Multimedia-Programmierung
Übung 8

Ludwig-Maximilians-Universität München

Sommersemester 2016

Today

- Sound
- Illustrated with pygame + COCOS2DX

- Particles
- Illustrated with + COCOS2DX

Sound in Pygame

Mixer

- Sounds are controlled using the pygame.mixer interface
- Mixer must be initialized pygame.mixer.init(frequency,size,channels,buffer)
- Automatically initialized with pygame.init() using the default values
- Default values can be changed using pygame.mixer.pre_init()
- The mixer mixes the sounds in background threads
  - Sounds are not blocking the rest of the application logic

Sound in Pygame

Sound Object

- pygame.mixer.Sound provides a class to load and control sound files (OGG and uncompressed WAV)
- Sound.play(loops=0, maxtime=0, fade_ms=0) plays the sound file
- Other methods: stop(), fadeout(time), set_volume(value) etc.

playing a sound file

```
click_sound = pygame.mixer.Sound("click.wav")
click_sound.play()
```

playing a sound file in a loop 4(!) times

```
click_sound = pygame.mixer.Sound("click.wav")
click_sound.play(3)
```

Sound in Pygame

Channels

- A channel represents one of the channels that are mixed by the soundcard
- Sound.play() returns a Channel object (or None if all channels are blocked)
- Provides methods to manipulate the sound and create useful effects (e.g. Channel.set_volume(left, right))

playing a sound file from the right speaker only

```
channel = click_sound.play()
channel.set_volume(0.0,1.0)
```

Sound in Pygame

Stereo Panning

- Create the illusion that sound is coming from a specific point at the screen
- Manipulate the volume of the different speakers
- Can be used to make a sound “move” over the screen

stereo panning function

```
def stereo_pan(x_coord, screen_width):
        right_volume = float(x_coord) / screen_width
        left_volume = 1.0 - right_volume
        return (left_volume, right_volume)
```

From: W. McGugan, Beginning Game Development with Python and Pygame, Apress 2007

Music in Pygame

- Don`t use pygame.mixer but pygame.mixer.music
- It enables **streaming** music which means that the file will be read in small chunks
- Supports MP3 and OGG files (OGG better supported across platforms)
- Other Methods include stop(), pause(), rewind() etc.
- Attention: only one song can be streamed at the same time

playing a song using pygame

```
pygame.mixer.music.load("music.ogg")
pygame.mixer.music.play()
```

Sound in Cocos2d-x

File Formats and Libraries

- The file format to use depends on the platform:
  - OGG is preferred for Android but does not work for iOS
  - Instead, use MP3 for iOS

- The library to use depends on the complexity:
  - The standard SimpleAudioEngine is easy to use but also very limited.
  - If more functionality is need, use AudioEngine or an external library.

Background Music

Initialize, load and play music

- To play any song, first initialize the audio engine

initialize audio engine

```
auto audio = SimpleAudioEngine::getInstance();
```

- Only one (background) music file can be played at a time
- Preloading is not required. It reads the file before it is played and ensures fluent music playing

load and play a sound file

```
audio->preloadBackgroundMusic("song.mp3");
audio->playBackgroundMusic("song.mp3");
```

Background Music

Pause, resume and stop music

- Pause and resume song playback

```
if (audio->isBackgroundMusicPlaying())
    audio->pauseBackgroundMusic();
else
    audio->resumeBackgroundMusic();
```

- Stop music playback

```
audio->stopBackgroundMusic();
```

Soundeffects

Load&play and stop&unload

- Multiple sound effects can be played simultaneously [redacted]
- Stop music and set memory free, if sound is not used anymore

load & play one file, copy & paste for multiple files

```
audio->preloadEffect("song1.mp3");
audio->playEffect("song1.mp3");
```

stop and unload file

```
audio->stopEffect ("song1.mp3");
audio->unloadEffect("song1.mp3");
```

Soundeffects

Pause and resume

- Pause and Resume one sound effect

```
audio->pauseEffect();
audio->resumeEffect();
```

- Pause and Resume all sound effects

```
audio->resumeAllEffects();
audio->pauseAllEffects();
```

Set the volume

- Tuning the volume with setBackgroundMusicVolume() does not work (in windows). This bug is known for a long time but was not fixed so far. So work around, e.g. by adjusting the volume of the sound file.

