MolecularFlipBook Code Overview

Initialization

- `framework.init()` *initializes the main scene*
- `framework.init2d()` *initializes the optional overlay scene*

Main Loop

- `framework.loop30()`

  > - `sleepHandler()` : Idles app to save CPU cycles
  > - `space3d.navigationCubeUpdate()` : Updates navigation cube if enabled
  > - `logic.gui.loop()` : Handles GUI event loop
  > - `space3d.cameraUpdate()` : Updates view camera
  > - `keyboardHandler()` : Handles global keyboard input (except for the GUI inputs)
  > - `playHandler()` : Plays back animation
  > - `manipulator.setManipulator()` : Handles the manipulation widget
  > - `shader.updateProtein()` : Update shader
  > - `shader.updateWidget()` : Update shader
  > - runs `logic.registeredFunctions[]` : Delayed or periodical function calls

- `framework.loop15()`

  > - `resizeHandler()` : Handles resize of window
  > - `updateSky()` : Update sky shader

- `framework.loop5()`

  > - `space3d.cameraManipulatorPulse()` : Handles slower pulses of the manipulator

> - runs `logic.deferredFunctions[]` : Delayed or periodical function calls

singleton class *logic.mvb*

- `mvb.objects = {Class MVBObject}`
- `mvb.slides = [Class Slides]`
- `mvb.activeObjs = set(kx_object)`
- `mvb.preActiveObj = kx_object or [kx_object]`
- `mvb.activeSlide = 0` onChange: Goes to that time, highlight slides and update slide number
- `mvb.time = 0` onChange: Calls viewTime()
- `mvb.playing = False`
- `mvb.rendering = False`
- `mvb._scrubbing = False`
- `mvb._frameCounter = 0`
- `mvb._rendering = False` onChange: Reset .mvb._frameCounter
- `mvb.looping = True`
- `mvb.snap()` snap to the closest slide
- `mvb.hoverObjectUpdate()`
- `mvb.selectObjectUpdate()`
- `mvb.addObject(name, obj, pdbFullPath)` instantiate new objects
- `mvb.deleteObject(mvbObj)` mark obj as deleted
- `mvb.getMVBObject(kx_object)` get mvb object from kx_obj
- `mvb.addSlide(index, silent)` create a new keyframe at index
- `mvb.deleteSlide(index)` delete keyframe at index
- `mvb.moveSlide(a, b)` move slide a to b
- `mvb.viewTime(time)` interpolate keyframes and display a particular time
- `mvb.getTotalTime()` return total time of animation

Namespace *logic*

- `logic.binaryPlayerPath` *Path of the blenderplayer binary*
- `logic.binaryBlenderPath` *Path of the blender binary*
- `logic.basePath` *Path of the mfb app*
- `logic.tempFilePath` *Path of the mfb cache folder*
- `logic.renderFilePath` *Path of the mfb render folder*

- `logic.scene`
    - `logic.viewCamObject`, `logic.widgetObject`, `logic.controllerObject`,
    - `logic.widgetList`, `logic.widgetRenderList`
- `logic.scene2D`
    - `logic.scene2DReady`, `logic.viewCamOrthoObject`
- `logic.watch{name}=var` *add to debug watcher*
- `logic.registeredFunctions=[func]` *run func every tic until returns False*
- `logic.objCounter = 1`
- `logic.activeContext()`
- `logic.gui`
- `logic.gate`
- `logic.outliner`
- `logic.helper`
- `logic.options`
- `logic.logger`
    - `logic.logger.new(msg, type='ERROR|WARNING|IMPORTANT|MESSAGE')`
- `logic.timeline`
    - `logic.timeline.playToggle()`
    - `logic.timeline.loopToggle()`
    - `logic.timeline.slideDelete()`
    - `logic.timeline.slideAdd()`
    - `logic.timeline.slideDelete()`
    - `logic.timeline.viewUpdate()`

singleton class *logic.gui*

- `gui.publishDialog`
- `gui.importDialog`
- `gui.viewport`
- `gui.showSimpleUI`

- `gui.showFullUI`
- `gui.showtoolTip(widget)`
- `gui.hideToolTip(widget)`
- `gui.showMenu()`
- `gui.hideMenu()`
- `gui.showModalMessage(subject, message, action)`
- `gui.showModalConfirm(subject, message, action, cancelAction)`
- `gui.onClick()`
- `gui.outlinerIsVisible()`
- `gui.outlinerVisible()`
- `gui.helpIsVisible()`
- `gui.helpVisible()`
- `gui.gridVisible()`
- `gui.initMultitouch()`
- `gui.loop()`

Actions module

- `actions.deleteObjs()`
- `actions.scatterObjs()`
- `actions.gatherObjs()`
- `actions.duplicateObjs()`

File Interface module

- `fileInterface.saveBrowse()`
- `fileInterface.loadBrowse()`
- `fileInterface.save()`
- `fileInterface.load()`
- `fileInterface.saveSession()`
- `fileInterface.loadSession()`

Helper module

- `helpers.profile(cmd, global, local)`
- `helpers.smoothstep(x)`

- `helpers.comptueFlatS(period, b, time)`
- `helpers.mix(a,b, factor)`
- `helpers.guiKill(widget)`
- `helpers.drawLine(x)`
- `helpers.createBusyBar(container, size, pos)`
- `helpers.updateBusyBar(container, time)`
- `helpers.guiKill(widget)`
- `helpers.themeRoot(filename)`
- `helpers.createPath(path)`
- `helpers.activeContext(regions)`
