, you need to add the component js files if you want to use particles.

For this time, we will be using the online component

```
<a-entity
```

```
  particles="texture:
```

```
/images/particles/star.png;
```

```
  color: blue;
```

```
  opacity: 1,1,1,0;
```

```
  velocity: 0 10 0;
```

```
  velocity-spread: 2 0 2;
```

```
  blending: normal;
```

```
  particle-count: 100">
```

```
</a-entity>
```

Also as seen before, particles are also entities and they should be declared under the `<a-entity>` tag.

Then they use a set of properties:

- texture: the image for every particle.
-