



Lego Animation 101

Simple Movement

Ahui Herrera

Jedi_agh@yahoo.com

February 2002

http://jedi_agh.tripod.com

http://am_production.tripod.com

Outline of Presentation

- Animation methods
- Stage 1: Storyboarding
- Stage 2: Modeling
- Stage 3: Rendering
- Stage 4: Production
- Additional Information

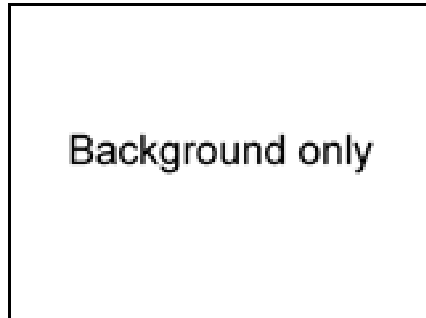
Animations Methods

- Method 1 - Object Movement
 - Camera position is fixed while object moves
 - Think of early cartoons
- Method 2 - Camera Movement
 - Object is fixed while camera moves
 - Think of spaceship movies
- Method 3 – Combination of 1 & 2

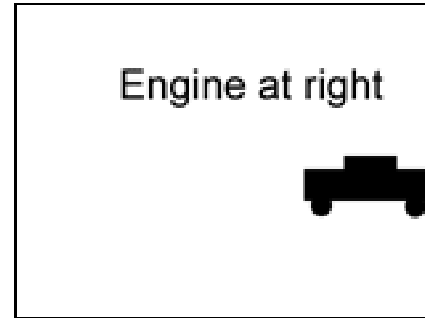
Stage 1: Storyboarding

- Putting ideas on papers
- Simple conceptual drawings will do
 - Draw key moments in the animation
- For our example...
 - To show a train engine moving
 - Three major actions
 - Train coming into the screen (right side)
 - Train stopping in middle of the screen
 - Train leaving the screen (left side)

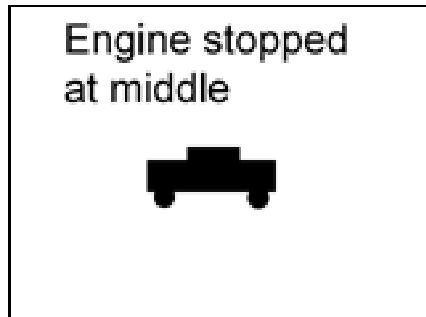
S1: Our Storyboard Images



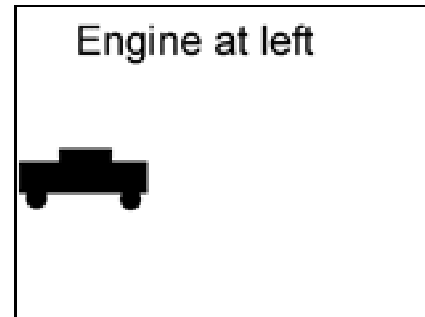
Board 1



Board 2



Board 3



Board 4

Stage 2: Modeling

- Each object that will move **MUST** have it's own LDraw file
- The final LDraw file will have the sub-models in it plus all the stationary objects
- Why is this needed?
 - Due to the way LDraw & L3P do conversions

S2: Our Example

- LDraw File 1
 - All pieces to build train & track pieces
- LDraw File 2
 - Track pieces & sub model of train

S2: Wrong Conversion

POV-Ray - C:\Documents and Settings\Administrator\Desktop\Presentation\SimpleMovement-Wrong.pov

