Day: Jan 25, 2023
Time: 10:30 – 11:30AM
Room: 56-154

Lecture 3: Scientific Animations and Movies Notes

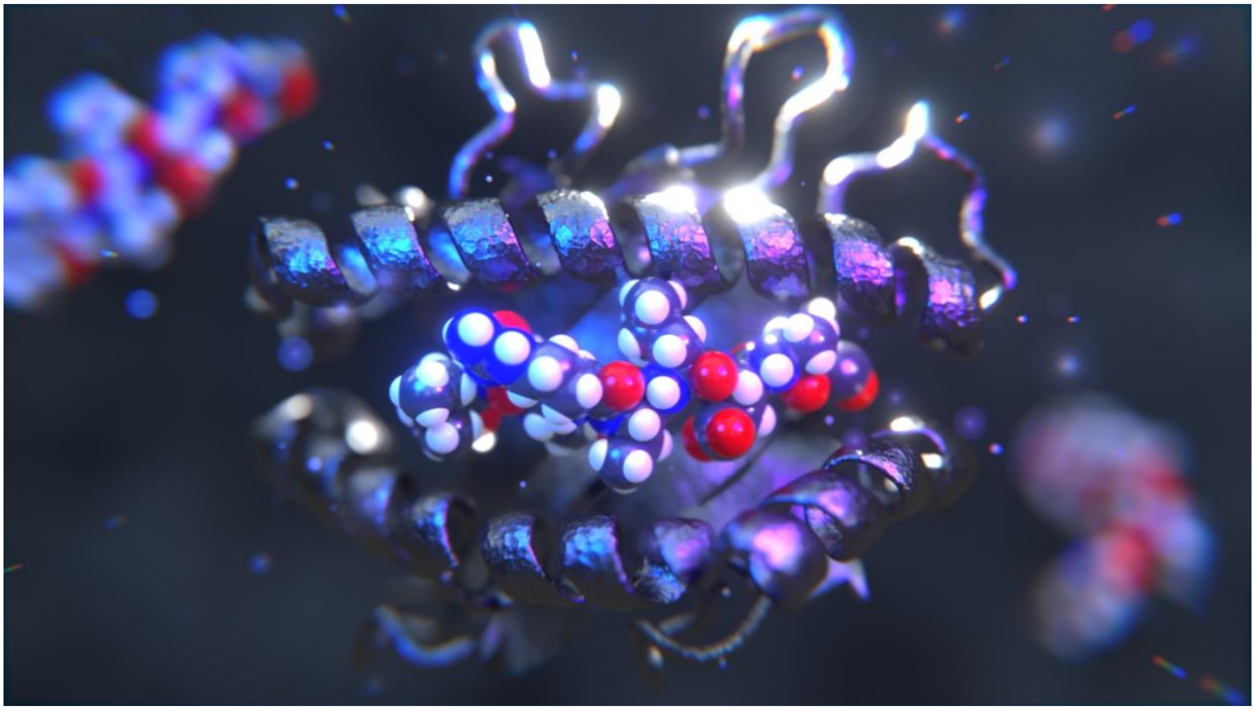

Figure 1. The result of this lesson will be a movie of the peptide ligand entering and leaving the MHC binding pocket.

Lesson Summary

I hope you appreciate how far you have come. Using the skills, you have learned in Lectures 1 & 2, you can now create realistic and interesting renders of molecular structures. However, there is one last topic I would like to introduce you to, and that is Animations. Being able to render scientific movies and animations can be a powerful tool for community outreach and getting your work seen. In this lesson, we will take the final scene from lesson 1 and turn it into a simple movie showcasing the binding of a peptide to the MHC pocket.

Class Schedule

1.	**Intro to animation and the scene**	5 min
2.	**Looping animation of a ligand binding**	20min
3.	**Creating looping dust particles that appear random**	15 min
4.	**Creating looping molecules that appear random**	5 min
5.	**Rendering and processing an animation in Blender**	10 min

Total Time 55 min

1. Intro to Animating and the Scene (15 min)

When you open the Start.blend file in the lecture3_files, you will recognize the scene as the end of lesson 1. So that we can focus on the topic of this lesson, animation, we will animate this scene with some minor tweaks. This first section will focus on those minor tweaks which I am adding in as a review of lesson 2. We are going to add a simple background, an HDRI, and procedural texture and then we'll get started animating in part 2!

