

Module Interface Specification for Digital Twin Forest

Team 8, Forest Mirror

Tingyu Shi

Jiacheng Wu

Junhong Chen

Yichen Jiang

Bowen Zhang

April 4, 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
DBH	Diameter at breast height
FR	Functional Requirement
GUI	Graphical User Interface
LAI	Leaf Area Index
M	Module
MG	Module Guide
MVC	Model, Viewer, Controller
NFR	Non-Functional Requirement
OS	Operating System
R	Requirement
SC	Scientific Computing
SRS	Software Requirements Specification
UC	Unlikely Change

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	2
6	MIS of Json File Reader (M1)	7
6.1	Module	7
6.2	Uses	7
6.3	Syntax	7
6.3.1	Exported Constants	7
6.3.2	Exported Access Programs	7
6.4	Semantics	7
6.4.1	State Variables	7
6.4.2	Environment Variables	8
6.4.3	Assumptions	8
6.4.4	Access Routine Semantics	8
6.4.5	Local Functions	9
7	MIS of Json File Writer (M2)	10
7.1	Module	10
7.2	Uses	10
7.3	Syntax	10
7.3.1	Exported Constants	10
7.3.2	Exported Access Programs	10
7.4	Semantics	10
7.4.1	State Variables	10
7.4.2	Environment Variables	10
7.4.3	Assumptions	11
7.4.4	Access Routine Semantics	11
7.4.5	Local Functions	12
8	MIS of Pause Manager (M3)	13
8.1	Module	13
8.2	Uses	13
8.3	Syntax	13
8.3.1	Exported Constants	13
8.3.2	Exported Access Programs	13

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

11.4.4	Access Routine Semantics	17
11.4.5	Local Functions	18
12	MIS of Instruction Button Controller (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	19
12.4.4	Access Routine Semantics	19
12.4.5	Local Functions	20
13	MIS of Update Data Button Controller (M8)	21
13.1	Module	21
13.2	Uses	21
13.3	Syntax	21
13.3.1	Exported Constants	21
13.3.2	Exported Access Programs	21
13.4	Semantics	21
13.4.1	State Variables	21
13.4.2	Environment Variables	21
13.4.3	Assumptions	21
13.4.4	Access Routine Semantics	22
13.4.5	Local Functions	22
14	MIS of Contact Us Button Controller (M9)	23
14.1	Module	23
14.2	Uses	23
14.3	Syntax	23
14.3.1	Exported Constants	23
14.3.2	Exported Access Programs	23
14.4	Semantics	23
14.4.1	State Variables	23
14.4.2	Environment Variables	23
14.4.3	Assumptions	23
14.4.4	Access Routine Semantics	23
14.4.5	Local Functions	24

15 MIS of Quit Button Controller (M10)	25
15.1 Module	25
15.2 Uses	25
15.3 Syntax	25
15.3.1 Exported Constants	25
15.3.2 Exported Access Programs	25
15.4 Semantics	25
15.4.1 State Variables	25
15.4.2 Environment Variables	25
15.4.3 Assumptions	25
15.4.4 Access Routine Semantics	25
15.4.5 Local Functions	26
16 MIS of Back Button Controller (M11)	27
16.1 Module	27
16.2 Uses	27
16.3 Syntax	27
16.3.1 Exported Constants	27
16.3.2 Exported Access Programs	27
16.4 Semantics	27
16.4.1 State Variables	27
16.4.2 Environment Variables	27
16.4.3 Assumptions	27
16.4.4 Access Routine Semantics	27
16.4.5 Local Functions	28
17 MIS of Plot Selection Drop Down Controller (M12)	29
17.1 Module	29
17.2 Uses	29
17.3 Syntax	29
17.3.1 Exported Constants	29
17.3.2 Exported Access Programs	29
17.4 Semantics	29
17.4.1 State Variables	29
17.4.2 Environment Variables	29
17.4.3 Assumptions	30
17.4.4 Access Routine Semantics	30
17.4.5 Local Functions	30
18 MIS of Tree Type Selection Drop Down Controller(M13)	31
18.1 Module	31
18.2 Uses	31
18.3 Syntax	31

18.3.1	Exported Constants	31
18.3.2	Exported Access Programs	31
18.4	Semantics	31
18.4.1	State Variables	31
18.4.2	Environment Variables	31
18.4.3	Assumptions	32
18.4.4	Access Routine Semantics	32
18.4.5	Local Functions	32
19	MIS of Show Environmental Data Button Controller (M14)	33
19.1	Module	33
19.2	Uses	33
19.3	Syntax	33
19.3.1	Exported Constants	33
19.3.2	Exported Access Programs	33
19.4	Semantics	33
19.4.1	State Variables	33
19.4.2	Environment Variables	33
19.4.3	Assumptions	33
19.4.4	Access Routine Semantics	33
19.4.5	Local Functions	34
20	MIS of Show Tree Parameter Button Controller(M15)	35
20.1	Module	35
20.2	Uses	35
20.3	Syntax	35
20.3.1	Exported Constants	35
20.3.2	Exported Access Programs	35
20.4	Semantics	35
20.4.1	State Variables	35
20.4.2	Environment Variables	35
20.4.3	Assumptions	35
20.4.4	Access Routine Semantics	35
20.4.5	Local Functions	36
21	MIS of Environmental Selection Button Controller(M16)	37
21.1	Module	37
21.2	Uses	37
21.3	Syntax	37
21.3.1	Exported Constants	37
21.3.2	Exported Access Programs	37
21.4	Semantics	37
21.4.1	State Variables	37

21.4.2	Environment Variables	37
21.4.3	Assumptions	37
21.4.4	Access Routine Semantics	37
21.4.5	Local Functions	38
22	MIS of Data Type Selection Buttons Controller(M17)	39
22.1	Module	39
22.2	Uses	39
22.3	Syntax	39
22.3.1	Exported Constants	39
22.3.2	Exported Access Programs	39
22.4	Semantics	39
22.4.1	State Variables	39
22.4.2	Environment Variables	39
22.4.3	Assumptions	39
22.4.4	Access Routine Semantics	39
22.4.5	Local Functions	40
23	MIS of Save Button Controller(M18)	41
23.1	Module	41
23.2	Uses	41
23.3	Syntax	41
23.3.1	Exported Constants	41
23.3.2	Exported Access Programs	41
23.4	Semantics	41
23.4.1	State Variables	41
23.4.2	Environment Variables	41
23.4.3	Assumptions	41
23.4.4	Access Routine Semantics	41
23.4.5	Local Functions	42
24	MIS of File Writer(M19)	43
24.1	Module	43
24.2	Uses	43
24.3	Syntax	43
24.3.1	Exported Constants	43
24.3.2	Exported Access Programs	43
24.4	Semantics	43
24.4.1	State Variables	43
24.4.2	Environment Variables	43
24.4.3	Assumptions	44
24.4.4	Access Routine Semantics	44
24.4.5	Local Functions	44

25 MIS of File Reader(M20)	46
25.1 Module	46
25.2 Uses	46
25.3 Syntax	46
25.3.1 Exported Constants	46
25.3.2 Exported Access Programs	46
25.4 Semantics	46
25.4.1 State Variables	46
25.4.2 Environment Variables	46
25.4.3 Assumptions	46
25.4.4 Access Routine Semantics	47
25.4.5 Local Functions	47
26 MIS of Season Change Controller(M21)	48
26.1 Module	48
26.2 Uses	48
26.3 Syntax	48
26.3.1 Exported Constants	48
26.3.2 Exported Access Programs	48
26.4 Semantics	48
26.4.1 State Variables	48
26.4.2 Environment Variables	48
26.4.3 Assumptions	48
26.4.4 Access Routine Semantics	48
26.4.5 Local Functions	49
27 MIS of move Panel Controller(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	51
27.4.5 Local Functions	51
28 MIS of Tree Planting Controller(M23)	52
28.1 Module	52
28.2 Uses	52
28.3 Syntax	52

28.3.1	Exported Constants	52
28.3.2	Exported Access Programs	52
28.4	Semantics	52
28.4.1	State Variables	52
28.4.2	Environment Variables	52
28.4.3	Assumptions	53
28.4.4	Access Routine Semantics	53
28.4.5	Local Functions	53
29	MIS of Tree Switch Button Controller(M24)	55
29.1	Module	55
29.2	Uses	55
29.3	Syntax	55
29.3.1	Exported Constants	55
29.3.2	Exported Access Programs	55
29.4	Semantics	55
29.4.1	State Variables	55
29.4.2	Environment Variables	55
29.4.3	Assumptions	55
29.4.4	Access Routine Semantics	55
29.4.5	Local Functions	56
30	MIS of Pie Chart Button Controller(M25)	57
30.1	Module	57
30.2	Uses	57
30.3	Syntax	57
30.3.1	Exported Constants	57
30.3.2	Exported Access Programs	57
30.4	Semantics	57
30.4.1	State Variables	57
30.4.2	Environment Variables	57
30.4.3	Assumptions	57
30.4.4	Access Routine Semantics	57
30.4.5	Local Functions	58
31	MIS of Pie Chart Controller(M26)	59
31.1	Module	59
31.2	Uses	59
31.3	Syntax	59
31.3.1	Exported Constants	59
31.3.2	Exported Access Programs	59
31.4	Semantics	59
31.4.1	State Variables	59

