3D Animation
Block –III : 3D Animation & Rigging (Practical)

Odisha State Open University
3D Animation

This course has been developed with the support of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Odisha State Open University, Sambalpur (OSOU) is the first Open and Distance learning institution in the State of Odisha, where students can pursue their studies through Open and Distance Learning (ODL) methodologies. Degrees, Diplomas, or Certificates awarded by OSOU are treated as equivalent to the degrees, diplomas, or certificates awarded by other national universities in India by the University Grants Commission.

© 2018 by the Commonwealth of Learning and Odisha State Open University. Except where otherwise noted, 3D Animation is made available under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) License: https://creativecommons.org/licenses/by-sa/4.0/legalcode

For the avoidance of doubt, by applying this license the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of Commonwealth of Learning and do not commit the organisation.

Odisha State Open University
G.M. University Campus
Sambalpur
Odisha
India
Fax: +91-0663-252 17 00
E-mail: info@osou.ac.in
Website: www.osou.ac.in

Commonwealth of Learning
4710 Kingsway, Suite 2500,
Burnaby, V5H 4M2, British,
Columbia
Canada
Fax: +1 604 775 8210
Email: info@col.org
Website: www.col.org
Acknowledgements

The Odisha State Open University and COL, Canada wishes to thank those Resource Persons below for their contribution to this DMA-04:

Concept / Advisor
Dr. Srikant Mohapatra  
Vice-Chancellor  
Odisha State Open University, Sambalpur

Course Writer
Praseed Nair  
Assistant Professor  
School of Communication, Manipal University

Course Editor
S.Anuradha  
Bangalore based Freelancer

Video Production
R. Mohana Sundaram  
Creative Director  
Jai Ram Institute of Visual Academy, Khurda, Odisha  
Guest Faculty, National Institute of Fashion Technology (NIFT), Bhubaneswar

Published by:
Dr. Jayanta Kar Sharma  
Registrar on behalf of Odisha State Open University, Sambalpur

Contribution of following staff members of Odisha State Open University is acknowledged:
- Sambit Mishra
- Debidatta Behera
- Prashansa Das
- Radhakanta Suna
- Abhinandan Tripathy

OSOU and COL acknowledge the support extended by Prof. Madhu Parhar, STRIDE, IGNOU, New Delhi in conducting several workshops in the process of preparation of course material for DMA.
## Contents

### Course overview 3
- Welcome to 3D Animation & Rigging .............................................................. 3
- Introduction to Rigging ............................................................................. 3
- Working with Armature ....................................................................... 4
- 3D Animation ....................................................................................... 4
- Advanced Animation-Tracking .............................................................. 4
- Course outcomes ............................................................................... 5
- Timeframe .......................................................................................... 5
- Study skills ........................................................................................ 6
- Need help? ......................................................................................... 6
- Assignments ....................................................................................... 7
- Assessments ....................................................................................... 7
- Video Resources ................................................................................. 7

### Getting around this Course material 8
- Margin icons .................................................................................... 8

### Unit 1 9
- Introduction to Rigging ........................................................................
  - Introduction ................................................................................ 9
  - Outcomes .................................................................................... 9
  - Terminology ............................................................................... 10
  - Working with Constraints ........................................................... 10
  - Adding/Removing a Constraint ..................................................... 12
  - Relationship ............................................................................... 23
  - Child of Constraint ................................................................... 26
  - Unit summary ............................................................................. 29
  - Assessment .................................................................................. 30
  - Resources ................................................................................... 31

### Unit 2 33
- Introduction to Working with Armature ............................................
  - Introduction ................................................................................ 33
  - Outcomes .................................................................................... 33
  - Terminology ............................................................................... 34
  - Working with Armature ............................................................... 35
  - If the Active Element is a Disconnected Root: .......................... 44
  - If the Active Element is a Connected Root: ............................... 45
  - Armature Deform Parent ............................................................... 49
<table>
<thead>
<tr>
<th>Unit 3</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Animation</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>55</td>
</tr>
<tr>
<td>Outcomes</td>
<td>55</td>
</tr>
<tr>
<td>Terminology</td>
<td>56</td>
</tr>
<tr>
<td>Introduction to KeyFrame</td>
<td>56</td>
</tr>
<tr>
<td>Keyframe Types</td>
<td>57</td>
</tr>
<tr>
<td>KeyFrame Animation</td>
<td>60</td>
</tr>
<tr>
<td>To test the animation, press Alt-A Play</td>
<td>61</td>
</tr>
<tr>
<td>Interpolation</td>
<td>64</td>
</tr>
<tr>
<td>Extrapolation</td>
<td>68</td>
</tr>
<tr>
<td>Using Dope Sheet in Animation</td>
<td>72</td>
</tr>
<tr>
<td>Unit summary</td>
<td>79</td>
</tr>
<tr>
<td>Assignment</td>
<td>79</td>
</tr>
<tr>
<td>Assessment</td>
<td>79</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit 4</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textures and Mapping</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>81</td>
</tr>
<tr>
<td>Outcomes</td>
<td>81</td>
</tr>
<tr>
<td>Brief the Process</td>
<td>81</td>
</tr>
<tr>
<td>Terminology</td>
<td>81</td>
</tr>
<tr>
<td>Introduction to Motion Tracking</td>
<td>82</td>
</tr>
<tr>
<td>2D Stabilization</td>
<td>88</td>
</tr>
<tr>
<td>Mask Editor</td>
<td>94</td>
</tr>
<tr>
<td>Understanding Layers</td>
<td>95</td>
</tr>
<tr>
<td>Control Points</td>
<td>97</td>
</tr>
<tr>
<td>Compositing Node</td>
<td>99</td>
</tr>
<tr>
<td>Animating Masks</td>
<td>100</td>
</tr>
<tr>
<td>Parenting to Motion Tracks</td>
<td>101</td>
</tr>
<tr>
<td>Unit summary</td>
<td>103</td>
</tr>
<tr>
<td>Assignment</td>
<td>103</td>
</tr>
<tr>
<td>Assessment</td>
<td>104</td>
</tr>
<tr>
<td>Resources</td>
<td>104</td>
</tr>
</tbody>
</table>
Welcome to 3D Animation & Rigging

Rigging is a process done prior to the Animation. Rigging is a process of taking a static mesh, creating an internal digital skeleton, creating a relationship between the mesh and the skeleton (known as skinning, enveloping or binding) and adding a set of controls that the animator can use to push and pull the character around as if he/she is a puppeteer.

An Armature in Blender is similar to the Armature of a real skeleton. Just like a real skeleton, an Armature can consist of many Bones. These Bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way. An “Armature” is a type of Object used for rigging.

In 3D Animation and other forms of Computer Animation, the frames are generated by interpolating between the numerical values that are defined in any two consecutive keyframes.

In the process of video production Tracking is used, such as 2D Tracking, 3D Motion Tracking, Camera Tracking and Object Tracking. Tracking allows you to import raw footage, track the footage, and mask areas using Camera movements in your 3D scene.

In Block 1 and Block 2, you have learnt about 3D Modelling and 3D Shading respectively. Now in this Block 3, you will learn about 3D Animation and Rigging.

Introduction to Rigging

Rigging is a process done prior to the Animation. Rigging is a process of taking a static mesh, creating an internal digital skeleton, creating a relationship between the mesh and the skeleton (known as skinning, enveloping or binding) and adding a set of controls that the animator can use to push and pull the character around as if he/she is a puppeteer. In this Unit, you will learn what is rigging and how it is important in designing 3D animation. You will also learn how to use, add and remove the
Constraints; Create IK Constraint & Spline IK Constraint; and also describes the functions of Header, Target, Space, and Influence.

Working with Armature

An Armature in Blender is similar to the Armature of a real skeleton. Just like a real skeleton, an Armature can consist of many Bones. These Bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way. An “Armature” is a type of Object used for rigging. In this Unit 2, you will learn to work with Armatures in Blender;

Use of Bones; Types of Armature structure; Linking Objects to Bone; Set up Mesh and Armature using Skinning; Create Poses for the Rigged Character; Skin the Mesh to the Bones and to Add and remove influence for a Bone.

3D Animation

In 3D Animation and other forms of Computer Animation, the frames are generated by interpolating between the numerical values that are defined in any two consecutive keyframes. Typically, in 3D Animation, this interpolation takes the form of 3D Beziers curves (paths) which are constructed as a series of control points, allowing for the interactive manipulation of smooth 3D curves. In this Unit, you will learn about 3D Animation and how animation is making an Object move or change shape over time. This will be done through different tools and techniques in Blender.

Advanced Animation-Tracking

In this Unit, you will learn the process of production of videos using Tracking, such as 2D Tracking, 3D Motion Tracking, Camera Tracking and Object Tracking. Tracking allows you to import raw footage, track the footage, and mask areas using Camera movements in your 3D scene. You will also learn about Stabilizing, Rotoscoping, Mask Editor, Mask Data block, Shape Keyframe and Layers.
This video will provide a brief overview of this course.

<table>
<thead>
<tr>
<th>Topic</th>
<th>YouTube link</th>
<th>QR Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1 – Keyframe Animation</td>
<td><a href="https://youtu.be/Y3qXNJKw8Tc">https://youtu.be/Y3qXNJKw8Tc</a></td>
<td><img src="#" alt="QR Code" /></td>
</tr>
<tr>
<td>Video 2 – Types of Rigging</td>
<td><a href="https://youtu.be/5a1-ctvwywk">https://youtu.be/5a1-ctvwywk</a></td>
<td><img src="#" alt="QR Code" /></td>
</tr>
</tbody>
</table>

**Course outcomes**

Upon completion of 3D Animation & Rigging you will be able to:

- Explain the Usage of Constraints
- Plan for Adding or Removing Constraints
- Explain the Usage of Bones
- List the types of Armature structure
- Describe Keyframe Animation
- Work with Timeline
- Use the Movie Clip Editor
- Brief the Process of Rotoscoping

**Timeframe**

This course will be completed within “2” classes.

This course is of “1” credits.

1 Hour of study time is required for this unit.
Study skills

This is a totally practical oriented course. Hence, you should have access to personal computer or personal laptop for better understanding of this unit.

Each and every options are explained step by step in the course material.

Apart from this course material, the learner has to adopt the tendency of learning from multiple sources i.e.,

- Internet tutorials
- Video tutorials on YouTube
- Collaboration with people working in the industry etc.

Only classroom study will not make you a professional. You have to be active to grab the opportunity of learning wherever you get a chance.

Need help?

In case of any help needed you can browse the internet sites like youtube.com for video tutorials about the subject.

Apart from that, you can contact the writer of this course material at praseed.nair@manipal.edu
Assignments

There will be some assignments at the end of each unit. These assignments are mostly practical based and should be submitted in CD or DVD. Theoretical assignments are to be submitted neatly written on A4 size sheet.

All assignments will be submitted to Regional centre of Odisha State Open University or as directed by Co-ordinator.

All assignment should be unit wise on separate CD/DVDs clearly mentioning course title and unit on Top. Theoretical Assignment will be neatly filed or spiral bind with cover clearly mentioning necessary information of course, student detain on top.

Assessments

There will be few assessment questions for each unit.

All practical assessment will be submitted to OSOU.

Assessment will take place once at the end of each unit.

Learner will be allowed to complete the assessment within stipulated time frame given by the university.

Video Resources

This study material comes with additional online resources in the form of videos. As videos puts in human element to e-learning at the same time demonstrating the concepts visually also improves the overall learning experience.

