

Module Interface Specification for Digital Twin Forest

Team 8, Forest Mirror

Tingyu Shi

Jiacheng Wu

Junhong Chen

Yichen Jiang

Bowen Zhang

January 18, 2023

1 Revision History

Date	Version	Notes
Jan 14	1.0	First Version

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at [here](#).

symbol	description
AC	Anticipated Change
DAG	Directed Acyclic Graph
M	Module
MG	Module Guide
OS	Operating System
R	Requirement
FR	Functional Requirement
NFR	Non-Functional Requirement
SC	Scientific Computing
SRS	Software Requirements Specification
Digital Twin Forest	Explanation of program name
UC	Unlikely Change
MVC	Model, Viewer, Controller
GUI	Graphical User Interface
LAI	Leaf Area Index
DBH	Diameter at breast height

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	2
6	MIS of Forest Trees (M1)	6
6.1	Module	6
6.2	Uses	6
6.3	Syntax	6
6.3.1	Exported Constants	6
6.3.2	Exported Access Programs	6
6.4	Semantics	6
6.4.1	State Variables	6
6.4.2	Environment Variables	6
6.4.3	Assumptions	6
6.4.4	Access Routine Semantics	6
6.4.5	Local Functions	7
7	MIS of Forest Sky (M2)	8
7.1	Module	8
7.2	Uses	8
7.3	Syntax	8
7.3.1	Exported Constants	8
7.3.2	Exported Access Programs	8
7.4	Semantics	8
7.4.1	State Variables	8
7.4.2	Environment Variables	8
7.4.3	Assumptions	8
7.4.4	Access Routine Semantics	8
7.4.5	Local Functions	8
8	MIS of Forest Terrain (M3)	9
8.1	Module	9
8.2	Uses	9
8.3	Syntax	9
8.3.1	Exported Constants	9
8.3.2	Exported Access Programs	9

8.4	Semantics	9
8.4.1	State Variables	9
8.4.2	Environment Variables	9
8.4.3	Assumptions	9
8.4.4	Access Routine Semantics	9
8.4.5	Local Functions	9
9	MIS of Red Pine (M4)	10
9.1	Module	10
9.2	Uses	10
9.3	Syntax	10
9.3.1	Exported Constants	10
9.3.2	Exported Access Programs	10
9.4	Semantics	10
9.4.1	State Variables	10
9.4.2	Environment Variables	10
9.4.3	Assumptions	11
9.4.4	Access Routine Semantics	11
9.4.5	Local Functions	12
10	MIS of Oak (M5)	13
10.1	Module	13
10.2	Uses	13
10.3	Syntax	13
10.3.1	Exported Constants	13
10.3.2	Exported Access Programs	13
10.4	Semantics	13
10.4.1	State Variables	13
10.4.2	Environment Variables	13
10.4.3	Assumptions	14
10.4.4	Access Routine Semantics	14
10.4.5	Local Functions	15
11	MIS of Beech (M6)	16
11.1	Module	16
11.2	Uses	16
11.3	Syntax	16
11.3.1	Exported Constants	16
11.3.2	Exported Access Programs	16
11.4	Semantics	16
11.4.1	State Variables	16
11.4.2	Environment Variables	16
11.4.3	Assumptions	17

11.4.4	Access Routine Semantics	17
11.4.5	Local Functions	18
12	MIS of Birch (M7)	19
12.1	Module	19
12.2	Uses	19
12.3	Syntax	19
12.3.1	Exported Constants	19
12.3.2	Exported Access Programs	19
12.4	Semantics	19
12.4.1	State Variables	19
12.4.2	Environment Variables	19
12.4.3	Assumptions	20
12.4.4	Access Routine Semantics	20
12.4.5	Local Functions	21
13	MIS of White Pine (M8)	22
13.1	Module	22
13.2	Uses	22
13.3	Syntax	22
13.3.1	Exported Constants	22
13.3.2	Exported Access Programs	22
13.4	Semantics	22
13.4.1	State Variables	22
13.4.2	Environment Variables	22
13.4.3	Assumptions	23
13.4.4	Access Routine Semantics	23
13.4.5	Local Functions	24
14	MIS of Red Maple (M9)	25
14.1	Module	25
14.2	Uses	25
14.3	Syntax	25
14.3.1	Exported Constants	25
14.3.2	Exported Access Programs	25
14.4	Semantics	25
14.4.1	State Variables	25
14.4.2	Environment Variables	25
14.4.3	Assumptions	26
14.4.4	Access Routine Semantics	26
14.4.5	Local Functions	27

15 MIS of Red Oak (M10)	28
15.1 Module	28
15.2 Uses	28
15.3 Syntax	28
15.3.1 Exported Constants	28
15.3.2 Exported Access Programs	28
15.4 Semantics	28
15.4.1 State Variables	28
15.4.2 Environment Variables	28
15.4.3 Assumptions	29
15.4.4 Access Routine Semantics	29
15.4.5 Local Functions	30
16 MIS of Environmental Data (M11)	31
16.1 Module	31
16.2 Uses	31
16.3 Syntax	31
16.3.1 Exported Constants	31
16.3.2 Exported Access Programs	31
16.4 Semantics	31
16.4.1 State Variables	31
16.4.2 Environment Variables	31
16.4.3 Assumptions	32
16.4.4 Access Routine Semantics	32
16.4.5 Local Functions	33
17 MIS of Plot Data (M12)	34
17.1 Module	34
17.2 Uses	34
17.3 Syntax	34
17.3.1 Exported Constants	34
17.3.2 Exported Access Programs	34
17.4 Semantics	35
17.4.1 State Variables	35
17.4.2 Environment Variables	35
17.4.3 Assumptions	35
17.4.4 Access Routine Semantics	35
17.4.5 Local Functions	38
18 MIS of First Person Player (M13)	39
18.1 Module	39
18.2 Uses	39
18.3 Syntax	39

18.4 Semantics	39
19 MIS of Json File (M14)	40
19.1 Module	40
19.2 Local Type	40
19.3 State Variables	40
19.4 Example	40
19.5 Local Functions	42
20 MIS of Main Page (M15)	43
20.1 Module	43
20.2 Uses	43
20.3 Syntax	43
20.3.1 Exported Constants	43
20.3.2 Exported Access Programs	43
20.4 Semantics	43
20.4.1 State Variables	43
20.4.2 Environment Variables	43
20.4.3 Assumptions	43
20.4.4 Access Routine Semantics	43
20.4.5 Local Functions	43
21 MIS of Start Button (M16)	44
21.1 Module	44
21.2 Uses	44
21.3 Syntax	44
21.3.1 Exported Constants	44
21.3.2 Exported Access Programs	44
21.4 Semantics	44
21.4.1 State Variables	44
21.4.2 Environment Variables	44
21.4.3 Assumptions	44
21.4.4 Access Routine Semantics	44
21.4.5 Local Functions	44
22 MIS of Instruction Button (M17)	45
22.1 Module	45
22.2 Uses	45
22.3 Syntax	45
22.3.1 Exported Constants	45
22.3.2 Exported Access Programs	45
22.4 Semantics	45
22.4.1 State Variables	45

22.4.2	Environment Variables	45
22.4.3	Assumptions	45
22.4.4	Access Routine Semantics	45
22.4.5	Local Functions	45
23	MIS of Contact Us Button (M18)	46
23.1	Module	46
23.2	Uses	46
23.3	Syntax	46
23.3.1	Exported Constants	46
23.3.2	Exported Access Programs	46
23.4	Semantics	46
23.4.1	State Variables	46
23.4.2	Environment Variables	46
23.4.3	Assumptions	46
23.4.4	Access Routine Semantics	46
23.4.5	Local Functions	46
24	MIS of Quit Button (M19)	47
24.1	Module	47
24.2	Uses	47
24.3	Syntax	47
24.3.1	Exported Constants	47
24.3.2	Exported Access Programs	47
24.4	Semantics	47
24.4.1	State Variables	47
24.4.2	Environment Variables	47
24.4.3	Assumptions	47
24.4.4	Access Routine Semantics	47
24.4.5	Local Functions	47
25	MIS of Instruction Page (M20)	48
25.1	Module	48
25.2	Uses	48
25.3	Syntax	48
25.3.1	Exported Constants	48
25.3.2	Exported Access Programs	48
25.4	Semantics	48
25.4.1	State Variables	48
25.4.2	Environment Variables	48
25.4.3	Assumptions	48
25.4.4	Access Routine Semantics	48
25.4.5	Local Functions	48

26 MIS of Contact Us Page (M21)	49
26.1 Module	49
26.2 Uses	49
26.3 Syntax	49
26.3.1 Exported Constants	49
26.3.2 Exported Access Programs	49
26.4 Semantics	49
26.4.1 State Variables	49
26.4.2 Environment Variables	49
26.4.3 Assumptions	49
26.4.4 Access Routine Semantics	49
26.4.5 Local Functions	49
27 MIS of Back Button (M22)	50
27.1 Module	50
27.2 Uses	50
27.3 Syntax	50
27.3.1 Exported Constants	50
27.3.2 Exported Access Programs	50
27.4 Semantics	50
27.4.1 State Variables	50
27.4.2 Environment Variables	50
27.4.3 Assumptions	50
27.4.4 Access Routine Semantics	50
27.4.5 Local Functions	50
28 MIS of Update Data Page (M23)	51
28.1 Module	51
28.2 Uses	51
28.3 Syntax	51
28.3.1 Exported Constants	51
28.3.2 Exported Access Programs	51
28.4 Semantics	51
28.4.1 State Variables	51
28.4.2 Environment Variables	51
28.4.3 Assumptions	51
28.4.4 Access Routine Semantics	51
28.4.5 Local Functions	51
29 MIS of Environmental Data Selection Button (M24)	52
29.1 Module	52
29.2 Uses	52
29.3 Syntax	52