31.4.2	Environment Variables	59
31.4.3	Assumptions	60
31.4.4	Access Routine Semantics	60
31.4.5	Local Functions	60
32	MIS of Season Change Button Controller(M27)	62
32.1	Module	62
32.2	Uses	62
32.3	Syntax	62
32.3.1	Exported Constants	62
32.3.2	Exported Access Programs	62
32.4	Semantics	62
32.4.1	State Variables	62
32.4.2	Environment Variables	62
32.4.3	Assumptions	62
32.4.4	Access Routine Semantics	62
32.4.5	Local Functions	62
33	MIS of Tree Switch Controller(M28)	63
33.1	Module	63
33.2	Uses	63
33.3	Syntax	63
33.3.1	Exported Constants	63
33.3.2	Exported Access Programs	63
33.4	Semantics	63
33.4.1	State Variables	63
33.4.2	Environment Variables	63
33.4.3	Assumptions	63
33.4.4	Access Routine Semantics	63
33.4.5	Local Functions	64
34	MIS of Forest Trees (M29)	65
34.1	Module	65
34.2	Uses	65
34.3	Syntax	65
34.3.1	Exported Constants	65
34.3.2	Exported Access Programs	65
34.4	Semantics	65
34.4.1	State Variables	65
34.4.2	Environment Variables	65
34.4.3	Assumptions	65
34.4.4	Access Routine Semantics	65
34.4.5	Local Functions	66

35 MIS of Forest Sky (M30)	67
35.1 Module	67
35.2 Uses	67
35.3 Syntax	67
35.3.1 Exported Constants	67
35.3.2 Exported Access Programs	67
35.4 Semantics	67
35.4.1 State Variables	67
35.4.2 Environment Variables	67
35.4.3 Assumptions	67
35.4.4 Access Routine Semantics	67
35.4.5 Local Functions	68
36 MIS of Forest Terrain (M31)	69
36.1 Module	69
36.2 Uses	69
36.3 Syntax	69
36.3.1 Exported Constants	69
36.3.2 Exported Access Programs	69
36.4 Semantics	69
36.4.1 State Variables	69
36.4.2 Environment Variables	69
36.4.3 Assumptions	69
36.4.4 Access Routine Semantics	69
36.4.5 Local Functions	70
37 MIS of Red Pine (M32)	71
37.1 Module	71
37.2 Uses	71
37.3 Syntax	71
37.3.1 Exported Constants	71
37.3.2 Exported Access Programs	71
37.4 Semantics	72
37.4.1 State Variables	72
37.4.2 Environment Variables	72
37.4.3 Assumptions	72
37.4.4 Access Routine Semantics	72
37.4.5 Local Functions	73
38 MIS of Oak (M33)	74
38.1 Module	74
38.2 Uses	74
38.3 Syntax	74

38.3.1	Exported Constants	74
38.3.2	Exported Access Programs	74
38.4	Semantics	74
38.4.1	State Variables	74
38.4.2	Environment Variables	74
38.4.3	Assumptions	75
38.4.4	Access Routine Semantics	75
38.4.5	Local Functions	76
39	MIS of Beech (M34)	77
39.1	Module	77
39.2	Uses	77
39.3	Syntax	77
39.3.1	Exported Constants	77
39.3.2	Exported Access Programs	77
39.4	Semantics	77
39.4.1	State Variables	77
39.4.2	Environment Variables	77
39.4.3	Assumptions	78
39.4.4	Access Routine Semantics	78
39.4.5	Local Functions	79
40	MIS of Birch (M35)	80
40.1	Module	80
40.2	Uses	80
40.3	Syntax	80
40.3.1	Exported Constants	80
40.3.2	Exported Access Programs	80
40.4	Semantics	80
40.4.1	State Variables	80
40.4.2	Environment Variables	80
40.4.3	Assumptions	81
40.4.4	Access Routine Semantics	81
40.4.5	Local Functions	82
41	MIS of White Pine (M36)	83
41.1	Module	83
41.2	Uses	83
41.3	Syntax	83
41.3.1	Exported Constants	83
41.3.2	Exported Access Programs	83
41.4	Semantics	83
41.4.1	State Variables	83

41.4.2	Environment Variables	83
41.4.3	Assumptions	84
41.4.4	Access Routine Semantics	84
41.4.5	Local Functions	85
42	MIS of Red Maple (M37)	86
42.1	Module	86
42.2	Uses	86
42.3	Syntax	86
42.3.1	Exported Constants	86
42.3.2	Exported Access Programs	86
42.4	Semantics	86
42.4.1	State Variables	86
42.4.2	Environment Variables	86
42.4.3	Assumptions	87
42.4.4	Access Routine Semantics	87
42.4.5	Local Functions	88
43	MIS of Red Oak (M38)	89
43.1	Module	89
43.2	Uses	89
43.3	Syntax	89
43.3.1	Exported Constants	89
43.3.2	Exported Access Programs	89
43.4	Semantics	89
43.4.1	State Variables	89
43.4.2	Environment Variables	89
43.4.3	Assumptions	90
43.4.4	Access Routine Semantics	90
43.4.5	Local Functions	91
44	MIS of Environmental Data (M39)	92
44.1	Module	92
44.2	Uses	92
44.3	Syntax	92
44.3.1	Exported Constants	92
44.3.2	Exported Access Programs	92
44.4	Semantics	92
44.4.1	State Variables	92
44.4.2	Environment Variables	93
44.4.3	Assumptions	93
44.4.4	Access Routine Semantics	93
44.4.5	Local Functions	94

45 MIS of Plot Data (M40)	95
45.1 Module	95
45.2 Uses	95
45.3 Syntax	95
45.3.1 Exported Constants	95
45.3.2 Exported Access Programs	95
45.4 Semantics	96
45.4.1 State Variables	96
45.4.2 Environment Variables	96
45.4.3 Assumptions	96
45.4.4 Access Routine Semantics	96
45.4.5 Local Functions	99
46 MIS of First Person Player (M41)	100
46.1 Module	100
46.2 Uses	100
46.3 Syntax	100
46.4 Semantics	100
47 MIS of Json File (M42)	101
47.1 Module	101
47.2 Local Type	101
47.3 State Variables	101
47.4 Example	101
47.5 Local Functions	103
48 MIS of Main Page (M43)	104
48.1 Module	104
48.2 Uses	104
48.3 Syntax	104
48.3.1 Exported Constants	104
48.3.2 Exported Access Programs	104
48.4 Semantics	104
48.4.1 State Variables	104
48.4.2 Environment Variables	104
48.4.3 Assumptions	104
48.4.4 Access Routine Semantics	104
48.4.5 Local Functions	104
49 MIS of Start Button (M44)	105
49.1 Module	105
49.2 Uses	105
49.3 Syntax	105

49.3.1	Exported Constants	105
49.3.2	Exported Access Programs	105
49.4	Semantics	105
49.4.1	State Variables	105
49.4.2	Environment Variables	105
49.4.3	Assumptions	105
49.4.4	Access Routine Semantics	105
49.4.5	Local Functions	105
50	MIS of Instruction Button (M45)	106
50.1	Module	106
50.2	Uses	106
50.3	Syntax	106
50.3.1	Exported Constants	106
50.3.2	Exported Access Programs	106
50.4	Semantics	106
50.4.1	State Variables	106
50.4.2	Environment Variables	106
50.4.3	Assumptions	106
50.4.4	Access Routine Semantics	106
50.4.5	Local Functions	106
51	MIS of Contact Us Button (M46)	107
51.1	Module	107
51.2	Uses	107
51.3	Syntax	107
51.3.1	Exported Constants	107
51.3.2	Exported Access Programs	107
51.4	Semantics	107
51.4.1	State Variables	107
51.4.2	Environment Variables	107
51.4.3	Assumptions	107
51.4.4	Access Routine Semantics	107
51.4.5	Local Functions	107
52	MIS of Quit Button (M47)	108
52.1	Module	108
52.2	Uses	108
52.3	Syntax	108
52.3.1	Exported Constants	108
52.3.2	Exported Access Programs	108
52.4	Semantics	108
52.4.1	State Variables	108

52.4.2	Environment Variables	108
52.4.3	Assumptions	108
52.4.4	Access Routine Semantics	108
52.4.5	Local Functions	108
53	MIS of Instruction Page (M48)	109
53.1	Module	109
53.2	Uses	109
53.3	Syntax	109
53.3.1	Exported Constants	109
53.3.2	Exported Access Programs	109
53.4	Semantics	109
53.4.1	State Variables	109
53.4.2	Environment Variables	109
53.4.3	Assumptions	109
53.4.4	Access Routine Semantics	109
53.4.5	Local Functions	109
54	MIS of Contact Us Page (M49)	110
54.1	Module	110
54.2	Uses	110
54.3	Syntax	110
54.3.1	Exported Constants	110
54.3.2	Exported Access Programs	110
54.4	Semantics	110
54.4.1	State Variables	110
54.4.2	Environment Variables	110
54.4.3	Assumptions	110
54.4.4	Access Routine Semantics	110
54.4.5	Local Functions	110
55	MIS of Back Button (M50)	111
55.1	Module	111
55.2	Uses	111
55.3	Syntax	111
55.3.1	Exported Constants	111
55.3.2	Exported Access Programs	111
55.4	Semantics	111
55.4.1	State Variables	111
55.4.2	Environment Variables	111
55.4.3	Assumptions	111
55.4.4	Access Routine Semantics	111
55.4.5	Local Functions	111