You can download any QR code reader from Google Play to view the videos embedded in the course or type the URL on a web browser.
Getting around this Course material

Margin icons

While working through this Course material you will notice the frequent use of margin icons. These icons serve to “signpost” a particular piece of text, a new task or change in activity; they have been included to help you to find your way around this Course material.

A complete icon set is shown below. We suggest that you familiarize yourself with the icons and their meaning before starting your study.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Assessment</th>
<th>Assignment</th>
<th>Case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discussion</td>
<td>Group activity</td>
<td>Help</td>
<td>Note it!</td>
</tr>
<tr>
<td>Outcomes</td>
<td>Reading</td>
<td>Reflection</td>
<td>Study skills</td>
</tr>
<tr>
<td>Summary</td>
<td>Terminology</td>
<td>Time</td>
<td>Tip</td>
</tr>
</tbody>
</table>
Unit-1

Introduction to Rigging

Introduction

Rigging is a process done prior to the Animation. Rigging is a process of taking a static mesh, creating an internal digital skeleton, creating a relationship between the mesh and the skeleton (known as skinning, enveloping or binding) and adding a set of controls that the animator can use to push and pull the character around as if he/she is a puppeteer.

Most commonly, characters are Rigged before they are animated because if a character model doesn't have a Rig, they can't be deformed and moved around. Process of Rigging also involves Rigging character or creatures and Objects like car, plane, or a robot.

In this Unit, you will learn about the process of Rigging and how it is important in designing 3D animation.

Outcomes

Upon completion of this unit you will be able to:

- Explain the Usage of Constraints
- Plan for Adding or Removing Constraints
- Describe the Functions of Header
- Explain the term Header, Target, Space, Influence
- Create IK Constraint
- Create Spline IK Constraint
Terminology

Constraints: Constraints are a fantastic way to add sophistication and complexity to a Rig

Header: A Header sits at the top of every Constraint.

Target: The Target field lets you link the Constraint to a Target Object of your choosing

Space: The frame of reference is called the “space” of the Constraint

World space: Transformation, Rotation and Scale are oriented to the world axes

Local Space: Transformation, Rotation and Scale are oriented to the parent Object axe

Influence: The influence slider determines how much the Constraint will affect the constrained Object

IK Solver Constraint: The Inverse Kinematics Constraint implements the inverse kinematics armature posing technique

Spline IK Constraint: The Spline IK Constraint aligns a chain of Bones along a curve

Stretch to Constraint: The Stretch To Constraint causes its owner to rotate and scale its Y axis towards its Target.

Action Constraint: It allows you to control an Action using the transformations of another Object.

Child of Constraint: Child of Constraint is the Constraint version of the standard parent/children relationship

Working with Constraints

Constraints control the behavior of one Object with data from another. It can make the eyes of a tennis player track a tennis ball bouncing across the court. It allows the wheels on a bus to all rotate together. It helps a dinosaur’s legs bend at the knee
automatically. It makes it easy for a hand to grip the hilt of a sword and the sword to swing with the hand.

Constraints, in Blender, work with **Object** and **Bone**.

- **Object Constraint.**

  ![Object Constraint](https://docs.blender.org/manual/en/dev/rigging/constraints/introduction.html)

  **Title**: Img 1. 1 Object Constraints

  **Source**: blender.org

- **Bone Constraint.**

  ![Bone Constraint](https://docs.blender.org/manual/en/dev/rigging/constraints/introduction.html)

  **Title**: Img 1. 2 Bone Constraints

  **Source**: blender.org

Constraints work in combination with each other to form a **Constraint Stack**.

![Constraint Stack](https://docs.blender.org/manual/en/dev/rigging/constraints/introduction.html)

**Title**: Img 1. 3 Constraint Stack

**Source**: blender.org

The **Constraint Stack** is evaluated from top to bottom.
Constraints are a fantastic way to add sophistication and complexity to a Rig. However, be careful not to rush in too quickly, piling up Constraint upon Constraint until you lose all sense of how they interact with each other.

Start simply. Get to know a single Constraint inside and out. Copy Location is a good first Constraint to explore. Take the time to understand every fundamental concept behind it, and the other Constraints will make far more sense.

Adding/Removing a Constraint

To add a Constraint in the Constraints Panel:

- **Step 1:** Click on the “Add Constraint” menu. (Refer [Img 1.4](#))

To add a Constraint in 3D View:

- **Step 2:** Select the Object you would like to constrain.

- **Step 3:** Press **Ctrl-Shift-C** and choose a Constraint from the pop-up menu.

If the chosen Constraint needs a Target, Blender will add an empty automatically as the Target and position it at the center of the constrained Object.
To add a Constraint in 3D View and simultaneously give it a Target:

- **Step 1:** Select the Target first and then shift-select the Object you would like to constrain.

- **Step 2:** Press Ctrl-Shift-C and choose a Constraint from the pop-up menu.

**To remove a Constraint:**

- **Step 1:** Click on the “X” button in the header.

**To remove all Constraints from all selected Object(s):**

- **Step 2:** Click Object ▶ Constraints ▶ Clear Object Constraints in 3D View Header.

- **Step 3:** Or Pose ▶ Constraints ▶ Clear Pose Constraints (for Bone Constraints).

- **Step 4:** Or, press Ctrl-Alt-C.

**Header**

Every Constraint has a header. The interface elements of the header are explained below using a Copy Location Constraint as an example. (Refer Img 1.5)

**Title**-Img 1. 2 A Header sits at the top of every Constraint.

**Source**-blender.org
Expansion Arrow (pointing down or Right)

Show or Hide the settings of the Constraint. Tidy up the Constraint stack by hiding Constraints that do not currently need attention. Constraints will continue to affect the scene even when hidden.

- “Copy Location” (first occurrence)
  The type of Constraint is determined when a new Constraint is created to help in the process of Rigging.

- “Copy Location” (second occurrence)
  Give the Constraint a meaningful name in this field, something that describes its intent. Meaningful names help to understand what each Constraint is supposed to do.

The red background is a warning that the Constraint is not yet functional. The background will turn grey when the Constraint is functioning. When this Copy Location Constraint has a valid Target in the “Target Field” it will turn grey and begin to function.

- Eyeball (open or closed)

  Enable or Disable (Mute/Unmute) the Constraint. Disabling a Constraint will stop its effect on the scene.

  Disabling a Constraint is useful for turning off a Constraint without losing all its settings. Disabling means you can enable the Constraint later with the settings intact. Disabling is like setting the influence slider to 0.0.

- Up/Down Arrows

  Move a Constraint up or down in the Constraint stack. Since the stack is evaluated from top to bottom, moving a Constraint in the stack can significantly affect the final outcome of the stack.
If there is only one Constraint in the stack, the arrows will not be drawn. If the Constraint is at the top of the stack, only the downarrow will be drawn. If the Constraint is at the bottom of the stack, only the up arrow will be drawn.

Delete the Constraint from the stack. The settings will be lost. The Constraint will no longer affect the final outcome of the stack.

Target

The Target field lets you link the Constraint to a Target Object of your choosing. This link provides data to the Constraint so that it can begin to function. For example, the Copy Location Constraint needs location data to function. Fill in the Target field, and the Copy Location Constraint will begin to use location data from the Target Object.

The Target field must be filled in for the Constraint to function. (Refer Img 1.6)

By default, the Target will use the Object Center as the Target point.

- If the Target field links to a Mesh or Lattice Object, a Vertex Group field will appear. Enter the name of a vertex group and the Constraint will Target the median point of this vertex group instead of the Object center.

- If the Target field links to an Armature, a Bone field will appear along with a Head or Tail slider. Enter the name of a Bone and the Constraint will Target the Bone instead of the entire armature Object center. Slide the slider and the Constraint will Target the head, the tail or somewhere in-between.
Space

Constraints need a **frame of reference** in order to function. This frame of reference is called the “**space**” of the Constraint. Choosing **one space vs. another** will change this frame of reference and substantially alter the behaviour of a Constraint. (Refer **Img 1.7**)

To understand how changing the space will change the behaviour of the Constraint, consider experimenting with two empties. Make sure they display **as arrows** so that you can see the **local axes for each empty**. Make sure to size one empty a little larger than the other so that they are both always visible even if directly on top of each other. Then add a Constraint to one empty that Targets the other and experiment thoroughly by moving, rotating and scaling the Target in many different ways.
This Constraint is set to use **World Space** as the frame of reference for both its **Target Space** and its **Owner Space**.

**Influence**

The influence slider determines **how much the Constraint** will affect the constrained Object.
Unit-1  Introduction to Rigging

- An influence of \(0.0\) will have no effect.
- An influence of \(1.0\) will have the full effect.
- Values between \((0.0\ and \ 1.0)\), will have a partial effect, however, be careful. These partial effects can be difficult to control, especially as the Constraint stack grows in complexity.

The influence value is animatable, allowing Constraints to be turned off, or partially on as needed. (Refer Img 1.8)

**IK Solver Constraint**

The Inverse Kinematics Constraint implements the inverse kinematics armature posing technique. Hence, it is only available for Bones. (Refer Img 1.9)

- To quickly create an IK Constraint with a Target, select a Bone in pose mode, and press Shift-I.

**Options**

![Inverse Kinematics Panel](https://docs.blender.org/manual/en/dev/rigging/constraints/tracking/ik_solver.html)

**Title**-Img 1.6Inverse Kinematics panel.

**Source**-blender.org

**Link**-


- **Target**
  
  Must be an armature.
- **Bone**
  A Bone in the armature.

- **Pole Target**
  Object for pole rotation.

- **Iterations**
  Maximum number of solving iterations.

- **Chain Length**
  How many Bones are included in the IK effect? Set to 0 to include all Bones.

- **Use Tail**
  Include Bone’s tail as last element in chain.

- **Stretch**
  Enable IK stretching.

- **Weight**

- **Position**
  - For Tree-IK: Weight of position control for this Target.

- **Rotation**
  - Chain follow rotation of Target.

- **Target**
  Disable for Target-less IK.

- **Rotation**
  Chain follows rotation of Target.

**Spline IK Constraint**

The Spline IK Constraint aligns a chain of Bones along a curve. By leveraging the ease and flexibility of achieving aesthetically pleasing shapes offered by curves and the predictability and well-integrated control offered by Bones, Spline IK is an invaluable tool in the Riggers’ toolbox. It is particularly well suited for Rigging flexible body parts such as tails, tentacles, and spines, as well as inorganic items such as ropes. (Refer Img 1.10)
To set up Spline IK, it is necessary to have a chain of connected Bones and a curve to constrain these Bones to:

- **Step 1**: With the last Bone in the chain selected, add a Spline IK Constraint from the Bone Constraints tab in the Properties Editor.

- **Step 2**: Set the ‘Chain Length’ setting to the number of Bones in the chain (starting from and including the selected Bone) that should be influenced by the curve.

- **Step 3**: Finally, set Target to the curve that should control the curve.

**Options**

![Spline IK panel](https://docs.blender.org/manual/en/dev/rigging/constraints/tracking/spline_ik.html)

**Title** - Img 1. 7. Spline IK panel.

**Source** - blender.org


- **Target**
  - The type of the Target curve.

- **Spline Fitting**
  - **Chain Length**
How many Bones are included in the chain.

- **Even Division**
  Ignore the relative length of the Bones when fitting to the curve.

- **Chain Offset**
  Offset the entire chain relative to the root joint.

- **Chain Scaling**
  - **Y stretch**
    Stretch the Y axis of the Bones to fit the curve.

- **XZ Scale Mode**
  - **None**
    Do not scale the X and X axes.
  - **Bone Original**
    Use the original scaling of the Bones.
  - **Volume Preservation**
    Scale of the X and Z axes is the inverse of the Y scale.

