

# DEAKIN UNIVERSITY

## OBJECT ORIENTED DEVELOPMENT

### ONTRACK SUBMISSION

---

# A Simple Reaction-Timer Controller

---

*Submitted By:*

Connor GENT

gentco

2021/06/02 12:15

*Tutor:*

Nayyar ZAIDI

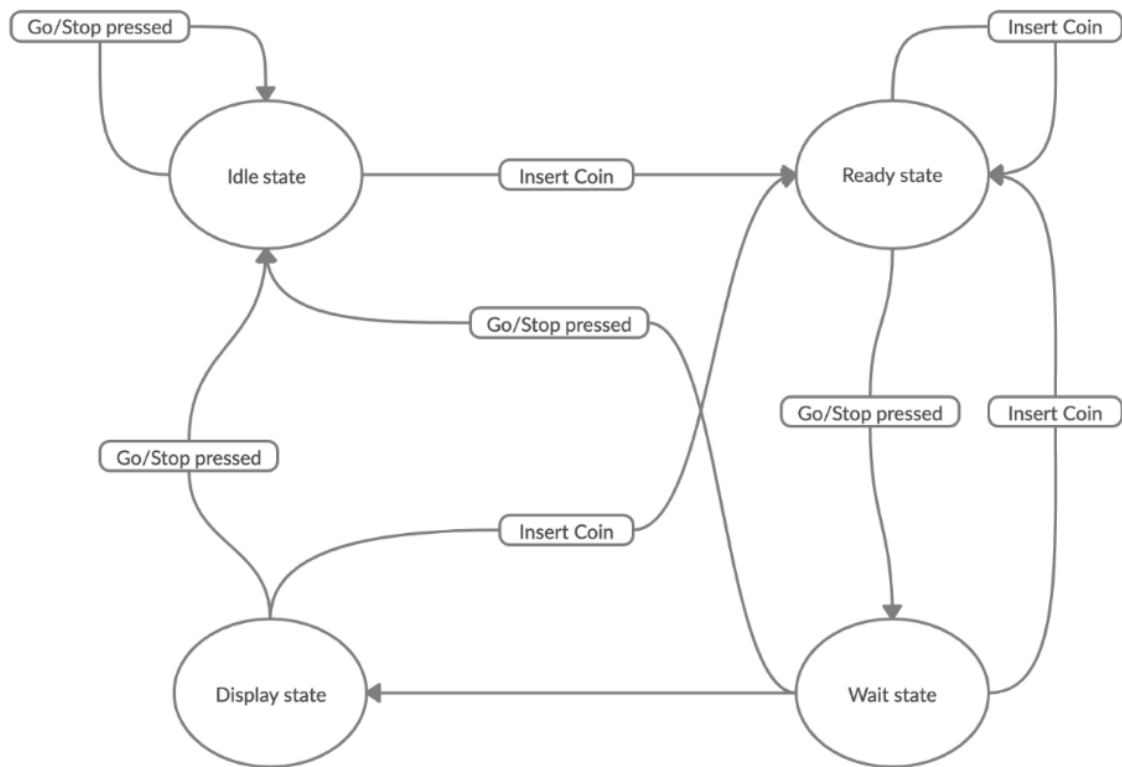
Outcome	Weight
Evaluate Code	◆◆◆◆◇
Principles	◆◆◆◆◇
Build Programs	◆◆◆◆◇
Design	◆◆◆◆◇
Justify	◆◆◆◆◇

This Task introduced us to State Design patterns which familiarise us with problems posed by event driven systems.

June 2, 2021



**Connor Gent 5.3D State Transition Diagram:**



---

	Go/Stop pressed	Insert Coin	Do nothing (Time out)
Idle	Idle	Ready	-
Ready	Wait	Ready	-
Wait	Idle	Ready	Display
Display	Idle	Ready	-

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Task5_3D
8  {
9      public interface IState
10     {
11         void CoinInserted();
12         void GoStopPressed();
13         void Tick();
14     }
15     public class SimpleReactionController : IController
16     {
17         public int count { get; set; }
18         IState CurrentState { get; set; }
19
20         public IGui gui { get; set; }
21         public IRandom random { get; set; }
22
23         public void Connect(IGui gui, IRandom rng)
24         {
25             this.random = rng;
26             this.gui = gui;
27             this.gui.Connect(this);
28             Init();
29         }
30
31         public void CoinInserted()
32         {
33             this.CurrentState.CoinInserted();
34         }
35
36         public void GoStopPressed()
37         {
38             this.CurrentState.GoStopPressed();
39         }
40
41
42         public void Tick()
43         {
44             CurrentState.Tick();
45         }
46
47         public void Init()
48         {
49             CurrentState = new IdleState(this);
50         }
51
52         public void ChangeState(IState state)
53         {
```

```
54         this.CurrentState = state;
55     }
56 }
57
58 public class IdleState : IState
59 {
60     private SimpleReactionController _controller;
61
62     public IdleState(SimpleReactionController controller)
63     {
64         this._controller = controller;
65         this._controller.gui.Init();
66     }
67
68     public void CoinInserted()
69     {
70         this._controller.ChangeState(new ReadyState(this._controller));
71     }
72
73     public void GoStopPressed()
74     {
75         this._controller.ChangeState(new IdleState(this._controller));
76     }
77
78
79     public void Tick()
80     {
81     }
82 }
83
84 public class ReadyState : IState
85 {
86     private SimpleReactionController _controller;
87
88     public ReadyState(SimpleReactionController controller)
89     {
90         this._controller = controller;
91         this._controller.gui.SetDisplay("Press Go/Stop");
92     }
93
94     public void CoinInserted()
95     {
96         this._controller.ChangeState(new ReadyState(this._controller));
97     }
98
99     public void GoStopPressed()
100    {
101        this._controller.ChangeState(new WaitState(this._controller));
102    }
103
104    public void Tick()
105    {
106    }
```

```
107
108     }
109 }
110
111 public class WaitState : IState
112 {
113     public SimpleReactionController _controller;
114     int timecounter;
115
116     public WaitState(SimpleReactionController controller)
117     {
118         this._controller = controller;
119         this._controller.gui.SetDisplay("Wait...");
120         this._controller.count = this._controller.random.GetRandom(150, 200);
121     }
122
123     public void CoinInserted()
124     {
125         this._controller.ChangeState(new ReadyState(this._controller));
126     }
127
128     public void GoStopPressed()
129     {
130         this._controller.ChangeState(new IdleState(this._controller));
131     }
132
133     public void Tick()
134     {
135         timecounter += 1;
136         if(timecounter >= this._controller.count)
137         {
138             this._controller.ChangeState(new DisplayState(this._controller));
139         }
140     }
141 }
142
143 public class DisplayState : IState
144 {
145     public SimpleReactionController _controller;
146
147     public DisplayState(SimpleReactionController controller)
148     {
149         this._controller = controller;
150         this._controller.gui.SetDisplay($"{Convert.ToString(this._controller.c_
151             ↪ ount)} Ticks has
152             ↪ passed");
153     }
154
155     public void CoinInserted()
156     {
157         this._controller.ChangeState(new ReadyState(this._controller));
158     }
159 }
```

```
158
159     public void GoStopPressed()
160     {
161         this._controller.ChangeState(new IdleState(this._controller));
162     }
163
164     public void Tick()
165     {
166
167     }
168 }
169
170
171
172 }
```