29.3.1	Exported Constants	52
29.3.2	Exported Access Programs	52
29.4	Semantics	52
29.4.1	State Variables	52
29.4.2	Environment Variables	52
29.4.3	Assumptions	52
29.4.4	Access Routine Semantics	52
29.4.5	Local Functions	52
30	MIS of Data Type Selection Button (M25)	53
30.1	Module	53
30.2	Uses	53
30.3	Syntax	53
30.3.1	Exported Constants	53
30.3.2	Exported Access Programs	53
30.4	Semantics	53
30.4.1	State Variables	53
30.4.2	Environment Variables	53
30.4.3	Assumptions	53
30.4.4	Access Routine Semantics	53
30.4.5	Local Functions	53
31	MIS of New Data Input Box (M26)	54
31.1	Module	54
31.2	Uses	54
31.3	Syntax	54
31.3.1	Exported Constants	54
31.3.2	Exported Access Programs	54
31.4	Semantics	54
31.4.1	State Variables	54
31.4.2	Environment Variables	54
31.4.3	Assumptions	54
31.4.4	Access Routine Semantics	54
31.4.5	Local Functions	54
32	MIS of Save Button (M27)	55
32.1	Module	55
32.2	Uses	55
32.3	Syntax	55
32.3.1	Exported Constants	55
32.3.2	Exported Access Programs	55
32.4	Semantics	55
32.4.1	State Variables	55

32.4.2	Environment Variables	55
32.4.3	Assumptions	55
32.4.4	Access Routine Semantics	55
32.4.5	Local Functions	55
33	MIS of Current Data Dispaly (M28)	56
33.1	Module	56
33.2	Uses	56
33.3	Syntax	56
33.3.1	Exported Constants	56
33.3.2	Exported Access Programs	56
33.4	Semantics	56
33.4.1	State Variables	56
33.4.2	Environment Variables	56
33.4.3	Assumptions	56
33.4.4	Access Routine Semantics	56
33.4.5	Local Functions	56
34	MIS of Plot Selection Drop Down (M29)	57
34.1	Module	57
34.2	Uses	57
34.3	Syntax	57
34.3.1	Exported Constants	57
34.3.2	Exported Access Programs	57
34.4	Semantics	57
34.4.1	State Variables	57
34.4.2	Environment Variables	57
34.4.3	Assumptions	57
34.4.4	Access Routine Semantics	57
34.4.5	Local Functions	57
35	MIS of Tree Type Selection Drop Down (M30)	58
35.1	Module	58
35.2	Uses	58
35.3	Syntax	58
35.3.1	Exported Constants	58
35.3.2	Exported Access Programs	58
35.4	Semantics	58
35.4.1	State Variables	58
35.4.2	Environment Variables	58
35.4.3	Assumptions	58
35.4.4	Access Routine Semantics	58
35.4.5	Local Functions	58

36 MIS of Update Data Button (M31)	59
36.1 Module	59
36.2 Uses	59
36.3 Syntax	59
36.3.1 Exported Constants	59
36.3.2 Exported Access Programs	59
36.4 Semantics	59
36.4.1 State Variables	59
36.4.2 Environment Variables	59
36.4.3 Assumptions	59
36.4.4 Access Routine Semantics	59
36.4.5 Local Functions	59
37 MIS of Forest Dispaly (M32)	60
37.1 Module	60
37.2 Uses	60
37.3 Syntax	60
37.3.1 Exported Constants	60
37.3.2 Exported Access Programs	60
37.4 Semantics	60
37.4.1 State Variables	60
37.4.2 Environment Variables	60
37.4.3 Assumptions	60
37.4.4 Access Routine Semantics	60
37.4.5 Local Functions	60
38 MIS of Show Environmental Data Button (M33)	61
38.1 Module	61
38.2 Uses	61
38.3 Syntax	61
38.3.1 Exported Constants	61
38.3.2 Exported Access Programs	61
38.4 Semantics	61
38.4.1 State Variables	61
38.4.2 Environment Variables	61
38.4.3 Assumptions	61
38.4.4 Access Routine Semantics	61
38.4.5 Local Functions	61
39 MIS of Show Tree Parameters Button (M34)	62
39.1 Module	62
39.2 Uses	62
39.3 Syntax	62

39.3.1	Exported Constants	62
39.3.2	Exported Access Programs	62
39.4	Semantics	62
39.4.1	State Variables	62
39.4.2	Environment Variables	62
39.4.3	Assumptions	62
39.4.4	Access Routine Semantics	62
39.4.5	Local Functions	62
40	MIS of Environment Data Display (M35)	63
40.1	Module	63
40.2	Uses	63
40.3	Syntax	63
40.3.1	Exported Constants	63
40.3.2	Exported Access Programs	63
40.4	Semantics	63
40.4.1	State Variables	63
40.4.2	Environment Variables	63
40.4.3	Assumptions	63
40.4.4	Access Routine Semantics	63
40.4.5	Local Functions	63
41	MIS of Tree Parameters Display (M36)	64
41.1	Module	64
41.2	Uses	64
41.3	Syntax	64
41.3.1	Exported Constants	64
41.3.2	Exported Access Programs	64
41.4	Semantics	64
41.4.1	State Variables	64
41.4.2	Environment Variables	64
41.4.3	Assumptions	64
41.4.4	Access Routine Semantics	64
41.4.5	Local Functions	64
42	MIS of Pause Indicator (M37)	65
42.1	Module	65
42.2	Uses	65
42.3	Syntax	65
42.3.1	Exported Constants	65
42.3.2	Exported Access Programs	65
42.4	Semantics	65
42.4.1	State Variables	65

42.4.2	Environment Variables	65
42.4.3	Assumptions	65
42.4.4	Access Routine Semantics	65
42.4.5	Local Functions	65
43	MIS of JSON File Reader Module (M38)	66
43.1	Module	66
43.2	Uses	66
43.3	Syntax	66
43.3.1	Exported Constants	66
43.3.2	Exported Access Programs	66
43.4	Semantics	66
43.4.1	State Variables	66
43.4.2	State Invariant	67
43.4.3	Environment Variables	67
43.4.4	Assumptions	67
43.4.5	Access Routine Semantics	67
43.4.6	Local Functions	68
44	MIS of JSON File Writer Module (M39)	69
44.1	Module	69
44.2	Uses	69
44.3	Syntax	69
44.3.1	Exported Constants	69
44.3.2	Exported Access Programs	69
44.4	Semantics	70
44.4.1	State Variables	70
44.4.2	State Invariant	70
44.4.3	Environment Variables	70
44.4.4	Assumptions	70
44.4.5	Access Routine Semantics	70
44.4.6	Local Functions	71
45	MIS of Pause Manager Module (M40)	72
45.1	Module	72
45.2	Uses	72
45.3	Syntax	72
45.3.1	Exported Constants	72
45.3.2	Exported Access Programs	72
45.4	Semantics	72
45.4.1	State Variables	72
45.4.2	State Invariant	72
45.4.3	Environment Variables	72

45.4.4	Assumptions	72
45.4.5	Access Routine Semantics	73
45.4.6	Local Functions	73
46	MIS of Player Movement Module (M41)	74
46.1	Module	74
46.2	Uses	74
46.3	Syntax	74
46.3.1	Exported Constants	74
46.3.2	Exported Access Programs	74
46.4	Semantics	74
46.4.1	State Variables	74
46.4.2	Environment Variables	74
46.4.3	Assumptions	74
46.4.4	Access Routine Semantics	74
46.4.5	Local Functions	75
47	MIS of New Data Input Box Controller Module (M42)	76
47.1	Module	76
47.2	Uses	76
47.3	Syntax	76
47.3.1	Exported Constants	76
47.3.2	Exported Access Programs	76
47.4	Semantics	76
47.4.1	State Variables	76
47.4.2	Environment Variables	76
47.4.3	Assumptions	76
47.4.4	Access Routine Semantics	76
47.4.5	Local Functions	77
48	MIS of Start Button Controller Module (M43)	78
48.1	Module	78
48.2	Uses	78
48.3	Syntax	78
48.3.1	Exported Constants	78
48.3.2	Exported Access Programs	78
48.4	Semantics	78
48.4.1	State Variables	78
48.4.2	Environment Variables	78
48.4.3	Assumptions	78
48.4.4	Access Routine Semantics	79
48.4.5	Local Functions	79