- **Use Curve Radius**
  Average radius of the endpoints is used to tweak the X and Z scaling of the Bones, on top of the X and Z scale mode.

**Stretch to Constraint**

The Stretch To Constraint causes its **owner to rotate** and **scale its Y axis** towards its Target. So, it has the same tracking behavior as the **Track To Constraint**. However, it assumes that the Y axis will be the tracking and stretching axis, and does not give you the option of using a different one.

It also optionally has some **raw volumetric features**, so the owner can **squash down** as the Target moves **closer**, or **thin out** as the Target moves **farther** away. Note that it is not the real volume of the owner which is thus preserved, however, rather the virtual one defined by its scale values. Hence, this feature works even with **non-volumetric Objects**, like empties, 2D meshes or surfaces, and curves.
With Bones, the “volumetric” variation scales them along their own local axes (remember that the local Y axis of a Bone is aligned with it, from root to tip). (Refer Img 1.11)

**Options**

![Stretch To panel]

**Title**-Img 1. 8 Stretch To panel.

**Source**-blender.org


- **Target (Mesh Object Type)**
  This Constraint uses one Target, and is not functional (red state) when it has none.

- **Vertex Group**
  When Target is a mesh, a new field is display where a vertex group can be selected.

- **Target (Armature Object Type)**
  This Constraint uses one Target, and is not functional (red state) when it has none.

- **Bone**
  When Target is an armature, a new field for a Bone is displayed.

- **Head/Tail**
When using a Bone Target, you can choose where along this Bone the Target point lies.

- **Rest Length**
  This number button sets the rest distance between the owner and its Target, i.e. the distance at which there is no deformation (stretching) of the owner.

- **Reset**
  When clicked, this small button will recalculate the Rest Length value, so that it corresponds to the actual distance between the owner and its Target (i.e. the distance before this Constraint is applied).

- **Volume**
  These buttons control which of the X and/or Z axes should be affected (scaled up/down) to preserve the virtual volume while stretching along the Y axis. If you enable the none button, the volumetric features are disabled.

- **Plane**
  These buttons are equivalent to the Up ones of the Track To Constraint: they control which of the X or Z axes should be maintained (as much as possible) aligned with the global Z axis, while tracking the Target with the Y axis.

### Relationship

**Action Constraint**

The Action Constraint is **powerful**. It allows you to **control an Action** using the transformations of another Object.

The underlying idea of the Action Constraint is very similar to the one behind the Drivers, except that the former uses a whole action (i.e. a bunch a F-Curves of the same type), while the latter controls a **single F-curve** of their “owner”

Note that even if the Constraint accepts the **Mesh action** type, only the Object, Pose and Constraint types are really working, as Constraints can only affect Objects’ or Bones’ transform properties, and **not meshes’ shapes**. Also, note that only the **Object transformation** (location, rotation, scale) is affected by the
action, if the action contains keyframes for other properties they are ignored, as Constraints do not influence those.

As an example, let us assume you have defined an Object action (it can be assigned to any Object, or even no Object at all), and have mapped it on your owner through an Action Constraint, so that moving the Target in the (0.0 to 2.0) range along its X-Axis maps the action content on the owner in the (0 to 100) frame range. This will mean that when the Target’s X property is 0.0 the owner will be as if in frame 0 of the linked action; with the Target’s X property at 1.0 the owner will be as if in frame 50 of the linked action, etc. (Refer Img 1.12)

Options


- **Title**-Img 1. 9 Action panel.

- **Source**-blender.org

- **Link**-

- **Target**
  This Constraint uses one Target, and is not functional (red state) when it has none.

- **Bone**
  When Target is an armature Object, use this field to select the Target Bone.

- **Transform Channel**
This selector controls which transform property (location, rotation or scale along/around one of its axes) from the Target to use as “action driver”.

- **Target Space**
  This Constraint allows you to choose in which space to evaluate its Target’s transform properties.

- **To Action**
  Select the name of the action you want to use.

  Even though it might not be in red state (UI refresh problems...), this Constraint is obviously not functional when this field does not contain a valid action.

- **Object Action**
  Bones only, when enabled, this option will make the constrained Bone use the “Object” part of the linked action, instead of the “same-named pose” part. This allows you to apply the action of an Object to a Bone.

- **Target Range Min/Max**
  The lower and upper bounds of the driving transform property value.

  Here again we find the **Constraints limitations**:  
  - When using a rotation property as “driver”, these values are “mapped back” to the (-180.0 to 180.0) range.  
  - When using a scale property as “driver”, these values are limited to null or positive values.

- **Action Range Start/End**
  The starting and ending frames of the action to be mapped.
Child of Constraint

Child of Constraint is the **Constraint version of the standard** parent/children relationship between Objects (the one established through the Ctrl-P shortcut, in **3D Views**).

Parenting with a Constraint has **several advantages** and enhancements, compared to the traditional method: (Refer [Img 1.13])

- You can have several **different parents** for the same Object (weighting their respective influence with the Influence slider).

- As with any Constraint, you can key (i.e. animate) its Influence setting. This allows the Object which has a Child of Constraint upon it to change over time which Target Object will be considered the parent, and therefore have influence over the Child of Constraint Object.

**Tip**

*Do not confuse this “basic” Object parenting with the one that defines the chains of Bones inside of an armature. This Constraint is used to parent an **Object to a Bone** (the so-called **Object skinning**), or even Bones to Bones. However, do not try to use it to define chains of Bones.*

**Options**

![Child Of panel](source-file)

**Title**-Img 1.10 Child Of panel.

**Source**-blender.org
Target
The Target Object that this Object will act as a child of. This Constraint uses one Target, and is not functional (red state) when it has none. If Target is an armature or a mesh, a new name field appears where a name of a Bone or a Vertex Group can be selected.

Location X, Y, Z
Each of these buttons will make the parent affect or not affect the location along the corresponding axis.

Rotation X, Y, Z
Each of these buttons will make the parent affect or not affect the rotation around the corresponding axis.

Scale X, Y, Z
Each of these buttons will make the parent affect or not affect the scale along the corresponding axis.

Set Inverse
By default, when you parent your owner to your Target, the Target becomes the origin of the owner’s space. This means that the location, rotation and scale of the owner are offset by the same properties of the Target. In other words, the owner is transformed when you parent it to your Target. This might not be desired! So, if you want to restore your owner to its before-parenting state, click on the Set Inverse button.

Clear Inverse
This button reverses (cancels) the effects of the above one, restoring the owner/child to its default state regarding its Target/parent.
When creating a new parent, relationship using this Constraint, it is usually necessary to click on the Set Inverse button after assigning the parent. As noted above, this cancels out any unwanted transform from the parent, so that the owner returns to the location/rotation/scale it was in before the Constraint was applied. Note that you should apply Set Inverse with all other Constraints disabled (their Influence set to 0.0) for a Child Of Constraint, and before transforming the Target/parent.

About the toggle buttons that control which Target’s (i.e. parent’s) individual transform properties affect the owner, it is usually best to leave them all enabled, or to disable all three of the given Location, Rotation or Scale transforms.

Technical Note

If you use this Constraint with all channels on, it will use a **straight matrix multiplication** for the parent relationship, not decomposing the parent matrix into loc/rot/size. This ensures any transformation correctly gets applied, also for combinations of rotated and non-uniform scaled parents.
Unit summary

In this Unit, you have learnt what is Rigging and how to

- Use the Constraints
- Do Adding or Removing Constraints
- Describe the Functions of Header
- Explain the term Header, Target, Space, Influence
- Edit the Properties of the Header panel, working on Parent and Child relationship
- Create IK Constraint
- Create Spline IK Constraint

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating Rig.

Assignment

- Create a basic Mechanical Rig referring to the YouTube video link
Assessment

- Define Constraints.
- State the Differentiate between Local space and World space.
- Write notes on Adding and Deleting Constraints with examples.
- Describe the Process of making a Parent Constraint.
- Write a brief note on the uses of Header.
- Explain Parent and Child Constraint.
- Write down the process of Creating Spline IK.
- Define Stretch Constraint

Fill in the Blanks

1. ________ is a good first Constraint to explore in the beginning.
2. __________ sits at the top of every Constraint.
3. __________ Enables or Disables (Mute/Unmute) the Constraint.
4. __________ allows you to control an Action using the transformations of another Object.
5. The __________ slider determines how much the Constraint will affect the constrained Object.
Resources

While studying this Unit, you can browse the internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- archive.org
- www.blender.org
- docs.blender.org
Unit 2

Introduction to Working with Armature

Introduction

An Armature in Blender is similar to the Armature of a real skeleton. Just like a real skeleton, an Armature can consist of many Bones. These Bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way. An “Armature” is a type of Object used for rigging. Armature Object borrows many ideas from real life skeletons.

As Armatures are designed to be posed, either for a static or animated scene, they have a specific state, called “rest position”. This is the Armature’s default “shape”, the default position/rotation/scale of its Bones, as set in Edit mode.

In Edit mode, you will always see your Armature in rest position, whereas in Object and Pose mode, you usually get the current “pose” of the Armature (unless you enable the Rest Position button of the Armature panel).

This unit will describe how you can work with Armatures in Blender.

Outcomes

Upon completion of this unit you will be able to:

- Explain the Usage of Bones
- List the types of Armature structure
- Manage to Edit an Armature
- Arrange Linking Objects to Bone
- Set up Mesh and Armature using Skinning
- Create Poses for the Rigged Character
## Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Armatures:</strong></td>
<td>Armature is the Object type used for rigging and it borrows many ideas from real life skeletons.</td>
</tr>
<tr>
<td><strong>Roll:</strong></td>
<td>Activating Axes checkbox on the will show local axes for each Bone’s tip.</td>
</tr>
<tr>
<td><strong>Bones Influence:</strong></td>
<td>Basically, a Bone controls geometry when vertices “follow” the Bone.</td>
</tr>
<tr>
<td><strong>Armature Layers:</strong></td>
<td>Each Armature has 32 “Armature layers” which allow you to organize your Armature by “regrouping” sets of Bones into layers.</td>
</tr>
<tr>
<td><strong>Octahedral Bone:</strong></td>
<td>This is the default visualization, well suited for most of editing tasks.</td>
</tr>
<tr>
<td><strong>Stick Bone:</strong></td>
<td>This is the simplest and most non-intrusive visualization.</td>
</tr>
<tr>
<td><strong>B-Bone Bone:</strong></td>
<td>This visualization shows the curves of “smooth” multi-segmented Bones; see the Bendy Bones for details.</td>
</tr>
<tr>
<td><strong>Envelope Bone:</strong></td>
<td>This visualization materializes the Bone deformation influence.</td>
</tr>
<tr>
<td><strong>Pose Library:</strong></td>
<td>The Pose Library panel is used to save, apply, and manage different Armature poses.</td>
</tr>
<tr>
<td><strong>Ghost:</strong></td>
<td>In traditional cartoon creation animators use tracing paper, to see several frames preceding the one they are working.</td>
</tr>
<tr>
<td><strong>Shadow:</strong></td>
<td>Controls how objects using this Material cast and receive shadows.</td>
</tr>
<tr>
<td><strong>Structure:</strong></td>
<td>Armatures mimic real skeletons. They are made from Bones, which are (by default) rigid elements.</td>
</tr>
<tr>
<td><strong>Chains of Bones:</strong></td>
<td>Bone can be the parent of several children, and hence be part of several chains at the same time.</td>
</tr>
</tbody>
</table>
Working with Armature

First, let us try to add the **default Armature** in Blender.

