Week3A – Viewing

- Viewing in 3D → 2D
- Parallel, orthogonal, perspective views
- set { Modeling, Viewing, Projection, Viewport } → transformations, for 3D to 2D
- OpenGL default: orthogonal projection, i.e PARALLEL to z axis
  - ☹ (p, q, r) → (p, q, 0) (embedded onto z=0 plane)
    (---- lines are ⊥ to z=0 plane AND Parallel to z axis)
  - same thing !!
  - all geometry THEN scan converted/rasterised with z-buffer applied
  - ☹ ORTHOGONAL, can't obtain perspective
    - solution: we must pre-distort stuff

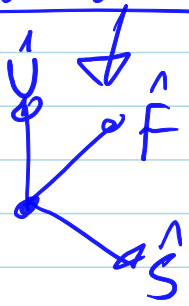
i.e  becomes

- Viewing: moving camera one way → the SAME shot as moving object the opposite
  - "Duality" of modelling and viewing
    - simulating "camera" by transforming object
      - NO CAMERA, we only transform a model

openGL default: "camera" looks down the z-axis at $(0,0,0)$

- eye point E
- centre of interest C
- view up vector V

→ use (E,C,V) to define a coordinate system



① $F = E - C$

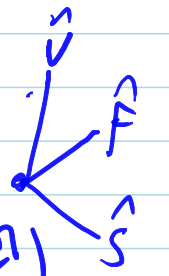
② $\hat{f}$ = normalised F (only direction needed)

② which way is UP for camera? → V

① CAN'T assume V is orthogonal to F → unreliable

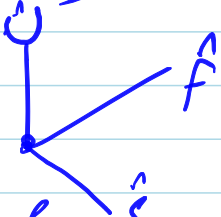
① Decouple $\hat{F}$ and $\hat{V}$ complete

→ $\hat{S} = \text{normalise}(\hat{F} \times \hat{V})$ →



→ THEN, $\hat{U} = \text{normalise}(\hat{F} \times \hat{S})$

→ Throw $\hat{V}$ away →



→ How to derive $(T_c)^{-1}$: get centre of above back to $(0,0,0)$