File Edit Search Text Editor Insert Render Options Tools GUI-Extensions Help

New Open Save Queue Rerun Show Ini Sel-Run Run Pause Tray 2:44

[320x240, No AA] ? POV-Win ? Scene IRTC CD POV Site IRTC Site

Messages light-source.pov Moving Train.pov SimpleMovement-Wrong.pov

```

    #if (version >= 3.1) material #else texture #end { Color0 }
  }
  object {
    _4593_dot_dat
    matrix <1-SW/8,0,0,0,1-SW/42,0,0,0,1-SW/8,0,SW/-2.47059,0>
    matrix <1,0,0,0,0.707107,-0.707107,0,0.707107,0.707107,10,-168,-10>
    #if (version >= 3.1) material #else texture #end { Color0 }
  }
  object {
    _298_dot_dat
    matrix <1-SW/16,0,0,0,1-SW/13.58,0,0,0,1-SW/16,0,SW/11.2231,0>
    matrix <1,0,0,0,1,0,0,0,1,-10,-168,-10>
    #if (version >= 3.1) material #else texture #end { Color0 }
  }
  object {
    _4593_dot_dat
    matrix <1-SW/8,0,0,0,1-SW/42,0,0,0,1-SW/8,0,SW/-2.47059,0>
    matrix <1,0,0,0,0.707107,-0.707107,0,0.707107,0.707107,-10,-168,-10>
    #if (version >= 3.1) material #else texture #end { Color0 }
  }
  // STEP
  object {
    _2865_dot_dat
    matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
    matrix <1,0,0,0,1,0,0,0,1,0,89,-105>
    #if (version >= 3.1) material #else texture #end { Color0 }
  }
  //
}

object { SimpleMovement_dash_wrong_dot_ldr #if (version >= 3.1) material #else texture #end { Color7 }

// Floor:
object {
  plane { y, 97 hollow }

```

S2: Correct Conversion

```

}
object {
  _2865_dot_dat
  matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
  matrix <-1,0,0,0,1,0,0,0,-1,0,0,-2880>
  #if (version >= 3.1) material #else texture #end { Color0 }
}
object {
  _2865_dot_dat
  matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
  matrix <-1,0,0,0,1,0,0,0,-1,0,0,-3200>
  #if (version >= 3.1) material #else texture #end { Color0 }
}
object {
  _2865_dot_dat
  matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
  matrix <-1,0,0,0,1,0,0,0,-1,0,0,-3520>
  #if (version >= 3.1) material #else texture #end { Color0 }
}
object {
  _2865_dot_dat
  matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
  matrix <-1,0,0,0,1,0,0,0,-1,0,0,-3840>
  #if (version >= 3.1) material #else texture #end { Color0 }
}
object {
  _2865_dot_dat
  matrix <1-SW/160,0,0,0,1-SW/24,0,0,0,1-SW/340,0,SW/-6,0>
  matrix <-1,0,0,0,1,0,0,0,-1,0,0,-4160>
  #if (version >= 3.1) material #else texture #end { Color0 }
}
// STEP
//
object { _4565__engine_dot_dat matrix <-1,0,0,0,1,0,0,0,-1,0,-88,-160> #if (version >= 3.1) materia
}

```

S2: Converting Ldr to POV

- Use L3P with L3PAO GUI
 - L3PAO Ver. 1.3.4 supports the .ldr file format
 - Use the following options
 - -f: Floor – Grey color is okay
 - -bu: Bumps
 - -sw: Seam Width
 - -q: Quality level set to 2
 - -o: Overwrite existing POV file

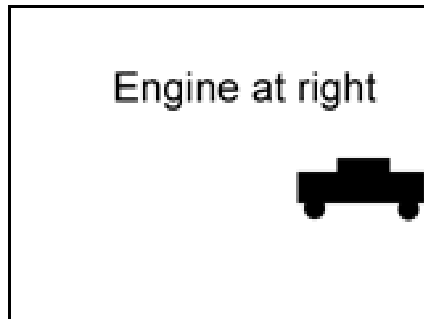
Stage 3: Rendering

- Create a folder for your project
- Place the POV file in the new folder
- Open the file and SURPRISE!
 - Code, Code, and more Code

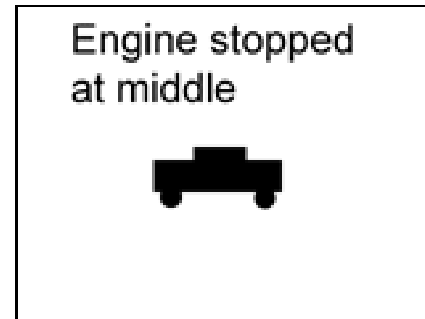
S3: Camera Placement

- Camera values
 - Use MLCAD to help determine location
 - Location $\langle 870, -700, -3080 \rangle$
 - Look_at $\langle 0, 0, -570 \rangle$
 - This is storyboard 1-2
 - Changes to storyboard to accommodate better view

