STARTING
PROGRAM

Main(Act.)

Construct Simulator( for Act.)

Construct Viewer3D( for Sim.)

Construct RenderProcess()

Viewer3D.Configure()

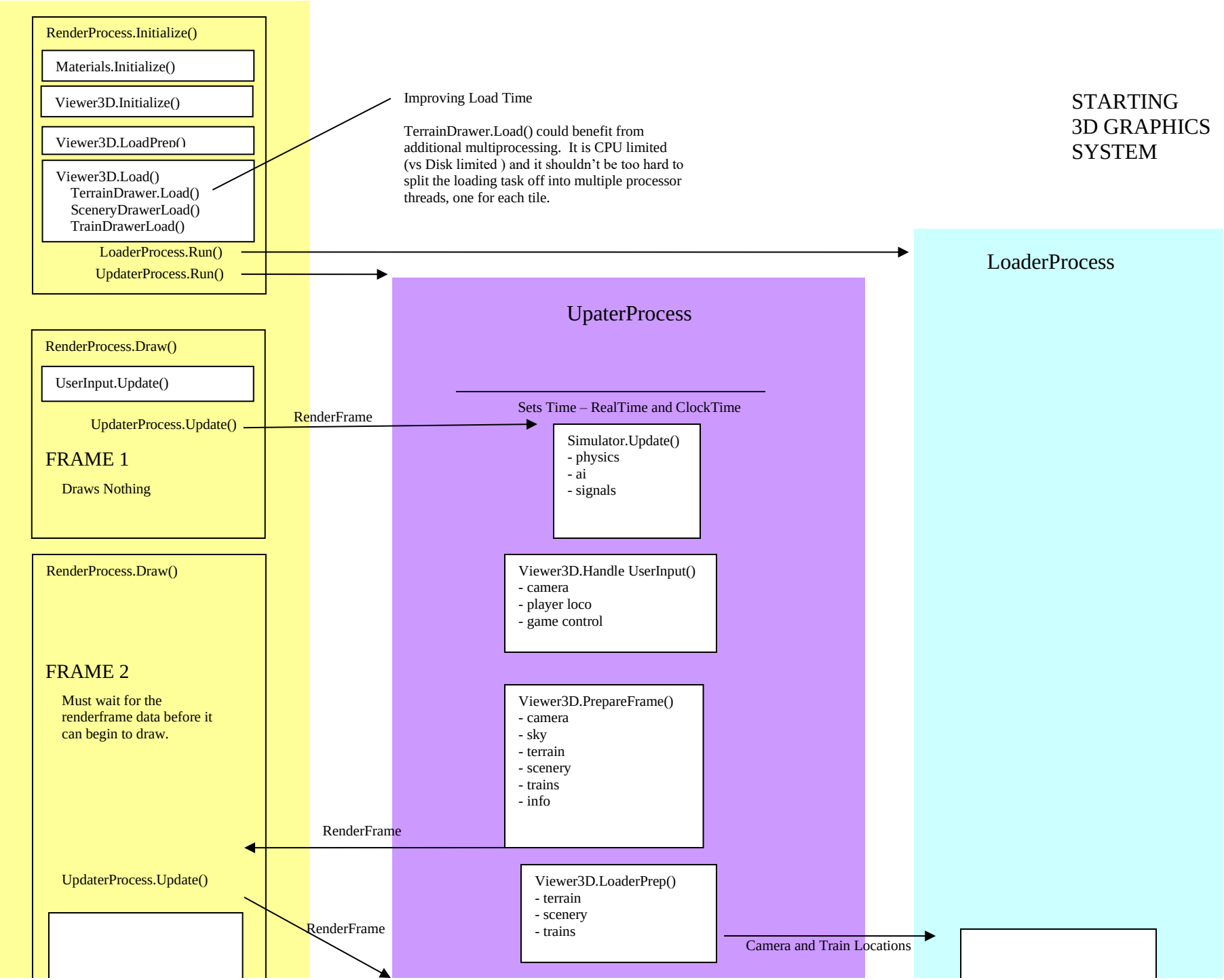
Construct LoaderProcess()