49 MIS of Instruction Button Controller Module (M44)	80
49.1 Module	80
49.2 Uses	80
49.3 Syntax	80
49.3.1 Exported Constants	80
49.3.2 Exported Access Programs	80
49.4 Semantics	80
49.4.1 State Variables	80
49.4.2 Environment Variables	80
49.4.3 Assumptions	80
49.4.4 Access Routine Semantics	80
49.4.5 Local Functions	81
50 MIS of Update Data Button Controller Module (M45)	82
50.1 Module	82
50.2 Uses	82
50.3 Syntax	82
50.3.1 Exported Constants	82
50.3.2 Exported Access Programs	82
50.4 Semantics	82
50.4.1 State Variables	82
50.4.2 Environment Variables	82
50.4.3 Assumptions	82
50.4.4 Access Routine Semantics	82
50.4.5 Local Functions	83
51 MIS of Contact Us Button Controller Module (M46)	84
51.1 Module	84
51.2 Uses	84
51.3 Syntax	84
51.3.1 Exported Constants	84
51.3.2 Exported Access Programs	84
51.4 Semantics	84
51.4.1 State Variables	84
51.4.2 Environment Variables	84
51.4.3 Assumptions	84
51.4.4 Access Routine Semantics	84
51.4.5 Local Functions	85
52 MIS of Quit Button Controller (M47)	86
52.1 Module	86
52.2 Uses	86
52.3 Syntax	86

52.3.1	Exported Constants	86
52.3.2	Exported Access Programs	86
52.4	Semantics	86
52.4.1	State Variables	86
52.4.2	Environment Variables	86
52.4.3	Assumptions	86
52.4.4	Access Routine Semantics	86
52.4.5	Local Functions	87
53	MIS of Back Button Controller (M48)	88
53.1	Module	88
53.2	Uses	88
53.3	Syntax	88
53.3.1	Exported Constants	88
53.3.2	Exported Access Programs	88
53.4	Semantics	88
53.4.1	State Variables	88
53.4.2	Environment Variables	88
53.4.3	Assumptions	88
53.4.4	Access Routine Semantics	88
53.4.5	Local Functions	89
54	MIS of Plot Selection Drop Down Controller (M49)	90
54.1	Module	90
54.2	Uses	90
54.3	Syntax	90
54.3.1	Exported Constants	90
54.3.2	Exported Access Programs	90
54.4	Semantics	90
54.4.1	State Variables	90
54.4.2	Environment Variables	90
54.4.3	Assumptions	91
54.4.4	Access Routine Semantics	91
54.4.5	Local Functions	91
55	MIS of Tree Type Selection Drop Down Controller(M50)	92
55.1	Module	92
55.2	Uses	92
55.3	Syntax	92
55.3.1	Exported Constants	92
55.3.2	Exported Access Programs	92
55.4	Semantics	92
55.4.1	State Variables	92

55.4.2	Environment Variables	92
55.4.3	Assumptions	93
55.4.4	Access Routine Semantics	93
55.4.5	Local Functions	93
56	MIS of Show Environmental Data Button Controller (M51)	94
56.1	Module	94
56.2	Uses	94
56.3	Syntax	94
56.3.1	Exported Constants	94
56.3.2	Exported Access Programs	94
56.4	Semantics	94
56.4.1	State Variables	94
56.4.2	Environment Variables	94
56.4.3	Assumptions	94
56.4.4	Access Routine Semantics	95
56.4.5	Local Functions	95
57	MIS of Show Tree Parameter Button Controller(M52)	96
57.1	Module	96
57.2	Uses	96
57.3	Syntax	96
57.3.1	Exported Constants	96
57.3.2	Exported Access Programs	96
57.4	Semantics	96
57.4.1	State Variables	96
57.4.2	Environment Variables	96
57.4.3	Assumptions	96
57.4.4	Access Routine Semantics	96
57.4.5	Local Functions	97
58	MIS of Environmental Selection Button Controller(M53)	98
58.1	Module	98
58.2	Uses	98
58.3	Syntax	98
58.3.1	Exported Constants	98
58.3.2	Exported Access Programs	98
58.4	Semantics	98
58.4.1	State Variables	98
58.4.2	Environment Variables	98
58.4.3	Assumptions	98
58.4.4	Access Routine Semantics	98
58.4.5	Local Functions	99

59 MIS of Data Type Selection Buttons Controller(M54)	100
59.1 Module	100
59.2 Uses	100
59.3 Syntax	100
59.3.1 Exported Constants	100
59.3.2 Exported Access Programs	100
59.4 Semantics	100
59.4.1 State Variables	100
59.4.2 Environment Variables	100
59.4.3 Assumptions	100
59.4.4 Access Routine Semantics	100
59.4.5 Local Functions	101
60 MIS of Save Button Controller(M55)	102
60.1 Module	102
60.2 Uses	102
60.3 Syntax	102
60.3.1 Exported Constants	102
60.3.2 Exported Access Programs	102
60.4 Semantics	102
60.4.1 State Variables	102
60.4.2 Environment Variables	102
60.4.3 Assumptions	102
60.4.4 Access Routine Semantics	103
60.4.5 Local Functions	103
61 Appendix	105

3 Introduction

The following document details the Module Interface Specifications for Digital Twin Forest. A digital twin is a virtual representation of the real world, including physical objects, processes, relationships, and behaviors. Elements of a digital twin include data capture and integration, visualization, and advanced analysis including AI, automation, and information sharing and collaboration. This project can be beneficial for two groups of users. The first group of users is forest owners. This project can help them to manage the forest. The second group of users is meteorologists. This project can help them to do research.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at [here](#).

4 Notation

The structure of the MIS for modules comes from [Hoffman and Strooper \(1995\)](#), with the addition that template modules have been adapted from [Ghezzi et al. \(2003\)](#). The mathematical notation comes from Chapter 3 of [Hoffman and Strooper \(1995\)](#). For instance, the symbol $:=$ is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by Digital Twin Forest.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbb{Z}	a number without a fractional component in $(-\infty, \infty)$
natural number	\mathbb{N}	a number without a fractional component in $[1, \infty)$
real	\mathbb{R}	any number in $(-\infty, \infty)$
Boolean	Boolean	a value that takes either True or False

The specification of Digital Twin Forest uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, Digital Twin Forest uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

Table 1: Module Hierarchy(Models)

Level 1	Level 2
Model Modules	M1: ForestTrees
	M2: ForestSky
	M3: ForestTerrain
	M4: RedPine
	M5: Oak
	M6: Beech
	M7: Birch
	M8: WhitePine
	M9: RedMaple
	M10: RedOak
	M11: EnvData
	M12: PlotData
	M13: FirstPersonPlayer
	M14: JsonFile

Table 2: Module Hierarchy(First Viewers Table)

Level 1	Level 2
Viewer Modules	M15: MainPageDisplay
	M16: StartButton
	M17: InstructionButton
	M18: ContactUsButton
	M19: QuitButton
	M20: InstructionInfoDisplay
	M21: ContactUsInfoDisplay
	M22: BackButton
	M23: UpdateDataDisplay
	M24: EnvDataSelectionButton
	M25: DataTypeSelectionButtons
	M26: NewDataInputBox
	M27: SaveButton

Table 3: Module Hierarchy(Second Viewers Table)

Level 1	Level 2
Viewer Modules	M28: CurrentDataDisplay
	M29: PlotSelectionDropDown
	M30: TreeTypeSelectionDropDown
	M31: UpdateDataButton
	M32: ForestDisplay
	M33: ShowEnvDataButton
	M34: ShowTreeParamButton
	M35: EnvDataDisplay
	M36: TreeParamDisplay
	M37: PauseIndicatorDisplay

Table 4: Module Hierarchy(Controllers)

Level 1	Level 2
Controller Modules	M38: JsonFileReader
	M39: JsonFileWriter
	M40: PauseManager
	M41: PlayerMovement
	M42: NewDataInputBoxController
	M43: StartButtonController
	M44: InstructionButtonController
	M45: UpdateDataButtonController
	M46: ContactUsButtonController
	M47: QuitButtonController
	M48: BackButtonController
	M49: PlotSelectionDropDownController
	M50: TreeTypeSelectionDropDownController
	M51: ShowEnvDataButtonController
	M52: ShowTreeParamButtonController
	M53: EnvDataSelectionButtonController
	M54: DataTypeSelectionButtonsController
	M55: SaveButtonController

6 MIS of Forest Trees (M1)

6.1 Module

ForestTrees

6.2 Uses

UnityPlaceTreeWizard

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
GenerateTree	s: Int; t: Double	TreeModels	IllegalArgumentException
DeleteTree	s: Int		

6.4 Semantics

6.4.1 State Variables

None

6.4.2 Environment Variables

TreeModel: the asset bundle of different types of tree models. Brush: the built-in brush to erase the trees.

6.4.3 Assumptions

The input parameters will match the given specification.

6.4.4 Access Routine Semantics

GenerateTree(s, t):

- transition: Unity generates tree models randomly based on the given number and tree height.
- output: None
- exception: None

DeleteTree(s):

- transition: Delete trees by clicking on the brush and erase the workspace.
- output: None
- exception: None

6.4.5 Local Functions

None

7 MIS of Forest Sky (M2)

7.1 Module

SkyBox

7.2 Uses

UnityLightning

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
SetSkyBox	s: Texture		

7.4 Semantics

7.4.1 State Variables

None

7.4.2 Environment Variables

SkyTexture: imported picture of the skybox.

7.4.3 Assumptions

Unity only takes valid texture file type as input.