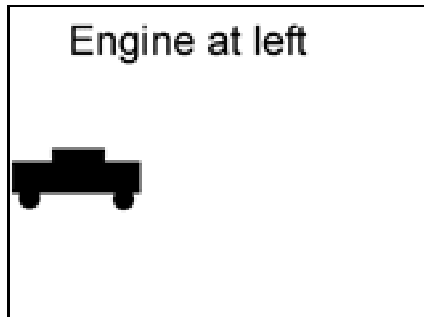
S3: New Storyboard #2



Board 1



Board 2



Board 3

S3: Moving the Engine

- Engine object values
 - Use the 'translate' command
 - Look for key actions scenes called key frames
 - Position 1 = translate <0,0,0>
 - Position 2 = translate <0,0,-1350>
 - Position 3 = translate <0,0,-2700>

S3: Adding Sky & Green Grass

- Open the sky.pov file and cut & paste the code into SM.pov
 - Remove the background code
- Change the background from $\langle .8, .8, .8 \rangle$ to $\langle 0, 1, 0 \rangle$ to get green.
 - Notice that you will LOSE the shadow effect of the engine in the green grass.

S3: Frame-Rates

- You choose but there are standards
 - Movies = 24 fps (23.976 fps)
 - Animation = 25 fps
 - TRON set this standard with a special Kodak film that is still used by animation films
 - NTSC (**N**ational **T**elevision **S**ystems **C**ommittee) 30 fps (29.976 fps)
 - North America, Mexico, Canada, & Japan
 - Due to 60 Hz cycle electrical system
 - PAL (**P**hase **A**lternating **L**ine) 25 fps
 - Europe, Hong Kong, & Middle East
 - Due to 50 Hz electrical system

S3: How many frames

- Train moves 1 upf @ 30 fps
 - Total frames 2700
 - Total time 90 secs
- Train moves in 10 secs @ 30 fps
 - Total frames 300
 - Total time 10 secs

What is the Difference?

LENGTH OF

RENDERING TIME!

S3: Rendering time

- Train moves 1 upf @ 30 fps
- 640 x 480 Video Size
- Total render time
 $2700 * 19 = 51300$
sec (14.25 hours)

- Train moves in 10 secs @ 30 fps
- 640 x 480 Video Size
- Total render time
 $300 * 19 = 5700$ sec
(1.6 hours)

Note: You can reduce the time with a 320x240 video size

S3: Creating the ini file

- Lets POV know the following
 - What file to render
 - Number of frames to render
 - Size of frames
 - Output file type
 - Clock values
 - This is the most important item as it does the animation
 - Change the pov file to use this.
 - Translate `<0,0,0>` to translate `<0,0, clock>`

S3: Decreasing Render Time

- Under Render options
 - Set GUI priority to lowest
 - Set render priority to highest
- Close ALL open applications except POV
- Close background applications
 - Anti-virus, scheduler, etc

Stage 4: Production

- Use the avi file format to house
 - video files (images from POV)
 - Video codec (DivX4.12)
 - Audio files
 - Audio codec
- What is a codec?
 - Compression algorithm
 - MP3 is a famous codec
 - Why DivX4.12
 - Produces high-quality small files
 - DivX4.xx are LEGAL versions & can play all previous versions. DivX3.11a/DivX3.11Alpha is cracked version of Microsoft's original DivX3

S4: The Basics – The MPEGs

■ MPEG-1

- Pro: Oldest codec can be played on any PC
- Con: Not suitable for web

■ MPEG-2

- Pro: Ideal for high quality video
- Con: Not suitable for web
- Con: CPU HOG!

■ MPEG-4

- Pro: Can handle a variety of codecs (DivX is here)
- Pro: Suitable for web based delivery
- Con: CPU HOG!

S4: For Apple users

- Sorenson Video 2.1
 - Pro: High Quality video and small files
 - Pro: Lots of custom control given to user
 - Con: Proprietary to QuickTime (Win or MacOS)
 - Con: CPU HOG!
 - Con: Developer's version NOT Free

S4: For Web Distribution

- RealVideo 8
 - Pro: Ideal for web distribution of 320x240 res. animations
 - Pro: RealAudio (sound codec) gives very crispy sound compression
 - Pro: Great color retention, especially of solid colors
 - Con: No good at 320x240+ resolutions
 - Con: RealMedia conversion “not really” possible to other formats.

S4: Which is best?

