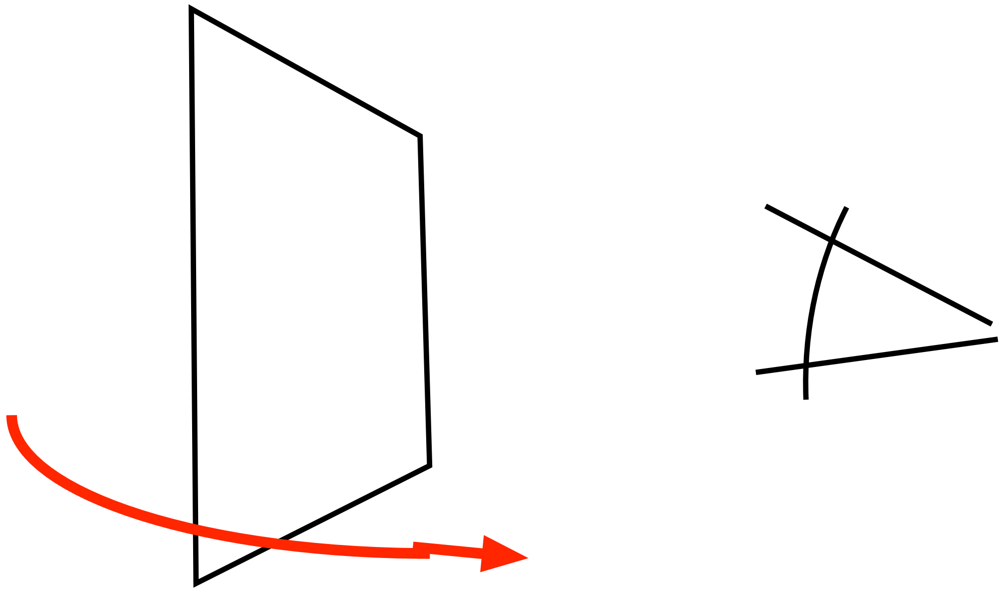
Billboards

A texture mapped polygon, which always faces the viewer



Billboards

2D images placed on surfaces that are always facing the camera

“Sprites” in older action games

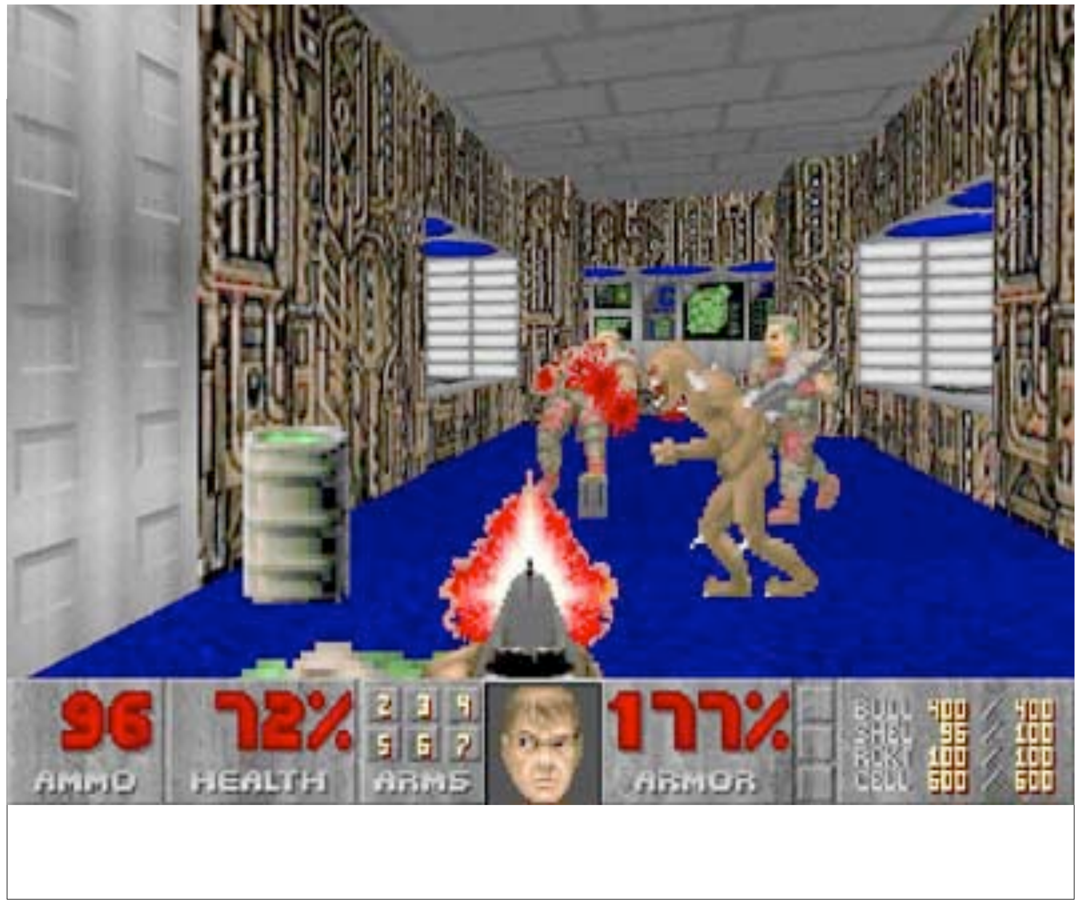
Advanced version: “impostors”