7.4.4 Access Routine Semantics

SetSkybox(s):

- transition: set the current skybox to the selected texture file.
- output: None
- exception: None

7.4.5 Local Functions

None

8 MIS of Forest Terrain (M3)

8.1 Module

ForestTerrain

8.2 Uses

UnityTerrain

8.3 Syntax

8.3.1 Exported Constants

None

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
SetHeight			

8.4 Semantics

8.4.1 State Variables

None

8.4.2 Environment Variables

Brush: brushes to set the shape and height of the terrain.

8.4.3 Assumptions

None

8.4.4 Access Routine Semantics

SetHeight():

- transition: Change the height of the current terrain with different Unity terrain brushes.
- output: None
- exception: None

8.4.5 Local Functions

None

9 MIS of Red Pine (M4)

9.1 Module

RedPine

9.2 Uses

None

9.3 Syntax

9.3.1 Exported Constants

None

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
new RedPine		RedPine	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

9.4 Semantics

9.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

9.4.2 Environment Variables

None

9.4.3 Assumptions

None

9.4.4 Access Routine Semantics

new RedPine():

- transition: $Treename, Density, DBH, Height, Age := \text{"Red Pine"}, \text{""}, \text{""}, \text{""}, \text{"}"$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg \text{isValidString}(newDensity) \implies \text{IllegalArgumentException}$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg \text{isValidString}(newDBH) \implies \text{IllegalArgumentException}$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

9.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

10 MIS of Oak (M5)

10.1 Module

Oak

10.2 Uses

None

10.3 Syntax

10.3.1 Exported Constants

None

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
new Oak		Oak	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

10.4 Semantics

10.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

10.4.2 Environment Variables

None

10.4.3 Assumptions

None

10.4.4 Access Routine Semantics

new Oak():

- transition: $Treename, Density, DBH, Height, Age := "Oak", "", "", "", ""$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg isValidString(newDensity) \implies IllegalArgumentException$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg isValidString(newDBH) \implies IllegalArgumentException$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

10.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

11 MIS of Beech (M6)

11.1 Module

Beech

11.2 Uses

None

11.3 Syntax

11.3.1 Exported Constants

None

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
new Beech		Beech	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

11.4 Semantics

11.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

11.4.2 Environment Variables

None

11.4.3 Assumptions

None

11.4.4 Access Routine Semantics

new Beech():

- transition: $Treename, Density, DBH, Height, Age := \text{"Beech"}, \text{""}, \text{""}, \text{""}, \text{"}"$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg isValidString(newDensity) \implies IllegalArgumentException$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg isValidString(newDBH) \implies IllegalArgumentException$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

11.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

12 MIS of Birch (M7)

12.1 Module

Birch

12.2 Uses

None

12.3 Syntax

12.3.1 Exported Constants

None

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
new Birch		Birch	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

12.4 Semantics

12.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

12.4.2 Environment Variables

None

12.4.3 Assumptions

None

12.4.4 Access Routine Semantics

new Birch():

- transition: $Treename, Density, DBH, Height, Age := \text{"Birch"}, \text{""}, \text{""}, \text{""}, \text{"}"$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg isValidString(newDensity) \implies IllegalArgumentException$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg isValidString(newDBH) \implies IllegalArgumentException$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

12.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

13 MIS of White Pine (M8)

13.1 Module

WhitePine

13.2 Uses

None

13.3 Syntax

13.3.1 Exported Constants

None

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
new WhitePine		WhitePine	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

13.4 Semantics

13.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

13.4.2 Environment Variables

None

13.4.3 Assumptions

None

13.4.4 Access Routine Semantics

new WhitePine():

- transition: $Treename, Density, DBH, Height, Age := \text{“White Pine”, “”, “”, “”, “”}$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg isValidString(newDensity) \implies IllegalArgumentException$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg isValidString(newDBH) \implies IllegalArgumentException$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

13.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

14 MIS of Red Maple (M9)

14.1 Module

RedMaple

14.2 Uses

None

14.3 Syntax

14.3.1 Exported Constants

None

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
new RedMaple		RedMaple	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

14.4 Semantics

14.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

14.4.2 Environment Variables

None

14.4.3 Assumptions

None

14.4.4 Access Routine Semantics

new RedMaple():

- transition: $Treename, Density, DBH, Height, Age := \text{"Red Maple"}, \text{""}, \text{""}, \text{""}, \text{"}"$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg \text{isValidString}(newDensity) \implies \text{IllegalArgumentException}$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg \text{isValidString}(newDBH) \implies \text{IllegalArgumentException}$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

14.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

15 MIS of Red Oak (M10)

15.1 Module

RedOak

15.2 Uses

None

15.3 Syntax

15.3.1 Exported Constants

None

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
new RedOak		RedOak	
getTreeName		String	
setDensity	String		IllegalArgumentException
getDensity		String	
setDBH	String		IllegalArgumentException
getDBH		String	
setHeight	String		IllegalArgumentException
getHeight		String	
setAge	String		IllegalArgumentException
getAge		String	

15.4 Semantics

15.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

15.4.2 Environment Variables

None

15.4.3 Assumptions

None

15.4.4 Access Routine Semantics

new RedOak():

- transition: $Treename, Density, DBH, Height, Age := \text{"Red Oak"}, \text{""}, \text{""}, \text{""}, \text{"}"$
- output: $out := self$
- exception: None

getTreeName():

- transition: None
- output: $out := Treename$
- exception: None

setDensity(newDensity):

- transition: $Density := newDensity$
- output: None
- exception: $\neg isValidString(newDensity) \implies IllegalArgumentException$

getDensity():

- transition: None
- output: $out := Density$
- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$
- output: None
- exception: $\neg isValidString(newDBH) \implies IllegalArgumentException$

getDBH():

- transition: None
- output: $out := DBH$
- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$
- output: None
- exception: $\neg \text{isValidString}(newHeight) \implies \text{IllegalArgumentException}$

getHeight():

- transition: None
- output: $out := Height$
- exception: None

setAge(newAge):

- transition: $Age := newAge$
- output: None
- exception: $\neg \text{isValidString}(newAge) \implies \text{IllegalArgumentException}$

getAge():

- transition: None
- output: $out := Age$
- exception: None

15.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

16 MIS of Environmental Data (M11)

16.1 Module

EnvData

16.2 Uses

None

16.3 Syntax

16.3.1 Exported Constants

None

16.3.2 Exported Access Programs

Name	In	Out	Exceptions
new EnvData		EnvData	
setHumility	String		IllegalArgumentException
getHumility		String	
setTemp	String		IllegalArgumentException
getTemp		String	
setSC	String		IllegalArgumentException
getSC		String	
setSN	String		IllegalArgumentException
getSN		String	
setLAI	String		IllegalArgumentException
getLAI		String	

16.4 Semantics

16.4.1 State Variables

Humility : String

Temp : String

SC : String

SN : String

LAI : String

16.4.2 Environment Variables

None

16.4.3 Assumptions

None

16.4.4 Access Routine Semantics

new EnvData():

- transition: $Humility, Temp, SC, SN, LAI := "", "", "", "", ""$
- output: $out := self$
- exception: None

getHumility():

- transition: None
- output: $out := Humility$
- exception: None

setHumility(newHumility):

- transition: $Humility := newHumility$
- output: None
- exception: $\neg isValidString(newHumility) \implies IllegalArgumentException$

getTemp():

- transition: None
- output: $out := Temp$
- exception: None

setTemp(newTemp):

- transition: $Temp := newTemp$
- output: None
- exception: $\neg isValidString(newTemp) \implies IllegalArgumentException$

getSC():

- transition: None
- output: $out := SC$

- exception: None

setSC(newSC):

- transition: $SC := newSC$
- output: None
- exception: $\neg \text{isValidString}(\text{newSC}) \implies \text{IllegalArgumentException}$

getSN():

- transition: None
- output: $out := SN$
- exception: None

setSN(newSN):

- transition: $SN := newSN$
- output: None
- exception: $\neg \text{isValidString}(\text{newSN}) \implies \text{IllegalArgumentException}$

getLAI():

- transition: None
- output: $out := LAI$
- exception: None

setLAI(newLAI):

- transition: $LAI := newLAI$
- output: None
- exception: $\neg \text{isValidString}(\text{newLAI}) \implies \text{IllegalArgumentException}$

16.4.5 Local Functions

ValidCharacters = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." }

isValidString(S): $\text{String} \rightarrow \mathbb{B}$

$\text{isValidString}(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in \text{ValidCharacters})$

17 MIS of Plot Data (M12)

17.1 Module

PlotData

17.2 Uses

M4, M5, M6, M7, M8, M9, M10, M11

17.3 Syntax

17.3.1 Exported Constants

None

17.3.2 Exported Access Programs

Name	In	Out	Exceptions
new PlotData		PlotData	
setRedPineObj	RedPine		
getRedPineObj		RedPine	
setOakObj	Oak		
getOakObj		Oak	
setBeechObj	Beech		
getBeechObj		Beech	
setBirchObj	Birch		
getBirchObj		Birch	
setWhitePineObj	WhitePine		
getWhitePineObj		WhitePine	
setRedMapleObj	RedMaple		
getRedMapleObj		RedMaple	
setRedOakObj	RedOak		
getRedOakObj		RedOak	
setEnvDataObj	EnvData		
getEnvDataObj		EnvData	

