

ffmpeg制作游戏视频

Subtitle

2022/09/29



Table of Contents

ffmpeg制作游戏视频	1
截取视频 :	1
视频信息	1
转场图片	1
合并视频	2
压缩	2
脚本	3

ffmpeg制作游戏视频

使用 Nvidia GeForce Experience 录制游戏，快捷键为 ALT+F9.

截取视频：

snippet.bash

```
ffmpeg.exe -ss 00:00:11 -t 00:02:10 -accurate_seek -i orig2.mp4 -codec copy -avoid_negative_ts 1 1.mp4
ffmpeg.exe -ss 00:13:52 -t 00:00:15 -accurate_seek -i orig.mp4 -codec copy -avoid_negative_ts 1 2.mp4
ffmpeg.exe -ss 00:24:54 -t 00:00:32 -accurate_seek -i orig.mp4 -codec copy -avoid_negative_ts 1 3.mp4
ffmpeg.exe -ss 00:27:54 -t 00:02:28 -accurate_seek -i orig.mp4 -codec copy -avoid_negative_ts 1 4.mp4
```

视频信息

snippet.bash

```
ffprobe -print_format json -show_streams video.mp4 2>/dev/null | jq .
```

转场图片

创建黑色背景，分辨率为 3840x2160 的图片，右键图片，选择创建新视频，使用 win10 自带工具创建开头视频。

创建完成之后转换为 4k 分辨率，并和 GeForce Experience 的参数保持一致，否则会出现视频音频不同步的问题。

```
$ ffmpeg -i 1.mp4 2>&1|grep tbn
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(tv,
smpte170m/smpte170m/bt470m), 3840x2160 [SAR 1:1 DAR 16:9], 22827 kb/s, 6 fps, 60 tbr, 90k
tbn, 120 tbc (default)

# 使用 90k tbn
$ ffmpeg.exe -i start.mp4 -s 3840x2160 -r 60 -video_track_timescale 90k start-4k.mp4
```

考：<https://stackoverflow.com/questions/15931437/ffmpeg-how-to-control-fps-tbr-tbn-tbc-parameters>

合并视频

创建视频列表文件 list.txt，内容为

```
file start-4k.mp4
file 1.mp4
file 2.mp4
file 3.mp4
file 4.mp4
```

然后执行命令

snippet.bash

```
$ ffmpeg.exe -f concat -i list.txt -c copy new.mp4
```

压缩

snippet.bash

```
$ ffmpeg.exe -i new.mp4 -r 30 -vf scale=-2:1080 new-1080p.mp4
$ ffmpeg.exe -i new-1080p.mp4
ffmpeg version 4.2.2 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 9.2.0 (Rev2, Built by MSYS2 project)
  configuration: --prefix=/mingw64 --target-os=mingw32 --arch=x86_64 --disable-debug --
  disable-static --enable-dxva2 --enable-d3d11va --enable-fontconfig --enable-gnutls --enable-gpl
  --enable-libaom --enable-libass --enable-libbluray --enable-libcaca --enable-libcelt --enable-
  libfreetype --enable-libgsm --enable-libmfx --enable-libmodplug --enable-libmp3lame --enable-
  libopencore_amrnb --enable-libopencore_amrwb --enable-libopenjpeg --enable-libopus --
  enable-librtmp --enable-libspeex --enable-libsrt --enable-libtheora --enable-libvorbis --enable-
  libx264 --enable-libx265 --enable-libxvid --enable-libvpx --enable-libwebp --enable-openal --
  enable-libwavpack --enable-pic --enable-postproc --enable-runtime-cpudetect --enable-shared --
  enable-static --enable-swresample --enable-version3 --enable-zlib --disable-doc
  libavutil 56. 31.100 / 56. 31.100
  libavcodec 58. 54.100 / 58. 54.100
  libavformat 58. 29.100 / 58. 29.100
  libavdevice 58.  8.100 / 58.  8.100
  libavfilter  7. 57.100 /  7. 57.100
  libswscale  5.  5.100 /  5.  5.100
  libswresample 3.  5.100 /  3.  5.100
  libpostproc 55.  5.100 / 55.  5.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'new-1080p.mp4':
```

```

Metadata:
  major_brand   : isom
  minor_version : 512
  compatible_brands: isomiso2avc1mp41
  encoder       : Lavf58.29.100
Duration: 00:02:52.90, start: 0.000000, bitrate: 2865 kb/s
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1920x1080 [SAR 1:1
  DAR 16:9], 2725 kb/s, 30 fps, 30 tbr, 15360 tbn, 60 tbc (default)
  Metadata:
    handler_name   : VideoHandler
  Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 128 kb/s
  (default)
  Metadata:
    handler_name   : SoundHandler
At least one output file must be specified

$ du -sh new*
688M  new.mp4
60M   new-1080p.mp4