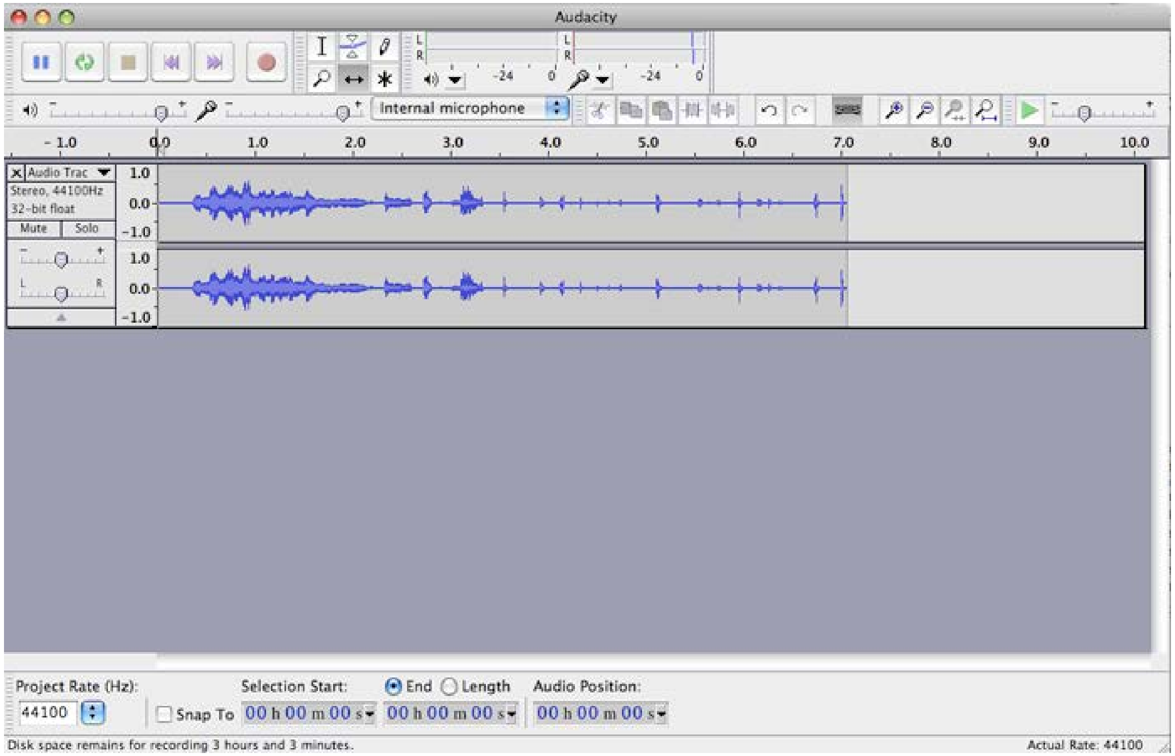
- Tuning the volume of effects works by setEffectsVolume(<float>):

set volume

```
audio->setEffectsVolume(5.0f);
```

Creating your own Sound

- Record real sounds and edit them
- Free sound editor Audacity (http://audacity.sourceforge.net/?lang=de)



Project Rate (Hz): 44100

Selection Start: □ Snap To 00 h 00 m 00 s

● End ○ Length 00 h 00 m 00 s

Audio Position: 00 h 00 m 00 s

Disk space remains for recording 3 hours and 3 minutes.

Actual Rate: 44100

Particles in Cocos2d-x

Particle Systems

- There is two basic ParticleSystems:
  - CCParticleSystemQuad
  - CCParticleSystemPoint
- CCParticleSystemQuad extends CCParticleSystemPoint
  - it supports spinning and particles of any size
- There is a set of predefined particles: e.g. ParticleFire, ParticleExplosion, ParticleSnow, ...

create a CCParticleFire based on CCParticleSystemQuad

```
CCParticleSystemQuad* m_emitter = new CCParticleSystemQuad();
m_emitter = CCParticleFire::create();
```

Particles in Cocos2d-x

Gravity Mode

- Gravity Mode lets particles fly toward or away from a center point.

Create a ParticleModeGravity

```
this->m_nEmitterMode = kCCParticleModeGravity;
this->modeA.gravity = ccp(0,-90);
```

- Additional Properties are e.g. speed, tangencialAccel, radialAccel
- Use tangencialAccelVar and RadialAccelVar for a more natural behavior

Particles in Cocos2d-x

Radius Mode

- Radius Mode causes particles to rotate in a circle.
- It also allows you to create spiral effects with particles either rushing inward or orating outward.

