



COMP2720: Automating Tools for New Media

Making movies in JES



Creating Movies in JES

- Movie as a sequence of still frames
- Frame rate
- Animation and video
- Movie formats
- Creating a simple animation

Movies



- Psychophysics: persistence of vision
 - humans (and others) can only see change with finite time resolution
 - if the change occurs fast enough ($1/16$ of a second or faster) the perception of change is continuous
 - if the successive images are not related this **montage**
 - if the successive images are related this change gives of motion
- Standard rate in today's motion picture is 24 fps
 - in older (“silver screen”) movies the rate was 16 fps
 - nowadays they're played at 24fps and perceived as “jerky”
(this is similar to the resolution loss when scaling up a picture)
- Digital video has 30 fps
- Some (“Tom Cruise” and other “top gun US fighter pilots” can recognize the change at $1/200$ of a sec
- Computer game players claim seeing the difference between 30 fps and 60 fps videos

Movie formats

- Compression is absolutely necessary:
 - 1sec of (RGB) colour 640x480 pixel frames at 30fps takes 27Mb
 - Which makes into 150Gb for 90 min film
 - Modern DVD have 6.47 Gb capacity
 - BlueRay has 25 Gb capacity (single layer), theoretical limit is 200Gb
- MPEG (current version MPEG-4) by *Moving Picture Expert Group*
- QuickTime by *Apple Inc*
- AVI (Audio Video Interleave File) by *Microsoft*
- These are all compressed formats – they store *key frames* record the *differences* between the successive frames (motion compensation), **or** they compress the individual frames (*interframe compression*)

Movie making in JES recipe

- First create an object `movie` by calling `makeMovie()`
- Start a loop (`for` or `while` loop) and
- Create a picture object by loading it from a file (like `.jpg` image file) or create a new one
- Manipulate and/or create additional imagery in the picture object, and save in a **properly named/numbered** file
- Continue with iteration until done
- As many loop iteration your program has performed as many image files are written and stored **in the same directory** (`frame001.jpg`, `frame002.jpg`, ...)
- Open the `movie` object in the MediaTool
- Add all images from a selected directory as frames to `movie`
- Play the movie `movie` and (on Macs, Windows) save it

Simple movie making program

- Bouncing ball example

```
import os
from string import zfill
rBall = 8
def oneBallMovie(directory, numFrames):
    movie = makeMovie()
    h = 650
    w = 650
    canvas = makeEmptyPicture(w,h)
    x = w/2 #start at the centre
    y = h/2
    vx = input("Enter horizontal velocity: ")
    vy = input("Enter vertical velocity: ")
    frame = 1 # start counting frames
```

Simple movie making program (cont)

- Bouncing ball example (cont)

```
while frame <= numFrames:
    setAllPixelsToAColor(canvas,white)
    if (x <= 1 or x >= w - 2*rBall): #horizontal rebound
        vx = -vx
    if (y <= 1 or y >= h - 2*rBall ): #vertical rebound
        vy = -vy
    # advance ball position
    x += vx
    y += vy
    # correction when going over the wall
    if x < 1:
        x = 1
```

Simple movie making program (concl)

- Bouncing ball example (concl)

```
elif x > w - 2*rBall:
    x = w - 2*rBall
if y < 1:
    y = 1
elif y > h - 2*rBall:
    y = h - 2*rBall
# drawing the ball
addOvalFilled(canvas,x,y,2*rBall,2*rBall,red) #in old JES colour goes second
# writing the frame to a file
frnum = str(frame).zfill(3) #padding zeros on the left
fname = directory + os.sep + frnum + ".jpg"
writePictureTo(canvas,fname)
frame +=1
return movie
```



What to do now, and next week

- Go to VPython Web site (www.vpython.org)
 - Read through *Overview* and *Tutorial Introduction*.
- Work on labs (the lab 6 in the week 11 is on VPython).
- Work on assignment 2 (due Saturday **18 October 6 pm**)
- Next week: Start the last module **Computer Science** *topics*