```

脚本

snippet.bash

```

#!/bin/bash

function _help() {
  echo "command"
  echo -e "\t split | merge | transition | compress"
  echo "param"
  echo -e "\t-f split or merge data file"
  echo -e "\t-i input video file"
  echo -e "\t-h help"
  exit 1
}

# 获取前缀
function _prefix() {
  [ "$DATA"x == "x ] && _help
  PREFIX=`echo $DATA |cut -f1 -d'.'`
}

function _checkInput() {
  [ "$INPUT"x == "x ] && _help
  FORMAT=`echo $INPUT |awk -F'.' '{print $NF}'`
  NAME=`echo $INPUT |awk -F'.' '{print $1}'`
}

```

```
# 将使用win10 创建的转场动画转为4k 分辨率, 和GeForce Experience 保持一致
function _transition() {
    _checkInput
    ffmpeg.exe -y -i $INPUT -s 3840x2160 -r 60 -video_track_timescale 90k $NAME-4k.$FORMAT
}

# DATA 格式
# 开始时间 持续时长, 如:
# 00:00:00 00:00:10
function _split() {
    _checkInput
    _prefix
    [ !-d $PREFIX ] && mkdir $PREFIX
    echo -n "" > $PREFIX/list.txt

    # 保证数据文件结尾换行
    cp $DATA $PREFIX/tmp.txt
    echo "" >> $PREFIX/tmp.txt

    ID=1
    while read line;do
        START=`echo $line |awk '{print $1}'`
        LENGTH=`echo $line |awk '{print $2}'`
        [ "$START"x == ""x ] && continue
        ffmpeg.exe -y -ss $START -t $LENGTH -accurate_seek -i $INPUT -codec copy -
        avoid_negative_ts 1 $PREFIX/$ID.$FORMAT
        echo "file $ID.$FORMAT" >> $PREFIX/list.txt
        ((ID=ID+1))
    done < $PREFIX/tmp.txt
    rm -f $PREFIX/tmp.txt
}

function _merge() {
    _prefix
    FORMAT=`head -n1 $DATA |awk -F'.' '{print $NF}'`
    ffmpeg.exe -y -f concat -i $DATA -c copy $PREFIX-merge.$FORMAT
}

function _compress() {
    _checkInput
    ffmpeg.exe -y -i $INPUT -r 30 -vf scale=-2:1080 $NAME-1080p.$FORMAT
    ffmpeg.exe -y -i $INPUT -r 30 -vf scale=-2:720 $NAME-720p.$FORMAT
}

COMMAND=$1

for i in "$@"
do
    key=$1
    case $key in
        -f) DATA=$2; shift 2;;
        -i) INPUT=$2; shift 2;;
        -h) _help;;
    esac
done
```



```
*) shift;;
esac
done

# 功能
case $COMMAND in
"split") _split;;
"transition") _transition;;
"merge") _merge;;
"compress") _compress;;
*) _help;;
esac
```

Printed on: 2022/09/29 01:17

Convert to img Failed!