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

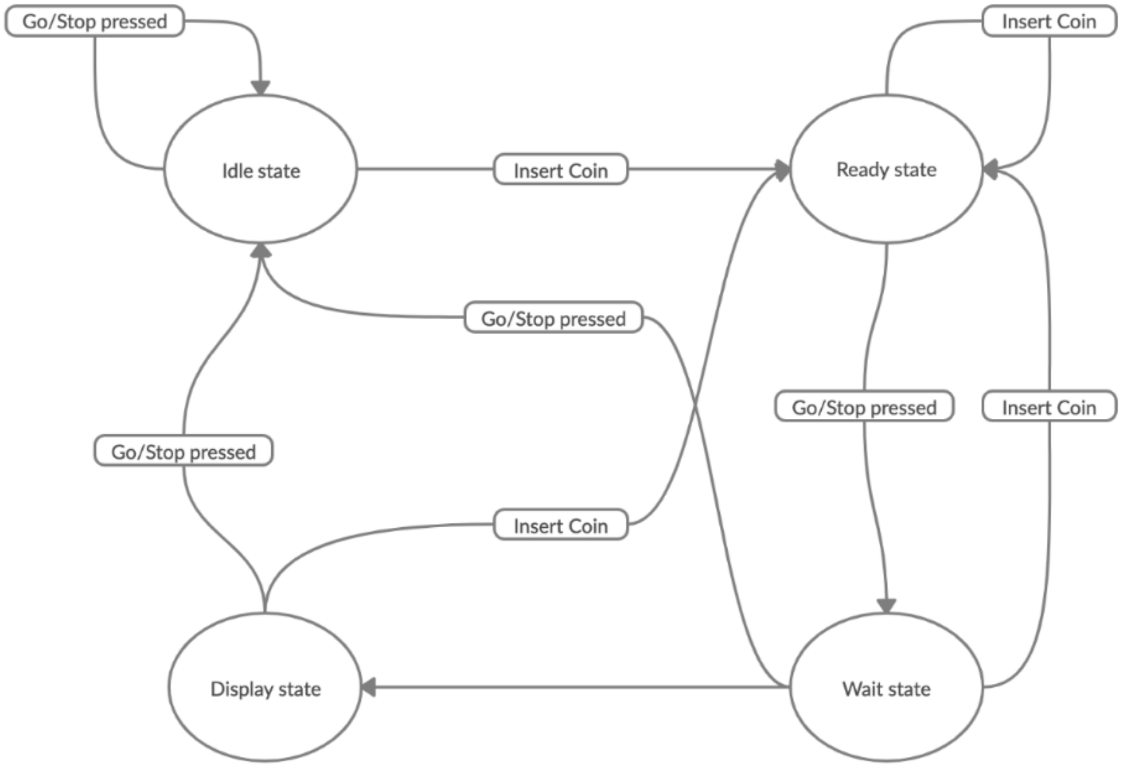
Outcome	**Weight**
Evaluate Code	♦♦♦♦◊
Principles	♦♦♦♦◊
Build Programs	♦♦♦♦◊
Design	♦♦♦♦◊
Justify	♦♦♦♦◊

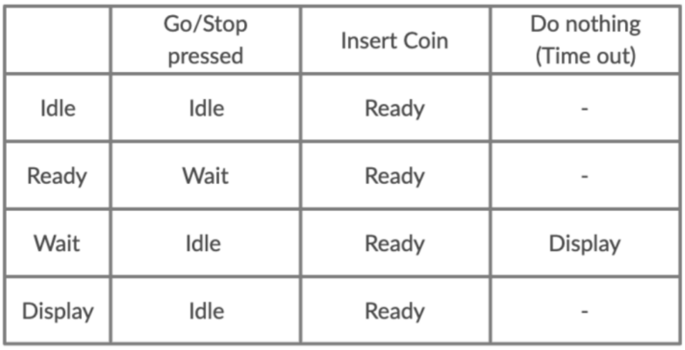
This Task introduced us to State Design patterns which familiarise us with problems posed by event driven systems.

June 2, 2021

Produced by Doubtfire

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	-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Task5_3D
{
    public interface IState
    {
        void CoinInserted();
        void GoStopPressed();
        void Tick();
    }
    public class SimpleReactionController : IController
    {
        public int count { get; set; }
        IState CurrentState { get; set; }

        public IGui gui { get; set; }
        public IRandom random { get; set; }

        public void Connect(IGui gui, IRandom rng)
        {
            this.random = rng;
            this.gui = gui;
            this.gui.Connect(this);
            Init();
        }

        public void CoinInserted()
        {
            this.CurrentState.CoinInserted();
        }

        public void GoStopPressed()
        {
            this.CurrentState.GoStopPressed();
        }


        public void Tick()
        {
            CurrentState.Tick();
        }

        public void Init()
        {
            CurrentState = new IdleState(this);
        }

        public void ChangeState(IState state)
        {
            this.CurrentState = state;
        }
    }

    public class IdleState : IState
    {
        private SimpleReactionController _controller;

        public IdleState(SimpleReactionController controller)
        {
            this._controller = controller;
            this._controller.gui.Init();
        }

        public void CoinInserted()
        {
            this._controller.ChangeState(new ReadyState(this._controller));
        }

        public void GoStopPressed()
        {
            this._controller.ChangeState(new IdleState(this._controller));

        }

        public void Tick()
        {

        }
    }

    public class ReadyState : IState
    {
        private SimpleReactionController _controller;

        public ReadyState(SimpleReactionController controller)
        {
            this._controller = controller;
            this._controller.gui.SetDisplay("Press Go/Stop");
        }

        public void CoinInserted()
        {
            this._controller.ChangeState(new ReadyState(this._controller));
        }

        public void GoStopPressed()
        {
            this._controller.ChangeState(new WaitState(this._controller));
        }

        public void Tick()
        {

        }
    }

    public class WaitState : IState
    {
        public SimpleReactionController _controller;
        int timecounter;

        public WaitState(SimpleReactionController controller)
        {
            this._controller = controller;
            this._controller.gui.SetDisplay("Wait...");
            this._controller.count = this._controller.random.GetRandom(150, 200);
        }

        public void CoinInserted()
        {
            this._controller.ChangeState(new ReadyState(this._controller));

        }

        public void GoStopPressed()
        {
            this._controller.ChangeState(new IdleState(this._controller));
        }

        public void Tick()
        {
            timecounter += 1;
            if(timecounter >= this._controller.count)
            {
                this._controller.ChangeState(new DisplayState(this._controller));
            }
        }
    }

    public class DisplayState : IState
    {
        public SimpleReactionController _controller;

        public DisplayState(SimpleReactionController controller)
        {
            this._controller = controller;
            this._controller.gui.SetDisplay($"{Convert.ToString(this._controller.count)} Ticks has passed");
        }

        public void CoinInserted()
        {
            this._controller.ChangeState(new ReadyState(this._controller));
        }

        public void GoStopPressed()
        {
            this._controller.ChangeState(new IdleState(this._controller));
        }

        public void Tick()
        {

        }
    }



}
```