THE VIRTUAL EYE: GAME CAMERAS

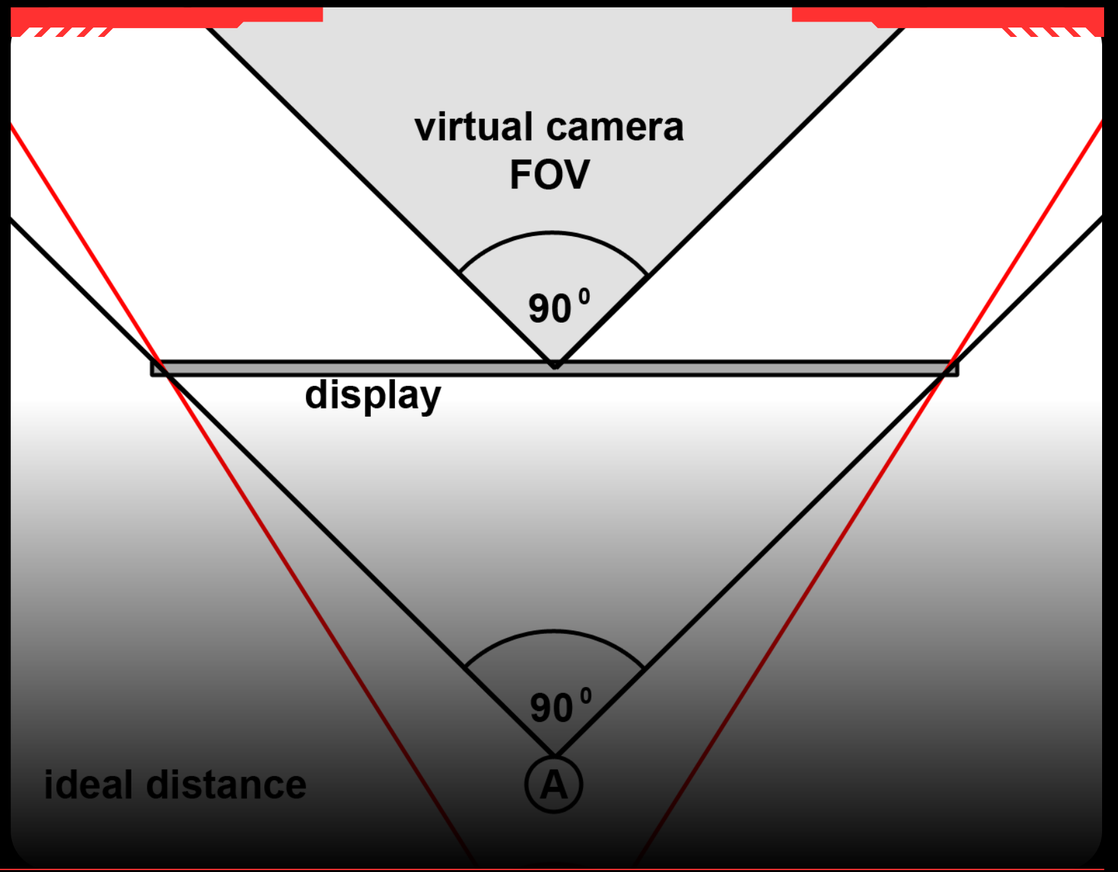


The geometry, psychology and code of virtual cinematography

Presented by Timbet

THE ~~FAKE~~ CAMERA IS A LIE

- In video games, there is no physical lens capturing photons.
- A virtual camera is essentially a single invisible point in 3D space with a defined direction vector.
- The computer engine does not actually move the camera around the world.
- To render a frame, the engine takes the camera's coordinates and moves the entire 3D universe in the exact opposite direction.





