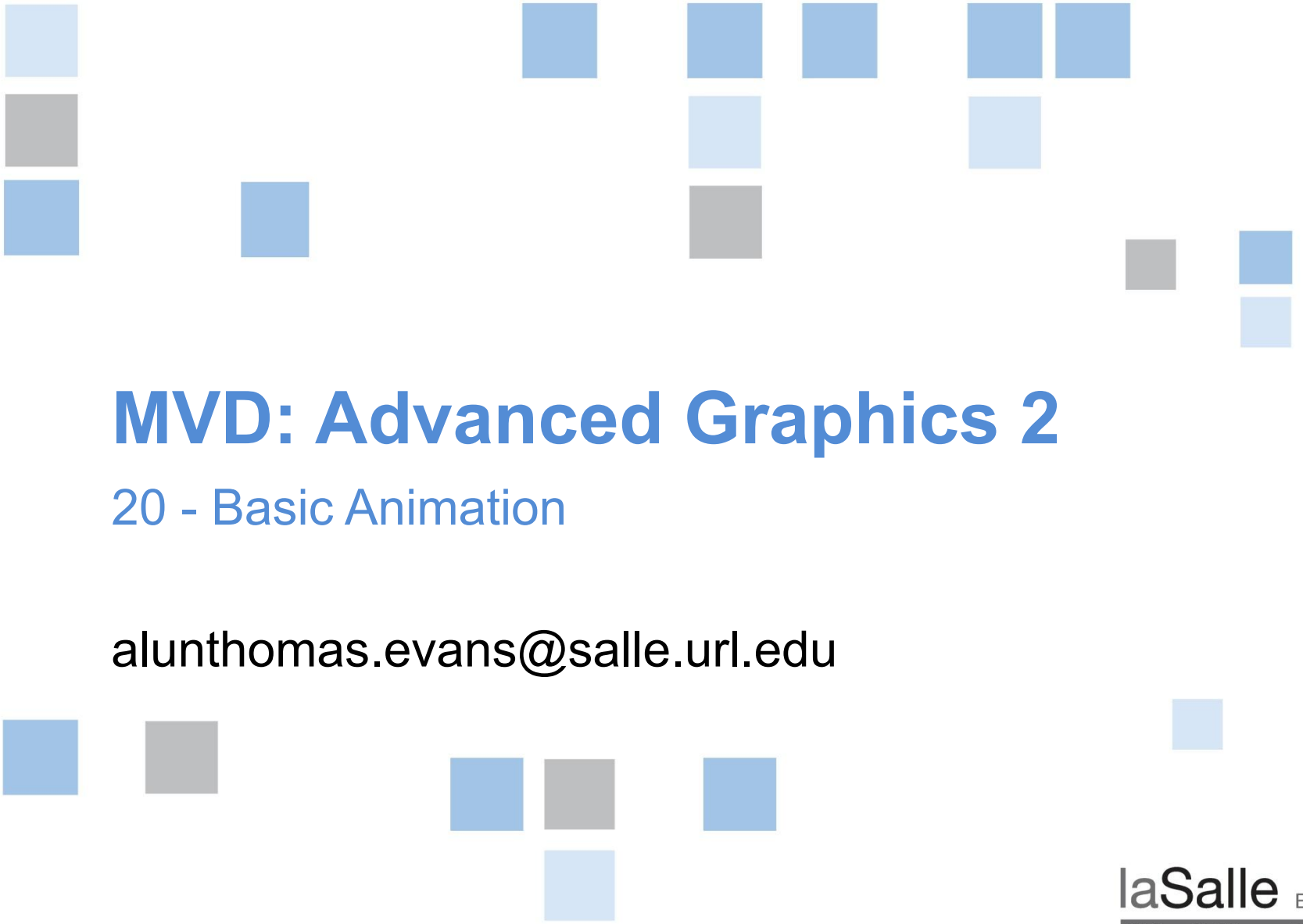

MVD: Advanced Graphics 2

20 - Basic Animation

alunthomas.evans@salle.url.edu

laSalle ENG
Universitat Ramon Llull

Bounce animation

What is an animation?

What parameters are changing?

What is an animation?

