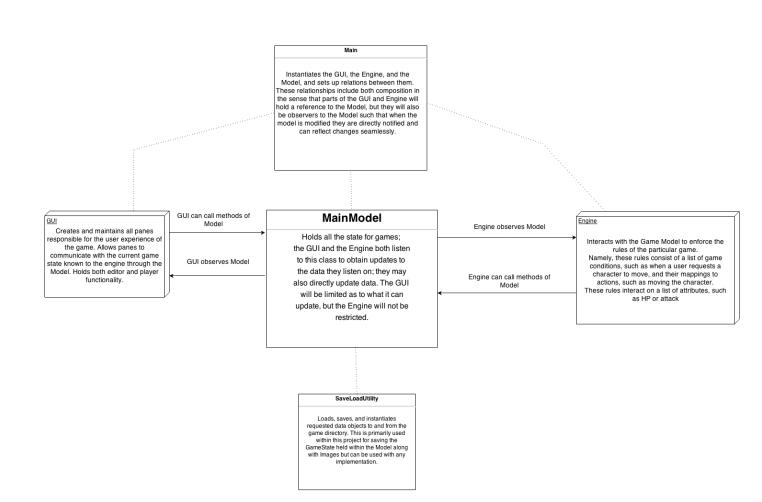
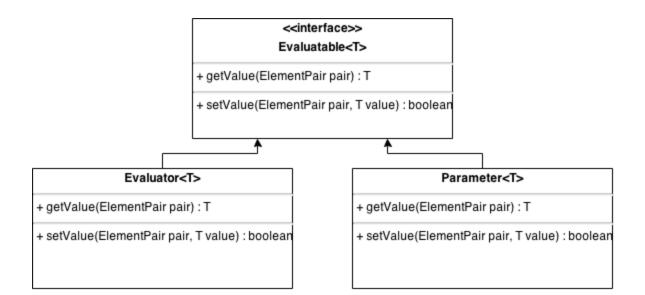
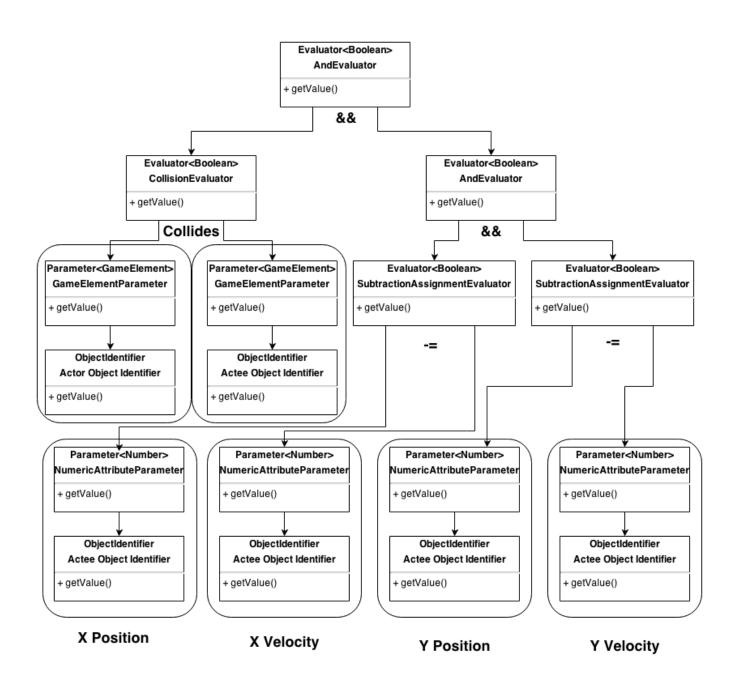
# VoogaSalad Demo

The Exception







	RuleLoader <t></t>		
	+ schemaPath: String + rules: List <t></t>		
	+ generateRules(String jsonPath, + validate(String jsonPath, boolea + getRules(): List <t></t>		
	A	$\overline{\nabla}$	
	Extends	Extends	
GrammarRuleLoade	er <lgrammarrule></lgrammarrule>	TokenR	uleLoader <ltokenrule></ltokenrule>