56 MIS of Update Data Page (M51)	112
56.1 Module	112
56.2 Uses	112
56.3 Syntax	112
56.3.1 Exported Constants	112
56.3.2 Exported Access Programs	112
56.4 Semantics	112
56.4.1 State Variables	112
56.4.2 Environment Variables	112
56.4.3 Assumptions	112
56.4.4 Access Routine Semantics	112
56.4.5 Local Functions	112
57 MIS of Environmental Data Selection Button (M52)	113
57.1 Module	113
57.2 Uses	113
57.3 Syntax	113
57.3.1 Exported Constants	113
57.3.2 Exported Access Programs	113
57.4 Semantics	113
57.4.1 State Variables	113
57.4.2 Environment Variables	113
57.4.3 Assumptions	113
57.4.4 Access Routine Semantics	113
57.4.5 Local Functions	113
58 MIS of Data Type Selection Button (M53)	114
58.1 Module	114
58.2 Uses	114
58.3 Syntax	114
58.3.1 Exported Constants	114
58.3.2 Exported Access Programs	114
58.4 Semantics	114
58.4.1 State Variables	114
58.4.2 Environment Variables	114
58.4.3 Assumptions	114
58.4.4 Access Routine Semantics	114
58.4.5 Local Functions	114
59 MIS of New Data Input Box (M54)	115
59.1 Module	115
59.2 Uses	115
59.3 Syntax	115

59.3.1	Exported Constants	115
59.3.2	Exported Access Programs	115
59.4	Semantics	115
59.4.1	State Variables	115
59.4.2	Environment Variables	115
59.4.3	Assumptions	115
59.4.4	Access Routine Semantics	115
59.4.5	Local Functions	115
60	MIS of Save Button (M55)	116
60.1	Module	116
60.2	Uses	116
60.3	Syntax	116
60.3.1	Exported Constants	116
60.3.2	Exported Access Programs	116
60.4	Semantics	116
60.4.1	State Variables	116
60.4.2	Environment Variables	116
60.4.3	Assumptions	116
60.4.4	Access Routine Semantics	116
60.4.5	Local Functions	116
61	MIS of Current Data Dispaly (M56)	117
61.1	Module	117
61.2	Uses	117
61.3	Syntax	117
61.3.1	Exported Constants	117
61.3.2	Exported Access Programs	117
61.4	Semantics	117
61.4.1	State Variables	117
61.4.2	Environment Variables	117
61.4.3	Assumptions	117
61.4.4	Access Routine Semantics	117
61.4.5	Local Functions	117
62	MIS of Plot Selection Drop Down (M57)	118
62.1	Module	118
62.2	Uses	118
62.3	Syntax	118
62.3.1	Exported Constants	118
62.3.2	Exported Access Programs	118
62.4	Semantics	118
62.4.1	State Variables	118

62.4.2	Environment Variables	118
62.4.3	Assumptions	118
62.4.4	Access Routine Semantics	118
62.4.5	Local Functions	118
63	MIS of Tree Type Selection Drop Down (M58)	119
63.1	Module	119
63.2	Uses	119
63.3	Syntax	119
63.3.1	Exported Constants	119
63.3.2	Exported Access Programs	119
63.4	Semantics	119
63.4.1	State Variables	119
63.4.2	Environment Variables	119
63.4.3	Assumptions	119
63.4.4	Access Routine Semantics	119
63.4.5	Local Functions	119
64	MIS of Update Data Button (M59)	120
64.1	Module	120
64.2	Uses	120
64.3	Syntax	120
64.3.1	Exported Constants	120
64.3.2	Exported Access Programs	120
64.4	Semantics	120
64.4.1	State Variables	120
64.4.2	Environment Variables	120
64.4.3	Assumptions	120
64.4.4	Access Routine Semantics	120
64.4.5	Local Functions	120
65	MIS of Forest Dispaly (M60)	121
65.1	Module	121
65.2	Uses	121
65.3	Syntax	121
65.3.1	Exported Constants	121
65.3.2	Exported Access Programs	121
65.4	Semantics	121
65.4.1	State Variables	121
65.4.2	Environment Variables	121
65.4.3	Assumptions	121
65.4.4	Access Routine Semantics	121
65.4.5	Local Functions	121

66 MIS of Show Environmental Data Button (M61)	122
66.1 Module	122
66.2 Uses	122
66.3 Syntax	122
66.3.1 Exported Constants	122
66.3.2 Exported Access Programs	122
66.4 Semantics	122
66.4.1 State Variables	122
66.4.2 Environment Variables	122
66.4.3 Assumptions	122
66.4.4 Access Routine Semantics	122
66.4.5 Local Functions	122
67 MIS of Show Tree Parameters Button (M62)	123
67.1 Module	123
67.2 Uses	123
67.3 Syntax	123
67.3.1 Exported Constants	123
67.3.2 Exported Access Programs	123
67.4 Semantics	123
67.4.1 State Variables	123
67.4.2 Environment Variables	123
67.4.3 Assumptions	123
67.4.4 Access Routine Semantics	123
67.4.5 Local Functions	123
68 MIS of Environment Data Display (M63)	124
68.1 Module	124
68.2 Uses	124
68.3 Syntax	124
68.3.1 Exported Constants	124
68.3.2 Exported Access Programs	124
68.4 Semantics	124
68.4.1 State Variables	124
68.4.2 Environment Variables	124
68.4.3 Assumptions	124
68.4.4 Access Routine Semantics	124
68.4.5 Local Functions	124
69 MIS of Tree Parameters Display (M64)	125
69.1 Module	125
69.2 Uses	125
69.3 Syntax	125

69.3.1	Exported Constants	125
69.3.2	Exported Access Programs	125
69.4	Semantics	125
69.4.1	State Variables	125
69.4.2	Environment Variables	125
69.4.3	Assumptions	125
69.4.4	Access Routine Semantics	125
69.4.5	Local Functions	125
70	MIS of Pause Indicator (M65)	126
70.1	Module	126
70.2	Uses	126
70.3	Syntax	126
70.3.1	Exported Constants	126
70.3.2	Exported Access Programs	126
70.4	Semantics	126
70.4.1	State Variables	126
70.4.2	Environment Variables	126
70.4.3	Assumptions	126
70.4.4	Access Routine Semantics	126
70.4.5	Local Functions	126
71	MIS of SeasonChangeButton (M66)	127
71.1	Module	127
71.2	Uses	127
71.3	Syntax	127
71.3.1	Exported Constants	127
71.3.2	Exported Access Programs	127
71.4	Semantics	127
71.4.1	State Variables	127
71.4.2	Environment Variables	127
71.4.3	Assumptions	127
71.4.4	Access Routine Semantics	127
71.4.5	Local Functions	127
72	MIS of pieChartButton (M67)	128
72.1	Module	128
72.2	Uses	128
72.3	Syntax	128
72.3.1	Exported Constants	128
72.3.2	Exported Access Programs	128
72.4	Semantics	128
72.4.1	State Variables	128

72.4.2	Environment Variables	128
72.4.3	Assumptions	128
72.4.4	Access Routine Semantics	128
72.4.5	Local Functions	128
73	MIS of TreeSwitchButton (M68)	129
73.1	Module	129
73.2	Uses	129
73.3	Syntax	129
73.3.1	Exported Constants	129
73.3.2	Exported Access Programs	129
73.4	Semantics	129
73.4.1	State Variables	129
73.4.2	Environment Variables	129
73.4.3	Assumptions	129
73.4.4	Access Routine Semantics	129
73.4.5	Local Functions	129

3 Introduction

The following document outlines the Module Interface Specifications for Digital Twin Forest, which is a virtual representation of the real world, including physical objects, processes, relationships, and behaviors. Components of a digital twin encompass data capture and integration, visualization, and advanced analysis including AI, automation, information sharing and collaboration. This project can benefit two groups of users: forest owners who can utilize the software for forest management and meteorologists who can use it for research purposes. Complementary documents include the [System Requirement Specifications](#) and [Module Guide](#).

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
string	String	a sequence of characters
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(First Controller Table)

Level 1	Level 2
Controller Modules	M1: JsonFileReader
	M2: JsonFileWriter
	M3: PauseManager
	M4: PlayerMovement
	M5: NewDataInputBoxController
	M6: StartButtonController
	M7: InstructionButtonController
	M8: UpdateDataButtonController
	M9: ContactUsButtonController
	M10: QuitButtonController
	M11: BackButtonController
	M12: PlotSelectionDropDownController
	M13: TreeTypeSelectionDropDownController
	M14: ShowEnvDataButtonController
	M15: ShowTreeParamButtonController
	M16: EnvDataSelectionButtonController
	M17: DataTypeSelectionButtonsController
	M18: SaveButtonController

Table 2: Module Hierarchy(Second Controller Table)

Level 1	Level 2
Controller Modules	M19: FileWriter
	M20: FileReader
	M21: SeasonChangeController
	M22: movePanelController
	M23: treePlantingController
	M24: TreeSwitchButtonController
	M25: pieChartButtonController
	M26: pieChartController
	M27: SeasonChangeButtonController
	M28: TreeSwitchController

Table 3: Module Hierarchy(Models)

Level 1	Level 2
Model Modules	M29: ForestTrees
	M30: ForestSky
	M31: ForestTerrain
	M32: RedPine
	M33: Oak
	M34: Beech
	M35: Birch
	M36: WhitePine
	M37: RedMaple
	M38: RedOak
	M39: EnvData
	M40: PlotData
	M41: FirstPersonPlayer
	M42: JsonFile

Table 4: Module Hierarchy(First Viewers Table)

Level 1	Level 2
Viewer Modules	M43: MainPageDisplay
	M44: StartButton
	M45: InstructionButton
	M46: ContactUsButton
	M47: QuitButton
	M48: InstructionInfoDisplay
	M49: ContactUsInfoDisplay
	M50: BackButton
	M51: UpdateDataDisplay
	M52: EnvDataSelectionButton
	M53: DataTypeSelectionButtons
	M54: NewDataInputBox
	M55: SaveButton

Table 5: Module Hierarchy(Second Viewers Table)

