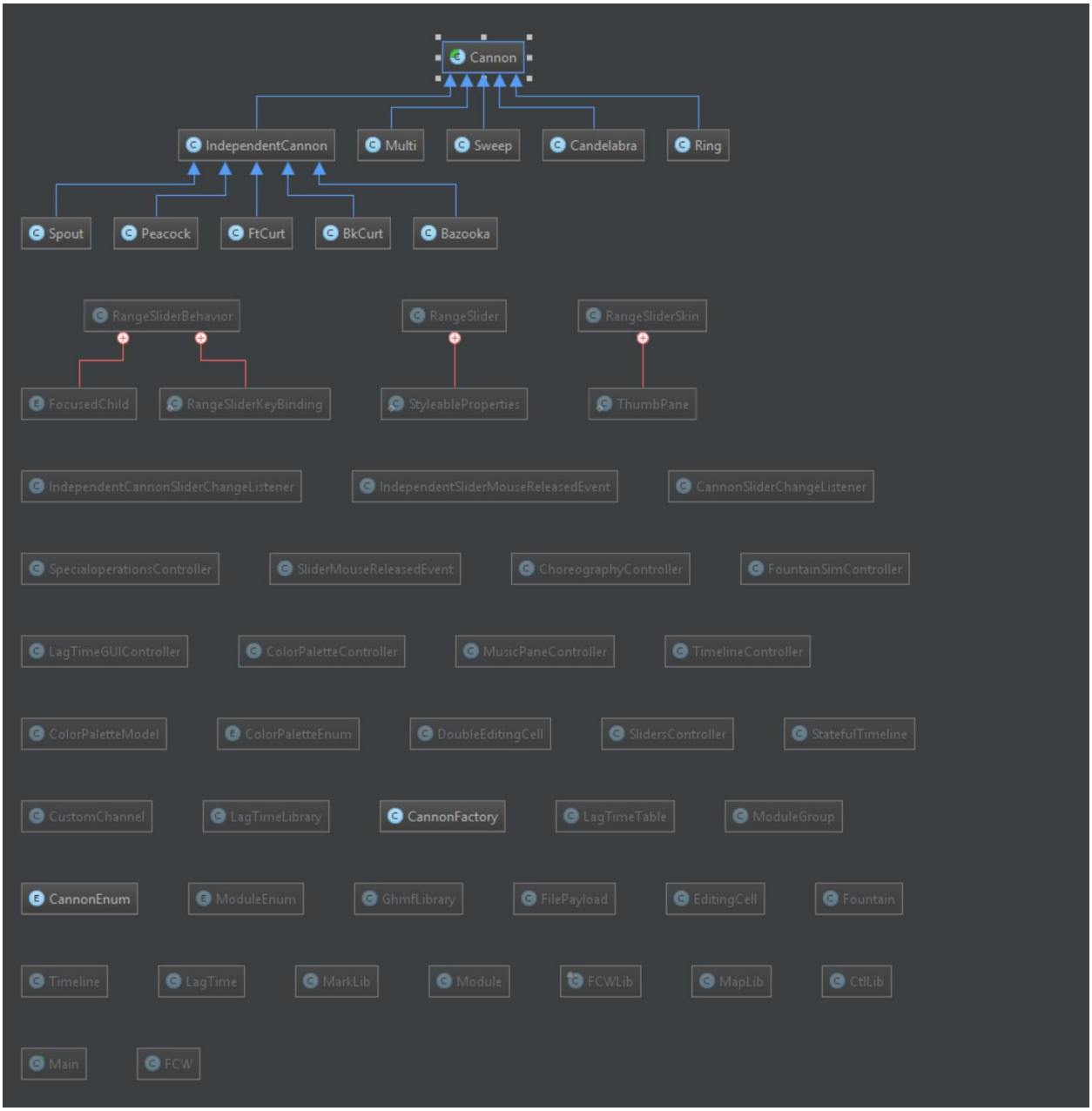
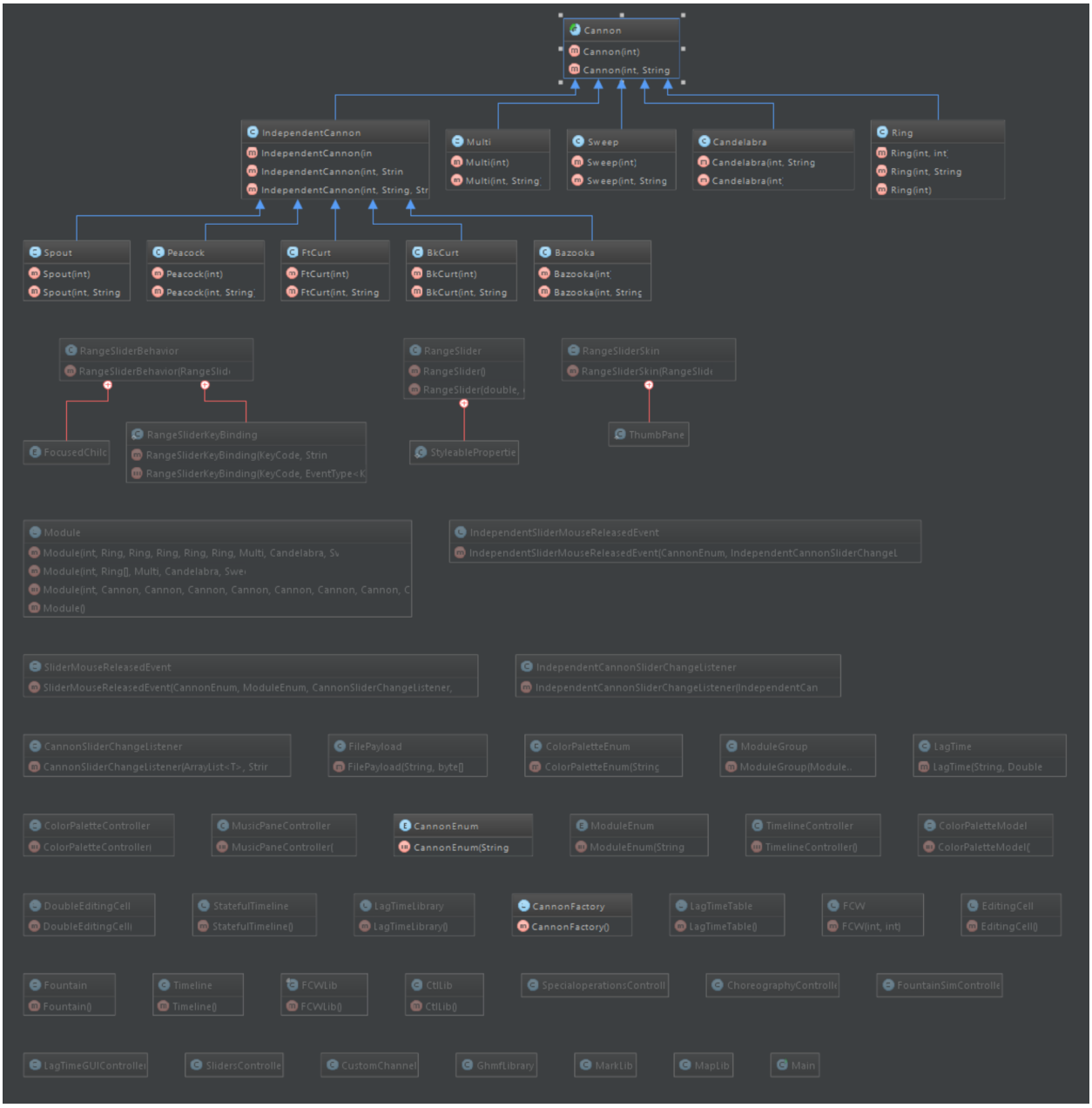