- **Step 1**: Open a default scene.
- **Step 2**: Delete all Objects in the scene.
- **Step 3**: Make sure the cursor is in the world origin with Shift-C.
- **Step 4**: Press Numpad1 to see the world in Front view.
- **Step 5**: Add a Single Bone (Add • Armature • Single Bone).
- **Step 6**: Press Numpad Delete to see the Armature at maximum zoom.

![Image 2.1 The default Armature.](https://docs.blender.org/manual/en/dev/rigging/armatures/introduction.html)

**Source**: blender.org  

**Armature Object**

As you can see, an Armature is like any other Object type in Blender:

- It has a **center, a position, a rotation and a scale factor**.
- It has an **Object Data data-block** that can be edited in **Edit Mode**.

- It can be linked to other scenes, and the same Armature data can be **reused** on multiple Objects.

- All animation you do in Object Mode is only working on the whole Object, not the **Armature’s Bones** (use the **Pose Mode** to do this).

- As Armatures are designed to be posed, either for a **static or animated scene**, they have a specific state, called “**rest position**”. This is the Armature’s default “**shape**”, the default position/rotation/scale of its Bones, as set in Edit Mode.

- In **Edit Mode**, you will always see your **Armature in rest position**, whereas in **Object Mode and Pose Mode**, you usually get the current “**pose**” of the **Armature** (unless you enable the Rest Position button of the Armature panel).

**Bones**

**Structure**

![Image of a Bone with labeled parts](https://docs.blender.org/manual/en/dev/rigging/armatures/bones/structure.html)

**Title-Img 2. 2 The elements of a Bone.**

**Source**- blender.org

**Link**- https://docs.blender.org/manual/en/dev/rigging/armatures/bones/structure.html

**They have three elements:**

1. “Start joint” named **Root** or Head,
2. **“Body”** itself,

3. **“End joint”** named **Tip** or Tail.

With the default Armature in edit-mode, you can select the root and the tip, and move them as you do with mesh vertices. Both root and tip (the “**joints**”) define the Bone by their respective position.

They also have a radius property, only useful for the envelope deformation method.

**Roll**

Activating Axes checkbox on the Armature tab • Display panel, will show local axes for each Bone’s tip. The Y axis is always aligned along the Bone, oriented from root to tip. So, this is the “**roll**” axis of the Bones.

**Bones Influence**

Title-Img 2. 3 A Bone in Envelope visualization, in Edit Mode.

**Source**: blender.org


Basically, a Bone **controls geometry** when vertices “follow” the Bone. This is like how the muscles and skin of your finger follow your **finger-Bone when you move a finger**.

To do this, you must define the strength of influences a Bone has on a certain vertex.
The simplest way is to have each Bone affecting those parts of the geometry that are within a given range from it. This is called the **Envelope Technique**, because each Bone can control only the geometry “enveloped” by its own influence area.

If a Bone is visualized as **Envelope**, in **Edit Mode** and in **Pose Mode** you can see the area of influence, which depends on:

- The Distance Property and
- The Root’s Radius and the Tip’s Radius.

![Our Armature in Envelope visualization, in Pose Mode.](image)

**Source-** blender.org  
**Link-**  

**Selection of Bones**

You can select and edit **Bones of Armatures** in Edit Mode and in Pose Mode. Here, we will see how to select Bones in Edit Mode. Selecting Bones in **Pose Mode** is similar to selecting in **Edit Mode** with a few specific differences that will be detailed in the posing part.

Similar to vertices/edges selection in meshes, there are **two ways** to select whole Bones in Edit Mode:

- Directly, by selecting the Bone’s body.
- Selecting both of its joints (roots and tip).
This is an important point to understand, because selecting Bones’ joints only might lead to non-obvious behavior, with respect to which Bone you actually select.

That unlike the mesh draw type the Armature draw type has no effect on selection behavior. In other words, you can select a Bone’s joint or body the same way regardless of the Bone visualization chosen.

Tip

Selecting Bone Joints

To select Bones’ joints, you have the standard selection methods.

- Inverse selection

As stated above, you must remember that these selection tools are for Bones’ joints only, not the Bones’ bodies.

For example, the Inverse selection option Ctrl-I inverts the selection of Bones’ joints, not of Bones.

Remember that a Bone is selected only if both its joints are selected. So, when the selection status of Bones’ joints is inverted, a new set of Bones is selected.

Source- blender.org

Link-

Selecting connected Bone Joints

Another Example: when you select the root of a Bone connected to its parent, you also implicitly select the tip of its parent (and vice versa).

Tip

Remember that when selecting Bones’ joints, the tip of the parent Bone is the “same thing” as the root of its children Bones.

Selecting Bones

By RMB clicking on a Bone’s body, you will select it (and hence you will implicitly select its root and tip).

Using Shift-RMB, you can add to/remove from the selection.

You also have some Advanced Selection options, based on their relations.

You can select at once all the Bones in the chain which the active (last selected) Bone belongs to by using the linked selection tool, L.

Linked Bones selection
Title-Img 2. 7 A single selected Bone.
Source- blender.org
Link-

Title-Img 2. 8 Its whole chain selected with L.

Source- blender.org
Link-

- Mirror **Shift-Ctrl-M**
- Flip the selection from one side to another.
- Pick Shortest Path **Ctrl-RMB**
- Selects the path from the active Bone to the Bone under the mouse.
- Deselecting connected Bones

There is a subtlety regarding connected Bones.
When you have several connected Bones selected, if you **deselect one Bone**, its tip will be deselected, however, **not its root**, if it is also the tip of another selected Bone.

To understand this, look at [Img 2.9](https://docs.blender.org/manual/en/dev/rigging/armatures/bones/selecting.html) Bone deselection in a selected chain.

Bone deselection in a selected chain.

- After **Shift-RMB**-clicking “**Bone.003**”:
• “Bone.003” ‘s tip (which is same as “Bone.004” ‘s root) is deselected.

• “Bone” is “Bone.003” ‘s parent. Therefore “Bone.003” ‘s root is same as the tip of “Bone”. Since “Bone” is still selected, its tip is selected. Thus, the root of “Bone.003” remains selected.

• Mouse Clicks

• Reference

• Mode: Edit Mode

• Hotkey: Ctrl-LMB

If at least one Bone is selected, Ctrl-LMB -clicking adds a new Bone.

**About the new Bone’s tip:**

After you Ctrl-LMB -clicked it becomes the Active Element in the Armature,

It appears to be right where you clicked, however, (As in mesh editing) it will be on the plane parallel to the view and passing through the 3D cursor.

The position of the Root and the parenting of the new Bone depends on the Active Element.

**Active Element**

**If the Active Element is a Bone**

The new Bone’s Root is placed on the Active Bone’s tip

The new Bone is parented and connected to the Active Bone (check the Outliner in [Img 2.11](#) Ctrl-clicking when the Active Element is a Bone.).
If the Active Element is a Tip

The new Bone’s root is placed on the **Active Tip**

The new Bone is parented and connected to the Bone owning the **Active Tip** (check the Outliner in img 2.12 Ctrl-clicking when the Active Element is a tip.).

If the Active Element is a Disconnected Root:

The new Bone’s root is placed on the **Active Root**
The new Bone is not parented to the Bone owning the Active Root (check the Outliner in *Img 2.13* Ctrl-clicking when the Active Element is a disconnected root.).

And hence the new Bone will not be connected to any Bone.

![Image 2.13 Ctrl-clicking when the Active Element is a disconnected root.](https://docs.blender.org/manual/en/dev/rigging/armatures/bones/editing/bones.html)

**Source:** blender.org

**Link:**

**If the Active Element is a Connected Root:**

The new Bone’s root is placed on the Active Root.

The new Bone is parented and connected to the parent of the Bone owning the Active Root (check the Outliner in *Img 2.14* Ctrl-clicking when the Active Element is a connected root.).

This should be obvious because if the Active Element is a connected root then the Active Element is also the tip of the parent Bone, so it is the same as the second case.

As the tip of the new Bone becomes the Active Element, you can repeat these Ctrl-RMB several times, to consecutively add several Bones to the end of the same chain.
Delete Selected Bone(s)

**Hotkey:** X

This tool deletes selected Bones, selected joints are ignored.

If you delete a Bone in a chain, its child(ren) will be automatically re-parented to its own parent, however, not connected, to avoid deforming the whole Armature.

Merge Bones

**Hotkey:** Alt-M

You can merge together several selected Bones, as long as they form a chain. Each sub-chain formed by the selected Bones will give one Bone, whose root will be the root of the root Bone, and whose tip will be the tip of the tip Bone.

Subdivide Bones

You can subdivide Bones, to get two or more Bones where there was just one Bone. The tool will subdivide all selected Bones, preserving the existing relationships: the Bones created from a subdivision always form a connected chain of Bones.
Naming Conventions

Naming conventions in Blender are not only useful for you in finding the right Bone, however, also to tell Blender when any two of them are counterparts.

In case your Armature can be mirrored in half (i.e. it is bilaterally symmetrical), it is worthwhile to stick to a left/right naming convention. This will enable you to use some tools that will probably save your time and effort (like the X-Axis Mirror editing tool we saw above...).

Title-Img 2. 15 An example of left/right Bone naming in a simple rig.

Source-blender.org
Link-

Structure

Armatures mimic real skeletons. They are made out of Bones, which are (by default) rigid elements. However, you have more possibilities than with real skeletons: In addition to the “natural” rotation of Bones, you can also translate and even scale them! And your Bones do not have to be connected to each other; they can be completely free if you want. However, the most natural and useful setups imply that some Bones are related to others, forming so-called “chains of Bones”.
Chains of Bones

The Bones inside an Armature can be completely independent from each other (i.e. the modification of one Bone does not affect the others). However, this is not often a useful set up: To create a leg, all Bones “after” the thigh Bone should move “with” it in a well-coordinated manner. This is exactly what happens in Armatures by parenting a Bone to the next one in the limb, you create a “chains of Bones”. These chains can be ramified. For example, five fingers attached to a single “hand” Bone.

Skinning

In Blender, you have two main skinning types:

1. You can Parent/Constrain Objects to Bones - then, when you transform the Bones in Pose Mode, their “children” Objects are also transformed, exactly as with a standard parent/children relationship... The “children” are never deformed when using this method.
2. You can use the **Armature Modifier on entire Mesh**, and then, some parts of this Object to some Bones inside this Armature. This is the more complex and powerful method, and the only way to really deform the geometry of the Object, i.e. to modify its vertices/control points relative positions.

**Armature Deform Parent**

**Hotkey: Ctrl-P**

Armature Deform Parenting is a way of **creating and setting up** an **Armature Modifier**.

To use Armature Deform Parenting, you must

- **Step 1:** First select all the child Objects that will be influenced by the Armature

- **Step 2:** Lastly, select the **Armature Object** itself.

- **Step 3:** Once all the child Objects and the Armature are selected press **Ctrl-P**

- **Step 4:** Select Armature Deform in the **Set Parent To** pop-up menu.

The Armature will be the parent Object of all the other child Objects and each child Object will have an **Armature Modifier** with the **Armature associated** (Object field).

Title-Img 2. 17 Bone associated with Mesh Object.

*Source-*
With Empty Groups

When parenting, it will create empty vertex groups on the child Objects (if they do not already exist) for and named after each deforming Bone in the Armature. The newly created vertex groups will be empty this means they will not have any weights assigned. Vertex groups will only be created for Bones which are setup as deforming (Properties Editor ▶ Bone ▶ Deform Panel). You can then manually select the vertices and assign them to a particular vertex group that you are choosing to have Bones in the Armature influence.