Create a ParticleModeRadius

```
this->m_nEmitterMode = kCCParticleModeRadius;
this->modeB.startRadius = 0;
this->modeB.startRadiusVar = 0;
```

- Additional Properties are e.g. startRadius, endRadius, rotatePerSecond
- Use startRadiusVar, endRadiusVar and rotatePerSecondVar for a more natural behavior

Click and Touch Events

- First step: Define what type of input you want to use. This influences which handler should be use and how the events are handled.
- Three input methods to differentiate:
  - (single) touch
  - multi-touch
  - mouse
- We will only look into single touch in the slides. You find more information about other events here:
  http://www.gamefromscratch.com/post/2014/10/03/Cocos2d-x-Tutorial-Series-Handling-Touch-and-Mouse-Input.aspx
  http://www.cocos2d-x.org/docs/programmers-guide/8/index.html

(Single) Touch Events

Header File

- Create all necessary functions in the header file

Within the header file (public section)

```
virtual bool onTouchBegan(cocos2d::Touch*, cocos2d::Event*);
virtual void onTouchEnded(cocos2d::Touch*, cocos2d::Event*);
virtual void onTouchMoved(cocos2d::Touch*, cocos2d::Event*);
virtual void onTouchCancelled(cocos2d::Touch*, cocos2d::Event*);
CREATE_FUNC(TouchScene);
```

(Single) Touch Events

CPP File

- Create an Event Listener in the cpp file and map each Touch Event to a function handler

Within the cpp file (init function)

```
auto touchLis= EventListenerTouchOneByOne::create();
touchLis>onTouchBegan = CC_CALLBACK_2(TouchScene::onTouchBegan, this);
touchLis>onTouchEnded = CC_CALLBACK_2(TouchScene::onTouchEnded, this);
...
_eventDispatcher->addEventListenerWithSceneGraphPriority(touchLis, this);
```

Create new methods

```
bool TouchScene::onTouchBegan(Touch* touch, Event* event) {
return true;  // return only for TouchBegan
}
```

(Single) Touch Events

Read out

- Read the touch coordinates

OpenGL x-coordinate

```
touch->getLocation().x;
```

UI x-coordinate

```
touch->getLocationInView().x;
```

- Read the movement of one touch

touch motion

```
touch->getDelta().x;
```

- Read the touched object

touch object

```
event->getCurrentTarget();
```

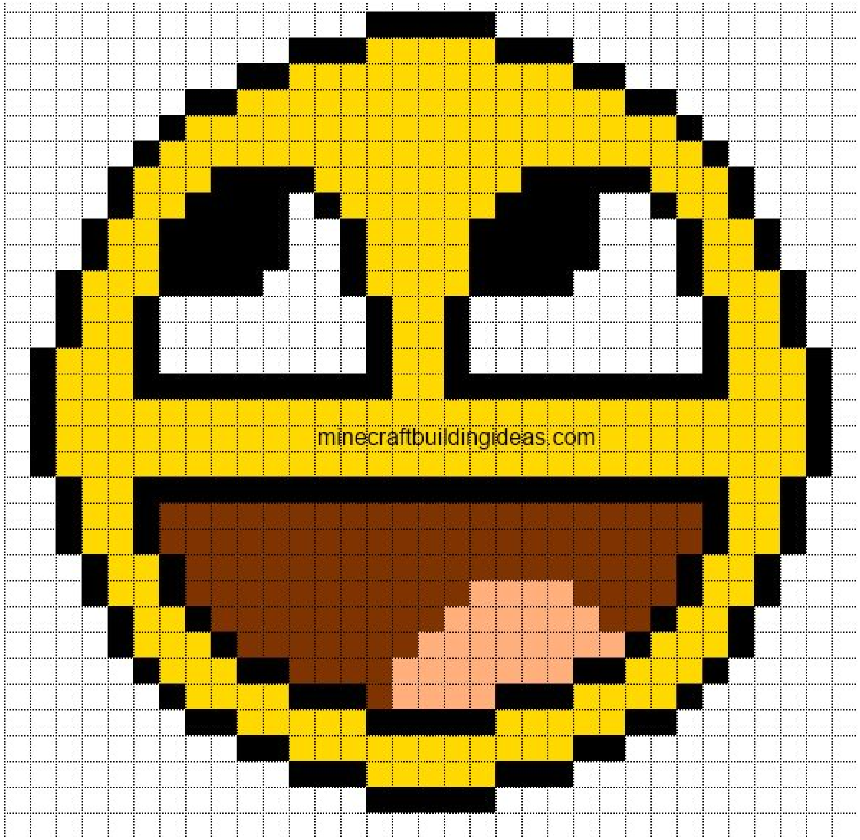
Pixel Art – How to Make a Sprite

Online-Editor:

http://www.piskelapp.com/

Offline-Editor

https://www.gimp.org



http://www.fotolip.com/wp-content/uploads/2016/05/Minecraft-Pixel-Art-Templates-5.jpg

More Useful Links

- http://www.cocos2d-x.org/docs/programmers-guide/12/index.html
- http://www.cocos2d-x.org/wiki/Particles
- http://www.gamefromscratch.com/post/2014/10/03/Cocos2d-x-Tutorial-Series-Handling-Touch-and-Mouse-Input.aspx
- http://www.cocos2d-x.org/docs/programmers-guide/8/index.html