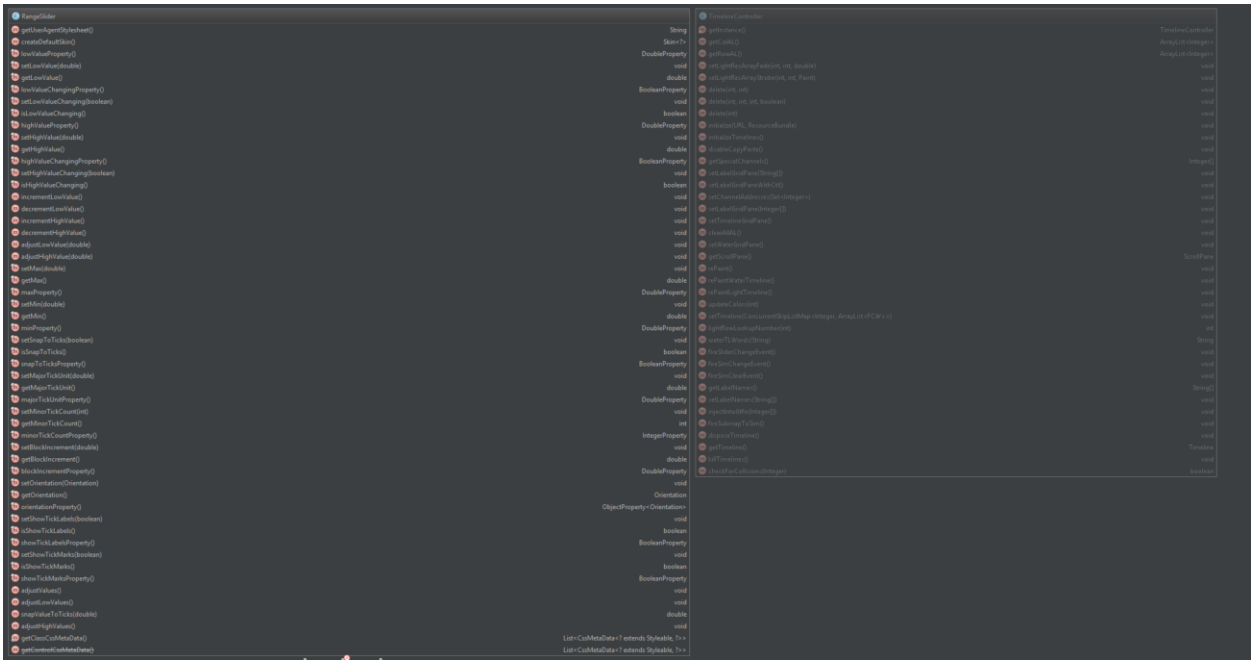
Methods and Inheritance