- Let's first add a simple background
  - **Shift+A** → **Mesh** → **Plane**
    - **Move**, **Scale**, and **Rotate** the plane to somewhere in the back of the scene
    - Place it as if it was a wall that covers the full camera
    - Rename it **Wall**
  - Let's change the material of the wall to the same cement color from lesson 2 but this time let's only use the color
  - Go to the **Timeline Window** at the bottom and drag it up
    - Click on the **Editor Type** (the clock in the left-hand corner of the **Timeline Window**
    - Change it to the **Shader Editor**
    - Click **New**
    - Add an **Image Texture** node with **Shift+A** → **Search** → **Image Texture**
      - Select **Open**
      - Navigate to the Lecture3-files and select **SurfaceImperfections015_2K_Color.jpg** from the last lesson
      - Connect the **Color** port in the **Image Texture** to the **Base Color** port in the **Principled BSDF**
      - It will add a nice blotchy color to the wall
- Now let's practice adding the HDRI from last lecture
  - In the **Shader Editor** space at the bottom, click on **Object** and change to **World**
  - Add an **Environment Texture** with **Shift+A** → **Search** → **Environment Texture**
  - Click **Open**
    - Navigate to the **Lesson3-files** and select **canary_wharf_4k.hdr**
    - Connect the **Color** port of the **Environment Texture** to the **Color** port of the **Background**
    - Change the **Strength** to **0.3**
- Let's now add a metal material to our protein. We are going to be playing with lights in this lesson, so I want to choose a material with an uneven surface for the light to reflect off
  - Click on the protein
  - Go to the BlenderKit website and under MATERIALS, search for a rough metal that you like

        - If you need help, check out this one! Metal Rough 01
        - Select Get this material
        - Copy link
        - Navigate back to Blender and you will see that the material is now in the top left-hand corner
        - Drag and drop it onto the protein
    - In the Shader Editor, switch from World back to Object. We are going to edit the material nodes to customize it! If you picked your own, go ahead and skip this step
        - Under the **Mapping** node, change the **X**, **Y**, and **Z** scale values from **1.000** to **0.2**
        - Go to the **Normal Map** node and change the **Strength** to **3.000**
- Lastly, the carbons on the peptides stand out now. Let's change them to a matching metal material
    - Select a carbon atom (any will work) go to its **Material Properties** and change its **Metallic** value to **0.75**
- Let's animate some lights!
    - Add two lights to the scene and place them in the top left and right corners of the scene
    - **Shift+A** → **Light** → **Point**
    - Click on one of the lights and go to the **Object Data Properties** tab (a picture of a light bulb) in the **Properties Window**
        - Change the **Radius** to **1.0**
        - Change the **Power** to **750** for each
        - Change of the **Color** of one of them to **Blue** and one to **Purple**
- Go to the **Timeline Window** (you may need to change from the **Shader Editor** using the **Editor Type** drop down
    - On the far-right hand corner, who will see a box that says end. This is the number of frames that will be in our animation
        - Change it to **150**
    - Make sure your cursor on the timeline is set to **1**
    - Click on one of the lights and open the **Sidebar**
    - Right-click **Location** and **Insert Keyframes**
        - Move the cursor to **151** on the timeline and do the same
            - This ensures that our animation states and ends at the same place
    - Repeat for the other light
    - Move the cursor to the middle (around **75**) and drag one of the lights in front of the peptide
        - Right-click **Location** and **Insert Keyframes**
        - Repeat for both lights
        - You can even make then criss-cross each other.
            - Animations are difficult to describe in words because they are so visual, so if this isn't making sense cross reference with the Lecture 3 Part 1 video

2. Animating a ligand binding (15 min)

We are ready to begin animating. We are going to create a simple animation of the ligand binding to the MHC.

- Select the peptide and combine all the atoms into a single object we can manipulate
  - Select an atom somewhere in the middle of the peptide
  - Select the whole peptide
    - `Outliner` → `Peptide Bound` → `Right-click` → `Select Objects`
  - Now if we move that one atom in the middle of the peptide, all of the atoms will move
- Expand the **`Timeline Window`**
- In the **`Timeline Window`**, make sure the **`End`** value is set to **`150`**
  - This represents the total number of frames of our render
- Move the timeline cursor to middle of the timeline (frame **`75`**)
  - Go to the **`Sidebar`** and **`Right-click`** **`Location values`** → **`Insert keyframes`**
  - Repeat on rotation
- This will ensure that the peptide is perfectly bound to the MHC at the middle of the animation
- Move the timeline cursor back to frame 1
- Move the peptide off to somewhere to the right using **`Move`**
- Insert new keyframes at **`Location`** and **`Rotation`** with **`Right-click`** **`Insert Keyframes`**
- Play the animation and see what you think
- This is the fun part of animation and why it is hard
- You can now add new **`Locations`** and **`Rotations`** at different keyframes to make it more natural. This can be as simple or as complex as you want it to be

In some cases, the goal for an animation or video is to showcase it on social media. In that case, the most effective type of animations is looping animations as it gives the illusion that your video is much longer and required more time than it did. In this section we are going to animate peptide completely leaving the screen in a way that allows for the animation to be looped.

- Go to frame **`150`** in the **`Timeline Window`**
- Select the **`Peptide`**
  - You can either select your central atom that you used as the parent or you can go to the **`Outliner`** and **`Right-click`** on **`Peptide`** → Select **`Objects`**
- Move the peptide out of the screen on the left side
  - Anywhere is fine but I just a position closer to the upper left rather than the lower left, which would add some interesting twisting
- Now run the animation
- Go to the frame when the peptide begins leaving the MHC binding site
  - Move the molecule slightly upward as if it is going “over” the helix
- Insert new keyframes at **`Location`** and **`Rotation`** with **`Right-click`** **`Insert Keyframes`**
- Add as many keyframes as you like

3. Create looping dust particles that appear random (10 min)

A small amount of dust is one of the best elements that you can add to your scene to improve the depth and atmosphere of your renders. However, dust usually moves randomly, which won't work in this scene since we are hoping to be able to loop the animation. This means we will need dust that is in the same place at the beginning and ends of the video but seems to move randomly in between.