# **Token Rule Definition**

# **Grammar Rule Definition**

```
{
          "pattern": [
                 {
                       "name": "parameter",
                       "level": 0
                },
                 {
                       "name": "evaluator",
                       "level": 1
                },
                       "name": "parameter",
                       "level": 0
                 }
          ],
          "additional": [
                 {
                       "name": "parameter",
                       "level: "2"
                }
   }
    [ "parameter", "evaluator", "parameter"]
         "parameter"
         "evaluator"
"parameter"
                   "parameter"
```

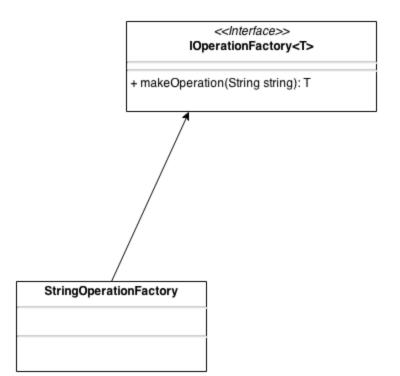
# # tokenize(Reader input): List<IToken> + loadTokenRules(List<ITokenRule rules): void</pre> Tokenizer + rules: List<ITokenRule>

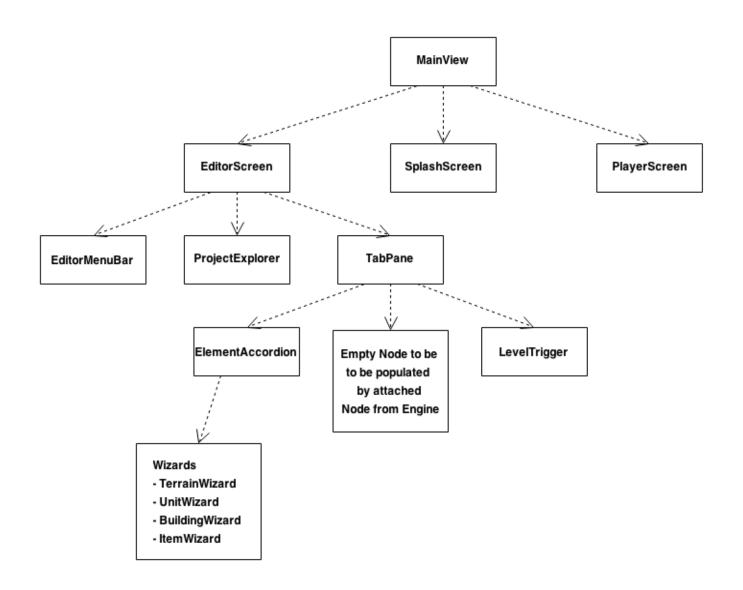
### <<Interface>> IParser<T>

- + parse(List<IToken> tokens): ISyntaxNode<T> + loadGrammar(List<IGrammarRule<T>> rules): void

### Parser<T>

- + rules: List<lGrammarRule<T>>
- + operationFactory: IOperationFactory<T>
- +tokensToNodes(List<IToken> tokens): List<ISyntaxNode<T>>





### GUIController (abstract)

Node getRoot() void initialize()

# GUIContainer (abstract) implements GUIController

void setModel(MainModel m)

- void attachChildContainers(GUIContainer ... children)
- void clearChildContainers()
- MainModel getMainModel()
- abstract void update()

### Wizard (abstract) implements GUIController

- Node getRoot()
- void initialize()
- void setSubmit(Consumer<WizardData> data)
- void addToData(String key, String value)
- abstract boolean checkCanSave()
- abstract void updateData()
- WizardData getWizardData()

### WizardData

Map<String, String> myData List<WizardData> myWizards String name

- void addDataPair(String key, String value)
- void addWizardData(WizardData wizData)
- Map<String,String> getData()
- List<WizardData> getWizards()