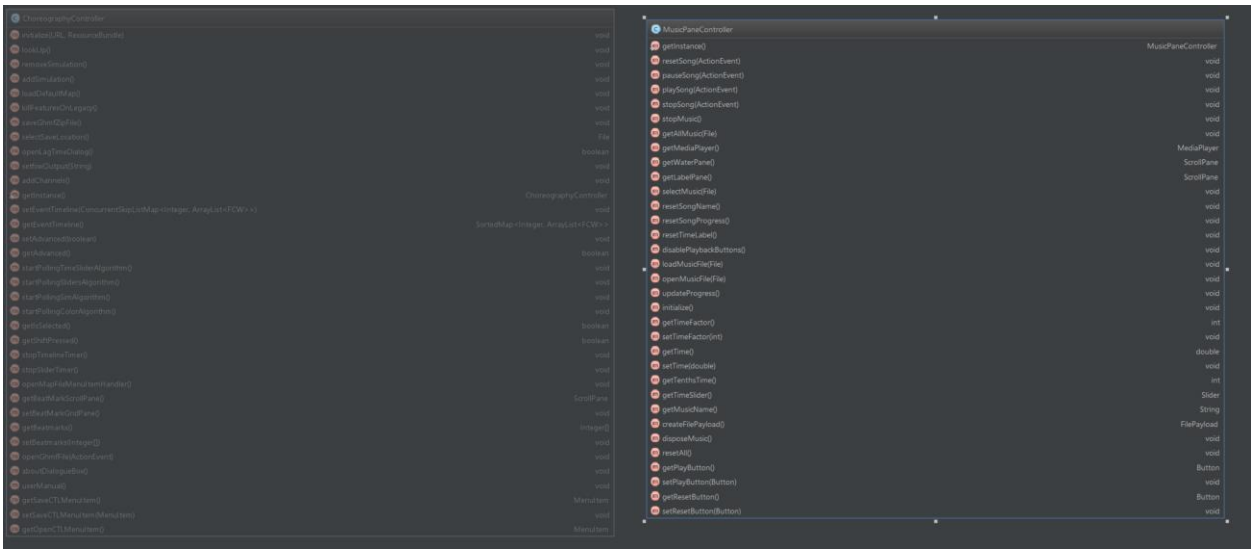


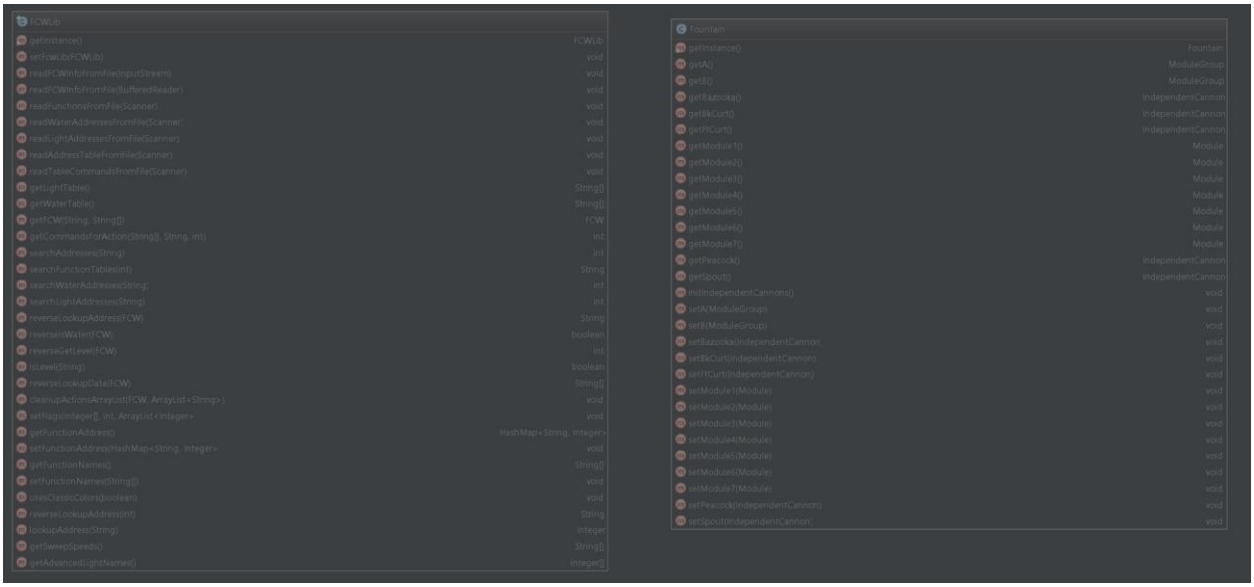
Method Constructors

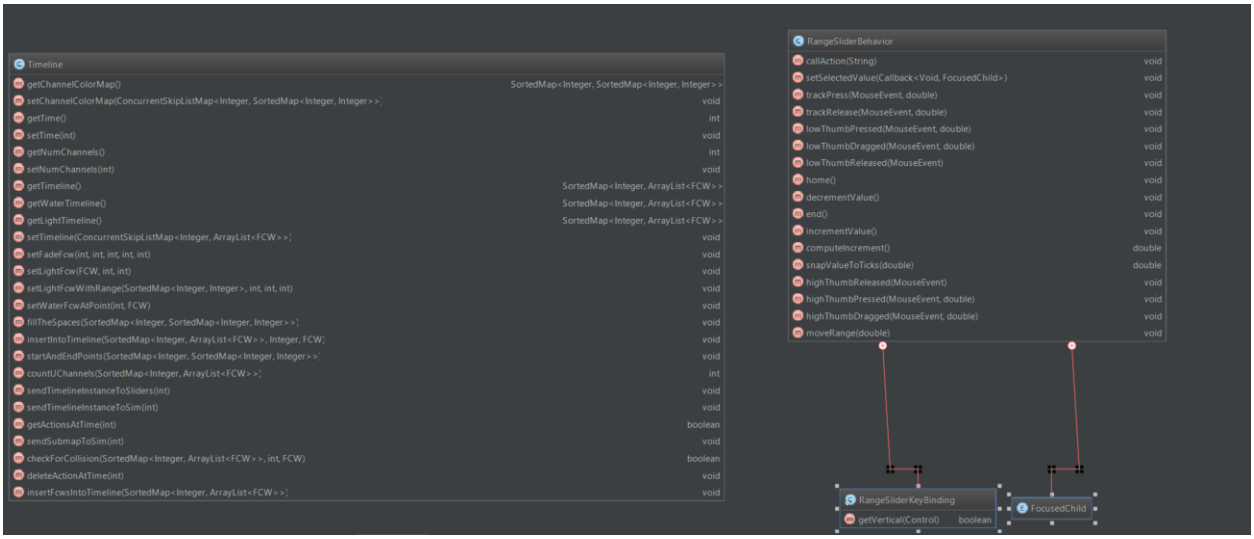


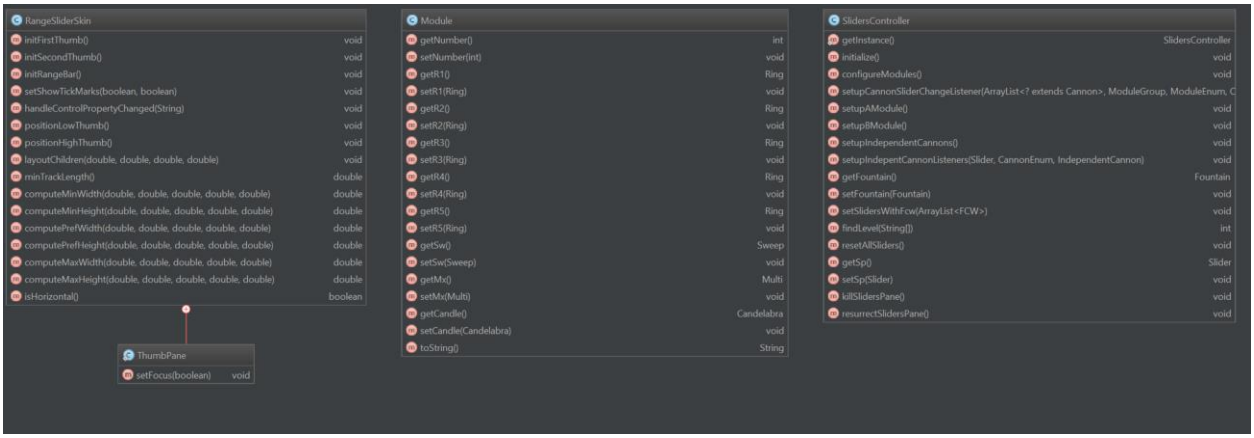
Methods of all Classes

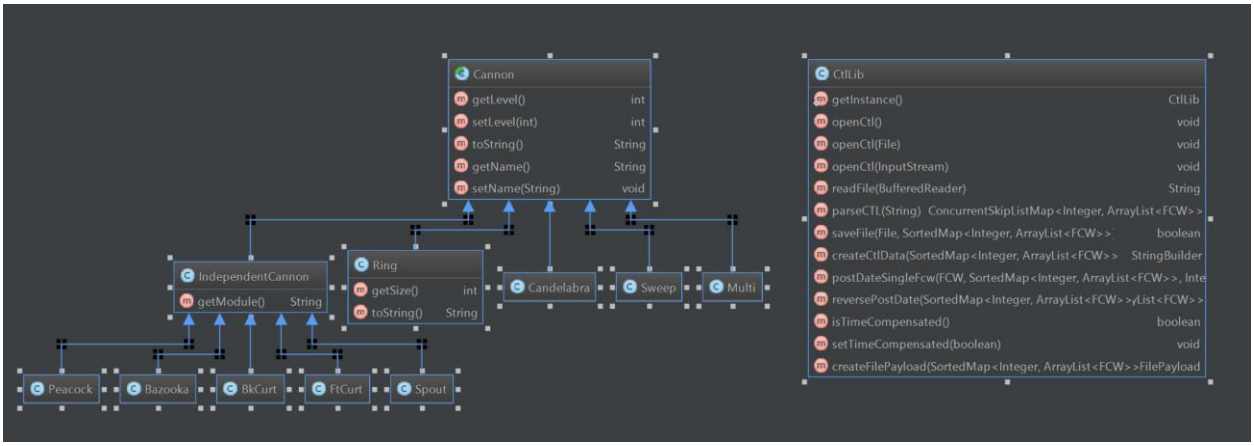


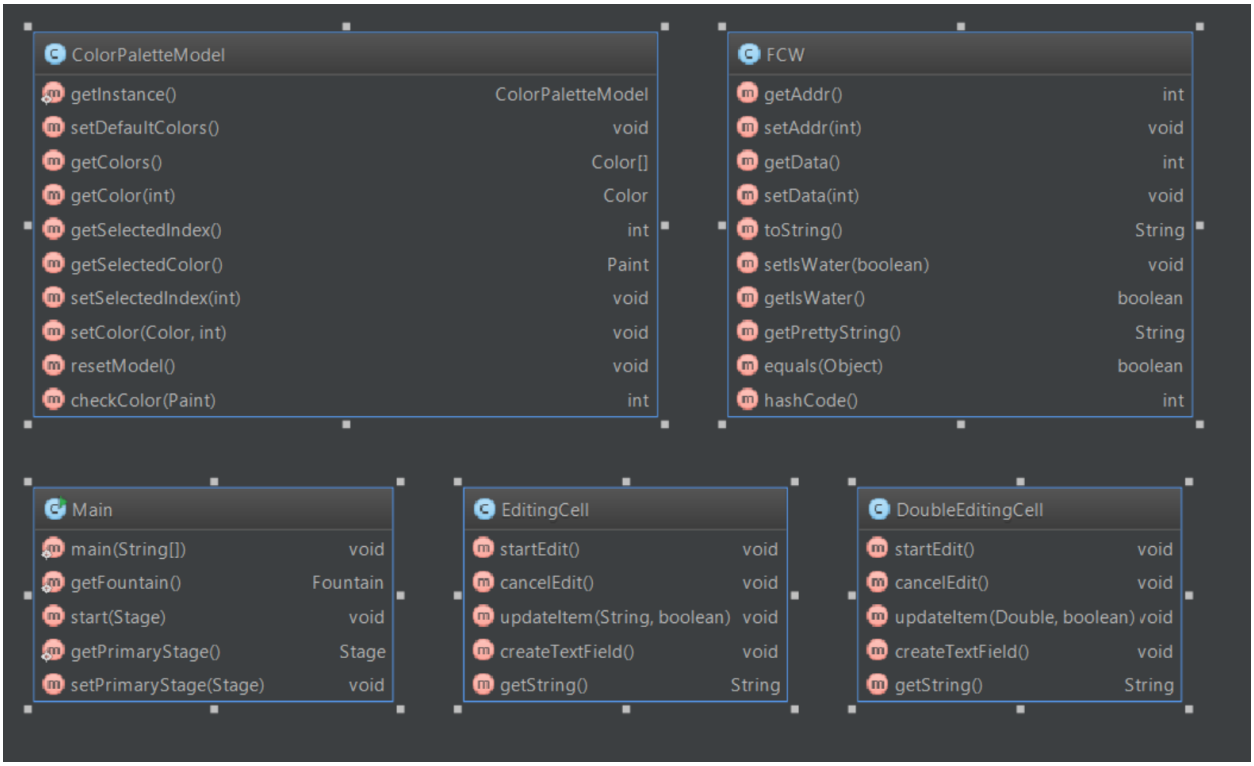


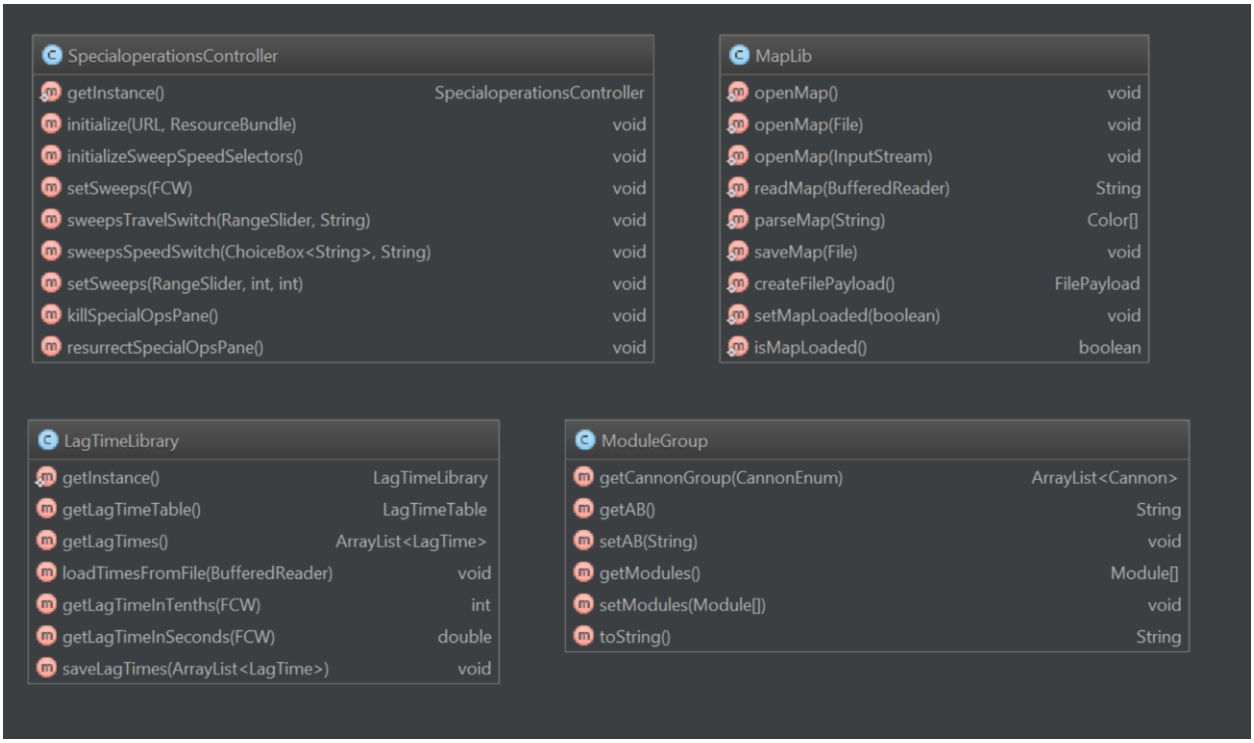

FCWLib
getInstance() FCWLib
setFcwLib(FCWLib) void
readFCWInfoFromFile(InputStream) void
readFCWInfoFromFile(BufferedReader) void
readFunctionsFromFile(Scanner) void
readWaterAddressesFromFile(Scanner) void
readLightAddressesFromFile(Scanner) void
readAddressTableFromFile(Scanner) void
readTableCommandsFromFile(Scanner) void
getLightTable() String[]
getWaterTable() String[]
getFCW(String, String[]) FCW
getCommandsForAction(String[], String, int) int
searchAddresses(String) int
searchFunctionTables(int) String
searchWaterAddresses(String) int
searchLightAddresses(String) int
reverseLookupAddress(FCW) String
reverseIsWater(FCW) boolean
reverseGetLevel(FCW) int
isLevel(String) boolean