- Create a new **Collection** called **Dust**
- Create an **Icosphere**
  - **Shift+A** → **Mesh** → **Icosphere**
  - Scale it up so that it contains the whole scene including the camera
  - All the dust particles will be generated within this object
- Add a circle
  - **Shift+A** → **Curve** → **Nurbs Circle**
  - Scale it up so that it completely surrounds the icosphere
- Create an Empty
  - **Shift+A** → **Empty** → **Plain Axes**
- Create a new **Icosphere** and scale it down
  - This will be the reference dust particle
- Rename each of the objects
  - Circle renamed to **Dust Path**
  - Large icosphere renamed to **Dust Container**
  - Small icosphere renamed to **Dust**
  - Empty renamed to **Dust Controller**
- Click on the **Dust Controller**
  - Go to **Object Constraint Properties** in the **Properties Window**
    - Add **Object Constraint** → **Follow Path**
    - Click on the eye dropper and then click on the **Dust Path** in the **3D Window**
      - The **Dust Controller** should snap to **Dust Path**
      - **Influe**nce = **0.03**
  - Go to frame 1 in the **Timeline Window**
    - Right click **Offset** in the **Object Constraint Properties**
      - Keep the value as **0.00**
      - **Right-click** → **Insert Keyframe**
  - Go to frame **151** in the **Timeline Window**
    - **Right-click** Offset (0.00) in the Object Constraint Properties
    - Change the value to **100**
    - **Right-click** → **Insert keyframe**
  - **Right-click** on each of the keyframes and change the **Interpolation Mode** → **Linear**
- Click on the **Dust Container**
  - Add a **Displace** modifier (wrench tab under Deform)

        - **Modifier Properties** in the **Properties Window** → **Add Modifier** → **Deform** → **Displace** → **New**
            - **Strength = 0.1**
        - **Coordinates** → **Object**
        - **Object** → **Eye dropper** → **Dust Controller**
    - With the **Dust Container** still selected, switch to the **Texture Properties** tab
        - **Type** → **Stucci**
            - **Size** = **1.75**
- We will no create a particle simulation
    - Click on **Dust Container**
        - Go to the **Particle Properties** tab in the **Properties Window**
        - Select the **+** to add a new **ParticleSystem**
    - Select **Hair**
    - Select **Advanced**
    - **Render** → **Path** → **Object** → **eye dropper** → select **Dust** in the **3D Window**
    - **Emission** → **Source** → Change **Faces** to **Volume**
    - **Render** → Uncheck **Show Emitter**
    - **Viewport Display** → Uncheck **Show Emitter**
        - This will hide the **Dust Container**
    - Change the size of the particles
        - **Render** → **Scale** = **0.03**
        - **Render** → Scale Randomn**ess** = **1.0**
- Click on **Dust** and go to its **Materials Properties**
    - Change **Transmission** to **0.9**

4. Looping molecules that appear random (5 min)

Originally, this was the end of this section. However, I found it distracting that the peptides in the background don’t move at all. We can change this by attaching them to the Dust Container. This is very easy to do and will automate the animation of the other peptides.

- First, we will need to parent all the atoms in the peptides to an atom in the center
    - Choose one of the two background peptides
    - Select one of the easy to find atoms
        - Go to the **Outliner** and **Right Click** the collection corresponding to that peptide for example:
        - **Right-click Peptide Front** → **Select Objects**
        - **Command+P** → **Object**
        - Repeat for the other peptide
- Now click on the **Dust Container** and press **Tab** to enter **Edit Mode**
    - Select three vertices somewhere on the icosphere
    - Press **Tab** to exit
    - Select your peptide parent atom then holding down **Shift** select **Dust Container** and parent then with **Command+P** → **Vertex (Triangle)**
    - Repeat for the other peptide

5. Rendering and processing an animation in Blender (10 min)

Now that the scene is finished, we can render it out. The render will take a while to finish depending on the number of frames. Once the still frames finish rendering, we can compile them into a final .mp4 right here in Blender. No need for external software.

- Render the Animation
    - **Render → Render Animation**
        - This will take around 30 min with a decent computer
- After the rendering is complete
- Go to the **Output Properties** tab in the **Properties Window**
    - **Output → File → Desktop**
        - This just tells Blender to save your files in some place other some cryptic temp folder
    - **Output → File Format → FFmpeg Video**
    - **Encoding → Video → Output Quality → High Quality**
    - **Encoding → Container → MPEG-4**
- Select **Video Sequencer** from within the **Editor Selector**
    - Click **Add** in the **Header** bar
        - **Image/Sequence**
        - Navigate to the folder with your frames
            - Select all the frames
            - Change **End Frame** to **150**
- You will see a tan bar which represents the new animation
- Now you can render the frames out together as a video
    - **Render → Render animation**
        - It will finish in less than a minute
- Congratulations of your first movie!