Level 1	Level 2
Viewer Modules	M56: CurrentDataDisplay
	M57: PlotSelectionDropDown
	M58: TreeTypeSelectionDropDown
	M59: UpdateDataButton
	M60: ForestDisplay
	M61: ShowEnvDataButton
	M62: ShowTreeParamButton
	M63: EnvDataDisplay
	M64: TreeParamDisplay
	M65: PauseIndicatorDisplay
	M66: SeasonChangeButton
	M67: pieChartButton
	M68: TreeSwitchButton

6 MIS of Json File Reader (M1)

6.1 Module

JsonFileReader

6.2 Uses

UnityEngine

System.IO

UnityEngine.UI

M64

M63

M23

M26

M40

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
Awake			
Start			
readfile	\mathbb{Z}		

6.4 Semantics

6.4.1 State Variables

treeParamDisplay: TreeParamDisplay

envDataDisplay: EnvDataDisplay

graphMaker: pieChartMaker

treePlanter: treePlanting

DataModelObj: DataModel

JsonModelObj: JsonModel

plotNumber: \mathbb{Z}

filePath: string

plotJsonData: string

6.4.2 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

6.4.3 Assumptions

None

6.4.4 Access Routine Semantics

Awake():

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

Start():

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

readfile(value):

- transition: Open the JSON file according to value, read all the content from the JSON file, update tree parameters of treeParamDisplay, update environmental data in envDataDisplay, invoke the markChart() function of graphMaker, and invoke plantTrees() function of treePlanter
- output: None
- exception: None

6.4.5 Local Functions

None

7 MIS of Json File Writer (M2)

7.1 Module

JsonFileWriter

7.2 Uses

UnityEngine

TMPro

Text

UnityEngine.UI

M1

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
Awake			
updateHandler			InvalidInputException
writeFile	String		
removeMsg			
isValid	String	Boolean	
isIN	char, List<char>	Boolean	

7.4 Semantics

7.4.1 State Variables

indicator: ValueIndicator

inputField: TMP.InputField

notify: Text

FR: FileReader

7.4.2 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

7.4.3 Assumptions

None

7.4.4 Access Routine Semantics

Awake():

- transition: Invoke removeMsg()
- output: None
- exception: None

updateHandler():

- transition: Validate the input of the text field and invoke writeFile function
- output: None
- exception: Throw InvalidInputException if the inputs are invalid

writeFile(newValue):

- transition: Import the data from the text field and use it to rewrite the data stored in the JSON files based on the attributes of the indicator
- output: None
- exception: None

removeMsg():

- transition: notify.text := ""

- output: None
- exception: None

isValid(s):

- transition: Check whether all the characters in s are in the pool
- output: True if all the characters in s are in the pool. False otherwise
- exception: None

isIN(target, pool):

- transition: Find an element of pool that matches target
- output: True if that element exists. False otherwise
- exception: None

7.4.5 Local Functions

None

8 MIS of Pause Manager (M3)

8.1 Module

PauseManager

8.2 Uses

UnityEngine

UnityEngine.UI

8.3 Syntax

8.3.1 Exported Constants

None

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
Start			
Update			

8.4 Semantics

8.4.1 State Variables

isPaused: Boolean

pauseIndicator: Image

8.4.2 Environment Variables

None

8.4.3 Assumptions

None

8.4.4 Access Routine Semantics

Start():

- transition: enable the visibility of the Pause image

- output: None
- exception: None

Update():

- transition: Capture the event of pressing the K-key, change the value of isPaused, and enable or disable the visibility of the Pause image
- output: None
- exception: None

8.4.5 Local Functions

None

9 MIS of Player Movement(M4)

9.1 Module

PlayerMovement

9.2 Uses

UnityEngine

9.3 Syntax

9.3.1 Exported Constants

None

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
Update			

9.4 Semantics

9.4.1 State Variables

controller: CharacterController speed: \mathbb{R}

9.4.2 Environment Variables

None

9.4.3 Assumptions

None

9.4.4 Access Routine Semantics

- transition: Change the position of the camera in each frame
- output: None
- exception: None

9.4.5 Local Functions

None

10 MIS of New Data Input Box Controller (M5)

10.1 Module

NewDataInputBoxController

10.2 Uses

UnityEngine

10.3 Syntax

10.3.1 Exported Constants

None

10.3.2 Exported Access Programs

None

10.4 Semantics

10.4.1 State Variables

None

10.4.2 Environment Variables

None

10.4.3 Assumptions

None

10.4.4 Access Routine Semantics

None

10.4.5 Local Functions

None

11 MIS of Start Button Controller (M6)

11.1 Module

StartButtonController

11.2 Uses

UnityEngine

11.3 Syntax

11.3.1 Exported Constants

None

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
OnClick			

11.4 Semantics

11.4.1 State Variables

None

11.4.2 Environment Variables

None

11.4.3 Assumptions

None

11.4.4 Access Routine Semantics

OnClick():

- transition: Enable the visibility of M60
- output: None
- exception: None

11.4.5 Local Functions

None

12 MIS of Instruction Button Controller (M7)

12.1 Module

InstructionButtonController

12.2 Uses

None

12.3 Syntax

12.3.1 Exported Constants

None

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick			

12.4 Semantics

12.4.1 State Variables

None

12.4.2 Environment Variables

None

12.4.3 Assumptions

None

12.4.4 Access Routine Semantics

onClick():

- transition: Enable the visibility of M48
- output: None
- exception: None

12.4.5 Local Functions

None

13 MIS of Update Data Button Controller (M8)

13.1 Module

UpdateDataButtonController

13.2 Uses

UnityEngine
UnityEngine.UI
Json File Reader
Value Indicator
M1

13.3 Syntax

13.3.1 Exported Constants

None

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
UpdateEnvDataButtonHandle UpdateTreeParamHandle onClick			

13.4 Semantics

13.4.1 State Variables

EnvDataOptions: Canvas
TreeParamOptions: Canvas
treeSelection: Dropdown
indicator: ValueIndicator
FR: FileReader

13.4.2 Environment Variables

None

13.4.3 Assumptions

None

13.4.4 Access Routine Semantics

UpdateEnvDataButtonHandle():

- transition: Change the isEnvData and isTreeParam variables in the indicator.
- output: None
- exception: None

UpdateTreeParamHande():

- transition: Change the isTreeParam and isEnvData, in the indicator, and also change the TreeType attribute in the indicator according to the value of the Dropdown class.
- output: None
- exception: None

onClick():

- transition: Invoke UpdateEnvDataButtonHandle() and UpdateTreeParamHande()
- output: None
- exception: None

13.4.5 Local Functions

None

14 MIS of Contact Us Button Controller (M9)

14.1 Module

ContactUsButtonController

14.2 Uses

UnityEngine

14.3 Syntax

14.3.1 Exported Constants

None

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick			

14.4 Semantics

14.4.1 State Variables

None

14.4.2 Environment Variables

None

14.4.3 Assumptions

None

14.4.4 Access Routine Semantics

onClick():

- transition: Enable the visibility of M49
- output: None
- exception: None

14.4.5 Local Functions

None

15 MIS of Quit Button Controller (M10)

15.1 Module

QuitButtonController

15.2 Uses

UnityEngine

15.3 Syntax

15.3.1 Exported Constants

None

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
QuitSoftware			
onClick			

15.4 Semantics

15.4.1 State Variables

None

15.4.2 Environment Variables

None

15.4.3 Assumptions

None

15.4.4 Access Routine Semantics

QuitSoftware():

- transition: Quit the software.
- output: None
- exception: None

onClick():

- transition: Invoke QuitSoftware()
- output: None
- exception: None

15.4.5 Local Functions

None

16 MIS of Back Button Controller (M11)

16.1 Module

BackButtonController

16.2 Uses

UnityEngine

UnityEngine.SceneManagement

16.3 Syntax

16.3.1 Exported Constants

None

16.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Back			

16.4 Semantics

16.4.1 State Variables

viewState

upperLevelPage

16.4.2 Environment Variables

Mouse

16.4.3 Assumptions

None

16.4.4 Access Routine Semantics

Back():

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

- exception: None

16.4.5 Local Functions

None

17 MIS of Plot Selection Drop Down Controller (M12)

17.1 Module

PlotSelectionDropDownController

17.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

System.Threading

17.3 Syntax

17.3.1 Exported Constants

None

17.3.2 Exported Access Programs

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

17.4 Semantics

17.4.1 State Variables

isActive: Boolean

s1: String

s2: String

s3: String

s4: String

s5: String

currentIndex: int

17.4.2 Environment Variables

Mouse

DataModelObj: The gameobject of the current script

EnvDisp: Interface that will be displayed in Unity

dropDown: Drop down menu to select plot

17.4.3 Assumptions

None

17.4.4 Access Routine Semantics

displayMenu():

- transition: isActive:= \neg 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

17.4.5 Local Functions

None

18 MIS of Tree Type Selection Drop Down Controller(M13)

18.1 Module

TreeTypeSelectionDropDownController

18.2 Uses

UnityEngine.UI

UnityEngine.SceneManagement

System.Threading

18.3 Syntax

18.3.1 Exported Constants

None

18.3.2 Exported Access Programs

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

18.4 Semantics

18.4.1 State Variables

isActive: Boolean

curIndex: int

s1: String

s2: String

s3: String

s4: String

18.4.2 Environment Variables

Mouse

DataModelObj: The gameobject of the current script

TreeParamDisp: Interface that will be displayed in Unity

dropdown: The drop down menu to select tree type

18.4.3 Assumptions

None

18.4.4 Access Routine Semantics

displayMenu():

- transition: isActive:= \neg isActive
- output: None
- exception: None

extractTreeParam(s):