- Depends on your need You must decided on
 - Quality Level
 - File Size
 - Distribution
- For Highest Quality DVD-like use DivX
- For Cross Platform compatibility use Sorenson
- For web distribution use RealVideo 8
- This is for cartoon videos **ONLY!**

S4: The Differences



MPEG-1



MPEG-2

S4: The Differences



DivX Fast Motion



DivX Low Motion

S4: The Differences



Sorenson



Windows Media 7

S4: The Differences

DivX 4



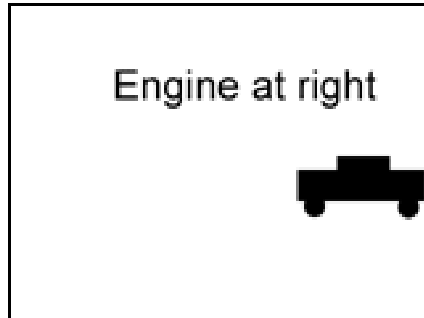
Real Media 8

Serenson

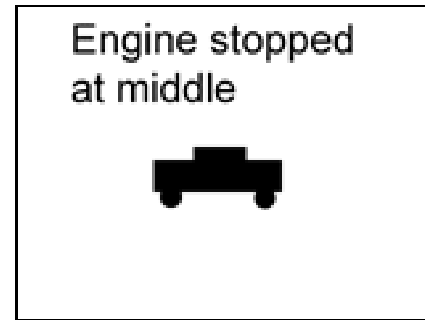
S4: Increase the film length

- Increase by 13 seconds
- Re-use footage
 - At middle stop for 3 seconds
 - Then go back to start (5 seconds)
 - You can reuse frames or render an additional 150 frame
 - Frame reuse is best but time-consuming in MediaStudio
 - Go forward until off screen (10 seconds)
- New total time
 - 5 sec (start to middle) + 3 sec (stopped) + 5 sec (middle to start) + 10 sec (start to end) = 23 seconds!

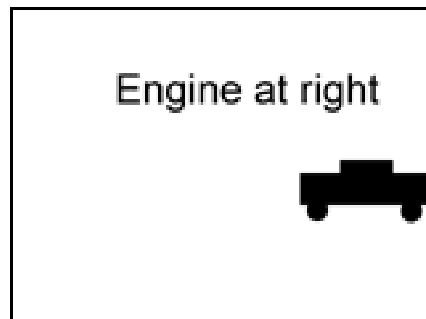
S4: New Storyboard #3



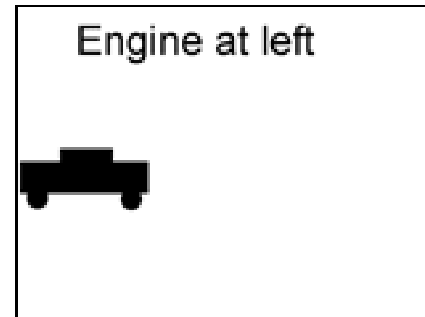
Board 1



Board 2



Board 3



Board 4

S4: Frame Order

- Storyboard 1 going to 2
 - SM001 - SM150
- Storyboard 2
 - Use SM150 for 3 seconds
- Storyboard 2 going to 3
 - SM150 – SM001
 - *SM001 – SM150 (From 2nd Render)*
- Storyboard 3 going to 4
 - SM001 – SM300

S4: Adding Credits

- Opening & Closing credits are optional but a good idea
 - They provide copyright protection for you
- Adding credits increasing the file size very very quickly!
 - Using DivX4.xx Codec experience only
 - Our final animation with credits = 4.18MB
 - Our final animation without credits = 1.97 MB

S4: The Timeline

- Video goes in
 - Va, Vb, V1, V2, V3
 - Fx is for special effects/ transitions between video sections
- Audio for
 - VA is Aa, Vb is Ab, etc...

Additional Information

- **Additional Tutorials**
 - <http://www.ldraw.org> (click on tutorial section)
- **Video Standards**
 - <http://168.144.91.167/nickyguides/interlace.htm>
- **Video Compressions for Cartoons**
 - <http://www.geocities.com/lukevideo>
- **Digital Film Resources (Sound Files)**
 - <http://brickfilms.topcities.com/resources.html>
- **LUGNET (CAD and Publishing groups)**
 - <http://www.lugnet.com>

Final Thoughts

- Use low quality levels 0 or 1 to save time when figuring things out.
- Change one line of code at a time!
- Post questions at LUGNET
- Experiment, Experiment, Experiment...
 - Worse you can do is render one 'bad' frame at a time! This is better than rendering 3 'bad' hours.

Questions?



LEGO Animation Sea