17.4 Semantics

17.4.1 State Variables

RedPineObj : *RedPine*
OakObj : *Oak*
BeechObj : *Beech*
BirchObj : *Birch*
WhitePineObj : *WhitePine*
RedMapleObj : *RedMaple*
RedOakObj : *RedOak*
EnvDataObj : *EnvData*

17.4.2 Environment Variables

None

17.4.3 Assumptions

None

17.4.4 Access Routine Semantics

new PlotData():

- transition:
 - *RedPineObj*, *OakObj*, *BeechObj*, *BirchObj* := *null*, *null*, *null*, *null*
 - *WhitePineObj*, *RedMapleObj*, *RedOakObj*, *EnvDataObj* := *null*, *null*, *null*, *null*
- output: *out* := *self*
- exception: None

getRedPineObj():

- transition: None
- output: *out* := *RedPineObj*
- exception: None

setRedPineObj(newRedPineObj):

- transition: *RedPineObj* := *newRedPineObj*
- output: None

- exception: None

getOakObj():

- transition: None
- output: $out := OakObj$
- exception: None

setOakObj(newOakObj):

- transition: $OakObj := newOakObj$
- output: None
- exception: None

getBeechObj():

- transition: None
- output: $out := BeechObj$
- exception: None

setBeechObj(newBeechObj):

- transition: $BeechObj := newBeechObj$
- output: None
- exception: None

getBirchObj():

- transition: None
- output: $out := BirchObj$
- exception: None

setBirchObj(newBirchObj):

- transition: $BirchObj := newBirchObj$
- output: None
- exception: None

getWhitePineObj():

- transition: None
- output: $out := WhitePineObj$
- exception: None

setWhitePineObj(newWhitePineObj):

- transition: $WhitePineObj := newWhitePineObj$
- output: None
- exception: None

getRedMapleObj():

- transition: None
- output: $out := RedMapleObj$
- exception: None

setRedMapleObj(newRedMapleObj):

- transition: $RedMapleObj := newRedMapleObj$
- output: None
- exception: None

getRedOakObj():

- transition: None
- output: $out := RedOakObj$
- exception: None

setRedOakObj(newRedOakObj):

- transition: $RedOakObj := newRedOakObj$
- output: None
- exception: None

getEnvDataObj():

- transition: None
- output: $out := EnvDataObj$
- exception: None

setEnvDataObj(newEnvDataObj):

- transition: $EnvDataObj := newEnvDataObj$
- output: None
- exception: None

17.4.5 Local Functions

None

18 MIS of First Person Player (M13)

18.1 Module

FirstPersonPlayer

18.2 Uses

Character Controller Module from Unity

18.3 Syntax

This is a module provided by UnityEngine.UI. Please click [here](#) to check official document from Unity. We have designed a controller for this module. The controller is PlayerMovement(M41).

18.4 Semantics

This is a module provided by UnityEngine.UI. Please click [here](#) to check official document from Unity. We have designed a controller for this module. The controller is PlayerMovement(M41).

19 MIS of Json File (M14)

19.1 Module

JsonFile. This is not a typical class. This section only aims to show how JSON files are organized formally.

19.2 Local Type

$X = \text{tuple}(\text{key} : \text{String}, \text{value} : \text{String}) \wedge \text{isValidString}(\text{value})$

$S : \text{set of } X$

$\text{TreeANDEnvData} = \text{tuple}(\text{key} : \text{String}, \text{values} : S)$

19.3 State Variables

$\text{JsonFile} : \text{set of TreeANDEnvData}$

19.4 Example

- First, define all the tuples that have type X .

- $x_1 = (\text{"DBH"}, \text{"10"}) : X$
- $x_2 = (\text{"Age"}, \text{"10"}) : X$
- $x_3 = (\text{"Height"}, \text{"10"}) : X$
- $x_4 = (\text{"Density"}, \text{"10"}) : X$

- $x_5 = (\text{"DBH"}, \text{"20"}) : X$
- $x_6 = (\text{"Age"}, \text{"20"}) : X$
- $x_7 = (\text{"Height"}, \text{"20"}) : X$
- $x_8 = (\text{"Density"}, \text{"20"}) : X$

- $x_9 = (\text{"DBH"}, \text{"30"}) : X$
- $x_{10} = (\text{"Age"}, \text{"30"}) : X$
- $x_{11} = (\text{"Height"}, \text{"30"}) : X$
- $x_{12} = (\text{"Density"}, \text{"30"}) : X$

- $x_{13} = (\text{"DBH"}, \text{"40"}) : X$
- $x_{14} = (\text{"Age"}, \text{"40"}) : X$
- $x_{15} = (\text{"Height"}, \text{"40"}) : X$

- $x_{16} = (\text{"Density"}, \text{"40"}) : X$
- $x_{17} = (\text{"DBH"}, \text{"50"}) : X$
- $x_{18} = (\text{"Age"}, \text{"50"}) : X$
- $x_{19} = (\text{"Height"}, \text{"50"}) : X$
- $x_{20} = (\text{"Density"}, \text{"50"}) : X$
- $x_{21} = (\text{"DBH"}, \text{"60"}) : X$
- $x_{22} = (\text{"Age"}, \text{"60"}) : X$
- $x_{23} = (\text{"Height"}, \text{"60"}) : X$
- $x_{24} = (\text{"Density"}, \text{"60"}) : X$
- $x_{25} = (\text{"DBH"}, \text{"70"}) : X$
- $x_{26} = (\text{"Age"}, \text{"70"}) : X$
- $x_{27} = (\text{"Height"}, \text{"70"}) : X$
- $x_{28} = (\text{"Density"}, \text{"70"}) : X$
- $x_{29} = (\text{"Humidity"}, \text{"10"}) : X$
- $x_{30} = (\text{"Temperature"}, \text{"20"}) : X$
- $x_{31} = (\text{"SC"}, \text{"10"}) : X$
- $x_{32} = (\text{"SN"}, \text{"95"}) : X$
- $x_{33} = (\text{"LAI"}, \text{"95"}) : X$

- Second, define all the sets that have type S

- $s_1 = \{x_1, x_2, x_3, x_4\} : S$
- $s_2 = \{x_5, x_6, x_7, x_8\} : S$
- $s_3 = \{x_9, x_{10}, x_{11}, x_{12}\} : S$
- $s_4 = \{x_{13}, x_{14}, x_{15}, x_{16}\} : S$
- $s_5 = \{x_{17}, x_{18}, x_{19}, x_{20}\} : S$
- $s_6 = \{x_{21}, x_{22}, x_{23}, x_{24}\} : S$
- $s_7 = \{x_{25}, x_{26}, x_{27}, x_{28}\} : S$
- $s_8 = \{x_{29}, x_{30}, x_{31}, x_{32}, x_{33}\} : S$

- Third, define all the tuples that have type $TreeANDEnvData$.

- $d_1 = ("RedPineData", s_1) : TreeANDEnvData$
- $d_2 = ("OakData", s_2) : TreeANDEnvData$
- $d_3 = ("BeechData", s_3) : TreeANDEnvData$
- $d_4 = ("BirchData", s_4) : TreeANDEnvData$
- $d_5 = ("WhitePineData", s_5) : TreeANDEnvData$
- $d_6 = ("RedMapleData", s_6) : TreeANDEnvData$
- $d_7 = ("RedOakData", s_7) : TreeANDEnvData$
- $d_8 = ("EnvData", s_8) : TreeANDEnvData$

- Finally, $JsonFile = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$.

19.5 Local Functions

$ValidCharacters = \{ "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "." \}$

$isValidString(S) : String \rightarrow \mathbb{B}$

$isValidString(S) = \forall (i : \mathbb{Z} | 0 \leq i < |S| : S[i] \in ValidCharacters)$

20 MIS of Main Page (M15)

20.1 Module

MainPageDisplay

20.2 Uses

M43 , UnityEngine.UI

20.3 Syntax

20.3.1 Exported Constants

None

20.3.2 Exported Access Programs

None

20.4 Semantics

This module is used to display the UI of the homepage. You can refer to Unity Canvas Documentation by clicking [here](#).

20.4.1 State Variables

None

20.4.2 Environment Variables

None

20.4.3 Assumptions

None

20.4.4 Access Routine Semantics

None

20.4.5 Local Functions

None

21 MIS of Start Button (M16)

21.1 Module

StartButton

21.2 Uses

M43 , UnityEngine.UI

21.3 Syntax

21.3.1 Exported Constants

None

21.3.2 Exported Access Programs

None

21.4 Semantics

This module is used to display the UI of the StartButton. You can refer to Unity Button Documentation by clicking [here](#).

21.4.1 State Variables

None

21.4.2 Environment Variables

None

21.4.3 Assumptions

None

21.4.4 Access Routine Semantics

None

21.4.5 Local Functions

None

22 MIS of Instruction Button (M17)

22.1 Module

InstructionButton

22.2 Uses

M44 , UnityEngine.UI

22.3 Syntax

22.3.1 Exported Constants

None

22.3.2 Exported Access Programs

None

22.4 Semantics

This module is used to display the UI of the InstructionButton. You can refer to Unity Button Documentation by clicking [here](#).