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

18.4.5 Local Functions

None

19 MIS of Show Environmental Data Button Controller (M14)

19.1 Module

ShowEnvDataButtoController

19.2 Uses

UnityEngine
UnityEngine.UI

19.3 Syntax

19.3.1 Exported Constants

None

19.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
EnvDataDispHandle			

19.4 Semantics

19.4.1 State Variables

displayEnvData: Boolean

19.4.2 Environment Variables

Mouse

19.4.3 Assumptions

None

19.4.4 Access Routine Semantics

EnvDataDispHandle():

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

- exception: None

19.4.5 Local Functions

None

20 MIS of Show Tree Parameter Button Controller(M15)

20.1 Module

ShowTreeParamButtonController

20.2 Uses

UnityEngine
UnityEngine.UI

20.3 Syntax

20.3.1 Exported Constants

None

20.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
TreeParamDispHandle			

20.4 Semantics

20.4.1 State Variables

isActive: Boolean

20.4.2 Environment Variables

Mouse

20.4.3 Assumptions

None

20.4.4 Access Routine Semantics

TreeParamDispHandle():

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

20.4.5 Local Functions

None

21 MIS of Environmental Selection Button Controller(M16)

21.1 Module

EnvDataSelectionButtonController

21.2 Uses

UnityEngine
UnityEngine.UI

21.3 Syntax

21.3.1 Exported Constants

None

21.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
displayEnvSel			

21.4 Semantics

21.4.1 State Variables

isActive: Boolean

21.4.2 Environment Variables

Mouse

21.4.3 Assumptions

None

21.4.4 Access Routine Semantics

displayEnvSel():

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

21.4.5 Local Functions

None

22 MIS of Data Type Selection Buttons Controller(M17)

22.1 Module

DataTypeSelectionButtonsController

22.2 Uses

UnityEngine
UnityEngine.UI

22.3 Syntax

22.3.1 Exported Constants

None

22.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
displayDataTypeSel			

22.4 Semantics

22.4.1 State Variables

isActive: Boolean

22.4.2 Environment Variables

Mouse

22.4.3 Assumptions

None

22.4.4 Access Routine Semantics

displayDataTypeSel():

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

22.4.5 Local Functions

None

23 MIS of Save Button Controller(M18)

23.1 Module

SaveButtonController

23.2 Uses

UnityEngine
UnityEngine.UI
M19

23.3 Syntax

23.3.1 Exported Constants

None

23.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick	mouse click		
Save	updatedData: float		

23.4 Semantics

23.4.1 State Variables

originalData: float
updatedData: float

23.4.2 Environment Variables

Mouse

23.4.3 Assumptions

None

23.4.4 Access Routine Semantics

Save():

- transition: originalData := updatedData
- output: None

- exception: None

23.4.5 Local Functions

None

24 MIS of File Writer(M19)

24.1 Module

FileWriter

24.2 Uses

UnityEngine

UnityEngine.UI

TMPro

System.IO

Newtonsoft.Json

24.3 Syntax

24.3.1 Exported Constants

None

24.3.2 Exported Access Programs

Name	In	Out	Exceptions
Awake			
updateHandler			InvalidInput
writeFile	inputText: String		

24.4 Semantics

24.4.1 State Variables

indicator: Forest data type of the input

inputField: Textbox of the input

inputText: input to the textbox

notify: text

FR: FileReader

JSON file

24.4.2 Environment Variables

None

24.4.3 Assumptions

None

24.4.4 Access Routine Semantics

Awake():

- transition: notify := NULL
- output: None
- exception: None

updateHandler():

- transition: notify := “Updated”
inputField := NULL
Update the JSON file if the input is valid.
- output: None
- exception: $inputText == NULL \vee inputText \in \{(a, A), (z, Z)\}$

writeFile(inputText):

- transition: Use inputText to update the JSON files
- output: None
- exception: $inputText == NULL \vee inputText \in \{(a, A), (z, Z)\}$

24.4.5 Local Functions

removemsg():

- transition: notify := NULL
- output: None
- exception: None

isValid(s: string):

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

isIN(target: char, pool: char[]):

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

25 MIS of File Reader(M20)

25.1 Module

FileWriter

25.2 Uses

UnityEngine

UnityEngine.UI

TMPro

System.IO

Newtonsoft.Json

25.3 Syntax

25.3.1 Exported Constants

None

25.3.2 Exported Access Programs

Name	In	Out	Exceptions
readEnvData			
readTreeParam			
clearText			

25.4 Semantics

25.4.1 State Variables

indicator: Forest data type of the input

JsonModelObj:

currentValueDisp: Output shown JSON file

25.4.2 Environment Variables

None

25.4.3 Assumptions

None

25.4.4 Access Routine Semantics

readEnvData():

- transition: filePath := "../dataCenter/plot" + plotNumber.ToString() + "data.json";
plotJsonData := File.ReadAllText(filePath);
JsonModelObj := Newtonsoft.Json.JsonConvert.DeserializeObject<JsonModel>(plotJsonData);
EnvDataType := indicator.EnvDataType;
currentValueDisp.text := currentValueDisp.text + JsonModelObj.envDataType.EnvDataType
- output: None
- exception: None

readTreeParam():

- transition: filePath := "../dataCenter/plot" + plotNumber.ToString() + "data.json";
plotJsonData := File.ReadAllText(filePath);
JsonModelObj := Newtonsoft.Json.JsonConvert.DeserializeObject<JsonModel>(plotJsonData);
TreeType := indicator.TreeType;
TreeParamType := indicator.TreeParamType;
currentValueDisp.text := currentValueDisp.text + JsonModelObj.TreeType.TreeParamType
- output: None
- exception: None

clearText():

- transition: currentValueDisp.text := None;
- output: None
- exception: None

25.4.5 Local Functions

None

26 MIS of Season Change Controller(M21)

26.1 Module

SeasonChangeController

26.2 Uses

UnityEngine

UnityEngine.UI

treePlantingController

SeasonChangeButtonController

26.3 Syntax

26.3.1 Exported Constants

None

26.3.2 Exported Access Programs

Name	In	Out	Exceptions
Awake			
changeSeason			

26.4 Semantics

26.4.1 State Variables

haveLeaves: If the trees have leaves

seasonChangeBut: Season Change button

Snow: Particle System for snowing effects

26.4.2 Environment Variables

None

26.4.3 Assumptions

None

26.4.4 Access Routine Semantics

Awake():

- transition: `seasonChangeBut.image.sprite := summerImage;`
`Snow.gameObject.SetActive(false)`
- output: `None`
- exception: `None`

`changeSeason():`

- transition: `haveLeaves := !haveLeaves;`
`(haveLeaves \implies Snow.gameObject.SetActive(false) \wedge (seasonChangeBut.image.sprite := summerImage));`
`(!haveLeaves \implies Snow.gameObject.SetActive(true) \wedge (seasonChangeBut.image.sprite := winterImage))`
- output: `None`
- exception: `None`

26.4.5 Local Functions

`None`

27 MIS of move Panel Controller(M22)

27.1 Module

movePanelController

27.2 Uses

27.3 Syntax

27.3.1 Exported Constants

None

27.3.2 Exported Access Programs

Name	In	Out	Exceptions
myClick			
Update			

27.4 Semantics

27.4.1 State Variables

27.4.2 Environment Variables

```
float speed = 2000f;  
float speed2 = -2000f;  
bool isActive = false;  
bool isShown = false;  
bool isActive2 = false;  
bool isShown2 = false;  
int l.boundary = -400;  
int r.boundary = 350;  
int l.boundary2 = 3100;  
int r.boundary2 = 3850;  
GameObject go;  
GameObject go2;
```

27.4.3 Assumptions

None

27.4.4 Access Routine Semantics

myClick():

- transition: $(\text{atBoundary}() \implies (\text{isActive} := \text{true} \wedge \text{isShown} := \text{false}))$
 $\wedge (!\text{atBoundary}() \implies (\text{isActive} := \text{false} \wedge \text{isShown} := \text{true}))$
- output: None
- exception: None

Update():

- transition: $((\text{isActive} \wedge \text{isShown}) \implies \text{go.transform.Translate}(\text{speed} * \text{Time.deltaTime}, 0, 0) \wedge ((\text{go.transform.position.x} \leq \text{r_boundary}) \implies (\text{isShown} := \text{true})))$
 $((!\text{isActive} \wedge \text{isShown}) \implies \text{go.transform.Translate}(-\text{speed} * \text{Time.deltaTime}, 0, 0);$
 $\wedge ((\text{go.transform.position.x} \leq \text{l_boundary}) \implies (\text{isShown} := \text{true})))$
 $((\text{isActive2} \wedge \text{isShown2}) \implies \text{go2.transform.Translate}(\text{speed2} * \text{Time.deltaTime}, 0,$
 $0) \wedge ((\text{go2.transform.position.x} \leq \text{l_boundary2}) \implies (\text{isShown} := \text{true})))$
 $((!\text{isActive2} \wedge \text{isShown2}) \implies \text{go2.transform.Translate}(-\text{speed2} * \text{Time.deltaTime}, 0,$
 $0); \wedge ((\text{go2.transform.position.x} \leq \text{r_boundary2}) \implies (\text{isShown} := \text{true})))$
- output: None
- exception: None

27.4.5 Local Functions

atBoundary():

- transition: $x := \text{go.transform.position.x}$
- output: $x \leq \text{l_boundary}$
- exception: None

