PUBLISHER/SUBSCRIBER PATTERN

Adil Salamat 2019

Adil Salamat
100879914

Table of Contents

Adil Salamat
100879914

Introduction

In this report I will be discussing what the Publish/Subscribe pattern is as well in what circumstances it is most appropriate to use. Finally, I will be talking about how I plan to implement the pattern for my project.

What is the Publisher/Subscribe Pattern?

The publish/subscribe pattern uses messages to communicate with objects. An object A can subscribe to another object B to receive messages from object A. This will allow object B to react accordingly. A real-life example of this would be a messaging system set up on your phone. You are subscribed to receive cricket scores from a sport mobile app, your phone will receive notifications and messages for the live cricket scores because you are a subscriber. The Publisher would be the reporter that updates the scores.

Why should you use the Publisher/Subscribe Pattern?

Using the publish/subscribe pattern has a number of advantages. The first would be how the pattern means more effective coding for applications that rely on real time events. Rather than checking to see if an event can take place, the publish/subscribe pattern utilises a push-based message system. This results in faster response times and means that an application would be more reliable especially for applications that cannot afford any latency or delay.

Another advantage would be how it makes the structure of the code more modular and flexible. Publishers and subscribers are decoupled from each other, meaning that you can develop and modify them separately. This allows changes to be made without having sever consequences on the systems as there is more control and freedom to make changes.

Another advantage would be the simplicity of the system. Rather than having multiple talking points, a publisher posts messages to a topic. Whatever objects need information can then subscribe to the topic to gain access to messages. This results in less call-backs, making the code cleaner and easier to read.

How will I use the Publisher/Subscribe Pattern?

In my game of Pacman, the publish/subscribe pattern can be used to help the Ghosts find Pacman. Pacman will update (publish) his location and the Ghosts who are subscribed will be able to then figure out how to get to Pacman based off his current location.

Another way this pattern would be used is by the map and the other objects. The map would keep track of the number of pellets on the screen. Once the last pellet is consumed, the map will publish to the other objects (Pacman and the Ghosts etc) that the stage is complete, they will then be reset to their starting position.

Conclusion

In summary, we have discussed what the publish/subscribe design pattern is as well as the ideal scenario to use it and when to avoid it. Finally, we acknowledged how I plan to utilize the publish/subscribe design pattern for my final project.

Bibliography

This source was used to gather more information on the pattern as well as studying an example using the images

Rana, P. (2019). *Publisher/Subscriber pattern with Event/Delegate and EventAggregator*. [online] Codeproject.com. Available at: https://www.codeproject.com/Articles/866547/Publisher-Subscriber-pattern-with-Event-Delegate-a [Accessed 17 Nov. 2019].

This source was used to gather more advantages of using this pattern

Amazon Web Services, Inc. (2019). *What are Benefits of Pub/Sub Messaging? – AWS*. [online] Available at: https://aws.amazon.com/pub-sub-messaging/benefits/ [Accessed 17 Nov. 2019].