

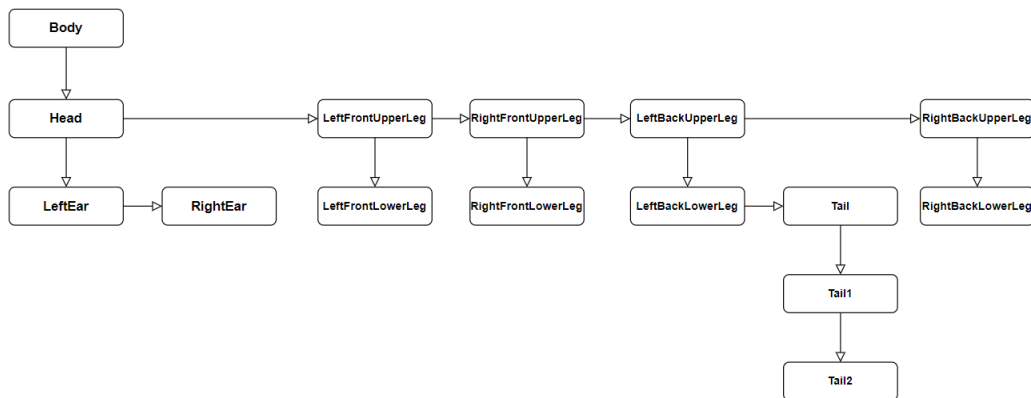
# Interactive Graphics

## Homework 2

Andrea Panceri 1884749

### 1 Hierarchical model of cat

For build the cat I design an Hierarchical Model: at first I put the body on the root and then with the createNode(transform, render, sibling, child) function I created the siblings and the children of the tree. Each part of the cat has associated an id and a function with the respective measures which allow to scale, translate or rotate it. The complete Hierarchical Model of the cat is the following:



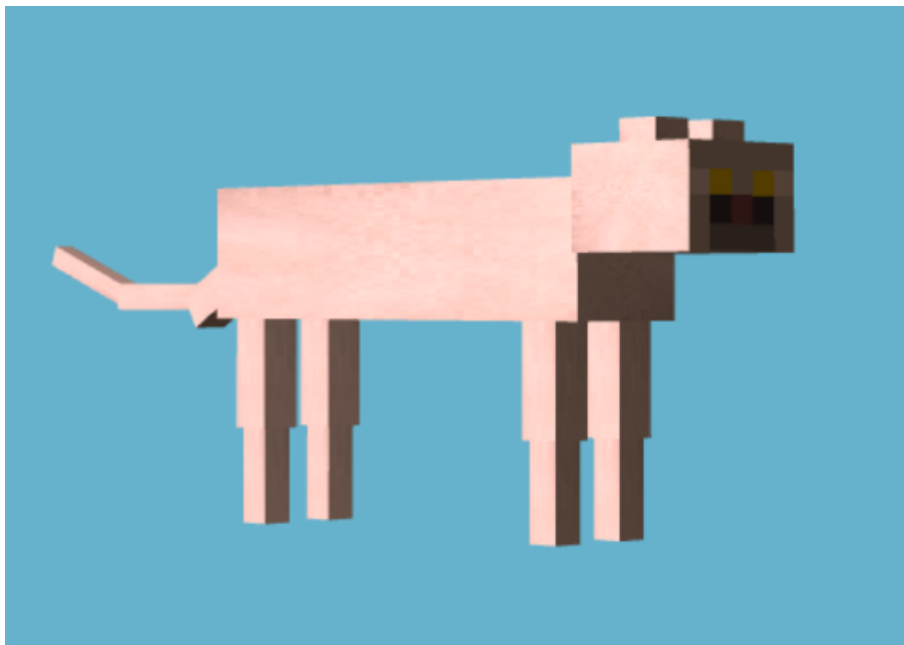
### 2 Carpet and Table

For build the carpet below the cat and the table, I added a cube that is separated component respect to the previous Hierarchical Model. Furthermore I attached to the carpet a color texture using a Normal Map to give a very realistic effect of vintage carpet. Furthermore I put a table at the center of the carpet, that is composed of five cubes, one for the upper part and four for the legs, each of one with his Id and function to scale. The table like the carpet is separated to the Hierarchical Model of the cat and also from the carpet and I decided to apply to the table a texture of the wood, to remind a wooden living room table. Also for the wood texture I used a Normal Map to give a very realistic effect of wood. the table is not really tall, since as mentioned I wanted to imitate a low and wide living room table. All the textures work on Google Chrome and also on Edge. Obviously for not have the cross origin error must use a web server. This is the result:



### 3 Texture of Cat

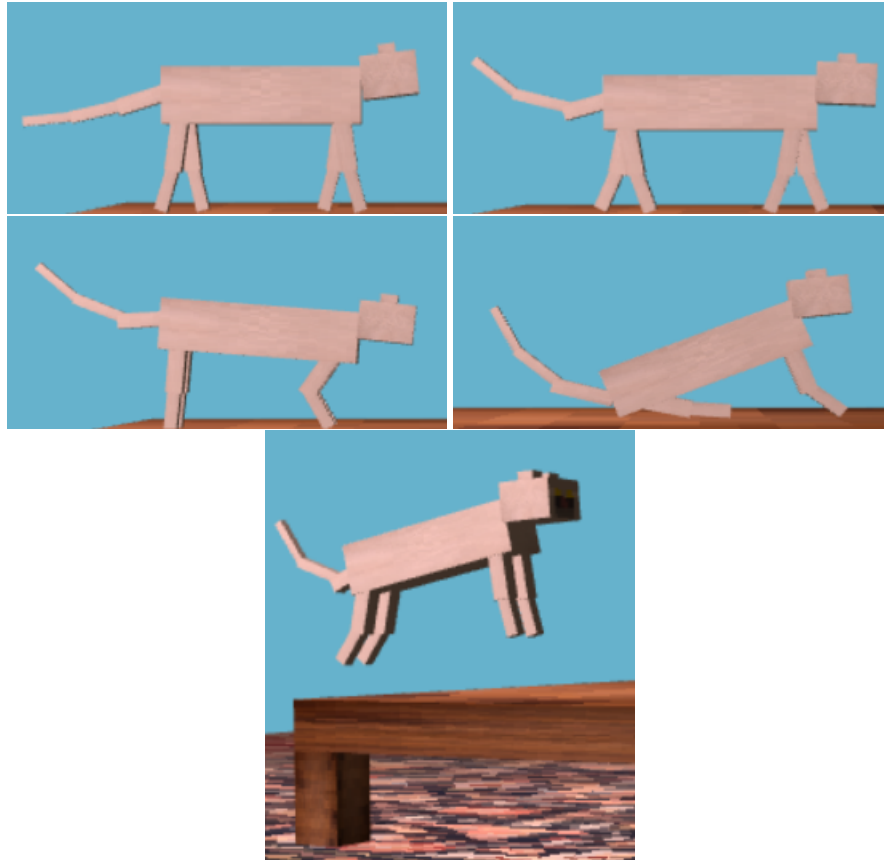
For the cat I used two more texture, already used two for the carpet and table. I used one texture for the body and one for the head. The texture of the body is attached on all the components of the cat and I use a Normal Map to obtain a effect of a realistic cat fur. For the head I attached on the front face only the color texture that I drew in Paint, editing the cat Minecraft face. The texture has a light gray color like it is requested. This is the result:



### 4 Cat Animation

For the animation I created a function called `catAnimation(flagForMotionBlur)` that starts when we click on the play button. I divided the animation of the cat in 7 steps: first 3 steps implement the walk, the fourth crouching cat, the fifth and the sixth implement the jump on the table and the last implement the cat that sit down. Each position of the cat belongs to

an if-condition of the function catAnimation. The walk is implemented with a loop on the second and third step. When the cat reaches the center of the table, it sits in that position and the loop ends terminating the animation. Each if-condition is adjusted with a counter parameter and the position of the cat is updated for all the duration of the animation function to describe the requested trajectory. When the cat is arrived to the center of the table and stops, to rerun the animation is enough to click on reset button and re-click the play button. This is the result:



For completeness I add also other buttons for changing the speed of the animation: 0.5x, 1x, 2x and 3x. I also want to emphasize the jump animation where I used parabolic motion to calculate the trajectory to follow, resulting in a very smooth and realistic animation.

## 5 Camera movement

Like requested I allow the user to move the camera before and during the animation, I put some sliders to modify the values of the `modelViewMatrix` and the `projectionMatrix`. These are near, far, radius, theta, phi, fovy and aspect, I use the same of the precedent Homework-1. All the initial values are set to see clearly the cat jump and walk on the table but these can be modified to obtain any wanted angle. Also for see the effect of Motion Blur.

## 6 Motion Blur

For implementing the Motion Blur, in the render function I draw many cat each at a different iteration of the animation to obtain a "Object"-motion blur effect. Moreover, each cat that is used for the effect has a variable called `motionBlurQuantity` that increase the

transparency to reach the desired effect and there is a button to switch on/off the motion blur effect and a slider to increase the intensity. If we speed up the animation, the effect increase like the real effect. The effect is very heavy to render, for my choice i do in the JavaScript using the CPU. To handle the problem of transparent objects, I first drew the table and the carpet using the Depth mask to handle the depth. Instead, to draw the cats with different degrees of opacity, I disabled the depth mask and enabled `gl.BLEND` in particular using `gl.blendFunc(gl.ONE, gl.ONE_MINUS_SRC_ALPHA)` to make the transparent objects not have a default white color, obtaining a more blurred effect. To make the animation proceed properly I save all the variables that are used in the animation and restore them after I finish rendering all the cats. Obviously for implement Motion Blur there are many choices like implement also Camera Motion Blur, or also use the shaders for implementing both instead done in JavaScript. I believe that object Motion blur is better for this particular case.



## 7 Final comments

Want to outline that the code is very modular and reusable in the JavaScript file, also the css could be added in a separated file, but we can not add other files. Also want to notice that the page is responsive we can see the scene and adjust all settings in any kind of device.