28 MIS of Tree Planting Controller(M23)

28.1 Module

treePlantingController

28.2 Uses

UnityEngine

UnityEngine.UI

seasonController

28.3 Syntax

28.3.1 Exported Constants

None

28.3.2 Exported Access Programs

Name	In	Out	Exceptions
plantTrees	plotNumberIndex		
cleanTrees			

28.4 Semantics

28.4.1 State Variables

28.4.2 Environment Variables

Related measures in unity tree editor:

double redPineHSR = 25.0;

double oakHSR = 17.2 / 2;

double beechHSR = 35.47 / 2;

double birchHSR = 22.18 / 2;

double redMapleHSR = 14.12 / 2;

double whitePineHSR = 64.4 / 2;

double redOakHSR = 18.95 / 2

Plot Information:

double sideLength = 100.0;

float startingCoordinate = 1f;

float endingCoordinate = 99f

Collection of tree positions:

Vector3[] treelocal

Treepfabs with leaves:

List< *GameObject* >treeprefabsWL

Treepfabs without leaves:

List< *GameObject* > treeprefabsWOL

ListOfCircles circles[plotNumberIndex]

28.4.3 Assumptions

None

28.4.4 Access Routine Semantics

plantTrees(plotNumberIndex):

- transition: generateCircleLocation(circles(plotNumberIndex),differentTreeNumbers.Sum());
(seasonController.haceLeaves \implies treeprefabWL);
(!seasonController.haveLeaves \implies treeprefabWOL);
(\forall species \implies standardScale := data.species.Height/HSR \wedge lowerBound := 0.9 *
standardScale \wedge upperBound := 1.1 * standardScale);
(\forall species \implies trees.Add(Instantiate(treeprefabs | treeinstance.transform.localscale
= Vector3(randomScale, randomScale, randomScale)))

- output: None
- exception: None

cleanTrees():

- transition: \forall tree \in trees \implies Destroy(tree);
- output: None
- exception: None

28.4.5 Local Functions

calculatedTreeNumbers():

- transition: area := sideLength *sideLength;
(int) treenumbers = $\sum_{species}$ density*area

- output: treenumbers
- exception: None

generateSquareLocation(numberOfPoints):

- transition: `treelocal[i] = Vector3(UnityEngine.Random.Range(startingCoordinate, endingCoordinate), 0f, UnityEngine.Random.Range(startingCoordinate, endingCoordinate))`
- output: None
- exception: None

generateCircleLocation(List<Circle> circles, int numberOfPoints):

- transition: `Point randomPoint := Point(UnityEngine.Random.Range(startingCoordinate, endingCoordinate), UnityEngine.Random.Range(startingCoordinate, endingCoordinate)); (isPointInCircle \implies treelocal[i] = Vector3((float)randomPoint.getX(),0f, (float) randomPoint.getY()));`
- output: None
- exception: None

isPointInCircles(List< Circle > circles, Point point):

- transition: None
- output: $(\exists \text{ circle} \in \text{circles} \mid \text{circle.isIn}(\text{point}) = \text{true})$
- exception: None

29 MIS of Tree Switch Button Controller(M24)

29.1 Module

TreeSwitchButtonController

29.2 Uses

UnityEngine
UnityEngine.UI

29.3 Syntax

29.3.1 Exported Constants

None

29.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick			

29.4 Semantics

29.4.1 State Variables

isActive: Boolean value indicating if the leaf information page is active.

29.4.2 Environment Variables

None

29.4.3 Assumptions

None

29.4.4 Access Routine Semantics

onClick():

- transition: isActive := !isActive
- output: None
- exception: None

29.4.5 Local Functions

None

30 MIS of Pie Chart Button Controller(M25)

30.1 Module

pieChartButtonController

30.2 Uses

UnityEngine
UnityEngine.UI

30.3 Syntax

30.3.1 Exported Constants

None

30.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick			

30.4 Semantics

30.4.1 State Variables

isActive: Boolean value indicating if the pie chart is active.

30.4.2 Environment Variables

None

30.4.3 Assumptions

None

30.4.4 Access Routine Semantics

onClick():

- transition: isActive := !isActive
- output: None
- exception: None

30.4.5 Local Functions

None

31 MIS of Pie Chart Controller(M26)

31.1 Module

pieChartController

31.2 Uses

pieChartButtonController

UnityEngine

UnityEngine.UI

31.3 Syntax

31.3.1 Exported Constants

None

31.3.2 Exported Access Programs

Name	In	Out	Exceptions
ChangeView			
markChart			

31.4 Semantics

31.4.1 State Variables

isActive: Boolean value indicating if the pie chart is active.

31.4.2 Environment Variables

GameObject graph

Image legend

EnvDataDisp

Data and Image for generating pieChart:

DataModel data

Image regPinePortion

Image oakPortion

Image beechPortion

Image birchPortion

Image redMaplePortion

Image whitePinePortion

Image redOakPortion

GameObject parentObject

31.4.3 Assumptions

None

31.4.4 Access Routine Semantics

ChangeView():

- transition: $\text{graph.gameObejct.SetActive(!isActive)} \wedge \text{legend.gameObject.SetActive(!isActive)} \wedge \text{EnvDataDisp.gameObject.SetActive(isActive)};$
 $\text{isActive} := \text{isActive}$
- output: None
- exception: None

markChart():

- transition: $\text{calculateValues()};$
 $\text{setImagePortions()}$
- output: None
- exception: None

31.4.5 Local Functions

calculateValues():

- transition: $\text{percentage}_{\text{species}} := \text{number}_{\text{species}} / \sum_{\text{all species}} \text{number}$
- output: None
- exception: None

setImagePortions():

- transition: $\text{pos} := \text{Vector3}(150\text{f}, 200\text{f}, 0\text{f});$
 $\text{size} := \text{Vector2}(600\text{f}, 600\text{f});$
 $\forall \text{species} \implies (\text{Portion}_{\text{species}}.\text{transform.localPosition} := \text{pos}$
 $\wedge \text{Portion}_{\text{species}}.\text{rectTransform.sizeData} := \text{size}$
 $\wedge \text{Portion}_{\text{species}}.\text{fillAmount} := \text{percentage}_{\text{species}}$
 $\wedge \text{Portion}_{\text{species}}.\text{transform.rotation} := \text{Quaternion.Euler}(\text{Vector3}(0\text{f}, 0\text{f}, \text{totalRot}))$
 $\wedge \text{totalRot} := \text{totalRot}(\text{percentage}_{\text{species}}))$

- output: None
- exception: None

32 MIS of Season Change Button Controller(M27)

32.1 Module

SeasonChangeButtonController

32.2 Uses

32.3 Syntax

32.3.1 Exported Constants

None

32.3.2 Exported Access Programs

Name	In	Out	Exceptions
onClick			

32.4 Semantics

32.4.1 State Variables

isActive: Boolean value indicating if the current season is summer.

32.4.2 Environment Variables

None

32.4.3 Assumptions

None

32.4.4 Access Routine Semantics

onClick():

- transition: isActive := !isActive
- output: None
- exception: None

32.4.5 Local Functions

None

33 MIS of Tree Switch Controller(M28)

33.1 Module

TreeSwitchController

33.2 Uses

UnityEngine

UnityEngine.UI

TreeSwitchButtonController

33.3 Syntax

33.3.1 Exported Constants

None

33.3.2 Exported Access Programs

Name	In	Out	Exceptions
ChangeView			

33.4 Semantics

33.4.1 State Variables

isActive: Boolean value indicating if leaf information is active.

TreeParamDisplay: UI of Tree parameters display.

LeafInfoDisplay: UI of Leaf information display.

33.4.2 Environment Variables

None

33.4.3 Assumptions

None

33.4.4 Access Routine Semantics

ChangeView():

- transition: $(\text{isActive} \implies (\text{LeafInfoDisplay.gameObject.SetActive(false)} \wedge \text{TreeParamDisplay.gameObject.SetActive(true)});$

$(!isActive \implies (LeafInfoDisplay.gameObject.SetActive(true) \wedge TreeParamDisplay.gameObject.SetActive(true)))$

$isActive := !isActive$

- output: None
- exception: None

33.4.5 Local Functions

None

34 MIS of Forest Trees (M29)

34.1 Module

ForestTrees

34.2 Uses

M32 , M33, M34, M35, M36, M37, M38

34.3 Syntax

34.3.1 Exported Constants

None

34.3.2 Exported Access Programs

Name	In	Out	Exceptions
new ForestTrees		ForestTree	
addTree	GameObject(This is unity built-in type)		
deleteTree			

34.4 Semantics

34.4.1 State Variables

trees : *GameObject*{}

34.4.2 Environment Variables

None

34.4.3 Assumptions

None

34.4.4 Access Routine Semantics

new ForestTrees():

- transition: None
- output: *out* := *self*

- exception: None

addTree(tree):

- transition: $trees := trees \cup tree$
- output: None
- exception: None

DeleteTree(s):

- transition: $\forall tree : GameObject | tree \in trees : tree.destory()$
- output: None
- exception: None

34.4.5 Local Functions

None

35 MIS of Forest Sky (M30)

35.1 Module

SkyBox

35.2 Uses

UnityLightning

35.3 Syntax

35.3.1 Exported Constants

None

35.3.2 Exported Access Programs

Name	In	Out	Exceptions
new ForestSky		ForestSky	
setSkyBox	Unity Texture		

35.4 Semantics

35.4.1 State Variables

None

35.4.2 Environment Variables

SkyTexture: imported picture of the skybox.

35.4.3 Assumptions

Unity only takes valid texture file type as input.