Often used for complex objects even today

Example: forests

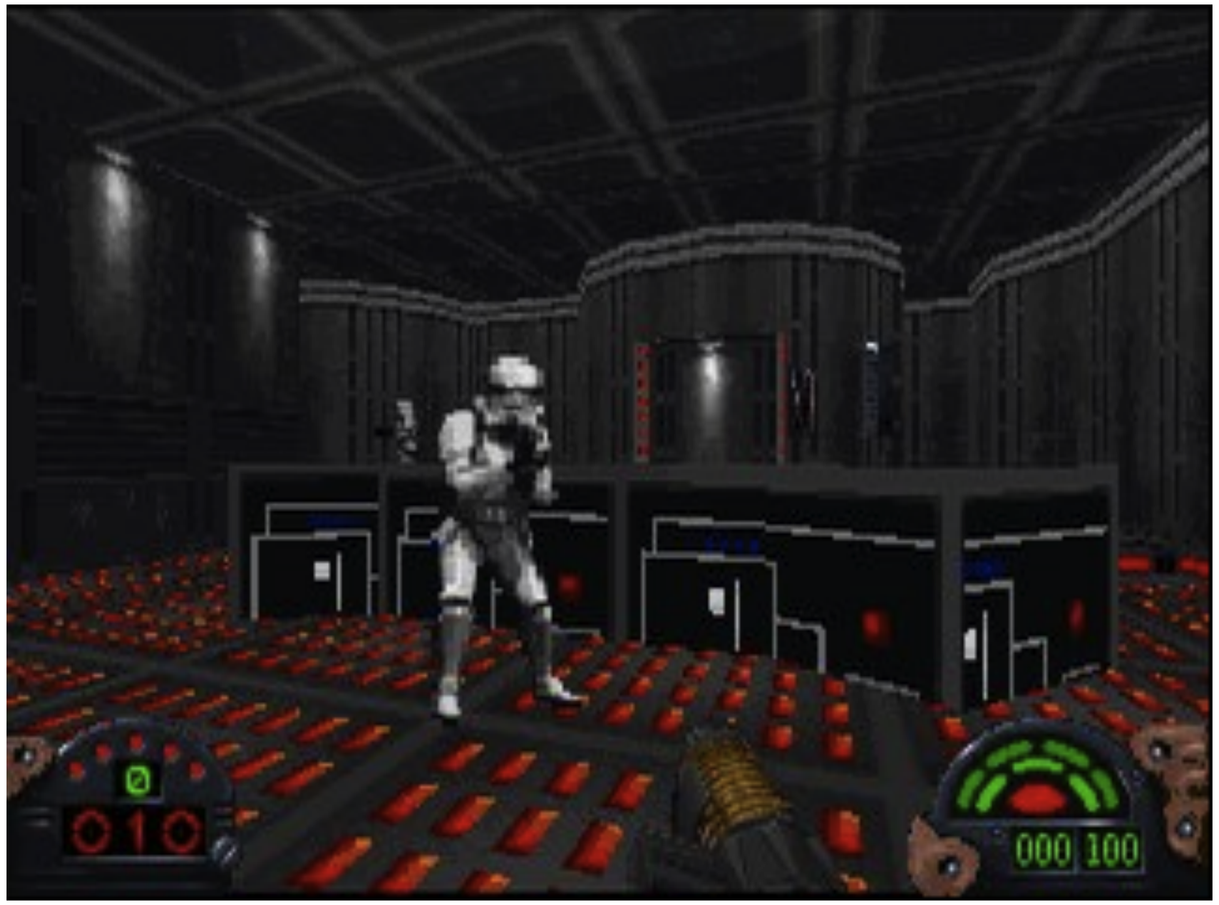
Classic example: Doom

Big billboards. No 3D models.