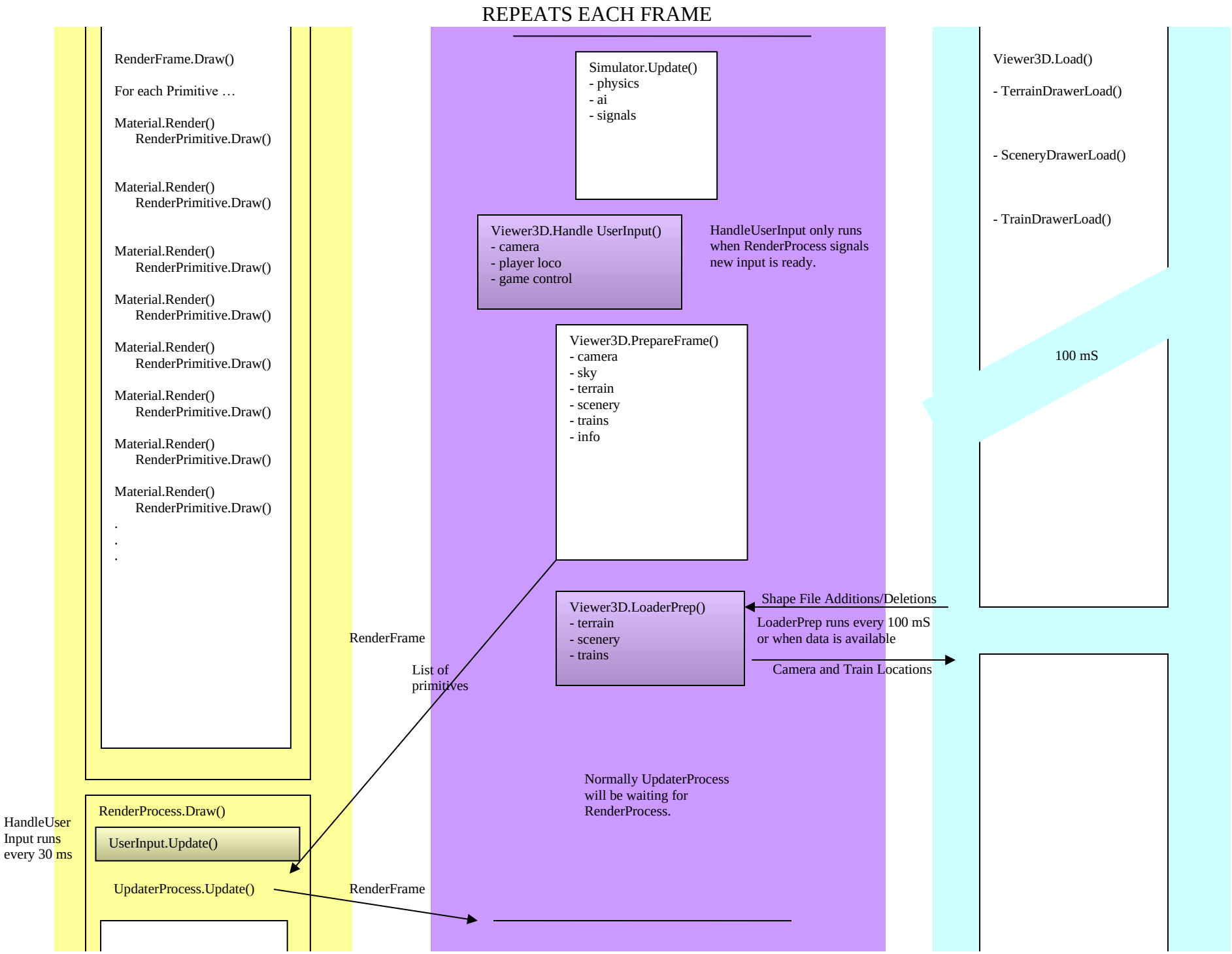
Construct UpdaterProcess()

Viewer3D.Run()

RenderProcess.Run()

RenderProcess
( XNA Game Class )


STARTING
3D GRAPHICS
SYSTEM
RenderProcess.Initialize()
Materials.Initialize()
Viewer3D.Initialize()
Viewer3D.LoadPrep()
Viewer3D.Load()
TerrainDrawer.Load()
SceneryDrawerLoad()
TrainDrawerLoad()
LoaderProcess.Run()
UpdaterProcess.Run()
Improving Load Time
TerrainDrawer.Load() could benefit from additional multiprocessing. It is CPU limited (vs Disk limited ) and it shouldn't be too hard to split the loading task off into multiple processor threads, one for each tile.
LoaderProcess
UpaterProcess
RenderProcess.Draw()
UserInput.Update()
UpdaterProcess.Update()
RenderFrame
Sets Time – RealTime and ClockTime
FRAME 1
Draws Nothing
Simulator.Update()
- physics
- ai
- signals
RenderProcess.Draw()
Viewer3D.Handle UserInput()
- camera
- player loco
- game control
FRAME 2
Must wait for the renderframe data before it can begin to draw.
Viewer3D.PrepareFrame()
- camera
- sky
- terrain
- scenery
- trains
- info
RenderFrame
UpdaterProcess.Update()
Viewer3D.LoaderPrep()
- terrain
- scenery
- trains
RenderFrame
Camera and Train Locations


REPEATS EACH FRAME
RenderFrame.Draw()
For each Primitive …
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
Material.Render()
RenderPrimitive.Draw()
RenderProcess.Draw()
UserInput.Update()
UpdaterProcess.Update()
HandleUser Input runs every 30 ms
RenderFrame
List of primitives
RenderFrame
Simulator.Update()
- physics
- ai
- signals
Viewer3D.Handle UserInput()
- camera
- player loco
- game control
HandleUserInput only runs when RenderProcess signals new input is ready.
Viewer3D.PrepareFrame()
- camera
- sky
- terrain
- scenery
- trains
- info
Viewer3D.LoaderPrep()
- terrain
- scenery
- trains
Shape File Additions/Deletions
LoaderPrep runs every 100 mS or when data is available
Camera and Train Locations
Normally UpdaterProcess will be waiting for RenderProcess.
Viewer3D.Load()
- TerrainDrawerLoad()
- SceneryDrawerLoad()
- TrainDrawerLoad()
100 mS