reverseLookupData(FCW) String[]
cleanupActionsArrayList(FCW, ArrayList<String>) void
setFlags(Integer[], int, ArrayList<Integer> void
getFunctionAddress() HashMap<String, Integer>
setFunctionAddress(HashMap<String, Integer> void
getFunctionNames() String[]
setFunctionNames(String[]) void
usesClassicColors(boolean) void
reverseLookupAddress(int) String
lookupAddress(String) Integer
getSweepSpeeds() String[]
getAdvancedLightNames() Integer[]
Fountain
getInstance() Fountain
getA() ModuleGroup
getB() ModuleGroup
getBazooka() IndependentCannon
getBkCurt() IndependentCannon
getFtCurt() IndependentCannon
getModule1() Module
getModule2() Module
getModule3() Module
getModule4() Module
getModule5() Module
getModule6() Module
getModule7() Module
getPeacock() IndependentCannon
getSpout() IndependentCannon
initIndependentCannons() void
setA(ModuleGroup) void
setB(ModuleGroup) void
setBazooka(IndependentCannon) void
setBkCurt(IndependentCannon) void
setFtCurt(IndependentCannon) void
setModule1(Module) void
setModule2(Module) void
setModule3(Module) void
setModule4(Module) void
setModule5(Module) void
setModule6(Module) void
setModule7(Module) void
setPeacock(IndependentCannon) void
setSpout(IndependentCannon) void

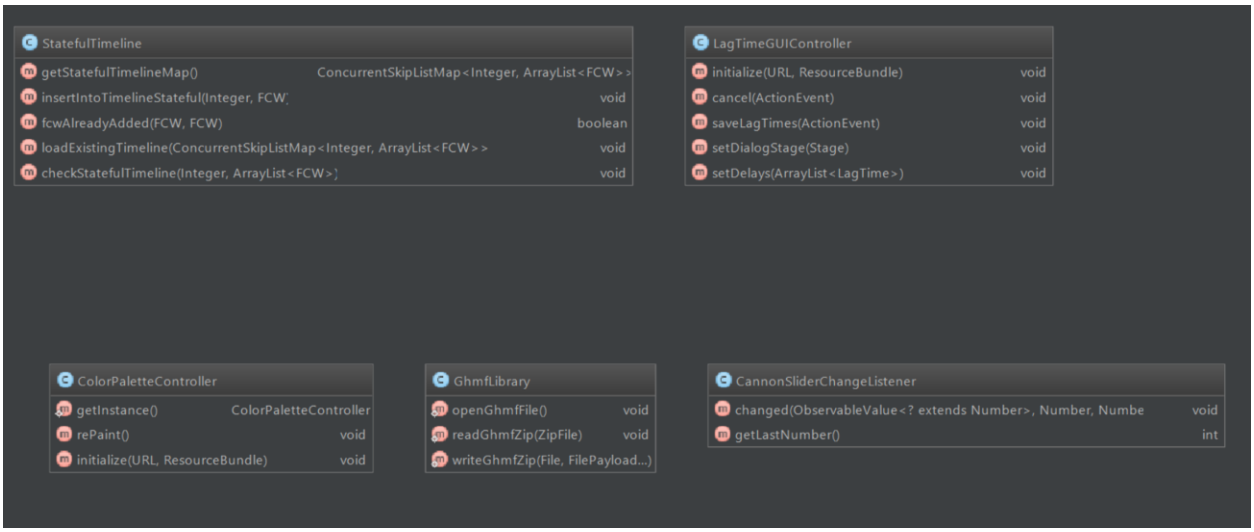

Timeline
getChannelColorMap() SortedMap<Integer, SortedMap<Integer, Integer>>
setChannelColorMap(ConcurrentSkipListMap<Integer, SortedMap<Integer, Integer>>, void
getTime() int
setTime(int) void
getNumChannels() int
setNumChannels(int) void
getTimeline() SortedMap<Integer, ArrayList<FCW>>
getWaterTimeline() SortedMap<Integer, ArrayList<FCW>>
getLightTimeline() SortedMap<Integer, ArrayList<FCW>>
setTimeline(ConcurrentSkipListMap<Integer, ArrayList<FCW>>) void
setFadeFcw(int, int, int, int, int) void
setLightFcw(FCW, int, int) void
setLightFcwWithRange(SortedMap<Integer, Integer>, int, int, int) void
setWaterFcwAtPoint(int, FCW) void
fillTheSpaces(SortedMap<Integer, SortedMap<Integer, Integer>>) void
insertIntoTimeline(SortedMap<Integer, ArrayList<FCW>>, Integer, FCW) void
startAndEndPoints(SortedMap<Integer, SortedMap<Integer, Integer>>) void
countUChannels(SortedMap<Integer, ArrayList<FCW>>) int
sendTimelineInstanceToSliders(int) void
sendTimelineInstanceToSim(int) void
getActionsAtTime(int) boolean
sendSubmapToSim(int) void
checkForCollision(SortedMap<Integer, ArrayList<FCW>>, int, FCW) boolean
deleteActionAtTime(int) void
insertFcwsIntoTimeline(SortedMap<Integer, ArrayList<FCW>>) void
RangeSliderBehavior
callAction(String) void
setSelectedValue(Callback<Void, FocusedChild>) void
trackPress(MouseEvent, double) void
trackRelease(MouseEvent, double) void
lowThumbPressed(MouseEvent, double) void
lowThumbDragged(MouseEvent, double) void
lowThumbReleased(MouseEvent) void
home() void
decrementValue() void
end() void
incrementValue() void
computeIncrement() double
snapValueToTicks(double) double
highThumbReleased(MouseEvent) void
highThumbPressed(MouseEvent, double) void
highThumbDragged(MouseEvent, double) void
moveRange(double) void
RangeSliderKeyBinding
getVertical(Control) boolean
FocusedChild