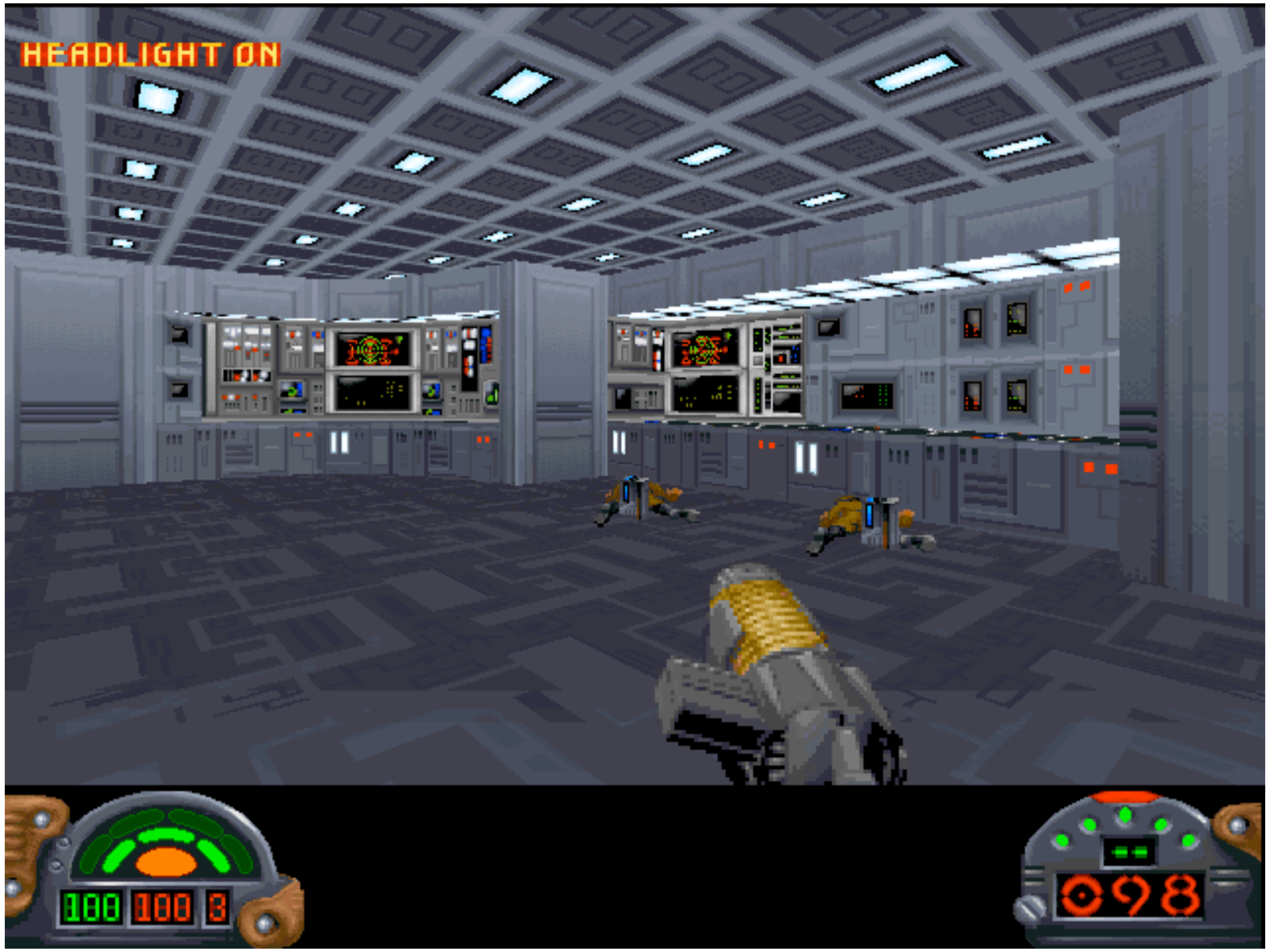


Classic example: Dark Forces

Big billboards. Some 3D models.

