Early topics in computer game development

While there are exceptions, most computer games require

things to be visually represented, some still and some moving, in a 2D or 3D world
SPRITES
and billboards and meshes and textures and colours and occlusion and perspective and haze and lighting and animation and screen refresh rate

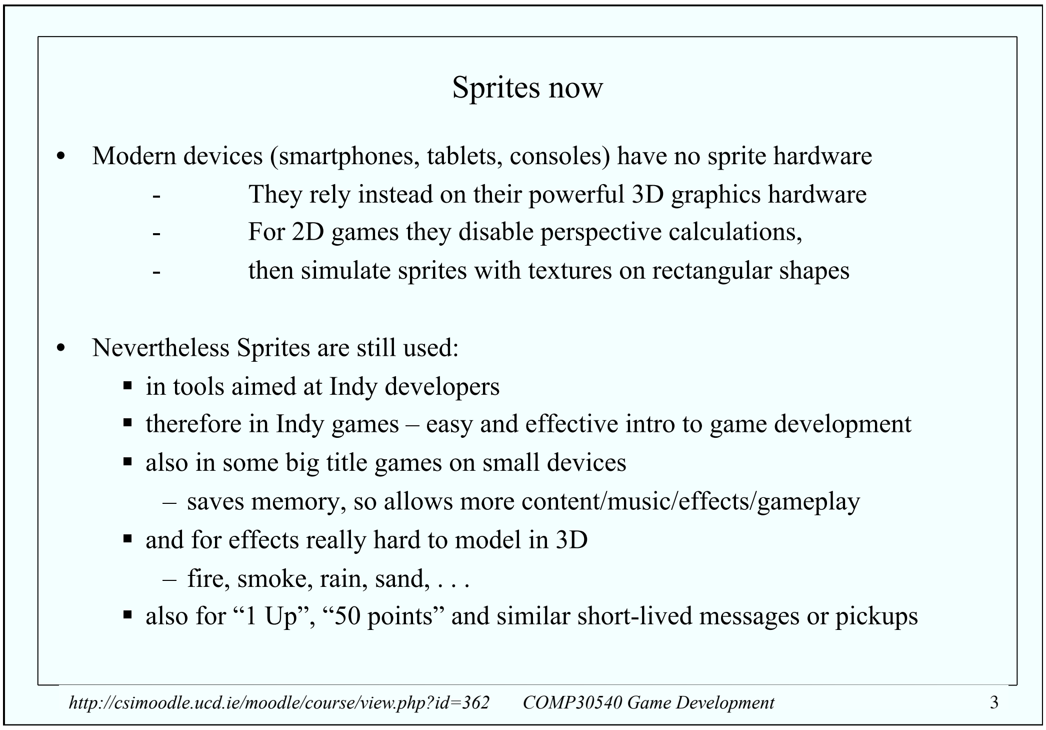
collisions of moving things with other things
COLLISION DETECTION and COLLISION RESPONSE
and spatial organisation and geometric calculations and physics and refresh

Sprites – in 2D, 2.5D, 3D

- Sprite: a 2-dimensional image or animation, generally with transparent parts
- overlaid onto other screen info, floating above it without altering it
  - though the term may also be applied to a background

- originally in hardware, like fixed-image screen cursors: limited number & variety
- now, with 3D graphics, applies to flat images superimposed on 3D screens

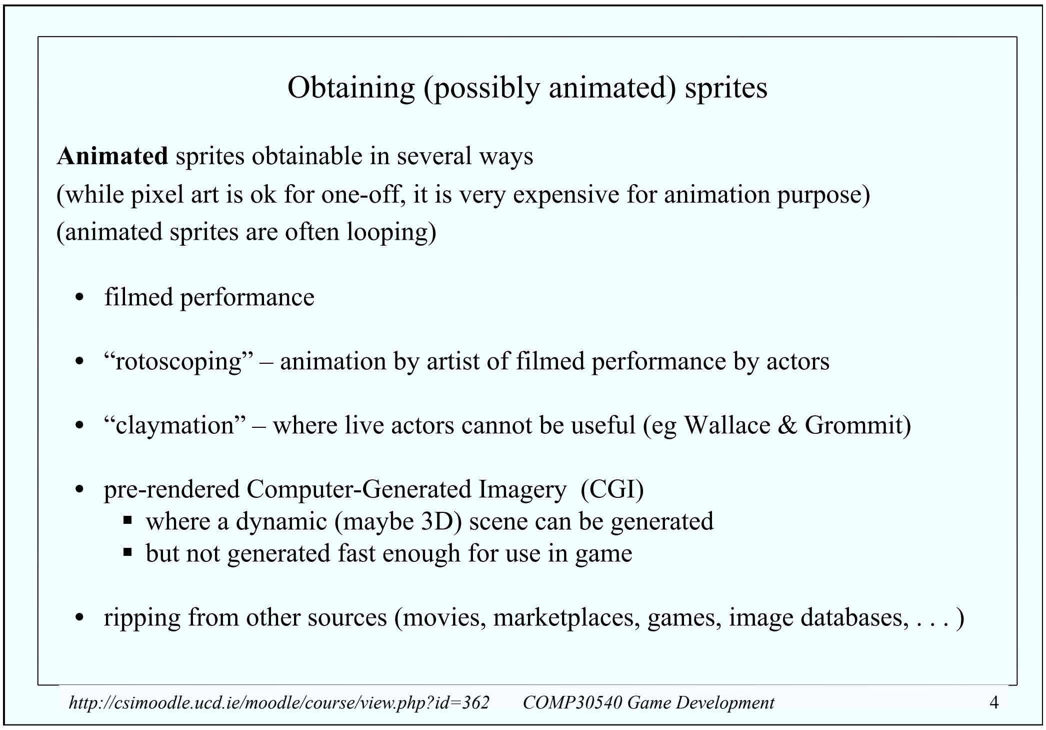
- transparency may involve special colour codes, or additional bit masks
- takes some time: double buffering often necessary to avoid flicker & tear