Choose this option if you have already created (and weighted) all the vertex groups the mesh requires.

Example

For example, if you have an Armature which consists of three Bones named “Bone A”, “Bone B” and “Bone C” and cube mesh called “Cube”. If you parent the cube to the Armature the cube will get three new vertex groups created on it called “Bone A”, “Bone B” and “Bone C”. Notice that each vertex group is empty.

![Image 2. 18 Cube in Edit Mode using Armature Deform with empty groups.](image)

Source-
With Automatic Weights

With Automatic Weights, parenting works similar to With Empty Groups, however, it will not leave the vertex groups empty. It calculates how much influence a particular Bone would have on vertices based on the distance from those vertices to a particular Bone (“Bone heat” algorithm). This influence will be assigned as weights in the vertex groups.

This method of parenting is certainly easier setup; however, it can often lead to Armatures which do not deform child Objects in ways you would want. Overlaps can occur when it comes to determining which Bones should influence certain vertices when calculating influences for more complex Armatures and child Objects. Symptoms of this confusion are that when transforming the Armature in Pose Mode parts of the child Objects do not deform as you expect; If Blender does not give you the results you require you will have to manually alter the weights of vertices in relation to the vertex groups they belong to and have influence in.

With Envelope Weights

With Envelope Weights, parenting works in a similar way like Automatic Weights. The difference is that the influences are calculated based on the Bone Envelopes settings. It will assign to each vertex groups the vertices that are inside its Bone’s influence volume, weighted depending on their distance to this Bone.

This means newly included/excluded vertices or new envelope settings will not be taken into account. You will have to apply Armature Deform with Envelope Weights parenting again.

Tip

If want the envelope setting to be used instantly bind the Armature Modifier to Bone Envelopes.
Title-Img 2. 19 Two sets of Armatures each with three Bones.

Source-
Unit summary

In this Unit, you have learnt

- To Create and edit different types of Bones and apply it to different mesh
- Types of Armature structure
- To Edit an Armature
- To Arrange Linking Objects to Bone
- To Set up Mesh and Armature using Skinning
- To Create Poses for the Rigged Character
- To Skin the mesh to the Bones
- To Add and remove influence for a Bone
- After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Objects.

Assignment

Create a Basic Primitive Human Rig referring to the Youtube video link.

Assessment

- Explain Armature in Blender
- Describe the Deforming Bones
- Write a brief note on Bone influence
- Explain five different Processes of Editing Bones
- Write a brief note on Bone structure
- Explain the process of skinning with examples
Fill in the Blanks

1. _______ are directly involved in altering the positions of vertices associated with their Bones.
2. __________ can be seen when the Bone is in edit mode.
3. Using ______________command on the keyboard, you can add to/remove from the selection.
4. If at least one Bone is selected, clicking ________ adds a new Bone.
5. In 3D View, clicking ___________ will add a new Bone to your Armature.

Resources

While studying this Unit, you can browse the internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- archive.org
- www.blender.org
- docs.blender.org

Study Skills
Unit 3

3D Animation

Introduction

In 3D Animation and other forms of Computer Animation, the frames are generated by interpolating between the numerical values that are defined in any two consecutive keyframes. Typically, in 3D Animation, this interpolation takes the form of 3D Beziers curves (paths) which are constructed as a series of control points, allowing for the interactive manipulation of smooth 3D curves.

In this Unit, you will learn about 3D Animation and how animation is making an object move or change shape over time. This will be done through different tools and techniques in Blender.

Outcomes

Upon completion of this unit you will be able to:

- Describe Keyframe Animation
- Work with Timeline
- Work with F - Curves
- Design the Dope Sheet
- Create Animation on a Motion Path
- Work with Interpolation and Extrapolation
Terminology

**Keyframe:** Normal keyframe for Animation.

**Breakdown:** Breakdown state. e.g. for transitions between key poses

**Bones Influence:** Basically, a bone controls a geometry when vertices “follow” the bone

**Moving Hold:** A keyframe that adds a small amount of motion around a holding pose. In the Dope Sheet, it will also draw a bar between them.

**Timeline Editor:** The Timeline editor, identified by a clock icon, is shown by default at the bottom of the screen.

**Time Cursor:** The Time Cursor is the green line, it is used to set and display the current time frame.

**F Curve Editor:** After animating some property in Blender using keyframes you can edit their corresponding curves in F-Curve editor.

**Constant:** There is no interpolation at all. The curve holds the value of its last keyframe, giving a discrete (stairway) “curve”.

**Pose Library:** The Pose Library panel is used to save, apply, and manage different armature poses.

**Linear:** This simple interpolation creates a straight segment, giving a non-continuous line.

**Beziers:** The more powerful and useful interpolation, and the default one. It gives nicely smoothed curves, i.e. smooth animations!

Introduction to KeyFrame

**Keyframe**

A **Keyframe** is a marker of time which stores the value of a property.

For example, a Keyframe might define that the horizontal position of a cube is at 3m on frame 1.
The purpose of a Keyframe is to **allow for interpolated animation**, meaning, for example, that the user could then add another key on frame 10, specifying the cube’s horizontal position at 20m, and Blender will automatically determine **the correct position** of the cube for all the frames between frame 1 and 10 depending on the **chosen interpolation method** (e.g. Linear, Bézier, Quadratic, etc).

**Visualization**

There are some important **visualization features** in 3D Views that can help animation.

When the current frame is a keyframe for the current active object, the name of this object (shown in **Img 3.1** the bottom left corner of 3D Views) turns **yellow**.

**Title-** Img 3. 1 Bottom: Current frame at 0. Top: Current frame is a keyframe for Cube.

**Source-** blender.org

**Link-**

**Keyframe Types**

For **visually distinguish** regular keyframes from different animation events or states (extremes, breakdowns, or other in between), there is the possibility of applying different colors on them for visualization.

- **Keyframe (yellow diamond)**
  
  Normal keyframe.
• **Breakdown (cyan small diamond)**
  Breakdown state. e.g. for transitions between key poses.

• **Moving Hold (slight orange diamond)**
  A keyframe that adds a small amount of motion around a holding pose. In the Dope Sheet, it will also draw a bar between them.

• **Extreme (red big diamond)**
  An ‘extreme’ state or some other purpose as needed.

• **Jitter (green tiny diamond)**
  A filler or baked keyframe for keying on ones, or some other purpose as needed.

**Insert Keyframe**

• **Mode:** Object Mode

• **Panel:** Tool Shelf ➔ Animation ➔ Animation ➔ Keyframes: Insert

• **Menu:** Object ➔ Animation ➔ Insert Keyframe...

• **Hotkey:** I

There are several methods of adding new keys, namely:

• In 3D View, pressing “I” will bring up a menu to choose what to add a keyframe to.

• Hovering over a property and pressing “I” or with the context menu by RMB a property and choose Insert Keyframe from the menu.

**Auto Keyframe**

Auto Keyframe is the **red record button** in the Timeline header. Auto Keyframe adds **keyframes automatically** to the set frame if the value for transform type properties changes.
Title- Img 3. 2 Timeline Auto Keyframe.

Source: blender.org
Link: https://docs.blender.org/manual/en/dev/animation/keyframes/editing.html

Delete Keyframe

Reference
- **Mode**: Object Mode
- **Panel**: Tool Shelf • Animation • Animation • Keyframes: Remove
- **Menu**: Object • Animation • Delete Keyframes...
- **Hotkey**: Alt-I

There are several methods of removing keyframes:
- In 3D View, Press **Alt-I** to remove keys on the current frame for selected objects.
- When the mouse is over a value press **Alt-I**.
- **RMB** a value and choose Delete Keyframe from the menu.

Clear Keyframe

Reference
- **Mode**: Object Mode
- **Menu**: Object • Animation • Clear Keyframes.
KeyFrame Animation

This example shows you how to animate a cube's location, rotation, and scale. (Refer \textit{Img 3.3})

- **Step 1:** First, in the Timeline, or other animation editors, set the \textbf{frame to 1}.

- **Step 2:** With the Cube selected in Object Mode, press I in 3D View.

- **Step 3:** From the Insert Keyframe Menu select \textbf{LocRotScale}. This will record the location, rotation, and scale, for the Cube on frame 1.

- **Step 4:** Set the \textbf{frame to 100}.

- **Step 5:** Use Grab/Move G, Rotate R, Scale S, to transform the cube.

- **Step 6:** Press I in 3D View. From the Insert Keyframe Menu, select \textbf{LocRotScale}.

![Insert Keyframes](https://docs.blender.org/manual/en/dev/animation/keyframes/editing.html)

\textit{Title- Img 3. 3 Insert Keyframes.}

\textit{Source-blender.org}

\textit{Link-}

\url{https://docs.blender.org/manual/en/dev/animation/keyframes/editing.html}
To test the animation, press Alt-A Play

Title- Img 3. 4 The animation on frames 1, 50, 100.

Source-blender.org

Timeline Editor

The Timeline editor, identified by a clock icon, is shown by default at the bottom of the screen.

Title- Img 3. 5 The Timeline.

Source-blender.org

The Timeline is not much of an editor, but more of an information and control.

Here, you can have an overview of the animation part of your scene.

- What is the current time frame, either in frames or in seconds?

- Where are the keyframes of the active object, the start and end frames of your animation, markers, etc.
The Timeline has **Player Controls**, to play, pause the animation, and to skip though parts of the scene. It also has **some tools** for Keyframes, Keying Sets, and Markers.

**Main View**

The Main Timeline region displays the animation frames over time.

[Image: Timeline Main Area]

Source: blender.org  

**Adjusting the View**

The Timeline can be panned by holding **MMB**, then dragging the area left or right.

You can zoom the Timeline by using **Ctrl-MMB**, the mouse Wheel, or pressing Numpad Minus and Numpad Plus.

**Time Cursor**

Time Cursor is the **green line**, it is used to **set and display** the current time frame.

[Image: Time Cursor]

Source: blender.org  

Time Cursor can be set or moved to a new position by pressing or holding **LMB** in the Timeline editor.

The current frame or second can be displayed on the **Time Cursor**, check the View menu for settings. The Time Cursor can be moved in steps by pressing **Left** or **Right**, or in steps of 10 frames by pressing **Shift-Up** or **Shift-Down**.
Using F Curve in Animation

After animating some property in Blender using keyframes, you can edit their corresponding curves. When something is “animated,” it changes over time. This curve is shown as something called an F-Curve. Basically, what an F-Curve does is an interpolation between two animated properties. In Blender, animating an object means changing one of its properties, such as the object’s location, or its scale.

As mentioned, Blender’s fundamental Unit of time is the “frame”, which usually lasts just a fraction of a second, depending on the frame rate of the scene. As animation is composed of incremental changes spanning multiple frames, usually these properties are not manually modified frame by frame, because:

- It would take ages!
- It would be very difficult to get smooth variations of the property (unless you compute mathematical functions and type a precise value for each frame, which would be crazy).

Therefore, nearly all direct animation is done using interpolation. The idea is simple: you define a few Keyframes, which are multiple frames apart. Between these keyframes, the properties’ values are computed (interpolated) by Blender and filled in. Thus, the animators’ workload is significantly reduced.

Title- Img 3. 8Example of interpolation.

Source-blender.org
Link-
For example, if you have:

- A control point of value 0 at frame 0,
- Another one of value 10 at frame 25,
- And you use linear interpolation,
- Then, at frame 5 we get a value of 2.

The same goes for all intermediate frames: with just two points, you get a smooth growth from (0 to 10) along the 25 frames. Obviously, if you would like the frame 15 to have a value of 9, you would have to add another control point (or keyframe)

Settings

F-Curves have three additional properties, which control the interpolation between points, extension behaviour, and the type of handles.