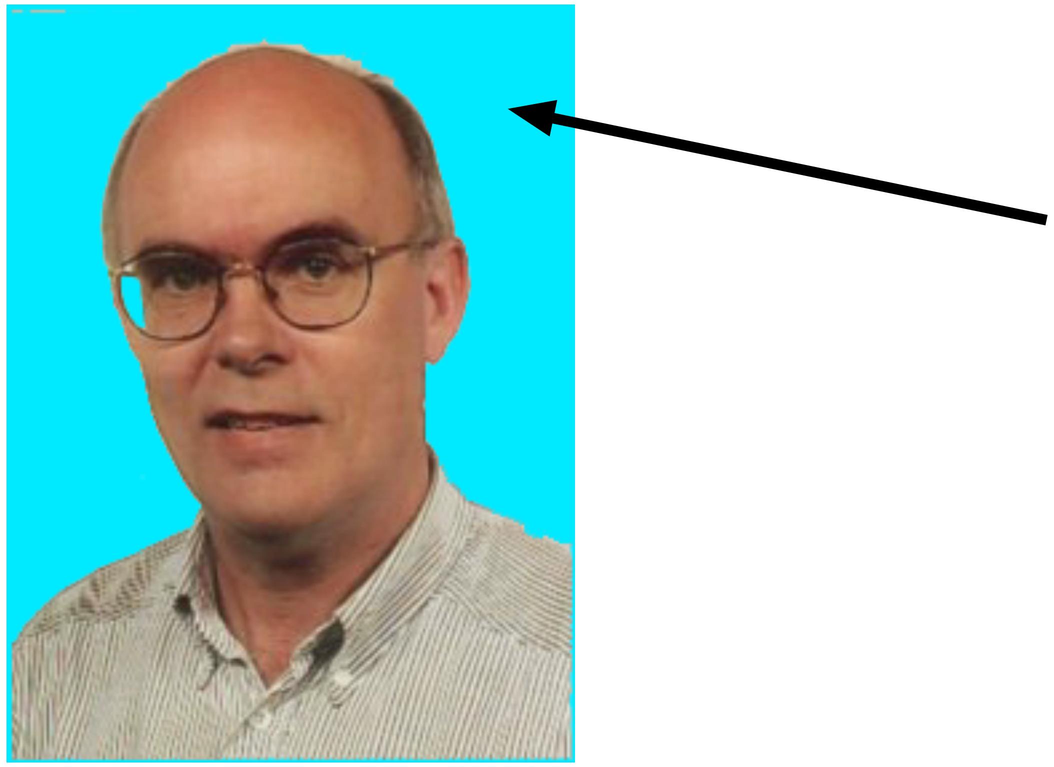


Problem with too few views.



Implementing billboards

- **Billboards generally need transparency!**



=> Special care is needed to avoid problems with Z-buffering!

Implementing billboards

- Billboards generally need transparency!

File formats:

TGA
PNG

TGA: simple format, good for demos

Libraries for PNG:
libPNG
PNGlite by Daniel Karling

Implementing billboards

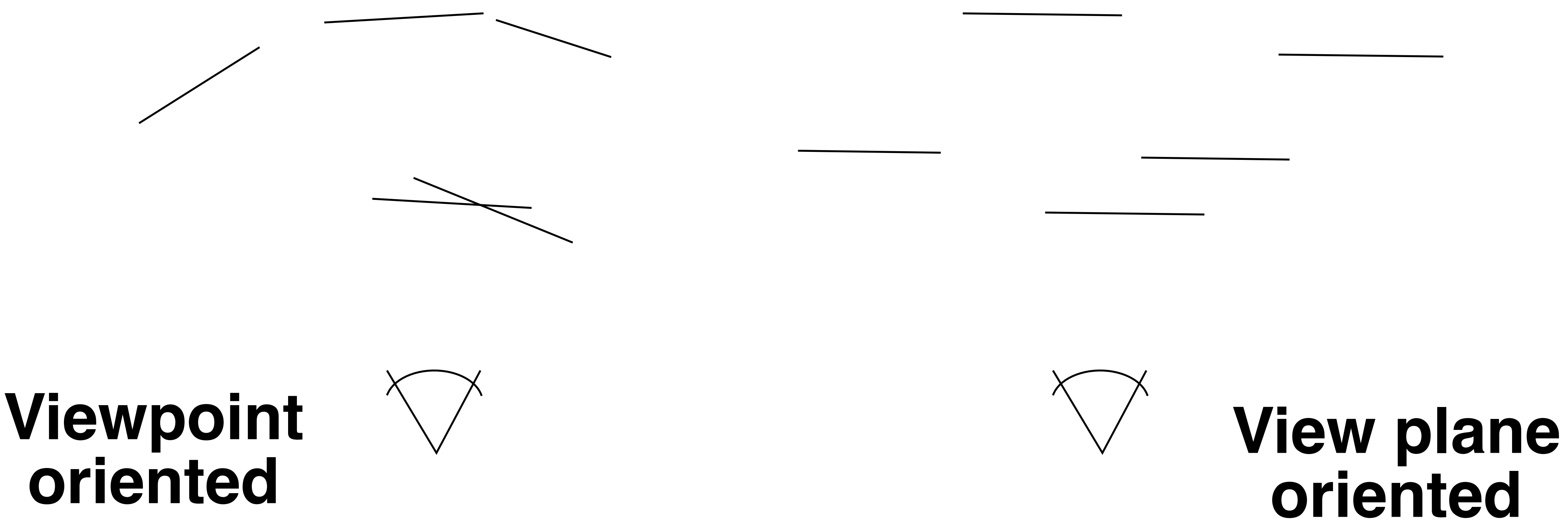
Variants:

- World oriented billboard
- Viewpoint oriented billboard (face the camera in full 3D)
- Axial (viewpoint) billboard
- View plane oriented billboard
- View plane oriented axial billboard