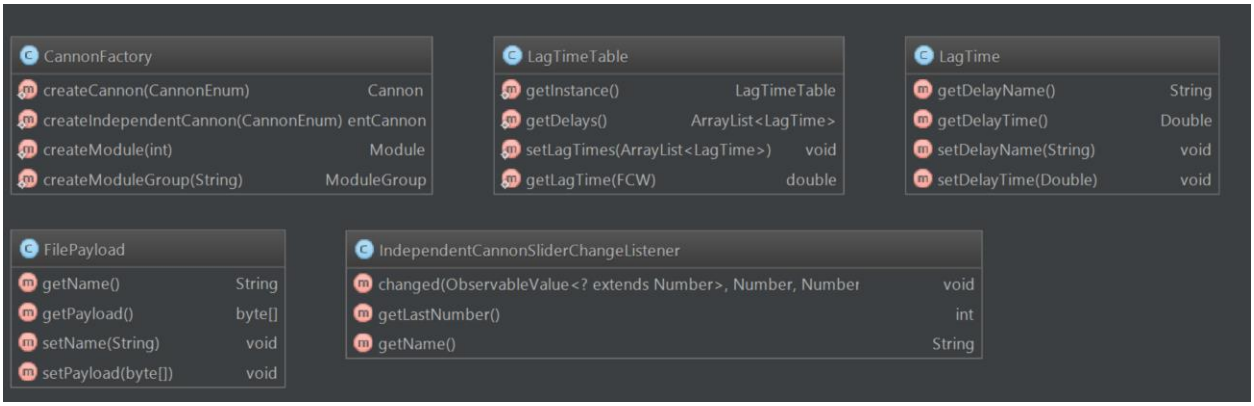

RangeSliderSkin
initFirstThumb() void
initSecondThumb() void
initRangeBar() void
setShowTickMarks(boolean, boolean) void
handleControlPropertyChanged(String) void
positionLowThumb() void
positionHighThumb() void
layoutChildren(double, double, double, double) void
minTrackLength() double
computeMinWidth(double, double, double, double, double) double
computeMinHeight(double, double, double, double, double) double
computePrefWidth(double, double, double, double, double) double
computePrefHeight(double, double, double, double, double) double
computeMaxWidth(double, double, double, double, double) double
computeMaxHeight(double, double, double, double, double) double
isHorizontal() boolean
ThumbPane
setFocus(boolean) void
Module
getNumber() int
setNumber(int) void
getR1() Ring
setR1(Ring) void
getR2() Ring
setR2(Ring) void
getR3() Ring
setR3(Ring) void
getR4() Ring
setR4(Ring) void
getR5() Ring
setR5(Ring) void
getSw() Sweep
setSw(Sweep) void
getMx() Multi
setMx(Multi) void
getCandle() Candelabra
setCandle(Candelabra) void
toString() String
SlidersController
getInstance() SlidersController
initialize() void
configureModules() void
setupCannonSliderChangeListener(ArrayList<? extends Cannon>, ModuleGroup, ModuleEnum, C
setupAModule() void
setupBModule() void
setupIndependentCannons() void
setupIndepentCannonListeners(Slider, CannonEnum, IndependentCannon) void
getFountain() Fountain
setFountain(Fountain) void
setSlidersWithFcw(ArrayList<FCW>) void
findLevel(String[]) int
resetAllSliders() void
getSp() Slider
setSp(Slider) void
killSlidersPane() void
resurrectSlidersPane() void