Change of entity attributes through *time*

Usually model matrix

Camera: target point or forward direction

Simple animation file

```
ball
24
0 0.0 6.75228563127 0.0 0.0 0.0 0.0 1.0 1.0 1.0
1 0.0 6.75228563127 0.0 0.0 0.0 0.0 1.0 1.0 1.0
2 0.0 6.7489703968 0.0 0.0 0.0 0.0 1.0 1.0 1.0
3 0.0 6.7397482757 0.0 0.0 0.0 0.0 1.0 1.0 1.0
4 0.0 6.72421696163 0.0 0.0 0.0 0.0 1.0 1.0 1.0
5 0.0 6.70192864454 0.0 0.0 0.0 0.0 1.0 1.0 1.0
6 0.0 6.67238273048 0.0 0.0 0.0 0.0 1.0 1.0 1.0
7 0.0 6.63501717702 0.0 0.0 0.0 0.0 1.0 1.0 1.0
8 0.0 6.58919780082 0.0 0.0 0.0 0.0 1.0 1.0 1.0
9 0.0 6.53420552274 0.0 0.0 0.0 0.0 1.0 1.0 1.0
10 0.0 6.46922026327 0.0 0.0 0.0 0.0 1.0 1.0 1.0
11 0.0 6.39330132687 0.0 0.0 0.0 0.0 1.0 1.0 1.0
12 0.0 6.30536307179 0.0 0.0 0.0 0.0 1.0 1.0 1.0
13 0.0 6.20414376024 0.0 0.0 0.0 0.0 1.0 1.0 1.0
14 0.0 6.08816741387 0.0 0.0 0.0 0.0 1.0 1.0 1.0
15 0.0 5.95569393932 0.0 0.0 0.0 0.0 1.0 1.0 1.0
16 0.0 5.80465840967 0.0 0.0 0.0 0.0 1.0 1.0 1.0
17 0.0 5.63259343446 0.0 0.0 0.0 0.0 1.0 1.0 1.0
18 0.0 5.43653401135 0.0 0.0 0.0 0.0 1.0 1.0 1.0
19 0.0 5.2129104958 0.0 0.0 0.0 0.0 1.0 1.0 1.0
20 0.0 4.95744115531 0.0 0.0 0.0 0.0 1.0 1.0 1.0
21 0.0 4.66507639999 0.0 0.0 0.0 0.0 1.0 1.0 1.0
22 0.0 4.33013850979 0.0 0.0 0.0 0.0 1.0 1.0 1.0
23 0.0 3.94700146449 0.0 0.0 0.0 0.0 1.0 1.0 1.0
24 0.0 3.51209746749 0.0 0.0 0.0 0.0 1.0 1.0 1.0
25 0.0 3.02858420309 0.0 0.0 0.0 0.0 1.0 1.0 1.0
26 0.0 2.51417596715 0.0 0.0 0.0 0.0 1.0 1.0 1.0
27 0.0 2.00666631674 0.0 0.0 0.0 0.0 1.0 1.0 1.0
```

Task:

- adapt our engine to be able to change attributes through time
- Read file to apply animation

Simple Animation component

What do we need?

Simple Animation component

What do we need?

```
struct Animation : public Component {
    std::string name = "";
    GLint target_transform = -1;
    GLuint num_frames = 0;
    GLuint curr_frame = 0;
    float ms_frame = 0;
    float ms_counter = 0;
    std::vector<lm::mat4> keyframes;
};
```

Task

Create two entities, each with an Animation Component, with a different **names** and different **ms_frame** values

Create an *Animation System* that updates each animation component and advances the frame at the correct time

Should also **loop** back to start of animation

Output to console when change frame:

name_framenumber

Exported animation files

```
ball
24
0 0.0 6.75228563127 0.0 0.0 0.0 0.0 1.0 1.0 1.0
1 0.0 6.75228563127 0.0 0.0 0.0 0.0 1.0 1.0 1.0
2 0.0 6.7489703968 0.0 0.0 0.0 0.0 1.0 1.0 1.0
3 0.0 6.7397482757 0.0 0.0 0.0 0.0 1.0 1.0 1.0
4 0.0 6.72421696163 0.0 0.0 0.0 0.0 1.0 1.0 1.0
5 0.0 6.70192864454 0.0 0.0 0.0 0.0 1.0 1.0 1.0
6 0.0 6.67238273048 0.0 0.0 0.0 0.0 1.0 1.0 1.0
7 0.0 6.63501717702 0.0 0.0 0.0 0.0 1.0 1.0 1.0
8 0.0 6.58919780082 0.0 0.0 0.0 0.0 1.0 1.0 1.0
9 0.0 6.53420552274 0.0 0.0 0.0 0.0 1.0 1.0 1.0
10 0.0 6.46922026327 0.0 0.0 0.0 0.0 1.0 1.0 1.0
11 0.0 6.39330132687 0.0 0.0 0.0 0.0 1.0 1.0 1.0
12 0.0 6.30536307179 0.0 0.0 0.0 0.0 1.0 1.0 1.0
13 0.0 6.20414376024 0.0 0.0 0.0 0.0 1.0 1.0 1.0
14 0.0 6.08816741387 0.0 0.0 0.0 0.0 1.0 1.0 1.0
15 0.0 5.95569393932 0.0 0.0 0.0 0.0 1.0 1.0 1.0
16 0.0 5.80465840967 0.0 0.0 0.0 0.0 1.0 1.0 1.0
17 0.0 5.63259343446 0.0 0.0 0.0 0.0 1.0 1.0 1.0
18 0.0 5.43653401135 0.0 0.0 0.0 0.0 1.0 1.0 1.0
19 0.0 5.2129104958 0.0 0.0 0.0 0.0 1.0 1.0 1.0
```

Reading the file

For each line.

read the translation and make a matrix

read the rotations into a quaternion, make a matrix

read the scale into a matrix

multiply the lot

save the keyframe

Reading text files (reminder!)

Open file with path ‘filename’, read line-by-line:

```
std::ifstream file(filename);
std::string line;

if (file.is_open())
{
    while (std::getline(file, line))
    {
        //do stuff with line...
    }
}
```

std::stoi() and std::stof() - convert string to int or float

Task: Parse animation file

Create an entity with a mesh component, using “ball.obj”

Optionally add a floor (“plane_20x20.obj”)

Call Parsers::parseAnimation to read bounce.anim file, and apply the animation to a ball entity in scene

Advanced task

Create new function to parse camera_aim.anim

Camera moves along a custom track

Read comments

Input animation into engine