Interpolation Mode

Reference

- **Menu**: Key › Interpolation Mode
- **Hotkey**: T
- **Mode**: for the Interpolation between the current and next keyframe.

Interpolation

Constant

There is no interpolation at all. The curve holds the value of its last keyframe, giving a discrete (stairway) “curve”. Usually only used during the initial “blocking” stage in pose-to-pose animation workflows.
3D Animation

Title- Img 3. 9Example of Constant.

Source-blender.org
Link- 

Linear

This simple interpolation creates a straight segment, giving a non-continuous line. It can be useful when using only two keyframes and the Extrapolation extend mode, to easily get an infinite straight line (i.e. a linear curve).

Title- Img 3. 10Example of Linear.

Source-blender.org
Link- 

Bezier

The more powerful and useful interpolation, and the default one. It gives nicely smoothed curves, i.e. smooth animations!
Title- Img 3. 11Example of Bezier.

Source-blender.org

Remember that some F-Curves can only take discrete values, in which case they are always shown as if constant interpolated, whatever option you chose.

Easing Interpolation (by strength)

There are different methods of easing interpolations for F-Curve segment. The “Robert Penner easing equations” (basically, equations which define some preset ways that one keyframe transitions to another) which reduce the amount of manual work (inserting and tweaking keyframes) to achieve certain common effects. For example, snappy movements.

- Linear
- Sinusoidal
- Quadratic
- Cubic
- Quartic
- Quintic
• Exponential
• Circular

• Dynamic Effects
  These additional easing types imitate (fake) physics-based effects like bouncing/springing effects. The corresponding settings can be found in the Properties region • Active Keyframe panel.

• Elastic
  Exponentially decaying sine wave, like an elastic band. This is like bending a stiff pole stuck to some surface, and watching it rebound and settle back to its original state.

• Amplitude
  The amplitude property controls how strongly the oscillation diverges from the basic curve. At 0.0, there is no oscillation (i.e. it just snaps to the B-value like an extreme exponential transition), and at 1.0 a profile similar to the one shown in the icon occurs.

• Period
  The period property controls the frequency with which oscillations occur. Higher values result in denser oscillations.

• Bounce
  Exponentially decaying parabolic bounce, like when objects collide. e.g. for Bouncing balls, etc.

• Back
  Cubic easing with overshoot and settle. Use this one when you want a bit of an overshoot coming into the next keyframe, or perhaps for some wind-up anticipation.

• Back
  The back property controls the size and direction (i.e. above/below the curve) of the overshoot.

Easing Type
Reference

- **Menu**: Key ➔ Easing Type

- **Hotkey**: Ctrl-E

The Easing Type controls which end of the segment between the two keyframes that the easing effects apply to.

**Automatic Easing**

The most commonly expected of the below behaviours is used. For the transitional effects, this is basically ease in, while for the physics effects it is ease out.

- **Ease In**
  Effect builds up to the second keyframe.

- **Ease Out**
  Effect fades out from the first keyframe.

- **Ease In Out**
  Effect occurs on both ends of the segment.

**Extrapolation**

Reference

- **Menu**: Channel ➔ Extrapolation Mode

- **Hotkey**: Shift-E

Extrapolation defines the behaviour of a curve before the first and after the last keyframes.

There are **two basic extrapolation** modes:

**Constant**

The default one, curves before their first keyframe and after their last one has a constant value (the one of these first and last keyframes).
Constant extrapolation.

Source-blender.org

Linear

Curves ends are **straight lines** (linear), as defined by their first two keyframes (respectively their last two keyframes).

Source-blender.org
Handle Types

There is another curve option quite useful for Bézier-interpolated curves. You can set the type of handle to use for the curve points

- Automatic
  
  Keyframes are automatically interpolated.

- Vector
  
  Creates linear interpolation between keyframes. The linear segments remain if keyframe centres are moved. If handles are moved, the handle becomes Free.
Handle maintain rotation when moved, and curve tangent is maintained.

Title- Img 3. 16 Aligned handles.

Source-blender.org

- Free
  Breaks handles tangents.

Title- Img 3. 17 Free handles.

Source-blender.org

- Auto Clamped
  Auto handles clamped to not overshoot.

Title- Img 3. 18 Auto clamped handles.
Using Dope Sheet in Animation

The Dope Sheet

Classical hand-drawn animators often made a chart, showing exactly when each drawing, sound and camera move would occur, and for how long. They nicknamed this the “dope sheet”. While CG foundations dramatically differ from classical hand-drawn animation, Blender’s Dope Sheet inherits a similar directive. It gives the animator a “birds-eye-view” of everything occurring within a scene.
Dope Sheet Modes

Title- Img 3. 20Dope Sheet Modes.

Source-blender.org

- **Dope Sheet**
  The Dope Sheet Mode allow you to edit multiple actions at once.

- **Action Editor**
  Action Editor is where you can define and control actions.

- **Shape Key Editor**
  ShapeKey Editor is dedicated to the shape key data-blocks.

- **Grease Pencil**
  Grease Pencil Mode is dedicated to the grease pencil tool’s keyframes for each grease pencil layer, you have a strip along which you can grab its keys, and hence easily re-time your animated sketches.

- **Mask**
  Mask Mode is dedicated to the mask data-blocks.

- **Cache File**
  To do.

- **Interface**
The Dope Sheet Editor interface is somewhat similar to the Graph Editor one, it is divided in three regions:

Title- Img 3. 21The Action Editor with object channels.

Source-blender.org

Motion Paths

Reference

- **Mode:** Object Mode
  
- **Panel:** Tool Shelf ▶ Animation ▶ Animation ▶ Motion Paths: Calculate
  
- **Panel:** Properties editor ▶ Object ▶ Motion Paths

Reference

- **Mode:** Pose Mode
  
- **Panel:** Tool Shelf ▶ Tools ▶ Pose Tools ▶ Motion Paths: Calculate
  
- **Panel:** Properties editor ▶ Armature ▶ Motion Paths
  
- **Menu:** Pose ▶ Motion Paths
Title: Img 3. 22 An animated cube with its motion path displayed.

Source: blender.org


This feature allows you to visualize the motion of points as paths over a series of frames. These points can be object origins and bone joints.

Before we look at its options, let us first see how to display/hide these paths. Unlike Ghost, you must do it manually and you have to first select the bones you want to show/hide the motion paths. Then,

- To show the paths (or update them, if needed), click on the Calculate Path button.

- To hide the paths, click on the Clear Paths button.

Tip

Remember that only selected bones and their paths are affected by these actions!

The paths are drawn in a light shade of gray for unselected points, and a slightly blueish gray for selected ones. Around the current frame a glow indicate the direction of movement: blue towards future frames and green towards the past. Each frame is displayed by a small white dot on the paths. As with ghosts, the paths are automatically updated when you edit your poses/keyframes, and they are
also active during animation playback. Alt-A is only useful when the Around Current Frame option is enabled.

Options

![Motion Paths Panel in the Armature tab](http://blender.org)

Source: blender.org


Type

- **Around Frame**
  
  Display paths of points within a fixed number of frames around the current frame. When you enable this button, you get paths for a given number of frames before and after the current one (again, as with ghosts).

- **In Range**
  
  Display paths of points within specified range.

Display Range

- **Before, After**
  
  Number of frames to show before and after the current frame (only for Around Current Frame Onion-skinning method).

- **Start, End**
Starting and Ending frame of range of paths to display/calculate (not for Around Current Frame Onion-skinning method).

- **Step**
  This is the same as the Step for ghosts. It allows you to only display on the path one frame for each n ones. Mostly useful when you enable the frame number display (see below), to avoid cluttering 3D Views.

**Cache/Cache for Bone**

- **From, To**
  These are the start/end frames of the range in which motion paths are drawn. You cannot modify this range without deleting the motion path first.

- **Calculate/Update Paths**
  If no paths have been calculated, Calculate Paths will create a new motion path in cache based on the options specified in the pop-up menu or Operator panel.

  If a path has already been calculated, Update Paths will update the path shape to the current animation. To change the frame range of the calculated path, you need to delete the path and calculate it again.

- **Start, End**
  These are the start/end frames of the range in which motion paths are drawn. You must Calculate Paths again if you modify this setting, to update the paths in 3D Views. Note that unlike with ghosts, the start frame is inclusive (i.e. if you set Start to 1, you will really see the frame 1 as starting point of the paths...).

- **Bake Location**
  Bones only – By default, you get the tips’ paths. By changing this setting to Tails, you will get the paths of the bone’s roots (remember that in Blender UI, bones’ roots are called “heads”). You must Calculate Paths again if you modify this setting, to update the paths in 3D Views.
• **Clear Paths X**
  Clears paths on all objects/bones or just the selected ones when holding Shift.

**Show**

• **Frame Numbers**
  When enabled, a *small number* appears next to each frame dot on the path, which is of course the number of the corresponding frame.

• **Keyframes**
  When enabled, **big yellow square dots** are drawn on motion paths, showing the keyframes of their bones (i.e. only the paths of keyed bones at a given frame get a yellow dot at this frame).

• **+ Non-Grouped Keyframes**
  For bone motion paths, it searches the whole Action for keyframes instead of in groups with matching name only (this is slower).

• **Keyframe Numbers**
  When enabled, you will see the numbers of the displayed keyframes, so this option is obviously only valid when Show Keys is enabled.

**Example**


**Title- Img 3. 24** An example of a motion path of an armature.

**Source-blender.org**

Unit summary

In this Unit, you have learnt how to

- Create and edit Animation and Keyframes using key editor
- Work effectively with F curve to change the timing of animation
- Edit the Animation curve using different Interpolation method
- Use Motion Path for animating objects
- Work on Dope Sheet to make your animation more precise and accurate

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Objects.

Assignment

- Create a Logo or Title Animation for an existing popular brand like Amazon, Flipkart, Zee Tv etc. Students need to submit the assignment in MP4 format with 1920 X 1080 Resolution

Assessment

- Explain Keyframe Animation
- Describe five types of Keyframes
- Examine the key Interpolation and its types
- Explain the importance of F curve in Animation
- Differentiate between Ease in and Ease out

- Write a note on the uses of Motion Path
Fill in the Blanks

1. The ______________ is identified by a clock icon at the bottom of the screen.

2. Pose-to-pose animation workflows can be achieved by ______________ Interpolation.

3. ______________ defines the behaviour of a curve before the first and after the last keyframe.

4. Press ______________ to test the animation.

5. The ______________ is the green line, it is used to set and display the current time frame.

Resources

While studying this Unit, you can browse the internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- archive.org
- www.blender.org
- docs.blender.org
Unit 4

Textures and Mapping

Introduction

In this Unit, you will be learning the process of production of videos using Tracking, such as 2D Tracking, 3D Motion Tracking, Camera Tracking and Object Tracking. Tracking allows you to import raw footage, track the footage, and mask areas using Camera movements in your 3D scene. You will also learn about Stabilizing, Rotoscoping, Mask Editor, Mask Data block, Shape Keyframe and Layers.

Outcomes

Upon completion of this unit you will be able to:

- Use the Masking
- Describe the Process of Motion Tracking
- Explain the Process of Stabilizing a Clip
- Use the Movie Clip Editor
- Brief the process of Rotoscoping

Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Tracking</td>
<td>To Track the motion of Objects and applying that data to 3D Object through the compositor</td>
</tr>
<tr>
<td>Speed</td>
<td>To control the speed of sequence Tracking.</td>
</tr>
<tr>
<td>Frames Limit</td>
<td>Controls how many frames can be Tracked when the Track Sequence operator is called.</td>
</tr>
<tr>
<td>Margin</td>
<td>To disable Tracks when they become too close to the image boundary.</td>
</tr>
<tr>
<td>Sensor Width</td>
<td>Width of the CCD sensor in the Camera.</td>
</tr>
</tbody>
</table>
Pixel Aspect Ratio: Is the pixel aspect of the CCD Sensor.