View plane oriented billboard

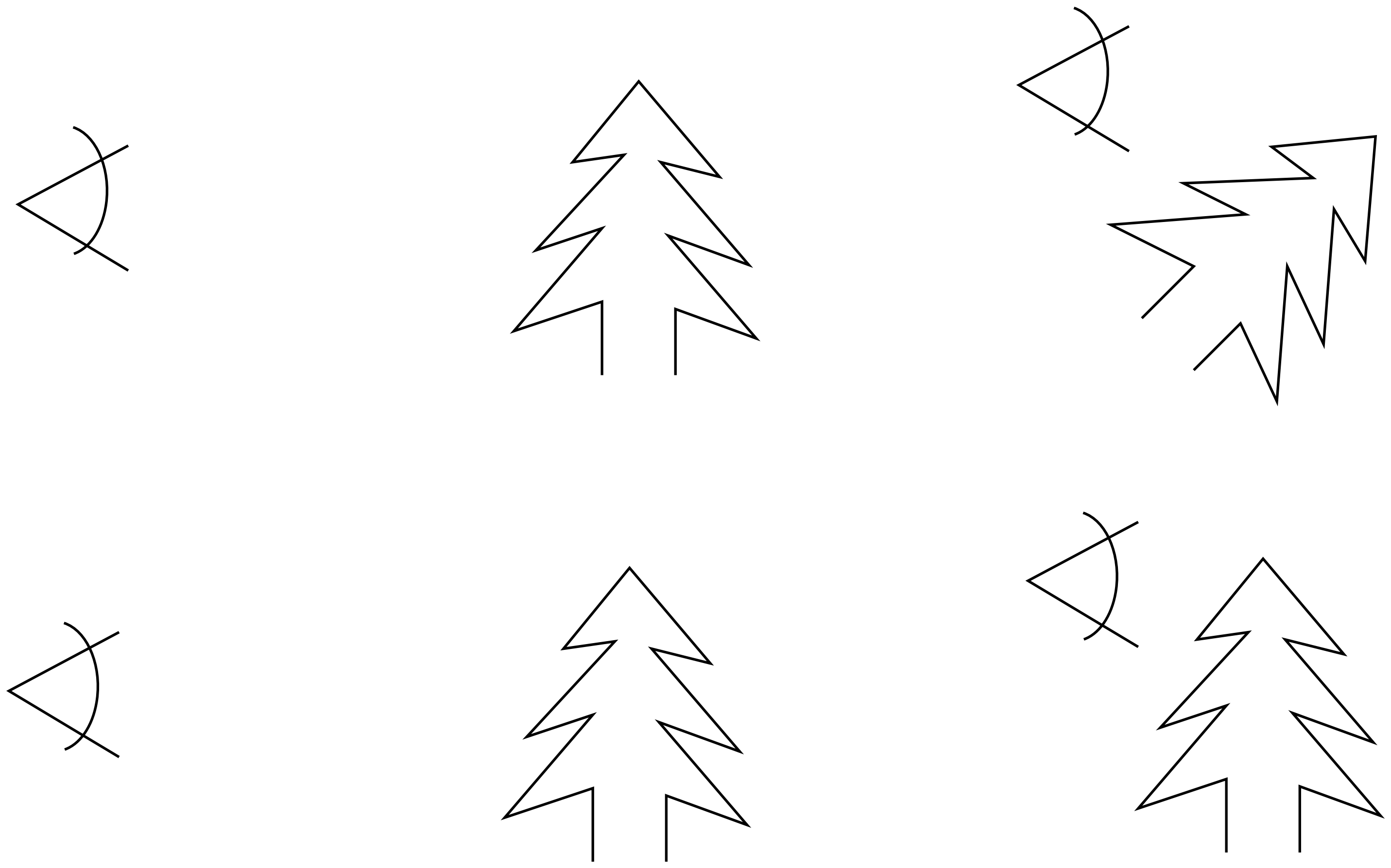
Easy! Zero out rotation!

Good - no overlaps!