35.4.4 Access Routine Semantics

new ForestSky()

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

SetSkybox(s):

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

35.4.5 Local Functions

None

36 MIS of Forest Terrain (M31)

36.1 Module

ForestTerrain

36.2 Uses

Unity Terrain Tool

36.3 Syntax

36.3.1 Exported Constants

None

36.3.2 Exported Access Programs

Name	In	Out	Exceptions
new ForestTerrain		ForestTerrain	
setLength	Double		
setWidth	Double		

36.4 Semantics

36.4.1 State Variables

None

36.4.2 Environment Variables

None

36.4.3 Assumptions

None

36.4.4 Access Routine Semantics

new ForestTerrain():

- transition: Create a new terrain in unity using the terrain tool
- output: None

- exception: None

setLength(x):

- transition: set the length of the terrain to be x meters
- output: None
- exception: None

setWidth(x):

- transition: set the width of the terrain to be x meters
- output: None
- exception: None

36.4.5 Local Functions

None

We want to address the following two points regarding creating different modules for different tree types:

- We created different modules for different types of trees because this is necessary when it comes to reading from JSON files by using the newtonsoft parser.
- Also, we deleted the “isValidString” local function here since we will check the validity of the string when users input from the GUI

37 MIS of Red Pine (M32)

37.1 Module

RedPine

37.2 Uses

None

37.3 Syntax

37.3.1 Exported Constants

None

37.3.2 Exported Access Programs

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

37.4 Semantics

37.4.1 State Variables

Treename : *String*

Density : *String*

DBH : *String*

Height : *String*

Age : *String*

37.4.2 Environment Variables

None

37.4.3 Assumptions

None

37.4.4 Access Routine Semantics

new RedPine():

- transition: *Treename*, *Density*, *DBH*, *Height*, *Age* := “Red Pine”, “”, “”, “”, “”
- output: *out* := *self*
- exception: None

getTreeName():

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

setDensity(newDensity):

- transition: *Density* := *newDensity*
- output: None
- exception: None

getDensity():

- transition: None

- output: $out := Density$

- exception: None

setDBH(newDBH):

- transition: $DBH := newDBH$

- output: None

- exception: None

getDBH():

- transition: None

- output: $out := DBH$

- exception: None

setHeight(newHeight):

- transition: $Height := newHeight$

- output: None

- exception: None

getHeight():

- transition: None

- output: $out := Height$

- exception: None

setAge(newAge):

- transition: $Age := newAge$

- output: None

- exception: None

getAge():

- transition: None

- output: $out := Age$

- exception: None

37.4.5 Local Functions

None

38 MIS of Oak (M33)

38.1 Module

Oak

38.2 Uses

None

38.3 Syntax

38.3.1 Exported Constants

None

38.3.2 Exported Access Programs

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

38.4 Semantics

38.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

38.4.2 Environment Variables

None

38.4.3 Assumptions

None

38.4.4 Access Routine Semantics

new Oak():

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

getTreeName():

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

setDensity(newDensity):

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

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

38.4.5 Local Functions

None

39 MIS of Beech (M34)

39.1 Module

Beech

39.2 Uses

None

39.3 Syntax

39.3.1 Exported Constants

None

39.3.2 Exported Access Programs

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

39.4 Semantics

39.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

39.4.2 Environment Variables

None

39.4.3 Assumptions

None

39.4.4 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: None

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

39.4.5 Local Functions

None

40 MIS of Birch (M35)

40.1 Module

Birch

40.2 Uses

None

40.3 Syntax

40.3.1 Exported Constants

None

40.3.2 Exported Access Programs

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

40.4 Semantics

40.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

40.4.2 Environment Variables

None

40.4.3 Assumptions

None

40.4.4 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: None

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

40.4.5 Local Functions

None

41 MIS of White Pine (M36)

41.1 Module

WhitePine

41.2 Uses

None

41.3 Syntax

41.3.1 Exported Constants

None

41.3.2 Exported Access Programs

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

41.4 Semantics

41.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

41.4.2 Environment Variables

None

41.4.3 Assumptions

None

41.4.4 Access Routine Semantics

new WhitePine():

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

getTreeName():

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

setDensity(newDensity):

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

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

41.4.5 Local Functions

None

42 MIS of Red Maple (M37)

42.1 Module

RedMaple

42.2 Uses

None

42.3 Syntax

42.3.1 Exported Constants

None

42.3.2 Exported Access Programs

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

42.4 Semantics

42.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

42.4.2 Environment Variables

None

42.4.3 Assumptions

None

42.4.4 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: None

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

42.4.5 Local Functions

None

43 MIS of Red Oak (M38)

43.1 Module

RedOak

43.2 Uses

None

43.3 Syntax

43.3.1 Exported Constants

None

43.3.2 Exported Access Programs

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

43.4 Semantics

43.4.1 State Variables

Treename : String

Density : String

DBH : String

Height : String

Age : String

43.4.2 Environment Variables

None

43.4.3 Assumptions

None

43.4.4 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: None

getDensity():

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

setDBH(newDBH):

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

getDBH():

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

- exception: None

setHeight(newHeight):

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

getHeight():

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

setAge(newAge):

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

getAge():

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

43.4.5 Local Functions

None

We deleted “isValidString” local function here because we will check the validity of string when users enter data from the GUI

44 MIS of Environmental Data (M39)

44.1 Module

EnvData

44.2 Uses

None

44.3 Syntax

44.3.1 Exported Constants

None

44.3.2 Exported Access Programs

Name	In	Out	Exceptions
new EnvData		EnvData	
setHumidity	String		
getHumidity		String	
setTemp	String		
getTemp		String	
setSC	String		
getSC		String	
setSN	String		
getSN		String	
setLAI	String		
getLAI		String	

44.4 Semantics

44.4.1 State Variables

Humidity : String

Temp : String

SC : String

SN : String

LAI : String

44.4.2 Environment Variables

None

44.4.3 Assumptions

None

44.4.4 Access Routine Semantics

new EnvData():

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

getHumidity():

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

setHumidity(newHumidity):

- transition: $Humidity := newHumidity$
- output: None
- exception: None

getTemp():

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

setTemp(newTemp):

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

getSC():

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

setSC(newSC):

- transition: $SC := newSC$
- output: None
- exception: None

getSN():

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

setSN(newSN):

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

getLAI():

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

setLAI(newLAI):

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

44.4.5 Local Functions

None

45 MIS of Plot Data (M40)

45.1 Module

PlotData

45.2 Uses

M32, M33, M34, M35, M36, M37, M38, M39

45.3 Syntax

45.3.1 Exported Constants

None

45.3.2 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	

45.4 Semantics

45.4.1 State Variables

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

45.4.2 Environment Variables

None

45.4.3 Assumptions

None

45.4.4 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

45.4.5 Local Functions

None

46 MIS of First Person Player (M41)

46.1 Module

FirstPersonPlayer

46.2 Uses

Character Controller Module from Unity

46.3 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(M4).

46.4 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(M4).

47 MIS of Json File (M42)

47.1 Module

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

47.2 Local Type

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

$S : \text{set of } X$

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

47.3 State Variables

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

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

47.5 Local Functions

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

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

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

48 MIS of Main Page (M43)

48.1 Module

MainPageDisplay

48.2 Uses

M6 , M7, M8, M9, M10, M44, M45, M46, M47 UnityEngine.UI

48.3 Syntax

48.3.1 Exported Constants

None

48.3.2 Exported Access Programs

None

48.4 Semantics

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

48.4.1 State Variables

None

48.4.2 Environment Variables

None

48.4.3 Assumptions

None

48.4.4 Access Routine Semantics

None

48.4.5 Local Functions

None

49 MIS of Start Button (M44)

49.1 Module

StartButton

49.2 Uses

M6 , UnityEngine.UI

49.3 Syntax

49.3.1 Exported Constants

None

49.3.2 Exported Access Programs

None

49.4 Semantics

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

49.4.1 State Variables

None

49.4.2 Environment Variables

windows: Computer screen used to display messages.

49.4.3 Assumptions

None

49.4.4 Access Routine Semantics

None

49.4.5 Local Functions

None

50 MIS of Instruction Button (M45)

50.1 Module

InstructionButton

50.2 Uses

M7 , UnityEngine.UI

50.3 Syntax

50.3.1 Exported Constants

None

50.3.2 Exported Access Programs

None

50.4 Semantics

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

50.4.1 State Variables

None

50.4.2 Environment Variables

windows: Computer screen used to display messages.

50.4.3 Assumptions

None

50.4.4 Access Routine Semantics

None

50.4.5 Local Functions

None

51 MIS of Contact Us Button (M46)

51.1 Module

ContactUsButton

51.2 Uses

M8 , UnityEngine.UI

51.3 Syntax

51.3.1 Exported Constants

None

51.3.2 Exported Access Programs

None

51.4 Semantics

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

51.4.1 State Variables

None

51.4.2 Environment Variables

windows: Computer screen used to display messages.

51.4.3 Assumptions

None

51.4.4 Access Routine Semantics

None

51.4.5 Local Functions

None

52 MIS of Quit Button (M47)

52.1 Module

QuitButton

52.2 Uses

M10 ,UnityEngine.UI

52.3 Syntax

52.3.1 Exported Constants

None

52.3.2 Exported Access Programs

None

52.4 Semantics

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

52.4.1 State Variables

None

52.4.2 Environment Variables

windows: Computer screen used to display messages.

52.4.3 Assumptions

None

52.4.4 Access Routine Semantics

None

52.4.5 Local Functions

None

53 MIS of Instruction Page (M48)

53.1 Module

InstructionInfoDisplay

53.2 Uses

M7 , UnityEngine.UI

53.3 Syntax

53.3.1 Exported Constants

None

53.3.2 Exported Access Programs

None

53.4 Semantics

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

53.4.1 State Variables

None

53.4.2 Environment Variables

windows: Computer screen used to display messages.

53.4.3 Assumptions

None

53.4.4 Access Routine Semantics

None

53.4.5 Local Functions

None

54 MIS of Contact Us Page (M49)

54.1 Module

ContactUsInfoDisplay

54.2 Uses

M9 UnityEngine.UI

54.3 Syntax

54.3.1 Exported Constants

None

54.3.2 Exported Access Programs

None

54.4 Semantics

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

54.4.1 State Variables

None

54.4.2 Environment Variables

windows: Computer screen used to display messages.