Track Path: It helps to determine if a Track jumps from its position or not.

Display Stabilization: This option makes the displayed frame be affected by 2D stabilization settings.

R, G, B: RGB And B/W buttons at the top of this panel are used to control color channels used for frame preview and to make the whole frame gray scale.

Clip Panel: This panel currently contains the single operator Set as background which sets the clip currently being edited as the Camera background for all visible 3D Views.

Solve Panel: Camera Motion operator solves the motion of Camera using all Tracks placed on the footage and two keyframes specified on this panel.

Cleanup Panel: This panel contains a single operator and its settings. This operator cleans up bad Tracks: Tracks which are not Tracked long enough or which failed to reconstruct accurately.

Introduction to Motion Tracking

Motion Tracking is used to Track the motion of Objects and applying that data to 3D Object through the compositor. Blender’s motion Tracker supports a couple of very powerful tools for 2D Tracking and 3D Motion Tracking, including Camera Tracking and Object Tracking, as well as some specific features like the Plane Track for compositing. Tracks can also be used to move and deform masks for Rotoscoping in the Mask Editor, which is available as a special mode in the Movie Clip Editor.

Manual Lens Calibration

All Cameras record distorted video. Nothing can be done about this because of the way optical lenses work. For accurate Camera
motion, the exact value of the **focal length** and the “**strength**” of distortion are needed.

Currently, **focal length** can be automatically obtained only from the **Camera’s settings** or from the **EXIF information**. There are some tools which can help to find approximate values to compensate for **distortion**. There are also fully manual tools where you can use a **grid** which is getting affected by distortion model and deformed cells define straight lines in the footage.

You can also use the **grease pencil** for this – just draw a line, which should be straight on the footage using **poly line brush** and adjust the distortion values to make the grease pencil match lines on the footage.

To calibrate your Camera more accurately, use the **Grid Calibration Tool** from **OpenCV**. OpenCV is using the same distortion model.

**Camera and Object Motion Solving**

Blender not only supports the solving of Camera motion, including tripod shots, however, also the solving of Object motion in relation to the motion of the Camera. In addition to that there is the **Plane Track**, which solves the motion of all markers on one plane.

There are also plans to add more tools in the future, for example more **Automatic Tracking** and solving, multi-Camera solving and constrained solutions.

**Tools for Scene Orientation and Stabilization**

After Camera solving, you need to **orient the real scene** in 3D scene for more convenient compositing. There are tools to define the floor, the scene origin, and the X/Y axes to perform scene orientation.

Sometimes, the video footage includes spurious jumps and tilting movements, e.g. hand-held Camera. Based on some Tracked image elements, **2D Stabilization** can detect and compensate such movements to improve the quality of the final result.

**Clip View**

The Clip View is used in the main part of the **Movie Clip Editor**. Almost all Motion Tracking tools are concentrated in the Movie Clip Editor.
It should be mentioned that the Camera solver consists of **three quite separate steps**:

**2D Tracking of footage**

- Camera intrinsic (focal length, distortion coefficients) specification/estimation/calibration.
- Solving Camera, scene orientation, and scene reconstruction.

Tools in the clip editor are split depending on which step they are used in, so the interface is not cluttered up with scene orientation tools when only 2D Tracking can be done. The currently displayed tool category can be changed using the Mode menu, which is in the editor header.

![Movie Clip Editor Mode Menu](http://blender-manual-i18n.readthedocs.io/ja/latest/motion_tracking/

**Title-Img 4. 1Movie Clip Editor Mode Menu.**

**Source**-blender.org


However, almost all operators can be called from menus, so it is not necessary to change the mode every time you want to use a tool which is associated with a different editor mode.

In Tracking mode, only tools related to Tracking and Camera solving are displayed. Camera solving tools are included here; because, after solving, you will most probably want to re-track existing Tracks or place new Tracks to make solving more accurate.

**Tracking Settings Panel**

This panel contains all settings for **2D tracking algorithms**. Depending on which algorithm is used, different settings are displayed, however, there are a few that are **common** for all Tracker settings:
• Adjust Frames’ controls which patterns get Tracked; to be more precise, the pattern from which frame is getting Tracked. Here is an example which should make things clearer.

• The Tracker algorithm receives two images inside the search area and the position of a point to be Tracked in the first image. The Tracker tries to find the position of that point from the first image in the second image.

Now, this is how Tracking of the sequence happens. The second image is always from a frame at which the position of marker is not known (next Tracking frame). However, a different first image (instead of the one that immediately precedes the second image in the footage) can be sent to the Tracker.

Most commonly used combinations:

• An image created from a frame on which the Track was keyframed. This configuration prevents sliding from the original position (because the position which best corresponds to the original pattern is returned by the Tracker), however, it can lead to small jumps and can lead to failures when the feature point is deformed due to Camera motion (perspective transformation, for example). Such a configuration is used if Adjust Frames is set to 0.

• An image created from the current frame is sent as first image to the Tracker. In this configuration, the pattern is Tracking between two neighbouring frames. It allows dealing with cases of large transformations of the feature point however, can lead to sliding from the original position, so it should be controlled. Such a configuration is used if Adjust Frames is set to 1.

• If Adjust Frames is greater than 1, the behaviour of Tracker is: keyframes for Tracks are creating every Adjust Frame, and Tracking between keyframed image and next image is used.

Movie Clip Properties
Objects Panel

Title-Img 4. 2Objects panel in clip editor

Source-blender.org
Link- http://blender-manual-i18n.readthedocs.io/ja/latest/motion_tracking/

This panel contains a list of all Objects which can be used for Tracking, Camera or Object solving. By default, there is only one Object in this list, which is used for Camera solving. It cannot be deleted and other Objects cannot be used for Camera solving; all added Objects are used for Object Tracking and solving only. These Objects can be referenced from Follow Track and Object Solver constraints.

- Follow Track uses the Camera Object by default.
- New Objects can be added using Plus and the active Object can be deleted with the Minus button. Text field at the bottom of this panel is used to rename the active Object.
- If some Tracks were added and Tracked to the wrong Object, they can be copied to another Object using Track • Copy Tracks and Track • Paste Tracks.

The usage for all kind of Objects (used for Camera and Object Tracking) is the same: Track features, set Camera data, solve motion. Camera data is sharing between all Objects and refining of Camera intrinsic happens when solving Camera motion only.

Track Panel
First of all, Track name can be changed in this panel. Track names are used for linking Tracking data to other areas, like a Follow Track constraint.

- The next thing that can be controlled here is the marker’s enabled flag (using the button with the eye icon). If a marker is disabled, its position is not used either by solver nor by constraints.

The button with the lock icon to the right of the button with the eye controls whether the Track is locked. Locked Tracks cannot be edited at all. This helps to prevent accidental changes to Tracks which are “finished” (Tracked accurate along the whole footage).

- The next widget in this panel is called “Track Preview” and it displays the content of the pattern area. This helps to check how accurately the feature is being tracked (controlling that there is no sliding off original position) and helps to move the Track back to the correct position. The Track can be moved directly using this widget by mouse dragging.

If an anchor is used (the position in the image which is Tracking is different from the position which is used for Parenting), a preview widget will display the area around the anchor position. This configuration helps in masking some things when there is no good
feature at position where the mask corner should be placed. Details of this technique will be written later.

There is small area below the preview widget which can be used to enlarge the vertical size of preview widget (the area is highlighted with two horizontal lines).

- The next setting is **channels control**. Tracking happens in gray-scale space, so a high contrast between the feature and its background yields more accurate Tracking. In such cases disabling some color channels can help.

When several Tracks are used for 3D Camera reconstruction or for 2D stabilization, it is possible to assign a reduced weight to some Tracks to control their influence on the solution result. The Weight parameter is used for 3D reconstruction, while the Stab Weight parameter is used to control 2D stabilization. This parameter can (and often need to be) animated.

- The last thing is **custom color**, and the preset for it. This setting overrides the default marker color used in the clip editor and 3D View, and it helps to distinguish different type of features (for example, features in the background vs. foreground and so on). Color also can be used for “grouping” Tracks so a whole group of Tracks can be selected by color using the Select Grouped operator.

**2D Stabilization**

**2D video stabilization** is a feature built on top of Blender’s image feature Tracking abilities: we use some Tracking points to remove shakiness, bumps and jerks from video footage. Typically, image stabilization is part of a 2D workflow to prepare and improve footage prior to further processing or modeling steps. This page helps to understand how it works, introduces related terms and concepts, describes the available interface controls in detail and finally gives some hints about usage in practice.

**Typical usage scenarios of the Stabilizer**

- fixing minor deficiencies (shaky tripod, jerk in Camera movement)
- “poor man’s steadycam” (when a real steadycam was not available, affordable or applicable)
- preparing for masking, matching and Rotoscopying
It is not uncommon for 2D stabilization to have to deal with somewhat imperfect and flawed footage.

**How it works**

To detect spurious movement in the given shot, we’ll assume a simplified model about this movement. We then try to fit the movement of Tracked features with this simplified model to derive a compensation. Of course, this works only to the degree our model is adequate – yet in practice, this simplified approach works surprisingly well even with rather complicated shots, where our basic assumption was just an approximation of much more elaborate movements.

This simplified model underlying 2D stabilization as implemented here assumes movement by an Affine-Linear Transform:

- the Camera is pushed up/down/sideways by some translation component
- the image is then tilted and scaled around a Pivot Point (rotation center)

To compensate movement according to this simplified model, 2D stabilizer proceeds in two steps. First, we try to detect the translation offset from the weighted average of all translation Tracking points. After compensating this translation component, we then use additional rotation/scale Tracking points to detect rotation around a given pivot point. Again, we detect rotation and scale changes through a weighted average of all the rotation/scale Tracking points given.

In the current version, the Pivot Point is anchored to the weight center of the translation Tracking points. So effectively the detected translation is already factored out. In some cases, this is not optimal, especially when Tracks have gaps or do not cover the whole duration of the footage – we plan further options to better control the Pivot Point in future releases.

**Stabilization Tracks**

Thus, as foundation for any image stabilization, we need Tracked image features to derive the movements. These Tracking points or “Tracks” can be established with Blender’s image feature Tracking component the right choice of points to Track is somewhat tricky, yet crucial for successful image stabilization. Often, we’re here because we’ll have to deal with imperfect footage. In such cases, the averaging of Tracks helps to work around image or Tracking
errors at some point. Moreover, when the footage contains perspective induced movements, symmetrically placed Tracking points above and below the horizon can be used to cancel out spurious movement and get stabilization to the focal area in between.

![2Diverging movements caused by perspective.](image)

**Source**: blender.org


### Footage, image and canvas

When talking about the movement stabilization video, we must distinguish several frames of reference. The image elements featured by the footage move around irregularly within the footage’s original image boundaries – this is the very reason why we are using the stabilizer. When our attempt at stabilization was successful, the image elements can be considered stable now, while in exchange the footage’s image boundaries have taken on irregular movement and jump around in the opposite way. This is the immediate consequence of the stabilizer’s activity.