Sprites now

- Modern devices (smartphones, tablets, consoles) have no sprite hardware
  - They rely instead on their powerful 3D graphics hardware
  - For 2D games they disable perspective calculations,
  - then simulate sprites with textures on rectangular shapes

- Nevertheless Sprites are still used:
  - in tools aimed at Indy developers
  - therefore in Indy games – easy and effective intro to game development
  - also in some big title games on small devices
    - saves memory, so allows more content/music/effects/gameplay
  - and for effects really hard to model in 3D
    - fire, smoke, rain, sand, . . .
  - also for “1 Up”, “50 points” and similar short-lived messages or pickups



Obtaining (possibly animated) sprites

Animated sprites obtainable in several ways
(while pixel art is ok for one-off, it is very expensive for animation purpose)
(animated sprites are often looping)

- filmed performance

- “rotoscoping” – animation by artist of filmed performance by actors

- “claymation” – where live actors cannot be useful (eg Wallace & Grommit)

- pre-rendered Computer-Generated Imagery (CGI)
  - where a dynamic (maybe 3D) scene can be generated
  - but not generated fast enough for use in game

- ripping from other sources (movies, marketplaces, games, image databases, . . .)

Acceptance of prites

Since a sprite is flat and is always seen face-on, it may seem odd in a 3D scene.

It may seem not so odd if

- what it depicts is a 3D object, unobscured and with proper perspective, or
- it rotates or changes or vanishes too quickly to be closely observed, or
- it appears almost the same from many angles (sphere is the extreme), or
- it appears so distant that perspective effects would not change it noticeably, or
- the viewer happily accepts it as totally artificial (eg “1-Up”), or
- the viewer accepts it would look the same from different directions (eg grass)

Billboarding

Billboarding is a technique that exploits the face-forward property of sprites

- Useful eg to create a glow or a halo around another image
- 3D rendering engines can process “3D sprites” faster than general 3D objects
- 3D polyhedral models cannot cheaply & realistically handle fire etc effects

- In 3D, Billboards may be oriented as if from a point, or just turn about an axis

- The same principles can be used to make an object (e.g. a head) appear to turn to face some other object in a 3D scene, not just the camera.

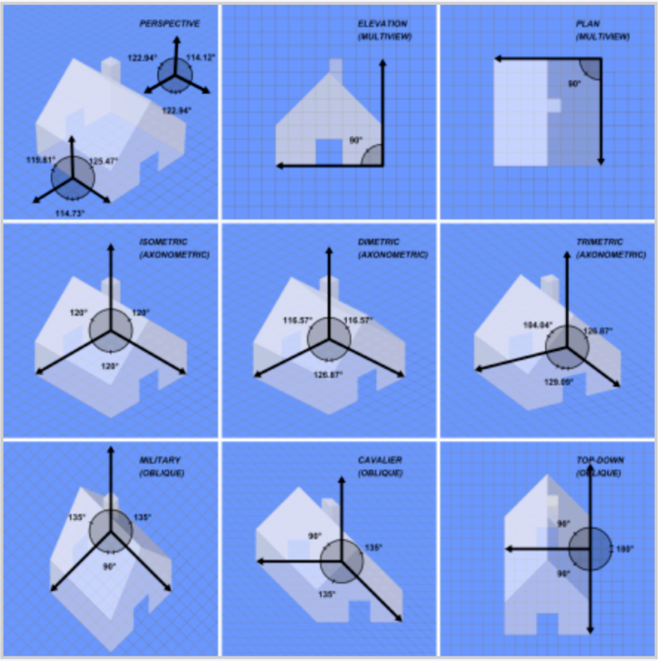
Sprites in 2.5D

- The term “2.5D” has 3 interpretations:
  - stereoscopic vision in depth perception (Marr, 2.5D sketch)
  - 2D gameplay set in a 3D-modelled scene
  - using projections or a series of images to simulate 3D appearance