THE MATHS OF SIGHT

- For the Maths and Further Maths students like most of us in this room, this is applied linear algebra running at 60 frames per second.
- To convert a 3D world into a 2D image on your monitor, the engine uses a 4x4 View Matrix.
- This matrix is calculated by finding the inverse of the camera's translation (position) and rotation.
- Every single vertex of every single 3D model is multiplied by this matrix to figure out where it sits relative to your screen.

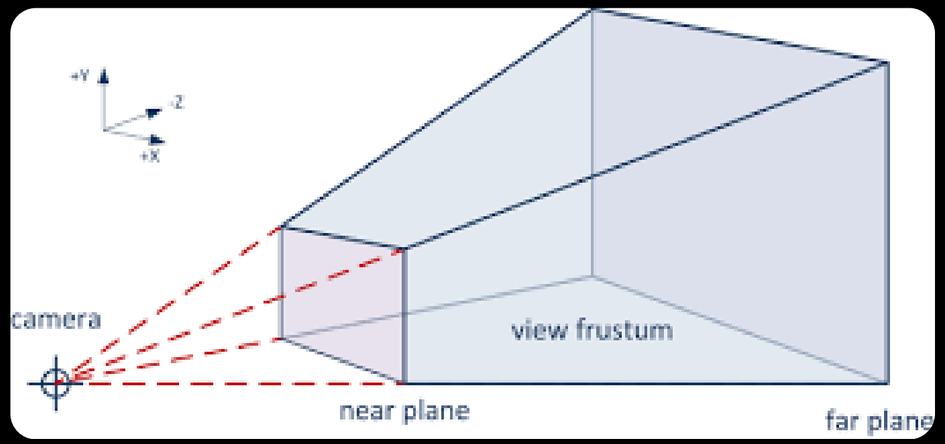


QUATERNIONS AND ROTATION

- Standard 3D rotation uses Euler angles (X, Y, Z).
- Rotating too far on one axis causes "Gimbal Lock", where two axes align and you lose a dimension of movement. This makes cameras wildly flip upside down.
- To fix this, game engines use Quaternions.
- Quaternions are a four dimensional complex number system used to calculate perfectly smooth spherical rotations without locking.

THE VIEW FRUSTUM

- Everything you actually see in a game is contained within a geometric shape called a View Frustum, which looks like a pyramid with the top chopped off.
- Near Plane: The closest point the camera renders. If an object passes this, it clips through your virtual "eyes" and disappears.
- Far Plane: The maximum distance the engine will calculate before giving up to save processing power.
- Anything outside this box is mathematically discarded via "Frustum Culling" to stop your GPU from melting.



LAKITU'S LEGACY [SUPER MARIO 64]

- Before 1996, developers had no idea how to design a controllable 3D camera.
- Nintendo's genius stroke was anthropomorphising the camera into a literal character: Lakitu floating on a cloud holding a camera on a fishing rod.
- By turning the camera into a physical entity within the game world, players instantly understood the spatial logic.
- It taught an entire generation how to navigate 3D space.



- In Super Mario 64, you can look in mirrors and see Lakitu hovering behind Mario.
- To save memory and processing power, the developers wrote code that stops Lakitu's model from rendering during normal gameplay.
- This means Mario is essentially being stalked by an invisible, sentient cloud ghost who controls what the player is allowed to see.

MARIO TRIVIA: THE GHOST CAMERAMAN



THE SPRING ARM

- The absolute hardest part of programming a third person camera is stopping it from phasing through solid brick walls.
- Developers use a "Spring Arm" component.
- The engine constantly fires an invisible laser (a raycast) from the player character directly backwards to the camera lens.
- If that laser hits a wall, the camera instantly snaps forward along the line to sit just in front of the collision point.



HERE'S THE SPRING ARM IN ACTION

LOW FOV / HIGH FOV



THE DRAMA OF FIELD OF VIEW

- FOV dictates the angle of the camera lens, but it is heavily used as a theatrical tool to manipulate player psychology.
- High FOV (90 to 120 degrees): Warps the edges of the screen, creating an aggressive sense of blistering speed and chaotic peripheral vision. Used in fast shooters like DOOM.
- Low FOV (60 degrees or less): Compresses the background and foreground together. It creates severe claustrophobia, intimacy, and cinematic tension. Used heavily in horror games.