Axial billboard

Rotate around Y



Non-axial

Axial
Suitable for trees etc

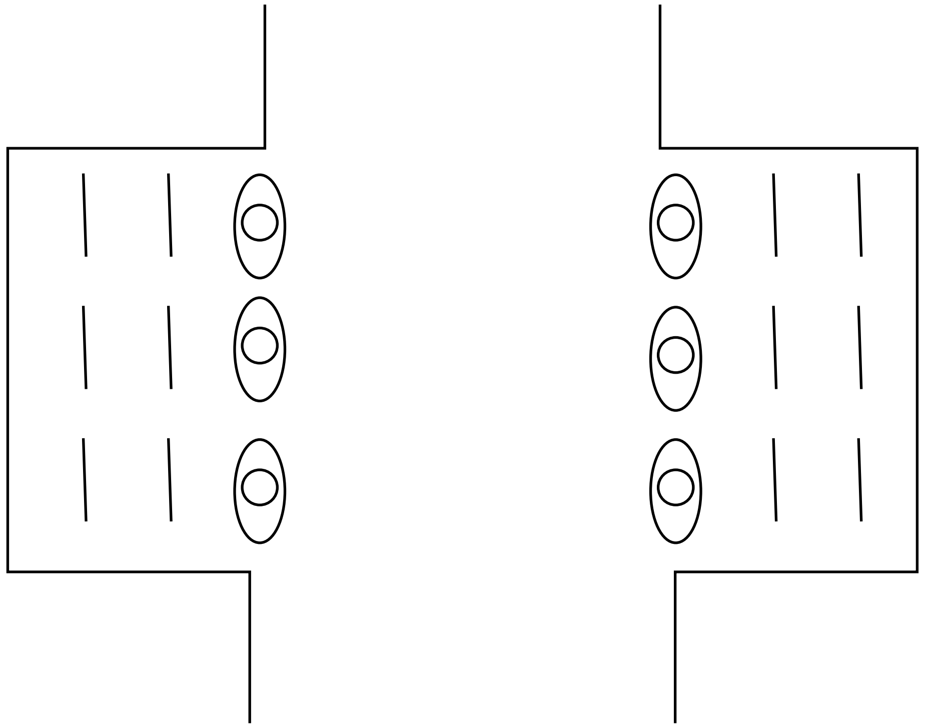
Full 3D viewpoint oriented billboard

Change of basis solution

Z vector from viewpoint, pick an up vector (usually Y axis), form basis with cross products

World oriented billboard

No camera dependent rotation



Example: Tomb Raider 2 Terracotta warriors (Temple of Xian)

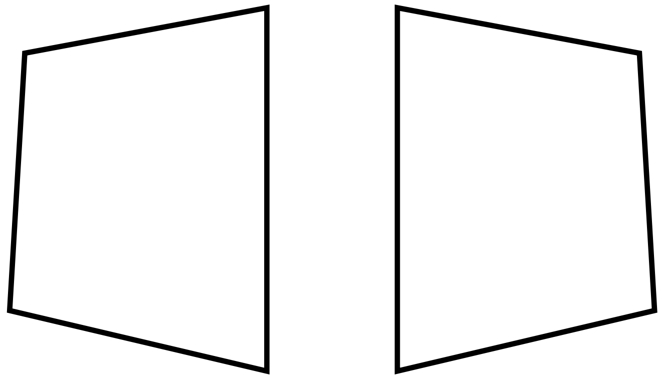
A grid of billboards

	Non-axial	Axial
Viewpoint oriented	Construct basis	Clear rotations
View plane oriented	Construct basis based on view plane	Construct axial rotation

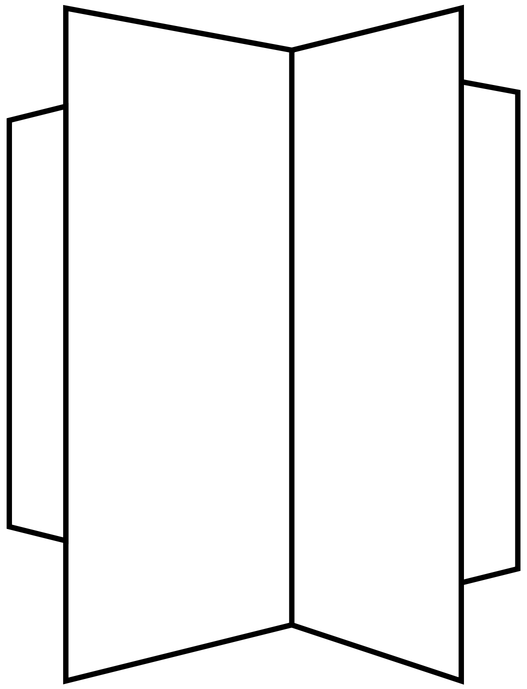
World oriented billboard not in grid

Billboards

1) Always facing the camera.
One polygon is enough!

