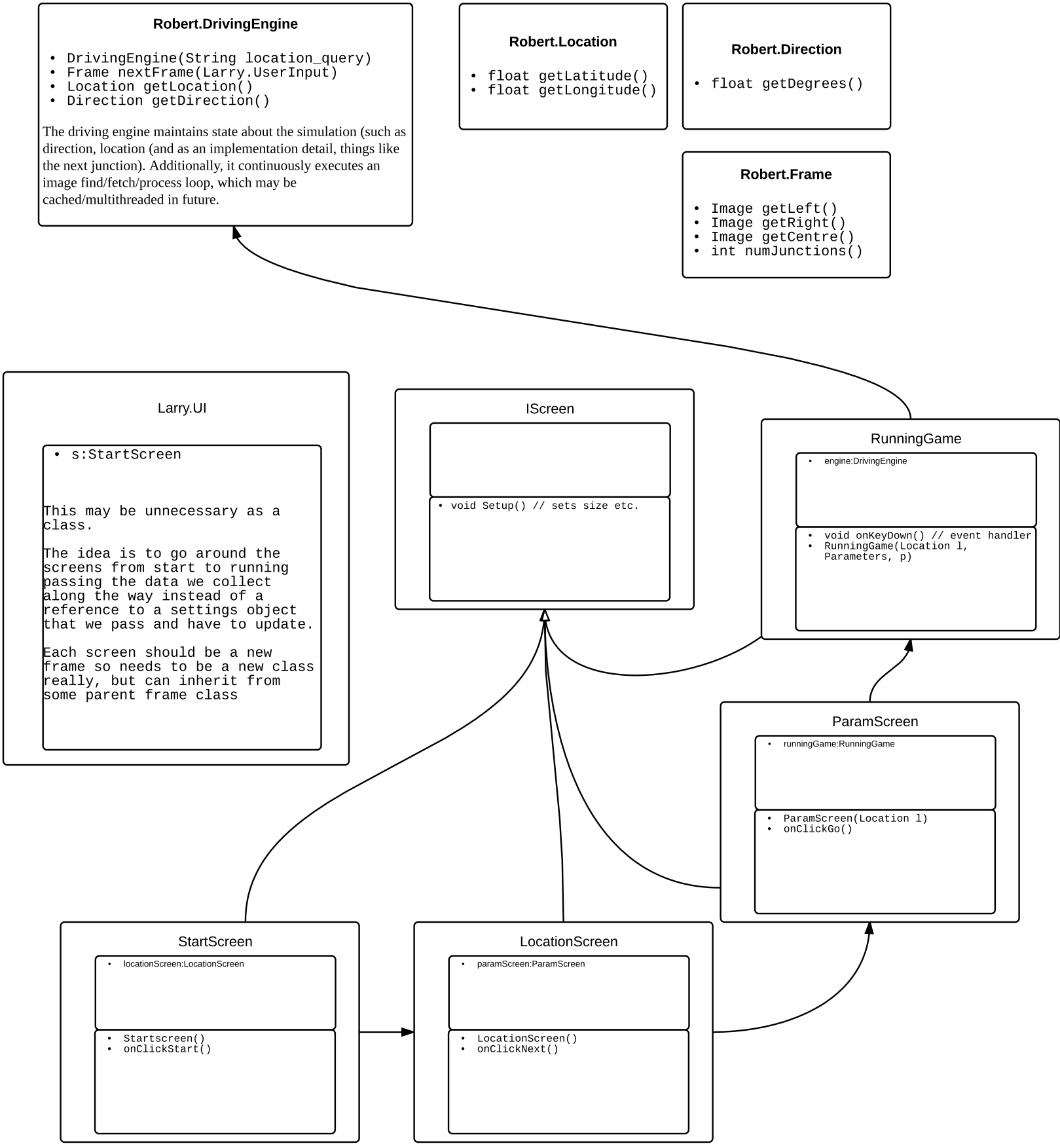
Robert.DrivingEngine
• DrivingEngine(String location_query)
• Frame nextFrame(Larry.UserInput)
• Location getLocation()
• Direction getDirection()
The driving engine maintains state about the simulation (such as direction, location (and as an implementation detail, things like the next junction). Additionally, it continuously executes an image find/fetch/process loop, which may be cached/multithreaded in future.
Robert.Location
• float getLatitude()
• float getLongitude()
Robert.Direction
• float getDegrees()
Robert.Frame
• Image getLeft()
• Image getRight()
• Image getCentre()
• int numJunctions()
Larry.UI
• s:StartScreen
This may be unnecessary as a class.
The idea is to go around the screens from start to running passing the data we collect along the way instead of a reference to a settings object that we pass and have to update.
Each screen should be a new frame so needs to be a new class really, but can inherit from some parent frame class
IScreen
• void Setup() // sets size etc.
RunningGame
• engine:DrivingEngine
• void onKeyDown() // event handler
• RunningGame(Location l, Parameters, p)
ParamScreen
• runningGame:RunningGame
• ParamScreen(Location l)
• onClickGo()
StartScreen
• locationScreen:LocationScreen
• Startscreen()
• onClickStart()
LocationScreen
• paramScreen:ParamScreen
• LocationScreen()
• onClickNext()