

# Motus Unitatis, an Animation Editor

Bryan Castillo, Timothy Elmer

## Purpose

The Motus Unitatis Animator Editor allows artists and designers to edit and create short animated clips. With MU, a designer has the ability to place objects at different positions with different rotations at different points in time. When the user plays the sequence, MU will create smooth animations between saved key frames. MU supports both a play mode and a simple editing mode. In addition to editing object position and rotation, users have the ability to edit, and delete keyframes.

## Project Requirements

The following section shows how the application fills the requirements of the project for the class.

### Working with graphical objects

There will be 4 main objects that can be worked with in the animation editor. A lamp will be one of the main objects in the scene, which will have 4 levels of scene hierarchy. The head, and the arms of the lamp can be rotated and the entire lamp can be moved. The play view camera, the ball, and the block can also be moved and rotated. All movements and rotations will be accomplished using a custom 3D manipulator.

### Interaction of objects

While the editor will allow the user to set objects in motion, the objects themselves will have built in collision detection, which will trigger default animations and or actions. For example, the head of the lamp will have the ability to deflect the ball or block. The application will also use custom meshes for the letter block and ball.

### Illumination and texture

The lamp will have its own light source and it will support illumination and the display of textures on lamp members.

### Two different views and camera manipulation

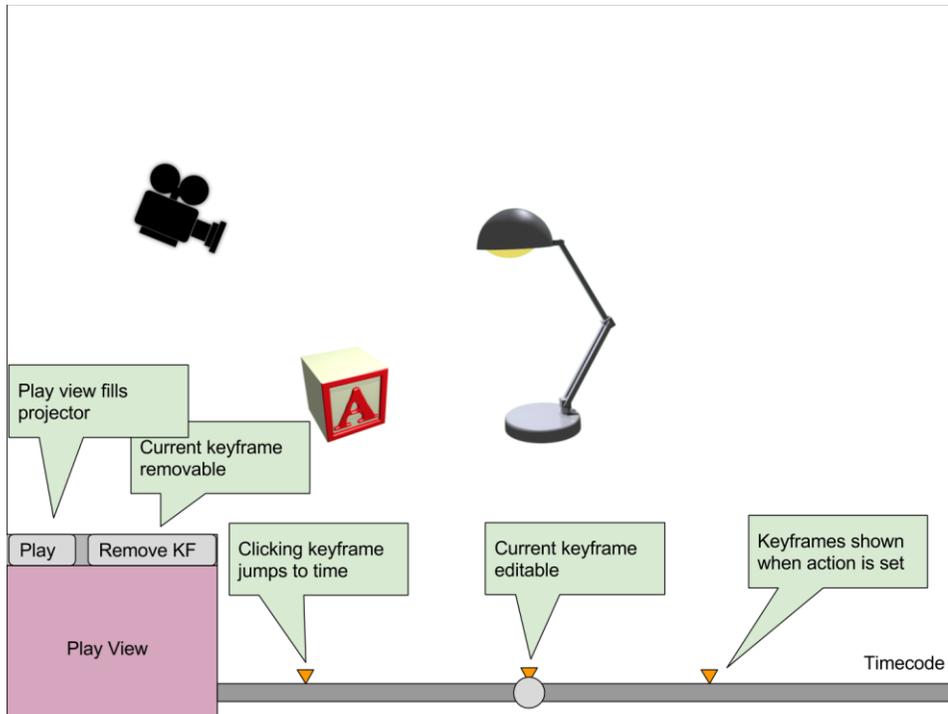
The animation editor will have a camera for the edit mode and one for play mode. In edit mode, the play mode camera will be displayed in the lower left hand corner with a smaller view. The position of the play camera can be manipulated in edit mode. When in play mode, the edit mode camera will be disabled, and the play mode will take up the entire screen.

## UI Design

MU will have 2 major modes for the user interface. There will be a play mode and an edit mode. The edit mode is used to create an animation, while the play mode runs the animation.

## Edit Mode

In edit mode the user sees the main editable objects in the scene hierarchy, the play view camera, a time slider with key frame indicators, a preview for the play view, and controls to perform actions such as going into play mode. In edit mode, the user has the ability to move through the scene using mouse controls to tumble, zoom, and track. When the user moves the slider the objects will rotate and move to their correct positions based on previously saved key frames.



## Play Mode

When play mode is started, the play view camera will fill the screen and the the animation will play. A limited number of controls will be available, mainly to stop the animation.



# Project Plan

We have organized our project plan to include 5 separate 1 week sprints. Each sprint will be focussed on critical milestones for the project and each sprint will run from Tuesday to Tuesday. We have setup our sprints to prioritize dependent tasks first, followed by high risk features, low risk features, and finally full release testing.

