

A.3 Practice 06 - Launch

A.3.1 parent.launch.py

```
1 from launch_ros.substitutions import FindPackageShare
2
3 from launch import LaunchDescription
4 from launch.actions import IncludeLaunchDescription
5 from launch.launch_description_sources import
  PythonLaunchDescriptionSource
6 from launch.substitutions import PathJoinSubstitution, TextSubstitution
7
8
9 def generate_launch_description():
10     colors = {
11         'background_r': '200'
12     }
13
14     return LaunchDescription([
15         IncludeLaunchDescription(
16             PythonLaunchDescriptionSource([
17                 PathJoinSubstitution([
18                     FindPackageShare('launch_test'),
19                     'launch',
20                     'substitutions1.launch.py'
21                 ])
22             ]),
23             launch_arguments={
24                 'turtlesim_ns': 'turtlesim2',
25                 'use_provided_red': 'True',
26                 'new_background_r': TextSubstitution(text=str(colors['
background_r']))
27             }.items()
28         )
29     ])
```

A.3.2 substitutions1.launch.py

```
1 from launch_ros.actions import Node
2
3 from launch import LaunchDescription
4 from launch.actions import DeclareLaunchArgument, ExecuteProcess,
  TimerAction
5 from launch.conditions import IfCondition
6 from launch.substitutions import LaunchConfiguration, PythonExpression
7
8
9 def generate_launch_description():
10     turtlesim_ns = LaunchConfiguration('turtlesim_ns')
11     use_provided_red = LaunchConfiguration('use_provided_red')
12     new_background_r = LaunchConfiguration('new_background_r')
13
```

```

14 turtlesim_ns_launch_arg = DeclareLaunchArgument(
15     'turtlesim_ns',
16     default_value='turtlesim1'
17 )
18 use_provided_red_launch_arg = DeclareLaunchArgument(
19     'use_provided_red',
20     default_value='False'
21 )
22 new_background_r_launch_arg = DeclareLaunchArgument(
23     'new_background_r',
24     default_value='200'
25 )
26
27 turtlesim_node = Node(
28     package='turtlesim',
29     namespace=turtlesim_ns,
30     executable='turtlesim_node',
31     name='sim'
32 )
33 spawn_turtle = ExecuteProcess(
34     cmd=[[
35         'ros2 service call ',
36         turtlesim_ns,
37         '/spawn ',
38         'turtlesim/srv/Spawn ',
39         "{x: 2, y: 2, theta: 0.2}"
40     ]],
41     shell=True
42 )
43 change_background_r = ExecuteProcess(
44     cmd=[[
45         'ros2 param set ',
46         turtlesim_ns,
47         '/sim background_r ',
48         '120'
49     ]],
50     shell=True
51 )
52 change_background_r_conditioned = ExecuteProcess(
53     condition=IfCondition(
54         PythonExpression([
55             new_background_r,
56             ' == 200',
57             ' and ',
58             use_provided_red
59         ])
60     ),
61     cmd=[[
62         'ros2 param set ',
63         turtlesim_ns,
64         '/sim background_r ',
65         new_background_r
66     ]],
67     shell=True

```

```

68     )
69
70     return LaunchDescription([
71         turtlesim_ns_launch_arg,
72         use_provided_red_launch_arg,
73         new_background_r_launch_arg,
74         turtlesim_node,
75         spawn_turtle,
76         change_background_r,
77         TimerAction(
78             period=2.0,
79             actions=[change_background_r_conditioned],
80         )
81     ])

```

A.3.3 eventhandler.launch.py

```

1  from launch_ros.actions import Node
2
3  from launch import LaunchDescription
4  from launch.actions import (DeclareLaunchArgument, EmitEvent,
5                               ExecuteProcess,
6                               LogInfo, RegisterEventHandler, TimerAction)
7  from launch.conditions import IfCondition
8  from launch.event_handlers import (OnExecutionComplete, OnProcessExit,
9                                     OnProcessIO, OnProcessStart, OnShutdown)
10 from launch.events import Shutdown
11 from launch.substitutions import (EnvironmentVariable, FindExecutable,
12                                   LaunchConfiguration, LocalSubstitution,
13                                   PythonExpression)
14
15 def generate_launch_description():
16     turtlesim_ns = LaunchConfiguration('turtlesim_ns')
17     use_provided_red = LaunchConfiguration('use_provided_red')
18     new_background_r = LaunchConfiguration('new_background_r')
19
20     turtlesim_ns_launch_arg = DeclareLaunchArgument(
21         'turtlesim_ns',
22         default_value='turtlesim1'
23     )
24     use_provided_red_launch_arg = DeclareLaunchArgument(
25         'use_provided_red',
26         default_value='False'
27     )
28     new_background_r_launch_arg = DeclareLaunchArgument(
29         'new_background_r',
30         default_value='200'
31     )
32
33     turtlesim_node = Node(
34         package='turtlesim',
35         namespace=turtlesim_ns,

```

```

36     executable='turtlesim_node',
37     name='sim'
38 )
39 spawn_turtle = ExecuteProcess(
40     cmd=[[
41         FindExecutable(name='ros2'),
42         ' service call ',
43         turtlesim_ns,
44         '/spawn ',
45         'turtlesim/srv/Spawn ',
46         "{x: 2, y: 2, theta: 0.2}"
47     ]],
48     shell=True
49 )
50 change_background_r = ExecuteProcess(
51     cmd=[[
52         FindExecutable(name='ros2'),
53         ' param set ',
54         turtlesim_ns,
55         '/sim background_r ',
56         '120'
57     ]],
58     shell=True
59 )
60 change_background_r_conditioned = ExecuteProcess(
61     condition=IfCondition(
62         PythonExpression([
63             new_background_r,
64             ' == 200 ',
65             ' and ',
66             use_provided_red
67         ])
68     ),
69     cmd=[[
70         FindExecutable(name='ros2'),
71         ' param set ',
72         turtlesim_ns,
73         '/sim background_r ',
74         new_background_r
75     ]],
76     shell=True
77 )
78
79 return LaunchDescription([
80     turtlesim_ns_launch_arg,
81     use_provided_red_launch_arg,
82     new_background_r_launch_arg,
83     turtlesim_node,
84     RegisterEventHandler(
85         OnProcessStart(
86             target_action=turtlesim_node,
87             on_start=[
88                 LogInfo(msg='Turtlesim started, spawning turtle'),
89                 spawn_turtle

```

```

90         ]
91     )
92 ),
93 RegisterEventHandler(
94     OnProcessIO(
95         target_action=spawn_turtle,
96         on_stdout=lambda event: LogInfo(
97             msg='Spawn request says "{}"'.format(
98                 event.text.decode().strip())
99         )
100     )
101 ),
102 RegisterEventHandler(
103     OnExecutionComplete(
104         target_action=spawn_turtle,
105         on_completion=[
106             LogInfo(msg='Spawn finished'),
107             change_background_r,
108             TimerAction(
109                 period=2.0,
110                 actions=[change_background_r_conditioned],
111             )
112         ]
113     )
114 ),
115 RegisterEventHandler(
116     OnProcessExit(
117         target_action=turtlesim_node,
118         on_exit=[
119             LogInfo(msg=(EnvironmentVariable(name='USER'),
120                 ' closed the turtlesim window')),
121             EmitEvent(event=Shutdown(
122                 reason='Window closed'))
123         ]
124     )
125 ),
126 RegisterEventHandler(
127     OnShutdown(
128         on_shutdown=[LogInfo(
129             msg=['Launch was asked to shutdown: ',
130                 LocalSubstitution('event.reason')]
131         )]
132     )
133 ),
134 ])
```