22.4.1 State Variables

None

22.4.2 Environment Variables

None

22.4.3 Assumptions

None

22.4.4 Access Routine Semantics

None

22.4.5 Local Functions

None

23 MIS of Contact Us Button (M18)

23.1 Module

ContactUsButton

23.2 Uses

M45 , UnityEngine.UI

23.3 Syntax

23.3.1 Exported Constants

None

23.3.2 Exported Access Programs

None

23.4 Semantics

This module is used to display the UI of the ContactUsButton. You can refer to Unity Button Documentation by clicking [here](#).

23.4.1 State Variables

None

23.4.2 Environment Variables

None

23.4.3 Assumptions

None

23.4.4 Access Routine Semantics

None

23.4.5 Local Functions

None

24 MIS of Quit Button (M19)

24.1 Module

QuitButton

24.2 Uses

M46 ,UnityEngine.UI

24.3 Syntax

24.3.1 Exported Constants

None

24.3.2 Exported Access Programs

None

24.4 Semantics

This module is used to display the UI of the QuitButton.You can refer to Unity Button Documentation by clicking [here](#).

24.4.1 State Variables

None

24.4.2 Environment Variables

None

24.4.3 Assumptions

None

24.4.4 Access Routine Semantics

None

24.4.5 Local Functions

None

25 MIS of Instruction Page (M20)

25.1 Module

InstructionInfoDisplay

25.2 Uses

M43 , UnityEngine.UI

25.3 Syntax

25.3.1 Exported Constants

None

25.3.2 Exported Access Programs

None

25.4 Semantics

This module is used to display the UI of the instruction page. You can refer to Unity Canvas Documentation by clicking [here](#).

25.4.1 State Variables

None

25.4.2 Environment Variables

None

25.4.3 Assumptions

None

25.4.4 Access Routine Semantics

None

25.4.5 Local Functions

None

26 MIS of Contact Us Page (M21)

26.1 Module

ContactUsInfoDisplay

26.2 Uses

M43 UnityEngine.UI

26.3 Syntax

26.3.1 Exported Constants

None

26.3.2 Exported Access Programs

None

26.4 Semantics

This module is used to display the UI of the Contact Us page. You can refer to Unity Canvas Documentation by clicking [here](#).

26.4.1 State Variables

None

26.4.2 Environment Variables

None

26.4.3 Assumptions

None

26.4.4 Access Routine Semantics

None

26.4.5 Local Functions

None

27 MIS of Back Button (M22)

27.1 Module

BackButton

27.2 Uses

M48 , UnityEngine.UI

27.3 Syntax

27.3.1 Exported Constants

None

27.3.2 Exported Access Programs

None

27.4 Semantics

This module is used to display the UI of the BackButton. You can refer to Unity Button Documentation by clicking [here](#).

27.4.1 State Variables

None

27.4.2 Environment Variables

None

27.4.3 Assumptions

None

27.4.4 Access Routine Semantics

None

27.4.5 Local Functions

None

28 MIS of Update Data Page (M23)

28.1 Module

UpdateDataDisplay

28.2 Uses

M43 UnityEngine.UI

28.3 Syntax

28.3.1 Exported Constants

None

28.3.2 Exported Access Programs

None

28.4 Semantics

This module is used to display the UI of the Update Data page. You can refer to Unity Canvas Documentation by clicking [here](#).

28.4.1 State Variables

None

28.4.2 Environment Variables

None

28.4.3 Assumptions

None

28.4.4 Access Routine Semantics

None

28.4.5 Local Functions

None

29 MIS of Environmental Data Selection Button (M24)

29.1 Module

EnvDataSelectionButton

29.2 Uses

M53 , UnityEngine.UI

29.3 Syntax

29.3.1 Exported Constants

None

29.3.2 Exported Access Programs

None

29.4 Semantics

This module is used to display the UI of the EnvDataSelectionButton. You can refer to Unity Button Documentation by clicking [here](#).

29.4.1 State Variables

None

29.4.2 Environment Variables

None

29.4.3 Assumptions

None

29.4.4 Access Routine Semantics

None

29.4.5 Local Functions

None

30 MIS of Data Type Selection Button (M25)

30.1 Module

DataTypeSelectionButton

30.2 Uses

M54 , UnityEngine.UI

30.3 Syntax

30.3.1 Exported Constants

None

30.3.2 Exported Access Programs

None

30.4 Semantics

This module is used to display the UI of the DataTypeSelectionButton. You can refer to Unity Button Documentation by clicking [here](#).

30.4.1 State Variables

None

30.4.2 Environment Variables

None

30.4.3 Assumptions

None

30.4.4 Access Routine Semantics

None

30.4.5 Local Functions

None

31 MIS of New Data Input Box (M26)

31.1 Module

NewDataInputBox

31.2 Uses

M43 , UnityEngine.UI

31.3 Syntax

31.3.1 Exported Constants

None

31.3.2 Exported Access Programs

None

31.4 Semantics

This module is used to display the UI of the new data input box. You can refer to Unity Input Field Documentation by clicking [here](#)

31.4.1 State Variables

None

31.4.2 Environment Variables

None

31.4.3 Assumptions

None

31.4.4 Access Routine Semantics

None

31.4.5 Local Functions

None

32 MIS of Save Button (M27)

32.1 Module

SaveButton

32.2 Uses

M55 , UnityEngine.UI

32.3 Syntax

32.3.1 Exported Constants

None

32.3.2 Exported Access Programs

None

32.4 Semantics

This module is used to display the UI of the SaveButton.You can refer to Unity Button Documentation by clicking [here](#).

32.4.1 State Variables

None

32.4.2 Environment Variables

None

32.4.3 Assumptions

None

32.4.4 Access Routine Semantics

None

32.4.5 Local Functions

None

33 MIS of Current Data Display (M28)

33.1 Module

CurrentDataDisplay

33.2 Uses

UnityEngine.UI

33.3 Syntax

33.3.1 Exported Constants

None

33.3.2 Exported Access Programs

None

33.4 Semantics

This module is used to display the UI of the current data. You can refer to Unity Text Documentation by clicking [here](#).

33.4.1 State Variables

None

33.4.2 Environment Variables

None

33.4.3 Assumptions

None

33.4.4 Access Routine Semantics

None

33.4.5 Local Functions

None

34 MIS of Plot Selection Drop Down (M29)

34.1 Module

PlotSelection

34.2 Uses

M49, UnityEngine.UI

34.3 Syntax

34.3.1 Exported Constants

None

34.3.2 Exported Access Programs

None

34.4 Semantics

This module is used to display the dropdown box of plot selection. You can refer to Unity Drop Down Documentation by clicking [here](#).

34.4.1 State Variables

None

34.4.2 Environment Variables

None

34.4.3 Assumptions

None

34.4.4 Access Routine Semantics

None

34.4.5 Local Functions

None

35 MIS of Tree Type Selection Drop Down (M30)

35.1 Module

TreeTypeSelection

35.2 Uses

M50 , UnityEngine.UI

35.3 Syntax

35.3.1 Exported Constants

None

35.3.2 Exported Access Programs

None

35.4 Semantics

This module is used to display the dropdown box of the tree type selection. You can refer to Unity Drop Down Documentation by clicking [here](#).

35.4.1 State Variables

None

35.4.2 Environment Variables

None

35.4.3 Assumptions

None

35.4.4 Access Routine Semantics

None

35.4.5 Local Functions

None

36 MIS of Update Data Button (M31)

36.1 Module

UpdateDataButton

36.2 Uses

M45 , UnityEngine.UI

36.3 Syntax

36.3.1 Exported Constants

None

36.3.2 Exported Access Programs

None

36.4 Semantics

The module is used to display the UI of UpdateDataButton. You can refer to Unity Button Documentation by clicking [here](#).

36.4.1 State Variables

None

36.4.2 Environment Variables

None

36.4.3 Assumptions

None

36.4.4 Access Routine Semantics

None

36.4.5 Local Functions

None

37 MIS of Forest Display (M32)

37.1 Module

ForestDisplay

37.2 Uses

UnityEngine.UI, M1, M2, M3

37.3 Syntax

37.3.1 Exported Constants

None

37.3.2 Exported Access Programs

None

37.4 Semantics

37.4.1 State Variables

This module is used to display the forest models.

37.4.2 Environment Variables

None

37.4.3 Assumptions

None

37.4.4 Access Routine Semantics

None

37.4.5 Local Functions

None

38 MIS of Show Environmental Data Button (M33)

38.1 Module

ShowEnvDataButton

38.2 Uses

M51 , UnityEngine.UI

38.3 Syntax

38.3.1 Exported Constants

None

38.3.2 Exported Access Programs

None

38.4 Semantics

This module is used to display the UI of the ShowEnvDataButton. You can refer to Unity Button Documentation by clicking [here](#).

38.4.1 State Variables

None

38.4.2 Environment Variables

None

38.4.3 Assumptions

None

38.4.4 Access Routine Semantics

None

38.4.5 Local Functions

None

39 MIS of Show Tree Parameters Button (M34)

39.1 Module

ShowTreeParamButton

39.2 Uses

M52 , UnityEngine.UI

39.3 Syntax

39.3.1 Exported Constants

None

39.3.2 Exported Access Programs

None

39.4 Semantics

This module is used to display the UI of the ShowTreeParamButton. You can refer to Unity Button Documentation by clicking [here](#).