54.4.3 Assumptions

None

54.4.4 Access Routine Semantics

None

54.4.5 Local Functions

None

55 MIS of Back Button (M50)

55.1 Module

BackButton

55.2 Uses

M11 , UnityEngine.UI

55.3 Syntax

55.3.1 Exported Constants

None

55.3.2 Exported Access Programs

None

55.4 Semantics

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

55.4.1 State Variables

None

55.4.2 Environment Variables

windows: Computer screen used to display messages.

55.4.3 Assumptions

None

55.4.4 Access Routine Semantics

None

55.4.5 Local Functions

None

56 MIS of Update Data Page (M51)

56.1 Module

UpdateDataDisplay

56.2 Uses

M8 UnityEngine.UI

56.3 Syntax

56.3.1 Exported Constants

None

56.3.2 Exported Access Programs

None

56.4 Semantics

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

56.4.1 State Variables

None

56.4.2 Environment Variables

windows: Computer screen used to display messages.

56.4.3 Assumptions

None

56.4.4 Access Routine Semantics

None

56.4.5 Local Functions

None

57 MIS of Environmental Data Selection Button (M52)

57.1 Module

EnvDataSelectionButton

57.2 Uses

M16 , UnityEngine.UI

57.3 Syntax

57.3.1 Exported Constants

None

57.3.2 Exported Access Programs

None

57.4 Semantics

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

57.4.1 State Variables

None

57.4.2 Environment Variables

windows: Computer screen used to display messages.

57.4.3 Assumptions

None

57.4.4 Access Routine Semantics

None

57.4.5 Local Functions

None

58 MIS of Data Type Selection Button (M53)

58.1 Module

DataTypeSelectionButton

58.2 Uses

M17 , UnityEngine.UI

58.3 Syntax

58.3.1 Exported Constants

None

58.3.2 Exported Access Programs

None

58.4 Semantics

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

58.4.1 State Variables

None

58.4.2 Environment Variables

windows: Computer screen used to display messages.

58.4.3 Assumptions

None

58.4.4 Access Routine Semantics

None

58.4.5 Local Functions

None

59 MIS of New Data Input Box (M54)

59.1 Module

NewDataInputBox

59.2 Uses

, UnityEngine.UI

59.3 Syntax

59.3.1 Exported Constants

None

59.3.2 Exported Access Programs

None

59.4 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](#)

59.4.1 State Variables

None

59.4.2 Environment Variables

windows: Computer screen used to display messages.

59.4.3 Assumptions

None

59.4.4 Access Routine Semantics

None

59.4.5 Local Functions

None

60 MIS of Save Button (M55)

60.1 Module

SaveButton

60.2 Uses

M18 , UnityEngine.UI

60.3 Syntax

60.3.1 Exported Constants

None

60.3.2 Exported Access Programs

None

60.4 Semantics

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

60.4.1 State Variables

None

60.4.2 Environment Variables

windows: Computer screen used to display messages.

60.4.3 Assumptions

None

60.4.4 Access Routine Semantics

None

60.4.5 Local Functions

None

61 MIS of Current Data Display (M56)

61.1 Module

CurrentDataDisplay

61.2 Uses

UnityEngine.UI

61.3 Syntax

61.3.1 Exported Constants

None

61.3.2 Exported Access Programs

None

61.4 Semantics

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

61.4.1 State Variables

None

61.4.2 Environment Variables

windows: Computer screen used to display messages.

61.4.3 Assumptions

None

61.4.4 Access Routine Semantics

None

61.4.5 Local Functions

None

62 MIS of Plot Selection Drop Down (M57)

62.1 Module

PlotSelection

62.2 Uses

M12, UnityEngine.UI

62.3 Syntax

62.3.1 Exported Constants

None

62.3.2 Exported Access Programs

None

62.4 Semantics

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

62.4.1 State Variables

None

62.4.2 Environment Variables

windows: Computer screen used to display messages.

62.4.3 Assumptions

None

62.4.4 Access Routine Semantics

None

62.4.5 Local Functions

None

63 MIS of Tree Type Selection Drop Down (M58)

63.1 Module

TreeTypeSelection

63.2 Uses

M13 , UnityEngine.UI

63.3 Syntax

63.3.1 Exported Constants

None

63.3.2 Exported Access Programs

None

63.4 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](#).

63.4.1 State Variables

None

63.4.2 Environment Variables

windows: Computer screen used to display messages.

63.4.3 Assumptions

None

63.4.4 Access Routine Semantics

None

63.4.5 Local Functions

None

64 MIS of Update Data Button (M59)

64.1 Module

UpdateDataButton

64.2 Uses

M8 , UnityEngine.UI

64.3 Syntax

64.3.1 Exported Constants

None

64.3.2 Exported Access Programs

None

64.4 Semantics

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

64.4.1 State Variables

None

64.4.2 Environment Variables

windows: Computer screen used to display messages.

64.4.3 Assumptions

None

64.4.4 Access Routine Semantics

None

64.4.5 Local Functions

None

65 MIS of Forest Display (M60)

65.1 Module

ForestDisplay

65.2 Uses

UnityEngine.UI, M29, M30, M31

65.3 Syntax

65.3.1 Exported Constants

None

65.3.2 Exported Access Programs

None

65.4 Semantics

65.4.1 State Variables

This module is used to display the forest models.

65.4.2 Environment Variables

windows: Computer screen used to display messages.

65.4.3 Assumptions

None

65.4.4 Access Routine Semantics

None

65.4.5 Local Functions

None

66 MIS of Show Environmental Data Button (M61)

66.1 Module

ShowEnvDataButton

66.2 Uses

M14 , UnityEngine.UI

66.3 Syntax

66.3.1 Exported Constants

None

66.3.2 Exported Access Programs

None

66.4 Semantics

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

66.4.1 State Variables

None

66.4.2 Environment Variables

windows: Computer screen used to display messages.

66.4.3 Assumptions

None

66.4.4 Access Routine Semantics

None

66.4.5 Local Functions

None

67 MIS of Show Tree Parameters Button (M62)

67.1 Module

ShowTreeParamButton

67.2 Uses

M15 , UnityEngine.UI

67.3 Syntax

67.3.1 Exported Constants

None

67.3.2 Exported Access Programs

None

67.4 Semantics

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

67.4.1 State Variables

None

67.4.2 Environment Variables

windows: Computer screen used to display messages.

67.4.3 Assumptions

None

67.4.4 Access Routine Semantics

None

67.4.5 Local Functions

None

68 MIS of Environment Data Display (M63)

68.1 Module

EnvDataDisplay

68.2 Uses

UnityEngine.UI

68.3 Syntax

68.3.1 Exported Constants

None

68.3.2 Exported Access Programs

None

68.4 Semantics

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

68.4.1 State Variables

None

68.4.2 Environment Variables

windows: Computer screen used to display messages.

68.4.3 Assumptions

None

68.4.4 Access Routine Semantics

None

68.4.5 Local Functions

None

69 MIS of Tree Parameters Display (M64)

69.1 Module

TreeParamDisplay

69.2 Uses

UnityEngine.UI

69.3 Syntax

69.3.1 Exported Constants

None

69.3.2 Exported Access Programs

None

69.4 Semantics

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

69.4.1 State Variables

None

69.4.2 Environment Variables

windows: Computer screen used to display messages.

69.4.3 Assumptions

None

69.4.4 Access Routine Semantics

None

69.4.5 Local Functions

None

70 MIS of Pause Indicator (M65)

70.1 Module

PauseIndicatorDisplay

70.2 Uses

UnityEngine.UI

70.3 Syntax

70.3.1 Exported Constants

None

70.3.2 Exported Access Programs

None

70.4 Semantics

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

70.4.1 State Variables

None

70.4.2 Environment Variables

windows: Computer screen used to display messages.

70.4.3 Assumptions

None

70.4.4 Access Routine Semantics

None

70.4.5 Local Functions

None

71 MIS of SeasonChangeButton (M66)

71.1 Module

SeasonChangeButton

71.2 Uses

UnityEngine.UI , M27

71.3 Syntax

71.3.1 Exported Constants

None

71.3.2 Exported Access Programs

None

71.4 Semantics

This module is used to display the seasonal change of the models of the forest

71.4.1 State Variables

None

71.4.2 Environment Variables

windows: Computer screen used to display messages.

71.4.3 Assumptions

None

71.4.4 Access Routine Semantics

None

71.4.5 Local Functions

None

72 MIS of pieChartButton (M67)

72.1 Module

pieChartButton

72.2 Uses

UnityEngine.UI

72.3 Syntax

72.3.1 Exported Constants

None

72.3.2 Exported Access Programs

None

72.4 Semantics

This module is providing a button GUI for users to switch between a pie chart and environmental data.

72.4.1 State Variables

None

72.4.2 Environment Variables

windows: Computer screen used to display messages.

72.4.3 Assumptions

None

72.4.4 Access Routine Semantics

None

72.4.5 Local Functions

None

73 MIS of TreeSwitchButton (M68)

73.1 Module

TreeSwitchButton

73.2 Uses

UnityEngine.UI

73.3 Syntax

73.3.1 Exported Constants

None

73.3.2 Exported Access Programs

None

73.4 Semantics

This module provides a button GUI for users to switch between tree parameters and leaf information

73.4.1 State Variables

None

73.4.2 Environment Variables

windows: Computer screen used to display messages.

73.4.3 Assumptions

None

73.4.4 Access Routine Semantics

None

73.4.5 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>.