Milestone	Sprint	Sprint Week	Status
Project Proposal	1	11/7 to 11/13	Complete
Requirements Specification	1	11/7 to 11/13	Complete
Base Project Structure	1	11/7 to 11/13	In Progress
Base Objects	2	11/14 to 11/20	Not Started
Edit Mode Skeleton	2	11/14 to 11/20	Not Started
Play Mode Skeleton	2	11/14 to 11/20	Not Started
Mode Switching	2	11/14 to 11/20	Not Started
Key Frame Object Model	3	11/21 to 11/27	Not Started
Time Line UI	3	11/21 to 11/27	Not Started
Object Edit Creates Key Frame	3	11/21 to 11/27	Not Started
Play Mode Animation	4	11/28 to 12/4	Not Started
Edit Mode Mouse Manipulation	4	11/28 to 12/4	Not Started
Delete Key Frame	5	12/4 to 12/11	Not Started
Play Camera Display in Edit	5	12/4 to 12/11	Not Started
Release Testing	5	12/4 to 12/11	Not Started

# Appendix

## Feature Requirements

The following section outlines the specific requirements and features of the application.

### Edit Mode

1. *Object Editing*
  - 1.1. Editing a scene object either edits or creates a new key frame.
  - 1.2. *Object Manipulation*
    - 1.2.1. When an object is selected for editing, a 3D manipulator will be displayed.
    - 1.2.2. *Play View Camera*
      - 1.2.2.1. Can be moved.
      - 1.2.2.2. Can be rotated.
      - 1.2.2.3. In edit mode the play camera has a visible object representation.
    - 1.2.3. *Lamp*
      - 1.2.3.1. Can be moved.
      - 1.2.3.2. Can be rotated.
      - 1.2.3.3. Individuals parts of the lamp can be rotated, but not moved.
    - 1.2.4. *Ball and Block*
      - 1.2.4.1. Can be moved.
      - 1.2.4.2. Can be rotated.
2. *Edit Mode Mouse Movement*
  - 2.1. In edit mode mouse movement support will include:
    - 2.1.1. Zooming
    - 2.1.2. Tumbling
    - 2.1.3. Tracking
  - 2.2. Mouse movements will be based on a look at position similar to MP4. (It will not show up in play mode though.)
3. *Time Slider*
  - 3.1. Can be moved to any point in time:
    - 3.1.1. When moved all objects change to their new calculated position and rotation.
    - 3.1.2. Shows current time value in label.
4. *Keyframe*
  - 4.1. All saved keyframes have a visual indicator above the time slider corresponding to the time of the key frame.
  - 4.2. Keyframes can be deleted.
5. *Entering Play Mode*
  - 5.1. A play button may be clicked to enter play mode starting at the given time position
    - 5.1.1. The user may choose to start at time 0
6. *Play Camera Display*
  - 6.1. Displays the view of the play camera at the given time code.

### Play Mode

1. *Display*

- 1.1. There will be only 1 view which will be attached to the play mode camera.
- 1.2. Animation starts automatically when play mode is entered starting at the current time code.
2. *Animation*
  - 2.1. Object will be smoothly animated between keyframes (tweening).
  - 2.2. The time position will be shown on screen.
  - 2.3. At the end of animation the application will go back into edit mode.
3. *Stop*
  - 3.1. There will be a button to stop the animation. The editor will go back into edit mode at the current time code.
    - 3.1.1. The user may choose to return to time 0

## Features / Components

*This table illustrates prerequisites for functional completion of components, and differs from the project plan above. The project plan involves the start of some features to enable parallel development, with the completion occurring later.*

Component	Description	Prerequisites
Model Fabrication	Model and texture meshes	
Hierarchy Placement	Import, arrange, texture, and light hierarchy elements	<ul style="list-style-type: none"> <li>● Model Fabrication</li> </ul>
Mesh Placement	Import, arrange, and texture standalone meshes	<ul style="list-style-type: none"> <li>● Model Fabrication</li> </ul>
Manipulation	Selection of objects, manipulator display, manipulator interaction	<ul style="list-style-type: none"> <li>● Hierarchy Placement</li> <li>● Mesh Placement</li> </ul>
Keyframe	Store transformation of objects	
Keyframe Update	Create/update keyframe on object manipulation	<ul style="list-style-type: none"> <li>● Manipulation</li> <li>● Keyframe</li> </ul>
Tweening	Interpolate transformation between keyframes based on timecode	<ul style="list-style-type: none"> <li>● Keyframe</li> </ul>
Timeline UI	Echo slider, keyframe marker display	<ul style="list-style-type: none"> <li>● Keyframe</li> </ul>
Mode Switching	Switch between play, edit modes	<ul style="list-style-type: none"> <li>● Tweening</li> </ul>
Editor Camera	Mouse manipulation of "edit mode" camera	