39.4.1 State Variables

None

39.4.2 Environment Variables

None

39.4.3 Assumptions

None

39.4.4 Access Routine Semantics

None

39.4.5 Local Functions

None

40 MIS of Environment Data Display (M35)

40.1 Module

EnvDataDisplay

40.2 Uses

UnityEngine.UI

40.3 Syntax

40.3.1 Exported Constants

None

40.3.2 Exported Access Programs

None

40.4 Semantics

This module is used to display the UI of the environment data. You can check Unity Text Documentation by clicking [here](#).

40.4.1 State Variables

None

40.4.2 Environment Variables

None

40.4.3 Assumptions

None

40.4.4 Access Routine Semantics

None

40.4.5 Local Functions

None

41 MIS of Tree Parameters Display (M36)

41.1 Module

TreeParamDisplay

41.2 Uses

UnityEngine.UI

41.3 Syntax

41.3.1 Exported Constants

None

41.3.2 Exported Access Programs

None

41.4 Semantics

This module is used to display the UI of the tree parameters. You can check Unity Text Documentation by clicking [here](#).

41.4.1 State Variables

None

41.4.2 Environment Variables

None

41.4.3 Assumptions

None

41.4.4 Access Routine Semantics

None

41.4.5 Local Functions

None

42 MIS of Pause Indicator (M37)

42.1 Module

PauseIndicatorDisplay

42.2 Uses

UnityEngine.UI

42.3 Syntax

42.3.1 Exported Constants

None

42.3.2 Exported Access Programs

None

42.4 Semantics

This module is used to display the status of pausing. You can check Unity Text Documentation by clicking [here](#).

42.4.1 State Variables

None

42.4.2 Environment Variables

None

42.4.3 Assumptions

None

42.4.4 Access Routine Semantics

None

42.4.5 Local Functions

None

43 MIS of JSON File Reader Module (M38)

43.1 Module

JsonFileReader

43.2 Uses

System.Collections

System.Collections.Generic

UnityEngine

System.IO

UnityEngine.UI

M35

M36

M12

M4

M5

M6

M7

M8

M9

M10

M11

43.3 Syntax

43.3.1 Exported Constants

None

43.3.2 Exported Access Programs

Name	In	Out	Exceptions
Awake			
Start			
readFile	Z		

43.4 Semantics

43.4.1 State Variables

treeParamDisplay: TreeParamDisplay

envDataDisplay: EnvDataDisplay

dataModelObj: DataModel
jsonModelObj: JsonModel
plotNumber: \mathbb{Z}
filePath: string
plotJsonData: string

43.4.2 State Invariant

DEFAULT = “./dataCenter/overalldata.json”
PATH = “./dataCenter/plot”
SUFFIX = “data.json”

43.4.3 Environment Variables

overalldata.json
plot1data.json
plot2data.json
plot3data.json
plot4data.json
plot5data.json
plot6data.json
plot7data.json
plot8data.json
plot9data.json
plot10data.json
plot11data.json
plot12data.json
plot13data.json
plot14data.json

43.4.4 Assumptions

Assume all the Json files are in the correct path.

43.4.5 Access Routine Semantics

Awake():

- transition: readFile(0)
- output: None
- exception: None

Start():

- transition: None
- output: None
- exception: None

readFile(value):

- transition: plotNumber:= value + 1,
 (plotNumber=15) \rightarrow (filePath:=DEFAULT) \vee (plotNumber \neq 15) \rightarrow (filePath:=psx)
 WHERE p:=PATH, s:=plotNumber.ToString(), f:= SUFFIX,
 plotJsonData:= File.ReadAllText(filePath),
 JsonModelObj:= Newtonsoft.Json.JsonConvert.DeserializeObject<JsonModel>(plotJsonData),
 DataModelObj.RedPineData:=JsonModelObj.redPine;
 DataModelObj.OakData:=JsonModelObj.oak;
 DataModelObj.BeechData:=JsonModelObj.beech;
 DataModelObj.BirchData:=JsonModelObj.birch;
 DataModelObj.RedMapleData:=JsonModelObj.redMaple;
 DataModelObj.WhitePineData:=JsonModelObj.whitePine;
 DataModelObj.RedOakData:=JsonModelObj.redOak;
 DataModelObj.EnvData:=JsonModelObj.envData;
- output: None
- exception: None

43.4.6 Local Functions

None

44 MIS of JSON File Writer Module (M39)

44.1 Module

JsonFileWriter

44.2 Uses

JsonFileReader

NewDataInpputBoxController

System.Collections

System.Collections.Generic

UnityEngine

System.IO

UnityEngine.UI

M35

M36

M12

M4

M5

M6

M7

M8

M9

M10

M11

44.3 Syntax

44.3.1 Exported Constants

None

44.3.2 Exported Access Programs

Name	In	Out	Exceptions
updateData	string, \mathbb{Z} , string		
getOldData	\mathbb{Z}		
changeData	\mathbb{Z} , string, string		
writeAndSave	string, string		

44.4 Semantics

44.4.1 State Variables

treeParamDisplay: TreeParamDisplay
envDataDisplay: EnvDataDisplay
dataModelObj: DataModel
jsonModelObj: JsonModel
plotNumber: \mathbb{Z}
filePath: string
plotJsonData: stri

44.4.2 State Invariant

DEFAULT = “./dataCenter/overalldata.json”
PATH = “./dataCenter/plot”
SUFFIX = “data.json”

44.4.3 Environment Variables

overalldata.json
plot1data.json
plot2data.json
plot3data.json
plot4data.json
plot5data.json
plot6data.json
plot7data.json
plot8data.json
plot9data.json
plot10data.json
plot11data.json
plot12data.json
plot13data.json
plot14data.json

44.4.4 Assumptions

Assume that all the JSON files are in the correct path.

44.4.5 Access Routine Semantics

findFilePath(value):

- transition: $\text{plotNumber} := \text{value} + 1,$
 $(\text{plotNumber} = 15) \rightarrow (\text{filePath} = \text{DEFAULT}) \vee (\text{plotNumber} \neq 15) \rightarrow (\text{filePath} = \text{psx})$
WHERE $p := \text{PATH}, s := \text{plotNumber.ToString()}, f := \text{SUFFIX}$

- output: None
- exception: None

`getOldData(value):`

- transition: `JsonFileReader.readFile(value)`
- output: None
- exception: None

`changeData(value, tree, p):`

- transition: $\text{treeData} := (\text{tree} \rightarrow \text{RedPineData} \vee \text{OakData} \vee \text{BeechData} \vee \text{RedPineData} \vee \text{RedMapleData} \vee \text{WhitePineData} \vee \text{RedOakData}),$
 $\text{treeData.p} := \text{value}, \text{plotJsonData} := \text{set of treeData}$
 $\text{content} := \text{serializeObject}(\text{plotJsonData});$
- output: None
- exception: None

`writeAndSave(content, filePath):`

- transition: `write(content)`
- output: None
- exception: None

44.4.6 Local Functions

None

45 MIS of Pause Manager Module (M40)

45.1 Module

PauseManager

45.2 Uses

System.Collections

System.Collections.Generic

UnityEngine

UnityEngine.UI

45.3 Syntax

45.3.1 Exported Constants

None

45.3.2 Exported Access Programs

Name	In	Out	Exceptions
Start			
Update			

45.4 Semantics

45.4.1 State Variables

isPaused: Boolean

pauseMessage: Text

45.4.2 State Invariant

PAUSE = "Pause"

BLANK = ""

45.4.3 Environment Variables

keyboard

45.4.4 Assumptions

None

45.4.5 Access Routine Semantics

Start():

- transition: `pauseMessage.text := BLANK`
- output: `None`
- exception: `None`

Update():

- transition: `Input.GetKeyDown(KeyCode.P) → (¬isPaused → (Time.timeScale := 0, pauseMessage.text = PAUSE, isPaused := ¬ isPaused) ∨ (isPaused → (Time.timeScale := 1, pauseMessage.text = BLANK, isPaused := ¬ isPaused)))`
- output: `None`
- exception: `None`

45.4.6 Local Functions

`None`

46 MIS of Player Movement Module (M41)

46.1 Module

PlayerMovement

46.2 Uses

CharacterController
System.Collections
System.Collections.Generic
UnityEngine
Time
Vector3

46.3 Syntax

46.3.1 Exported Constants

None

46.3.2 Exported Access Programs

Name	In	Out	Exceptions
Update			

46.4 Semantics

46.4.1 State Variables

speed: float
controller: CharacterController transform: Transform

46.4.2 Environment Variables

mouse

46.4.3 Assumptions

Assume that users press the right keys.

