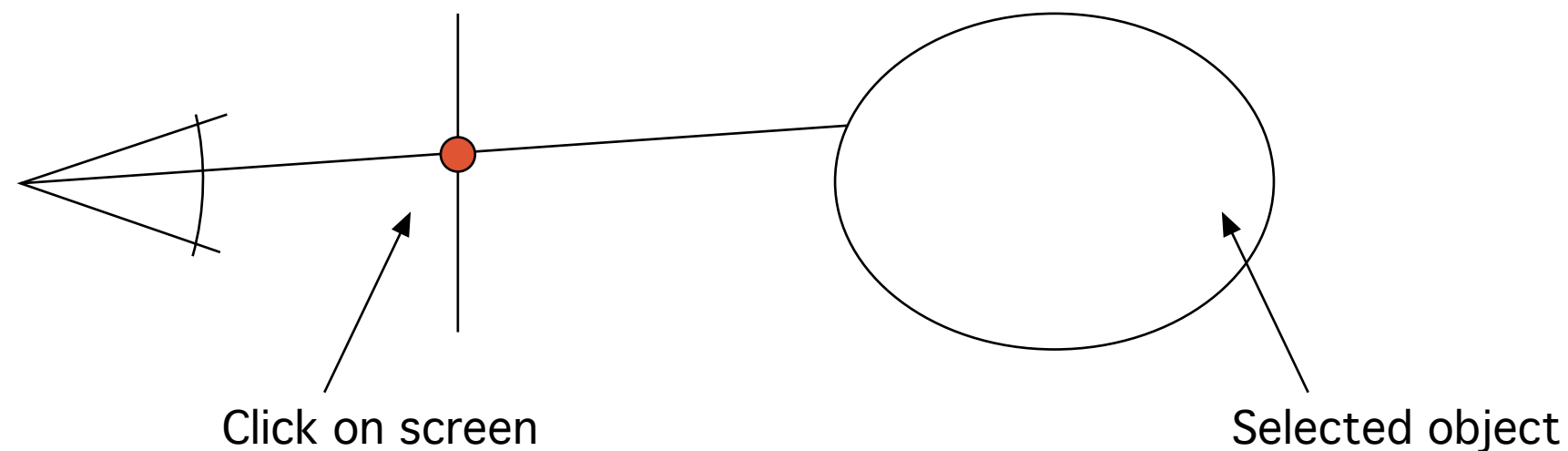




Picking

Interactively selecting objects with a mouse

Can be solved with raycasting!





Information Coding / Computer Graphics, ISY, LiTH



But in our case... picking = selecting.



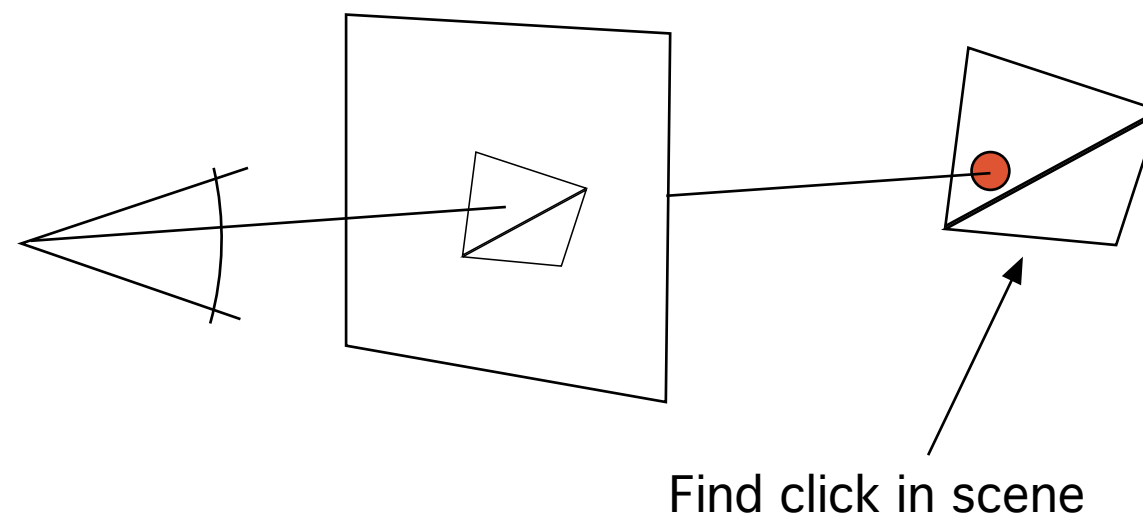
Picking in model space

Point in viewing plane

Create line through origin and point

Transform by inverse model-to-view

Find intersections with models





Picking in view space

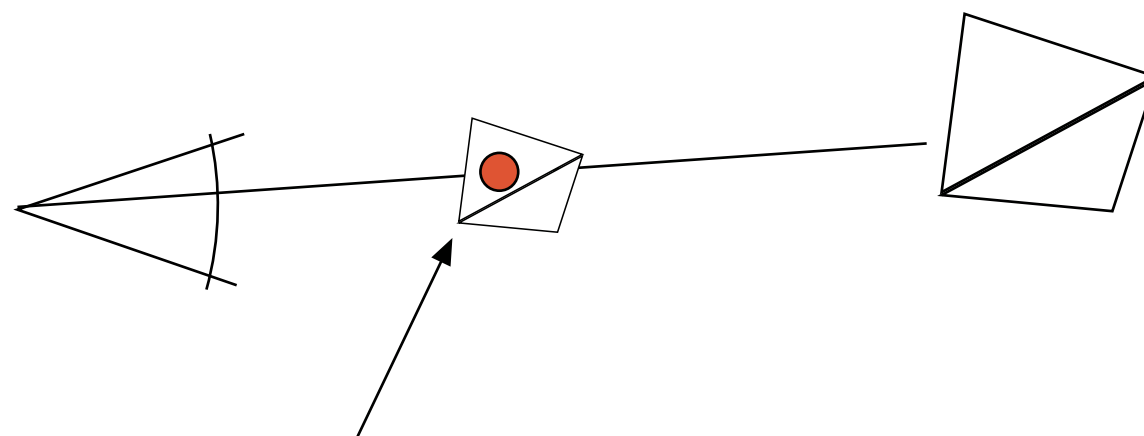
Point in viewing plane

Transform models by model-to-view

Find intersections with models

Cumbersome if all models need to be transformed

Cheap if done as part of drawing



Find click in projected triangles



Picking in image space

Draw all models with individual indices (colors)

Get resulting color at click from frame buffer

**Requires drawing entire scene an extra time - but
you can restrict drawing to a very small area**





Colors for indexing

The standard setting allows 256 shades per channel, red, green and blue

Assign each model a unique index, split to R, G, B, 8 bits each, divide each by 255 to get color values.

Draw. Get pixel at mouse position with the not very common call "glReadPixels".



Method in old OpenGL, GL_SELECTION

Requires you to enter a special mode, selection mode, and assign indices to each model.

Similar in usage and result to the color indexing method. Obsolete! Use any of the ones above instead.



Best picking method?

Picking in model space: Typically done on CPU (somewhat slow but capable).

Picking in view space: Efficient if done in the geometry stage (needs some tricks to output result to host)

Picking by index colors: Easy but can not identify local parts of a model.