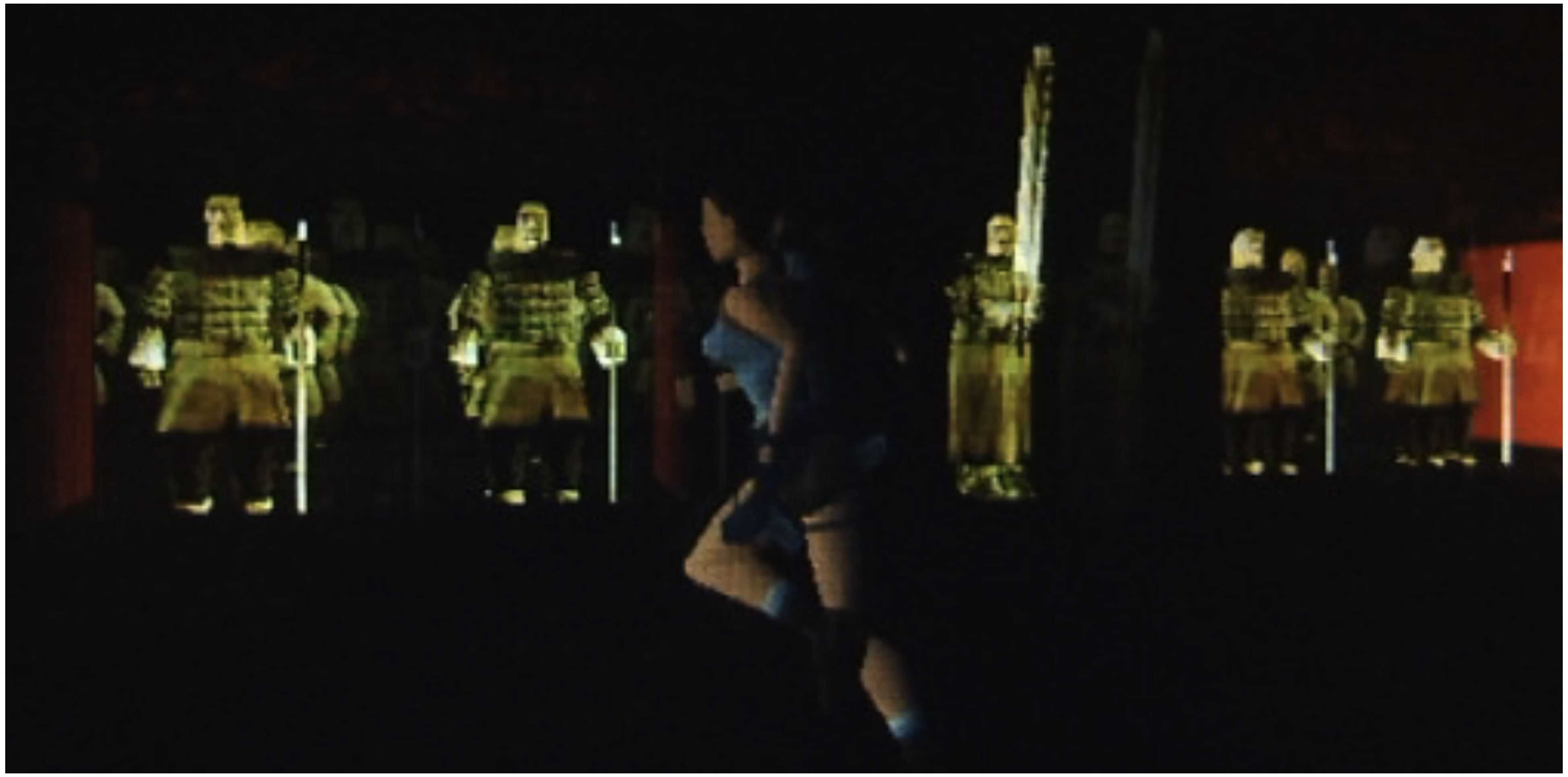


2) 2-3 polygons.
Higher realism.