However, when the Camera was moved intentionally, we must consider yet another frame of reference beyond the canvas: namely the frame (or “cadre”) of the final image we want to create. To understand this distinction, let’s consider a hand-held, panning shot to the right: Since our Camera was turned towards the right side, the actual image contents move towards the left side within the original image frame. However, let’s assume the stabilizer was successful with “fixing” any image contents relative to the canvas – which in turn means, that the original image boundaries start to move irregularly towards the right side, and the contents of the image will begin to disappear gradually behind the left boundary of the original image. After some amount of
panning, we’ll have lost all our original contents and just see an empty black image backdrop. The only solution to deal with that problem is to move the final image frame along to the right, thus following the originally intended panning movement. Of course, this time, we do want to perform this newly added panning movement in a smooth and clean way.

Title-img 4. 3 Stabilizing a panning shot.

Source-blender.org

Link:
Title-Img 4. Restoring the expected Camera movement.

Source-blender.org
Link-

To allow for such compensation and to reintroduce deliberate panning, or tilting and zoom of the resulting image, the stabilizer offers a dedicated set of controls:

- Expected position,
- Expected rotation and
- Expected scale

These act like the controls of a virtual Camera filming the contents we have fixed onto the canvas. By animating those parameters, we’re able to perform all kinds of deliberate Camera movements in a smooth fashion.

The “dancing” black borders

As explained above, when we succeed with stabilizing the image contents, the boundaries of the original footage start to jump
around in the opposite direction of the movements compensated. This is inevitable – yet very annoying, since due to the irregular nature of these movements, these “dancing black borders” tend to draw away attention from the actual subject and introduce an annoying restlessness. Thus, our goal must be to hide those dancing borders as good as possible. A simple solution is to add a small amount of zoom. Sometimes we’ll also need to animate the parameter Expected position in order to keep the image centered as good as we can – this helps to reduce the amount of zoom necessary to remove those annoying borders.

The Autoscale function can be used to find the minimal amount of zoom just sufficient to remove those black borders completely. However, if the Camera jumps a lot, the autoscale function often zooms in too much, especially since this calculation aims at finding a single, static zoom factor for the whole duration of the footage. When this happens, you'll typically get overall better results with animating both the zoom factor and the expected position manually.

**Tip**

To activate 2D stabilizer, you need to set the toggle in the panel, and additionally you need to enable Display Stabilization in the Display panel.
Mask Editor

Previously there was no simple workflow for masks in Blender, compositing a rendered scene with real footage was possible, however, when it came to mask out Objects, defining areas of influence and other scenarios, the workflow was cumbersome. Masks are now natively supported, which allow you to draw masks using splines, and then have them rasterized for use in the compositor or sequencer.

This feature consists of **different parts**:

- Mask data block containing multiple Mask Layers and splines.
- Mask editing in the image and Movie Clip Editor space using various tools.
- Animation of masks with keyframes, drivers and Tracking data.
- Compositing Node and sequencer strip to use mask.

**Mask Datablock: Points, Splines and Layers**

**Point**

A **Point** is the most low-level entity used to define mask. It's a simple point with a coordinate, handles and set of feather points. Points can be parented to **markers** from Motion Tracking.

**Spline**

Sets of points define a **Spline**. Currently only **Bezier splines** are supported. They create a **smooth curve** from the first to the last point in the spline. Splines, by default, will create a filled area, however, can also create **non-closed**
curves with a thickness to mask out Objects such as wires or hair. (Refer Img 4.8)

Layer

One or several splines can belong to the same Layer. Splines belonging to the same layer can be animated together, for example by an item from motion Tracker footage. By creating overlapping splines holes can be created, and it's the layer membership that defines which splines interact to create holes.

Mask datablocks are the most high-level entity used for masking purposes. They can be reused in various places, and hold global parameters for all the entities they consist of.

Understanding Layers

The purpose of Mask Layers can be explained with an example. Suppose there are two unwanted people in the footage, and one of them goes from left to right, and the other in the opposite direction. Two Mask Layers can then be used to mask them separately using a Single Mask Data block. At the point of intersection of these shapes they will be added together rather than creating a hole, as would happen if they were on the same layer. If the motion is simple enough, a single motion Tracked point can be used to drive the location of the entire Mask Layer.
Each Mask Layer can consist of **multiple splines** to fit more complex shapes.

**Editing Masks**

Masks can be created in the image and Movie Clip Editors, by changing the mode from **View to Mask** in the header. This will add various tools and properties to the editor panels, while hiding others that are not needed for interacting with masks. The tools and panels available to edit masks are the same in both editors, with the exception that linking masks to Motion Tracking data is only possible in the Movie Clip Editor.

Once set to Mask mode, a **Mask datablock** can be added. Any image, movie clip, render or compositing result can be used as a backdrop to draw masks over. To get interactive feedback on the resulting mask, a **Mask Node** can be connected directly to a Viewer Node in the compositor, which will then keep updating the compositing result while editing.

![Mask editing in image editor.](https://wiki.blender.org/index.php/File:Blender2.64_mask_editor_overview.png)

**Source**-Brecht

**Link**-

[https://wiki.blender.org/index.php/File:Blender2.64_mask_editor_overview.png](https://wiki.blender.org/index.php/File:Blender2.64_mask_editor_overview.png)
Control Points

Editing of mask splines happens in the same way of editing Bezier curves or paths in GIMP or other curve editors: control points are added to define the spline itself, and handles of several types are used to create smooth bends. This makes it possible to define a mask with few points to easily follow an Object in footage.

- **Ctrl + LMB** is used to place new control points and define handle orientations (click to place control point, click followed with slide to place new control point and set smoothness for it).

- **Alt + C**: to close the mask by joining the last control point to the first.

- Existing control points can be translated, scaled and rotated with the usual **G, S, R shortcuts**.

- **X** or **Delete** removes control points.

Selection

The usual **Selection and Hide/Reveal tools** are available:

- **A**: toggle select all

- **B, C**: border and circle Select

- **Ctrl + L** select linked from selection, **L**: select linked with mouse

- **Ctrl + Alt + LMB**: lasso select

- **H** hide selected, **⇧ Shift + H** hide unselected, **Alt + H** reveal

Curve Handles

- **Alt + C**: cycle toggle spline, to create a close curve or open it again

- **V**: set handle type for selected spline points

- **Ctrl + N**: make normals (handle directions) consistent
• Switch Direction handle directions in/out.

Feather

It’s possible to control feather of mask, including a way to define non-linear feather. Linear feather is controlled by a slider, non-linear feather is controlled in the same curve-based way to define feather falloff.

• Shift + LMB Template-LMB.png is used to define a feathering outline curve. To create an initial feather, sliding from a spline control point outside or inside will create and position feather points. After this Shift + LMB Template-LMB.png will insert new feather point and mouse sliding can be used to move them around.

• Alt + S will scale the feather size.

Using Masks

Masks have many purposes. They can be used in a Motion Tracking workflow to mask out, or influence a particular Object in the footage. They can be used for manual Rotoscoping to pull a particular Object out of the footage, or as a rough matte for green screen keying. Masks are independent from a particular image of movie clip, and so they can just as well be used for creating motion graphics or other effects in the compositor.
Title-Img 4. 8Using the Mask Node to isolate an Object in compositing.

Source-Brecht
Link- https://wiki.blender.org/index.php/File:Blender2.64_mask_compositor_node.png

Compositing Node

In the compositing Nodes, the Mask Input Node can be used to select a mask datablock, with as output the raster mask image. This image can be used with other Nodes, for example to Invert, Multiply or Mix, or use as a factor input. The Node options are:

- **Anti-Alias**
  Create smooth mask edges rather than hard ones.

- **Feather**
  Use or ignore feather points defined for splines.

- **Size**
  Scene Size will give an image the size of the render resolution for the scene, scaling along when rendering with different resolutions. Fixed gives a fixed size in pixels.
Fixed/Scene gives a size in pixels that still scales along when changing the render resolution percentage in the scene.

- **Motion Blur**
  For animated masks, creating a motion blurred mask from the surrounding frames, with a given number of samples (higher gives better quality), and a Camera shutter time in seconds.

### Animating Masks

Masks can be driven over the time so that they follow some Object from the footage, e.g. a **running actor**. This animation can be done in several ways:

- **Control points** can be parented to motion Tracks. This way is the main way to interact with masks in a Motion Tracking workflow.

- **Keyframe Animation** of control points using a shape keying system. This can be useful when there are not enough good feature points to Track in the footage, or the mask is not based on footage.

- For **animation**, more complex mask shapes, it is also possible to do more high-level animation:

  - **Splines and Mask Layers** can be animated as whole, instead of individual control points.

  - Masks can be parented to Motion Tracking data. Works for both individual mask point parenting and for overall spline. To select motion Track to be parented to use Ctrl + RMB to parent selected mask points to active motion Track use Ctrl + P.

  - **Mask Animation** timing can be edited from the **Dope Sheet** where there is a mask mode where mask keyframes can be selected and edited.
Shape Keyframe

Masks can be animated with Shape Keyframing. This works on the level of Mask Layers, so inserting a shape key will keyframe all the splines and points contained in it.

- **Step 1:** Insert a shape key for the active Mask Layer at the current frame.
- **Step 2:** Alt + I will clear the shape key for the active Mask Layer at the current frame.

**Feather Reset Animation:** Resets the feather offset across all animated frames useful if you animate first then add feather after.

**Re-Key Points of Selected Shapes:** Re-interpolate selected points on across the range of keys selected in the dope sheet. This has the same effect of removing and re-inserting keys - however it can be applied selectively to the points you need.

Parenting to Motion Tracks

In the Movie Clip Editor, it's possible to parent spline points to motion Tracks.

- Ctrl + P parents one or more selected spline points to the active motion Tracker.
• Alt + P clears any parenting relationship for the selected spline points.

• S-Curves

Title-Img 4. 10S-curves

Source-Brecht

Link-

The curve type used for creating mask splines is almost a Bezier curve, however, with some differences. The curve needed to support feathering in a way that stuck to the curve as you edited it, for ease of editing an animation. We call these S-Curves.

Besides the handles, every control point also has points that define the feather between the current point and the next point on the spline. Each feather point is stored in UV space, where U means position across spline segment, and V means distance between main spline and feather points.

This allows for deforming the main spline in almost any way, and the feather will be updated automatically to reflect that change. For example, if there's just rotation of the spline, feather would stay completely unchanged. If one point's feather is moved, the other feathers will be automatically stretched uniformly along that segment and the overall shape will be almost the same as artists would want it to be.
Unit summary

In this Unit, you have learnt how to

- Work effectively with Motion Tracking
- Create scenes for production
- Work with 2D Stabilization in Blender
- Use Rotoscopy in Blender
- Use of S curve
- Animate the mask
- Work with videos of Tracking using 3D virtual Camera and Camera Tracking

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Objects.

Assignment

- Create a Camera Tracking Shot in Blender watching the video
- Submit the assignment in MP4 format with 1920 X 1080 Resolution
Assessment

1. Explain Motion Tracking in Blender
2. Describe the process of 2D Stabilizing
3. Write a brief note on Mask Editing and Animation
4. Explain the importance of S curve
5. Write a brief note on Track Panel
6. Write a note on Clip view

Fill in the Blanks

1. The __________________ is identified by a clock icon at the bottom of the screen.
2. Pose-to-pose animation workflows can be achieved by________________ Interpolation.
3. _____________defines the behaviour of a curve before the first and after the last keyframe.
4. Press _____________ to test the animation.
5. The_________________ is the green line, it is used to set and display the current time frame.

Resources

While studying this Unit, you can browse the internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

Study Skills
- wiki.blender.org
- archive.org
- www.blender.org
- docs.blender.org