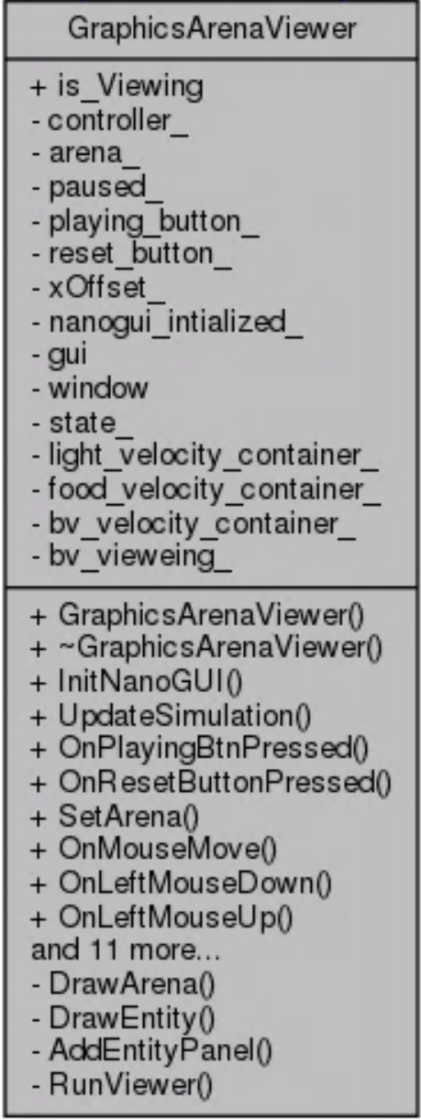
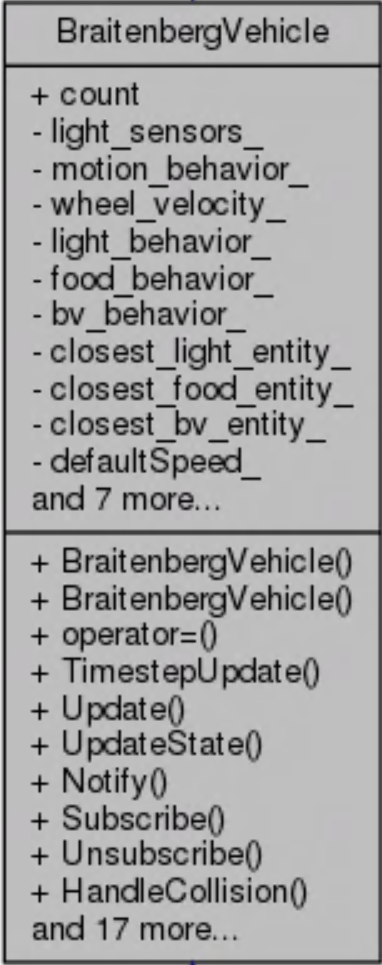
Moti Begna

CSCI 3081

4/5/2019

Iteration 2 Preliminary 2

**Doxy Generated UML for Observer Pattern**

My implementation of the Observer pattern involved adding necessary methods and elements to the BraiternbergVehicle and GraphicsArenaViewer classes in order to make them a Subject and Observer respectively.

+ Update() is not shown

Notify() calls Update()

Uses Unsubscribe()

Uses Subscribe()

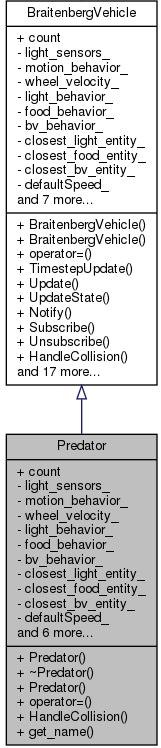
**Coode Snippet of GraphicArenaViewer’s Update Method**

|  |
| --- |
| void GraphicsArenaViewer::Update(const State& state){ |
| //setting GAV's state variable |
| state\_.light\_l = state.light\_l; |
| state\_.light\_r = state.light\_r; |
| state\_.food\_l = state.food\_l; |
| state\_.food\_r = state.food\_r; |
| state\_.bv\_l = state.bv\_l; |
| state\_.bv\_r = state.bv\_r; |
|  |
| //Converting state values |
| std::stringstream ll; |
| ll << std::setprecision(2) << state\_.light\_l; |
| std::stringstream lr; |
| lr << std::setprecision(2) << state\_.light\_r; |
| std::stringstream fl; |
| fl << std::setprecision(2) << state\_.food\_l; |
| std::stringstream fr; |
| fr << std::setprecision(2) << state\_.food\_r; |
| std::stringstream bl; |
| bl << std::setprecision(2) << state\_.bv\_l; |
| std::stringstream br; |
| br << std::setprecision(2) << state\_.bv\_r; |
|  |
| //setting values displayed on GUI |
| light\_velocity\_container\_[0]->setValue(std::string (ll.str())); |
| light\_velocity\_container\_[1]->setValue(std::string (lr.str())); |
| food\_velocity\_container\_[0]->setValue(std::string (fl.str())); |
| food\_velocity\_container\_[1]->setValue(std::string (fr.str())); |
| bv\_velocity\_container\_[0]->setValue(std::string (bl.str())); |
| bv\_velocity\_container\_[1]->setValue(std::string (br.str())); |
| } |

**Coode Snippet of BraitenbergVehicle’s Notify Method**

|  |
| --- |
| void BraitenbergVehicle::Notify(){ |
| for (auto obs : observers\_) { |
| obs->Update(state\_); |
| } |
| } |

**Doxy Generated UML for Predator Class**

****