CRC Models

List of CRC Cards

Entities:

- Expression
    - NumberExpression
    - FunctionExpression
    - VariableExpression
    - OperatorExpression
- Axes/Space
- Viewpoint

Controllers/Presenters:

- ExpressionReader
- Renderer

Use Cases:

- ExpressionCreator
- RenderUseCase
- ViewpointEditor

UserInterface

- CommandLineRunner
- GUIRunner (for later)

Expression (Abstract)

Responsibilities:

- Stores various items (Number, Variable, Operator, etc.)
- Provides method signature for evaluate method (to be overridden by subclasses)

Collaborators:

- NumberExpression
- FunctionExpression
- VariableExpression
- OperatorExpression
- ExpressionCreator

Expression

NumberExpression

Responsibilities:

- Extends Expression class
- Handles Number inputs
- Stores numbers (calls the constructor for the Parent class Expression to store the item and initiate the item variable)
- Evaluates its expression (override from abstract method)

Collaborators:

- ExpressionCreator

Expression

OperatorExpression

Responsibilities:

- Input to initializer: an operation and 2 expressions
- Stores the operation and 2 expressions
- Overrides evaluate from parent class Expression.
- Can evaluate the expression for appropriate operations (addition, subtraction, multiplication, division, exponents). Returns type double.

Collaborators:

- ExpressionCreator

Expression

VariableExpression

- Extends Expression class
- Accepts and stores strings “x”, “y”, and/or “z”.
- Overrides evaluate by checking a map to determine which value has been assigned to it.

Collaborators:

Expression (abstract parent)

FunctionExpression (abstract)

Responsibilities:

- Extends Expression class
- For functions like cos, sin, sqrt, etc.
- Handles multiple inputs of any kind (Number, Operator, etc.)
- Stores the name of the function (calls the constructor for the Parent class Expression to store the item and initiate the item variable)
- Stores an Expression for how a functions should be evaluated (e.g. if a user defines f(x) = x^2 + 5, then a FunctionExpression will be created that is named f and stores the expression “x^2 + 5”.)

Collaborators:

- Expression (Abstract Parent Class)
- ExpressionCreator

Axes/Space

Responsibilities:

- The Euclidean space in which graphs are graphed
- Handles scale of the axes
- Stores a list of all expressions that have been input

Interacts with:

- Expression

Viewpoint

- Stores attributes of viewpoint e.g. zoom level, scale, orientation, etc.

Interacts with:

Renderer

ViewpointEditor

Responsibilities:

- Changes attributes of viewpoint e.g. zoom level, scale, orientation, etc.

Interacts with:

Viewpoint

RendererUseCase

ExpressionCreator

Responsibilities:

Converts a list into an Expression, where the elements corresponds to a ‘unit’ of information that implements Abstract Syntax Trees

Inputs are of the format that ExpressionReader ‘spits’ out

Interacts with:

Expression (and its subclasses)

ExpressionReader

Responsibilities:

Takes string input and converts to a list that expression creator can use to create an expression with

E.g.

“x^2 + 5” -> [“x”, “^”, “3”, “+”, “5”]

“cos(x + y)” -> [“cos”, “(“, “x”, “+”, “y”, “)”]

Interacts with:

ExpressionCreator

UserInterface (when upgrading to graphical interface)

RendererUseCase

Responsibilities:

“Main method”, calls renderer with axes + viewpoint etc

Is bridge between core renderer(s) and UI layers

Pass parameters to Renderer

Interacts with:

- Viewpoint
- Renderer
- Axes

Renderer (interface)

Responsibilities:

- Generates image from pixel array
- Save image

Interacts with:

- RendererUseCase

UserInterface

Responsibilities:

- Accept user input and passes it to ExpressionReader
- Although we will be starting out with a command line interface, we ideally want to transition to a graphical user interface

Interacts with:

ExpressionReader