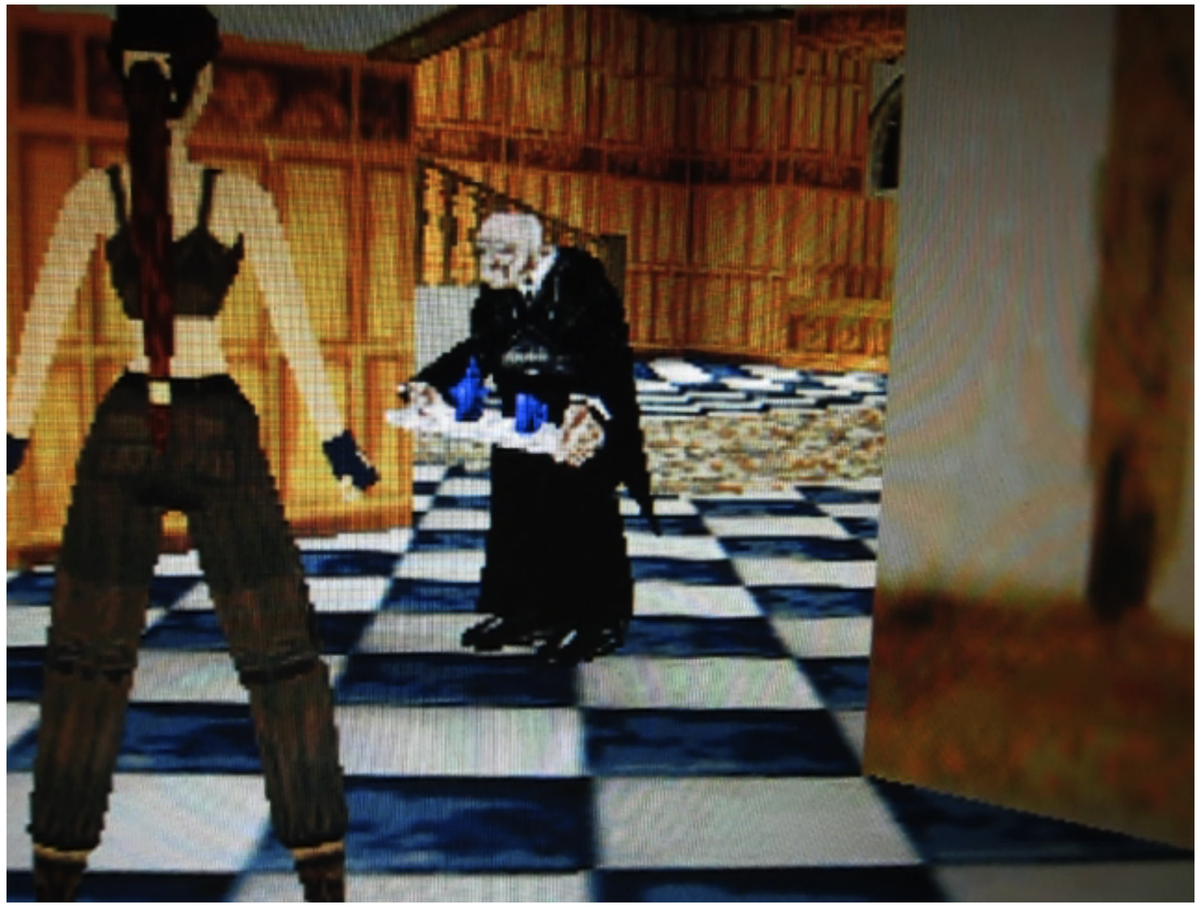
Temple of Xian

Statues behind first row are billboards!



Billboard variants

The objects on the plate are billboards - some 2-poly!

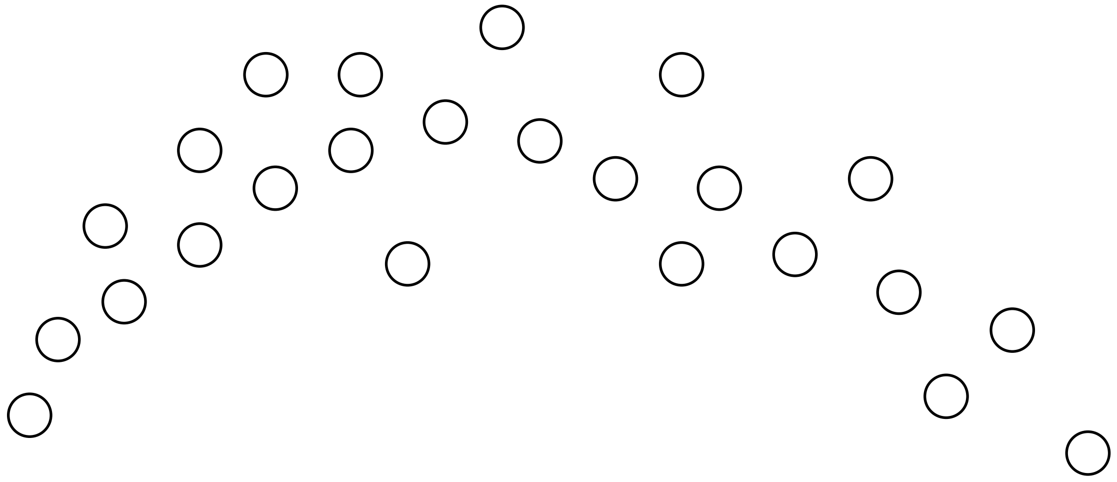


Application: Particle systems

Explosions, rain, fountains, smoke...

Excellent billboarding application

Many small objects - good opportunities to ”cheat” with transparency problems



Impostors

”Live” billboards

Render to texture, update sometimes

Render as other billboards

Decide when to update

Large worlds, conclusions

HIgh-level VSD to limit processing to visible parts - start with frustum culling

Level-of-detail to reduce unneccessary processing of detailed models

Use billboards for extreme simplification on large distance, particle effects etc

If we can’t see the difference, use the cheaper solution!