- Axonometric projection – non use of perspective – foreshortening of 3 axes
  - isometric (all 3 the same) as in engineering drawings; all 120deg – SimCity
  - dimetric (2 of 3 the same) eg 2:1 pixel ratio, lessens anti-aliasing problems
  - trimetric (no 2 the same)

with perspective

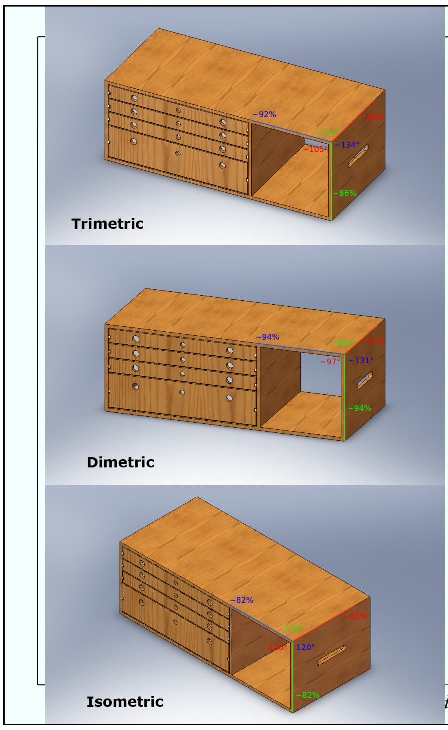
with projection



Various projection used for different purposes

Source:

https://en.wikipedia.org/wiki/Axonometric_projection



https://upload.wikimedia.org/wikipedia/commons/4/4c/Axonometric_projections.png

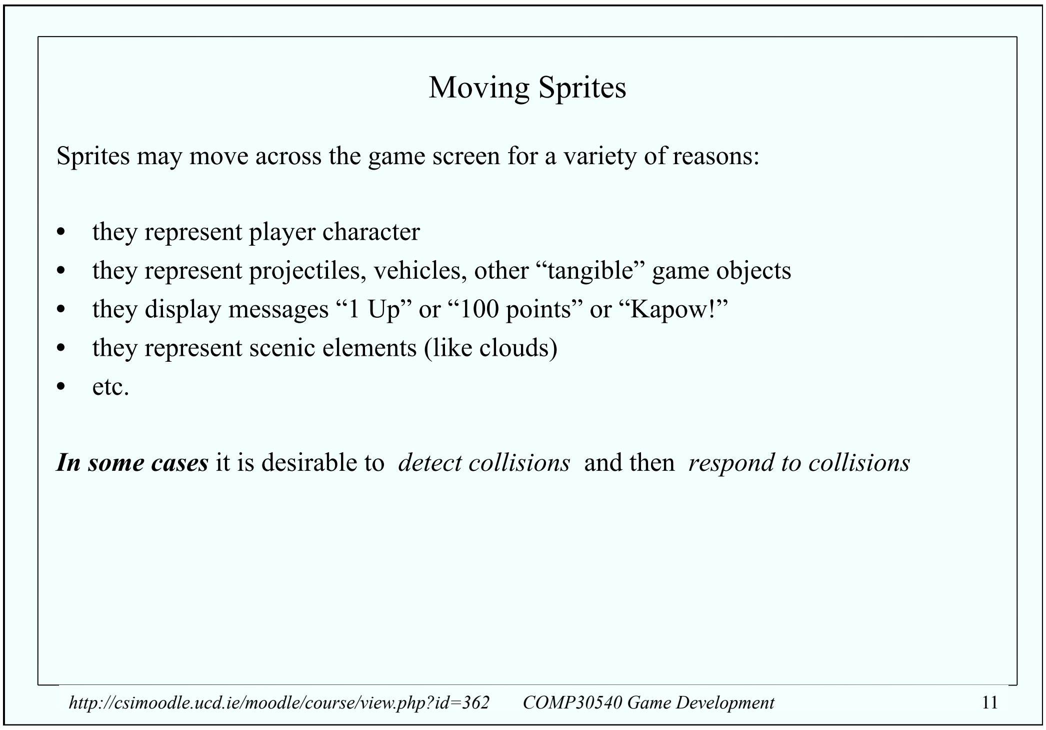
Simulating 3D with 2D

Strictly, the 2D images used in 2.5D (projections or image series) may be not sprites

- they may float over each other, in layers, as in Apple screensaver
- they may be in the background, rather than foreground, as in skyboxes/skydomes
- they may contain no transparent parts

Such 2D images are useful when the power of available graphics processing

- ***is enough*** to do calculations on bitmaps (translation, rotation, scaling, shearing, parallaxing, bump mapping, normal mapping)
- but ***is not enoug***h to handle everything through 3D modelling



Moving Sprites

Sprites may move across the game screen for a variety of reasons:

- they represent player character
- they represent projectiles, vehicles, other “tangible” game objects
- they display messages “1 Up” or “100 points” or “Kapow!”
- they represent scenic elements (like clouds)
- etc.

In some cases it is desirable to *detect collisions* and then *respond to collisions*