CAMERA "JUICE" AND IMPACT

- Static, perfect code feels entirely dead. "Juice" refers to the subtle mathematical imperfections added to create visceral feedback.
- A basic screen shake works by feeding random noise values into the camera's X and Y coordinates for a quarter of a second.
- Better screen shake uses "Perlin Noise", generating smooth, organic waves of displacement that mimic a physical camera being hit by a shockwave.



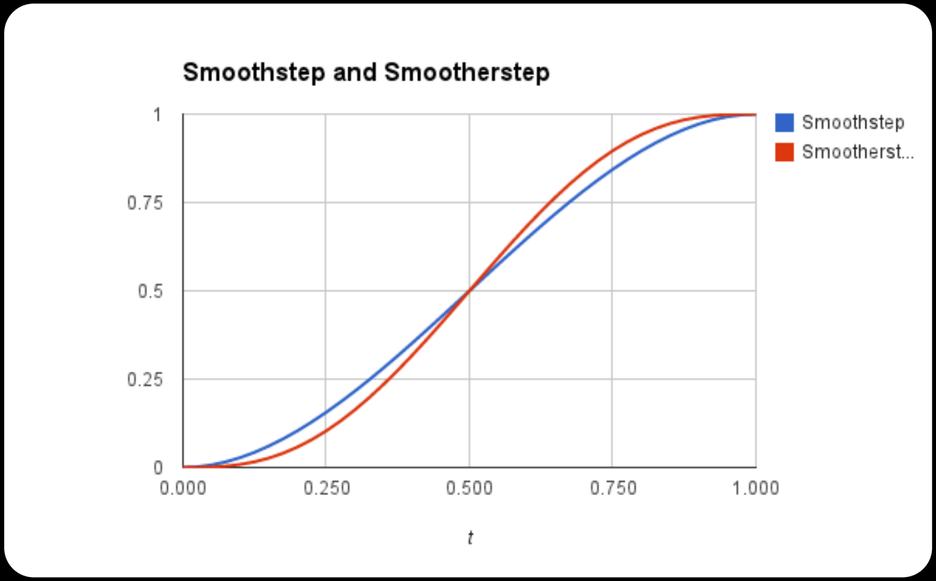
SCREEN SHAKE IN ACTION



Screen Shake

Filu

- Even in purely 2D games, locking the camera dead centre to the player character makes the game feel rigid and nauseating.
- Developers use "Lerp" (Linear Interpolation).
- The code tells the camera to constantly travel a percentage of the distance towards the player every frame, creating a smooth, rubber band effect that naturally eases in and out of movement.



2D CAMERA LOGIC AND LERPING

PLAYER PSYCHOLOGY AND NAUSEA

- Made famous by the film Vertigo, the Dolly Zoom creates a deeply unsettling visual warping effect where the subject stays the same size while the world stretches.
- In physical drama and cinema, you achieve this by physically pulling a camera backwards on a track while simultaneously zooming the lens in.
- In code, we replicate this by simultaneously moving the camera coordinates backwards on the Z axis while dynamically decreasing the Field of View variable.









SPHERICAL GRAVITY CAMERAS

- Super Mario Galaxy completely broke standard camera rules by placing players on tiny spherical planets.
- The camera’s "Up" vector, which normally points firmly towards the sky, had to be completely rewritten.
- The camera constantly calculates a vector between Mario and the exact core of the current planet, dynamically shifting the entire axis of the game world so "down" is always relative to gravity.

ANY QUESTIONS?



Feel free to ask about any specific games, maths, or your worst experiences getting a camera stuck inside a wall. **Also ask about Mario, PLEASE.**

Presented by Timbet