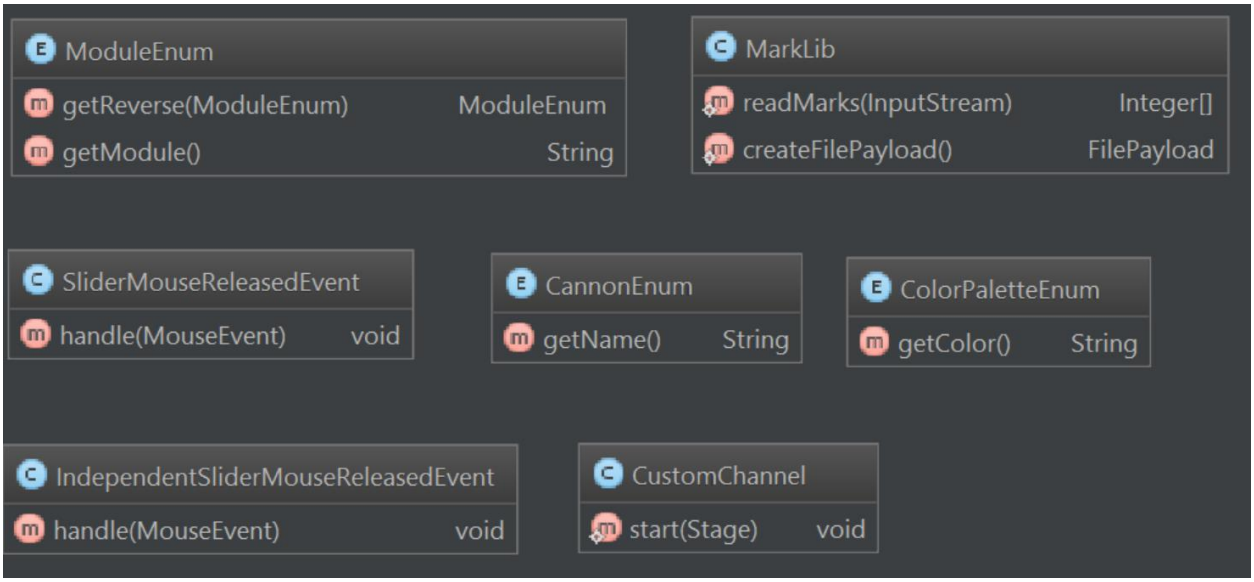

Cannon
getLevel() int
setLevel(int) int
toString() String
getName() String
setName(String) void
IndependentCannon
getModule() String
Ring
getSize() int
toString() String
Candelabra
Sweep
Multi
Peacock
Bazooka
BkCurt
FtCurt
Spout
CtlLib
getInstance() CtlLib
openCtl() void
openCtl(File) void
openCtl(InputStream) void
readFile(BufferedReader) String
parseCTL(String) ConcurrentSkipListMap<Integer, ArrayList<FCW>>
saveFile(File, SortedMap<Integer, ArrayList<FCW>>) boolean
createCtlData(SortedMap<Integer, ArrayList<FCW>> StringBuilder
postDateSingleFcw(FCW, SortedMap<Integer, ArrayList<FCW>>, Inte
reversePostDate(SortedMap<Integer, ArrayList<FCW>>yList<FCW>>
isTimeCompensated() boolean
setTimeCompensated(boolean) void
createFilePayload(SortedMap<Integer, ArrayList<FCW>>FilePayload


ColorPaletteModel
getInstance() ColorPaletteModel
setDefaultColors() void
getColors() Color[]
getColor(int) Color
getSelectedIndex() int
getSelectedColor() Paint
setSelectedIndex(int) void
setColor(Color, int) void
resetModel() void
checkColor(Paint) int
FCW
getAddr() int
setAddr(int) void
getData() int
setData(int) void
toString() String
setIsWater(boolean) void
getIsWater() boolean
getPrettyString() String
equals(Object) boolean
hashCode() int
Main
main(String[]) void
getFountain() Fountain
start(Stage) void
getPrimaryStage() Stage
setPrimaryStage(Stage) void
EditingCell
startEdit() void
cancelEdit() void
updateItem(String, boolean) void
createTextField() void
getString() String
DoubleEditingCell
startEdit() void
cancelEdit() void
updateItem(Double, boolean) void
createTextField() void
getString() String


SpecialoperationsController
getInstance() SpecialoperationsController
initialize(URL, ResourceBundle) void
initializeSweepSpeedSelectors() void
setSweeps(FCW) void
sweepsTravelSwitch(RangeSlider, String) void
sweepsSpeedSwitch(ChoiceBox<String>, String) void
setSweeps(RangeSlider, int, int) void
killSpecialOpsPane() void
resurrectSpecialOpsPane() void
MapLib
openMap() void
openMap(File) void
openMap(InputStream) void
readMap(BufferedReader) String
parseMap(String) Color[]
saveMap(File) void
createFilePayload() FilePayload
setMapLoaded(boolean) void
isMapLoaded() boolean
LagTimeLibrary
getInstance() LagTimeLibrary
getLagTimeTable() LagTimeTable
getLagTimes() ArrayList<LagTime>
loadTimesFromFile(BufferedReader) void
getLagTimeInTenths(FCW) int
getLagTimeInSeconds(FCW) double
saveLagTimes(ArrayList<LagTime>) void
ModuleGroup
getCannonGroup(CannonEnum) ArrayList<Cannon>
getAB() String
setAB(String) void
getModules() Module[]
setModules(Module[]) void
toString() String


StatefulTimeline
getStatefulTimelineMap() ConcurrentSkipListMap<Integer, ArrayList<FCW>>
insertIntoTimelineStateful(Integer, FCW)
fcwAlreadyAdded(FCW, FCW) boolean
loadExistingTimeline(ConcurrentSkipListMap<Integer, ArrayList<FCW>>
checkStatefulTimeline(Integer, ArrayList<FCW>)
LagTimeGUIController
initialize(URL, ResourceBundle) void
cancel(ActionEvent) void
saveLagTimes(ActionEvent) void
setDialogStage(Stage) void
setDelays(ArrayList<LagTime>) void
ColorPaletteController
getInstance() ColorPaletteController
rePaint() void
initialize(URL, ResourceBundle) void
GhmfLibrary
openGhmfFile() void
readGhmfZip(ZipFile) void
writeGhmfZip(File, FilePayload...)
CannonSliderChangeListener
changed(ObservableValue<? extends Number>, Number, Numbe void
getLastNumber() int


CannonFactory
createCannon(CannonEnum) Cannon
createIndependentCannon(CannonEnum) entCannon
createModule(int) Module
createModuleGroup(String) ModuleGroup
LagTimeTable
getInstance() LagTimeTable
getDelays() ArrayList<LagTime>
setLagTimes(ArrayList<LagTime>) void
getLagTime(FCW) double
LagTime
getDelayName() String
getDelayTime() Double
setDelayName(String) void
setDelayTime(Double) void
FilePayload
getName() String
getPayload() byte[]
setName(String) void
setPayload(byte[]) void
IndependentCannonSliderChangeListener
changed(ObservableValue<? extends Number>, Number, Number void
getLastNumber() int
getName() String


ModuleEnum
getReverse(ModuleEnum) ModuleEnum
getModule() String
MarkLib
readMarks(InputStream) Integer[]
createFilePayload() FilePayload
SliderMouseReleasedEvent
handle(MouseEvent) void
CannonEnum
getName() String
ColorPaletteEnum
getColor() String
IndependentSliderMouseReleasedEvent
handle(MouseEvent) void
CustomChannel
start(Stage) void