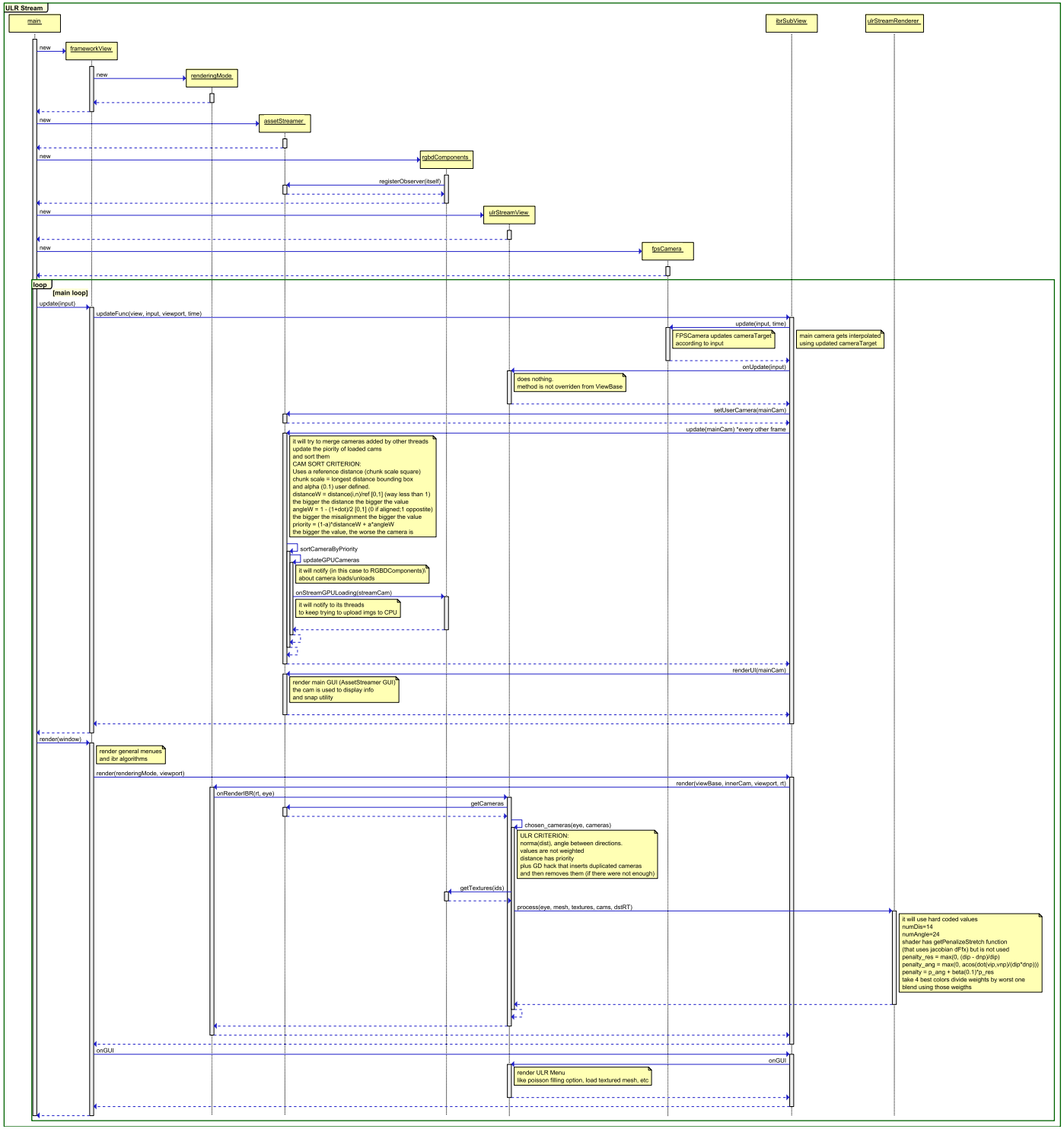
ULR Stream
main
ibrSubView
ulrStreamRenderer
new
frameworkView
new
renderingMode
new
assetStreamer
new
rgbdComponents
registerObserver(itself)
new
ulrStreamView
new
fpsCamera
loop
[main loop]
update(input)
updateFunc(view, input, viewport, time)
update(input, time)
FPSCamera updates cameraTarget
according to input
main camera gets interpolated
using updated cameraTarget
onUpdate(input)
does nothing.
method is not overriden from ViewBase
setUserCamera(mainCam)
update(mainCam) *every other frame
it will try to merge cameras added by other threads
update the piority of loaded cams
and sort them
CAM SORT CRITERION:
Uses a reference distance (chunk scale square)
chunk scale = longest distance bounding box
and alpha (0.1) user defined.
distanceW = distance(i,n)/ref [0,1] (way less than 1)
the bigger the distance the bigger the value
angleW = 1 - (1+dot)/2 [0,1] (0 if aligned;1 oppostite)
the bigger the misalignment the bigger the value
priority = (1-a)*distanceW + a*angleW
the bigger the value, the worse the camera is
sortCameraByPriority
updateGPUCameras
it will notify (in this case to RGBDComponents)\
about camera loads/unloads
onStreamGPULoading(streamCam)
it will notify to its threads
to keep trying to upload imgs to CPU
renderUI(mainCam)
render main GUI (AssetStreamer GUI)
the cam is used to display info
and snap utility
render(window)
render general menues
and ibr algorithms
render(renderingMode, viewport)
render(viewBase, innerCam, viewport, rt)
onRenderIBR(rt, eye)
getCameras
chosen_cameras(eye, cameras)
ULR CRITERION:
norma(dist), angle between directions.
values are not weighted
distance has priority
plus GD hack that inserts duplicated cameras
and then removes them (if there were not enough)
getTextures(ids)
process(eye, mesh, textures, cams, dstRT)
it will use hard coded values
numDis=14
numAngle=24
shader has getPenalizeStretch function
(that uses jacobian dFfa) but is not used
penalty_res = max(0, (dip - dnp)/dip)
penalty_ang = max(0, acos(dot(vip,vnp)/(dip*dnp)))
penalty = p_ang + beta(0.1)*p_res
take 4 best colors divide weights by worst one
blend using those weigths
onGUI
onGUI
render ULR Menu
like poisson filling option, load textured mesh, etc