46.4.4 Access Routine Semantics

Update():

- transition: `x, z:= Input.GetAxis("Horizontal"),Input.GetAxis("Vertical"),`
`direction:= transform.right * x + Camera.main.transform.forward * z,`
`controller.Move(d,s,t): Vector3 × float × Time`
- output: None
- exception: None

46.4.5 Local Functions

None

47 MIS of New Data Input Box Controller Module (M42)

47.1 Module

NewDataInputBoxController

47.2 Uses

M39

47.3 Syntax

47.3.1 Exported Constants

None

47.3.2 Exported Access Programs

Name	In	Out	Exceptions
isValid	string		
storeData	string		

47.4 Semantics

47.4.1 State Variables

number: string

47.4.2 Environment Variables

None

47.4.3 Assumptions

Assume that the contents of the string type input are all numbers

47.4.4 Access Routine Semantics

isValid(number):

- transition: check if the input is valid or not
- output: True if the input is valid, False otherwise
- exception: None

storeData(number):

- transition: call isValid(number) and pass the new data to the Update Data Button Module.
- output: None
- exception: None

47.4.5 Local Functions

None

48 MIS of Start Button Controller Module (M43)

48.1 Module

StartButtonController

48.2 Uses

System.Collections

System.Collections.Generic

UnityEngine

UnityEngine.SceneManagement

LoadSceneMode

48.3 Syntax

48.3.1 Exported Constants

newData: String

48.3.2 Exported Access Programs

Name	In	Out	Exceptions
Start			
Update			
goToForestScene	String, LoadSceneMode		

48.4 Semantics

48.4.1 State Variables

None

48.4.2 Environment Variables

mouse

Forest

48.4.3 Assumptions

None

48.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

goToForestScene():

- transition: timescale := 1, load the Forest scene
- output: None
- exception: None

48.4.5 Local Functions

None

49 MIS of Instruction Button Controller Module (M44)

49.1 Module

InstructionButtonController

49.2 Uses

None

49.3 Syntax

49.3.1 Exported Constants

None

49.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
setActive	Boolean		

49.4 Semantics

49.4.1 State Variables

value: Boolean

active: Boolean

49.4.2 Environment Variables

Mouse

InstructionPage

49.4.3 Assumptions

None

49.4.4 Access Routine Semantics

onClick():

- transition: $\text{value} := \neg \text{value}$
- output: None
- exception: None

setActive(value):

- transition: active:= value
- output: None
- exception: None

49.4.5 Local Functions

None

50 MIS of Update Data Button Controller Module (M45)

50.1 Module

UpdateDataButtonController

50.2 Uses

JsonFileWriter

50.3 Syntax

50.3.1 Exported Constants

None

50.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
updateData	string, \mathbb{Z} , string		

50.4 Semantics

50.4.1 State Variables

value: Boolean

plot: \mathbb{Z}

tree: string

50.4.2 Environment Variables

Mouse

UpdateDataPage

50.4.3 Assumptions

None

50.4.4 Access Routine Semantics

onClick():

- transition: updateData(value, plot, tree)

- output: None
- exception: None

updateData(value, plot, tree):

- transition: active:= JsonFileWriter.write(value, plot, tree)
- output: None
- exception: None

50.4.5 Local Functions

None

51 MIS of Contact Us Button Controller Module (M46)

51.1 Module

ContactUsButtonController

51.2 Uses

None

51.3 Syntax

51.3.1 Exported Constants

None

51.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
setActive	Boolean		

51.4 Semantics

51.4.1 State Variables

value: Boolean

active: Boolean

51.4.2 Environment Variables

Mouse

TeamInfoPage

51.4.3 Assumptions

None

51.4.4 Access Routine Semantics

onClick():

- transition: $\text{value} := \neg \text{value}$
- output: None
- exception: None

setActive(value):

- transition: active:= value
- output: None
- exception: None

51.4.5 Local Functions

None

52 MIS of Quit Button Controller (M47)

52.1 Module

QuitButtonController

52.2 Uses

UnityEngine.UI (UI Library)

52.3 Syntax

52.3.1 Exported Constants

None

52.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
QuitSoftware		terminate program	

52.4 Semantics

52.4.1 State Variables

None

52.4.2 Environment Variables

Mouse

52.4.3 Assumptions

None

52.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

QuitSoftware():

- transition: Application.Quit()
- output: None
- exception: None

52.4.5 Local Functions

None

53 MIS of Back Button Controller (M48)

53.1 Module

BackButtonController

53.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

53.3 Syntax

53.3.1 Exported Constants

None

53.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
Back			

53.4 Semantics

53.4.1 State Variables

viewState

upperLevelPage

53.4.2 Environment Variables

Mouse

53.4.3 Assumptions

None

53.4.4 Access Routine Semantics

Start():

- transition: None

- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

Back():

- transition: $\text{upperLevelPage} \implies (\text{viewState} := \text{upperLevelPage})$
- output: None
- exception: None

53.4.5 Local Functions

None

54 MIS of Plot Selection Drop Down Controller (M49)

54.1 Module

PlotSelectionDropDownController

54.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

54.3 Syntax

54.3.1 Exported Constants

None

54.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
displayMenu			
extractTreeParam	s: int		

54.4 Semantics

54.4.1 State Variables

isActive: Boolean

s1: String

s2: String

s3: String

s4: String

s5: String

currentIndex: int

54.4.2 Environment Variables

Mouse

DataModelObj: The gameobject of the current script

EnvDisp: UI text that will be displayed in Unity

54.4.3 Assumptions

None

54.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

displayMenu():

- transition: $\text{isActive} := \neg \text{isActive}$
- output: None
- exception: None

extractTreeParam(s):

- transition: Get the mouse click, assign different values to s1,s2,s3,s4,s5 based on the value of curIndex
- output: None
- exception: None

54.4.5 Local Functions

None

55 MIS of Tree Type Selection Drop Down Controller(M50)

55.1 Module

TreeTypeSelectionDropDownController

55.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

55.3 Syntax

55.3.1 Exported Constants

None

55.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
displayMenu			
extractTreeParam	s: int		

55.4 Semantics

55.4.1 State Variables

isActive: Boolean

currentIndex: int

s1: String

s2: String

s3: String

s4: String

s5: String

55.4.2 Environment Variables

Mouse

DataModelObj: The gameobject of the current script

TreeParamDisp: UI test that will be displayed in Unity dropdown: The drop down menu to select tree type

55.4.3 Assumptions

None

55.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

displayMenu():

- transition: $\text{isActive} := \neg \text{isActive}$
- output: None
- exception: None

extractTreeParam(s):

- transition: Get the mouse click, assign different values to s1,s2,s3,s4,s5 based on the value of curIndex
- output: None
- exception: None

55.4.5 Local Functions

None

56 MIS of Show Environmental Data Button Controller (M51)

56.1 Module

ShowEnvDataButtoController

56.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

56.3 Syntax

56.3.1 Exported Constants

None

56.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
EnvDataDispHandle			

56.4 Semantics

56.4.1 State Variables

displayEnvData: Boolean

56.4.2 Environment Variables

Mouse

56.4.3 Assumptions

None

56.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

EnvDataDispHandle():

- transition: $\text{displayEnvData} := \neg \text{displayEnvData}$
- output: None
- exception: None

56.4.5 Local Functions

None

57 MIS of Show Tree Parameter Button Controller(M52)

57.1 Module

ShowTreeParamButtonController

57.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

57.3 Syntax

57.3.1 Exported Constants

None

57.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
TreeParamDispHandle			

57.4 Semantics

57.4.1 State Variables

isActive: Boolean

57.4.2 Environment Variables

Mouse

57.4.3 Assumptions

None

57.4.4 Access Routine Semantics

Start():

- transition: None

- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

TreeParamDispHandle():

- transition: $\text{isActive} := \neg \text{isActive}$
- output: None
- exception: None

57.4.5 Local Functions

None

58 MIS of Environmental Selection Button Controller(M53)

58.1 Module

EnvDataSelectionButtonController

58.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

58.3 Syntax

58.3.1 Exported Constants

None

58.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
displayEnvSel			

58.4 Semantics

58.4.1 State Variables

isActive: Boolean

58.4.2 Environment Variables

Mouse

58.4.3 Assumptions

None

58.4.4 Access Routine Semantics

Start():

- transition: None

- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

displayEnvSel():

- transition: $\text{isActive} := \neg \text{isActive}$
- output: None
- exception: None

58.4.5 Local Functions

None

59 MIS of Data Type Selection Buttons Controller(M54)

59.1 Module

DataTypeSelectionButtonsController

59.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

59.3 Syntax

59.3.1 Exported Constants

None

59.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
displayDataTypeSel			

59.4 Semantics

59.4.1 State Variables

isActive: Boolean

59.4.2 Environment Variables

Mouse

59.4.3 Assumptions

None

59.4.4 Access Routine Semantics

Start():

- transition: None

- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

displayDataTypeSel():

- transition: $\text{isActive} := \neg \text{isActive}$
- output: None
- exception: None

59.4.5 Local Functions

None

60 MIS of Save Button Controller(M55)

60.1 Module

SaveButtonController

60.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

M39

60.3 Syntax

60.3.1 Exported Constants

None

60.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Start			
Update			
Save	originalData		
	updatedData		

60.4 Semantics

60.4.1 State Variables

originalData: float

updatedData: float

60.4.2 Environment Variables

None

60.4.3 Assumptions

None

60.4.4 Access Routine Semantics

Start():

- transition: None
- output: None
- exception: None

Update():

- transition: None
- output: None
- exception: None

Save():

- transition: originalData := updatedData
- output: None
- exception: None

60.4.5 Local Functions

None

References

- Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.
- Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.

61 Appendix