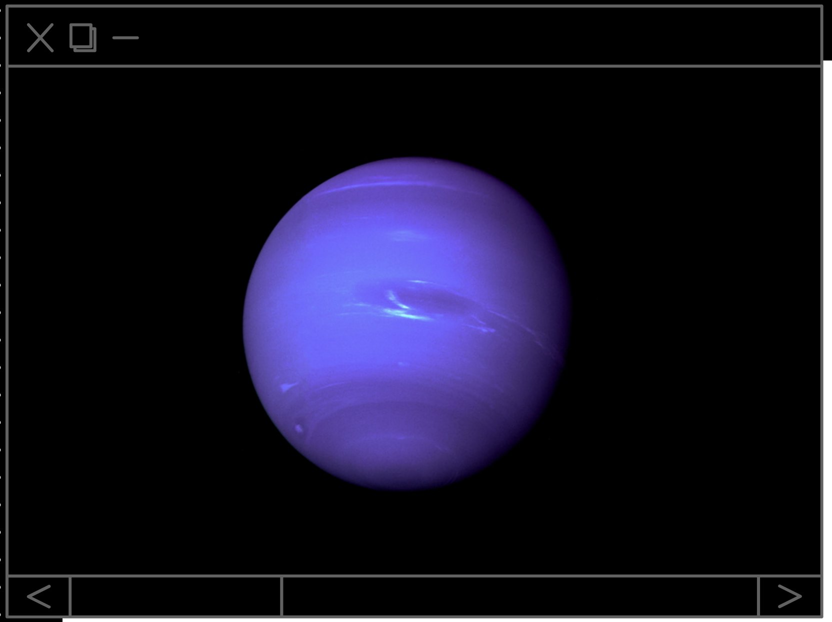
Understanding

MVC Architecture



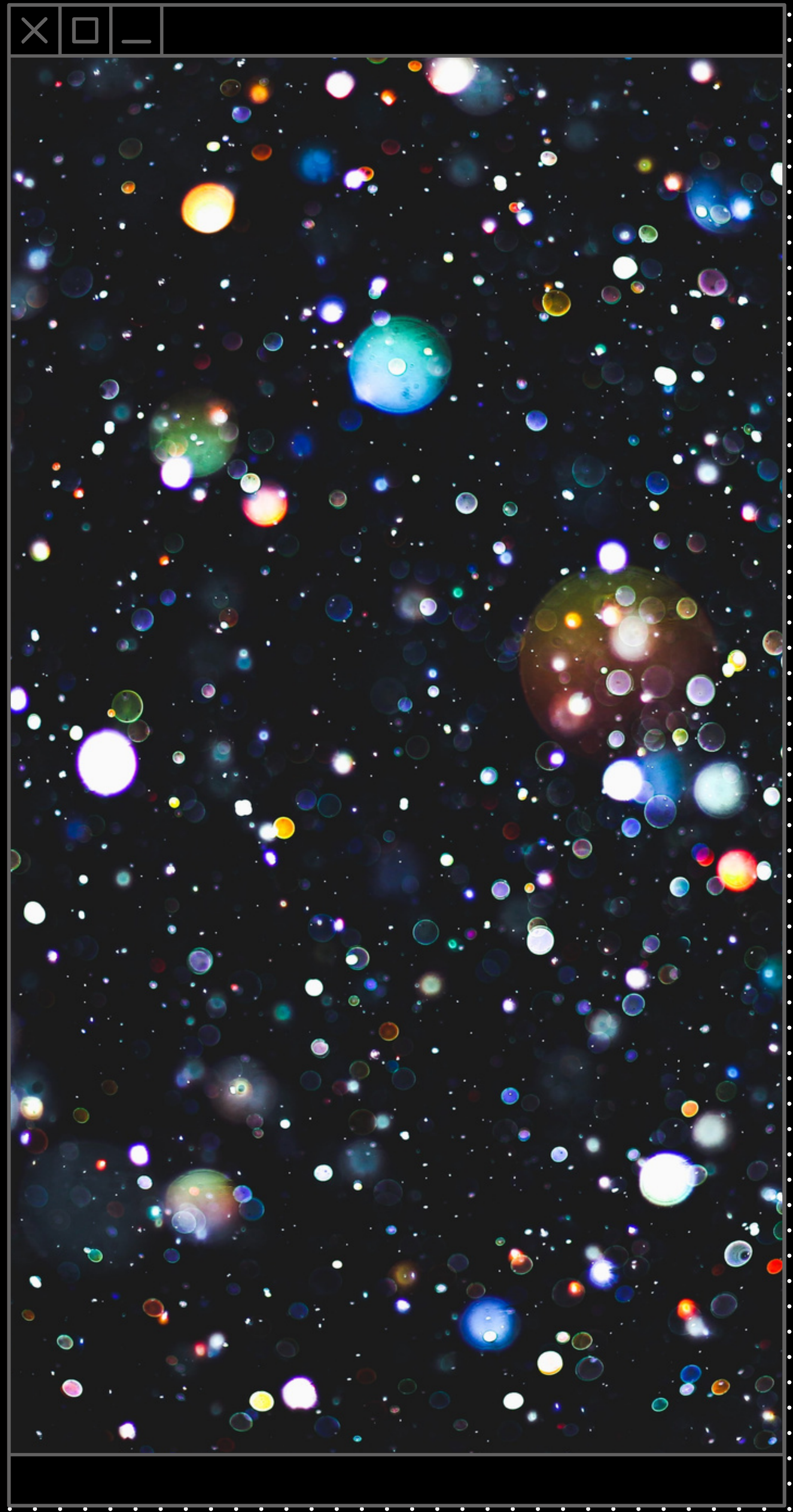
Confused? You're in the right place!

Start →

What is MVC?

Model View Controller (MVC) is a software design pattern that distinguishes between the data model, user interface and logic of an application.

This design pattern was introduced in the 1980's and used in graphical user-interfaces (GUIs). Today, it is commonly used in software and web applications.



The Model

The model is the central component of the pattern. It manages the data, logic and rules of an application.

Ideally, the model handles all database interactions.

The View

The view is responsible for what is visible on the client side. This often includes the user interface, rendering page layouts and graphics.

All logic should be abstracted away from the view.

The Controller

The controller is the intermediary coordinator between the model and the view.

It responds to user input, directs data transmission between the model and the view, and performs any validation required.

Putting It All Together

Model

How your data is handled and structured.

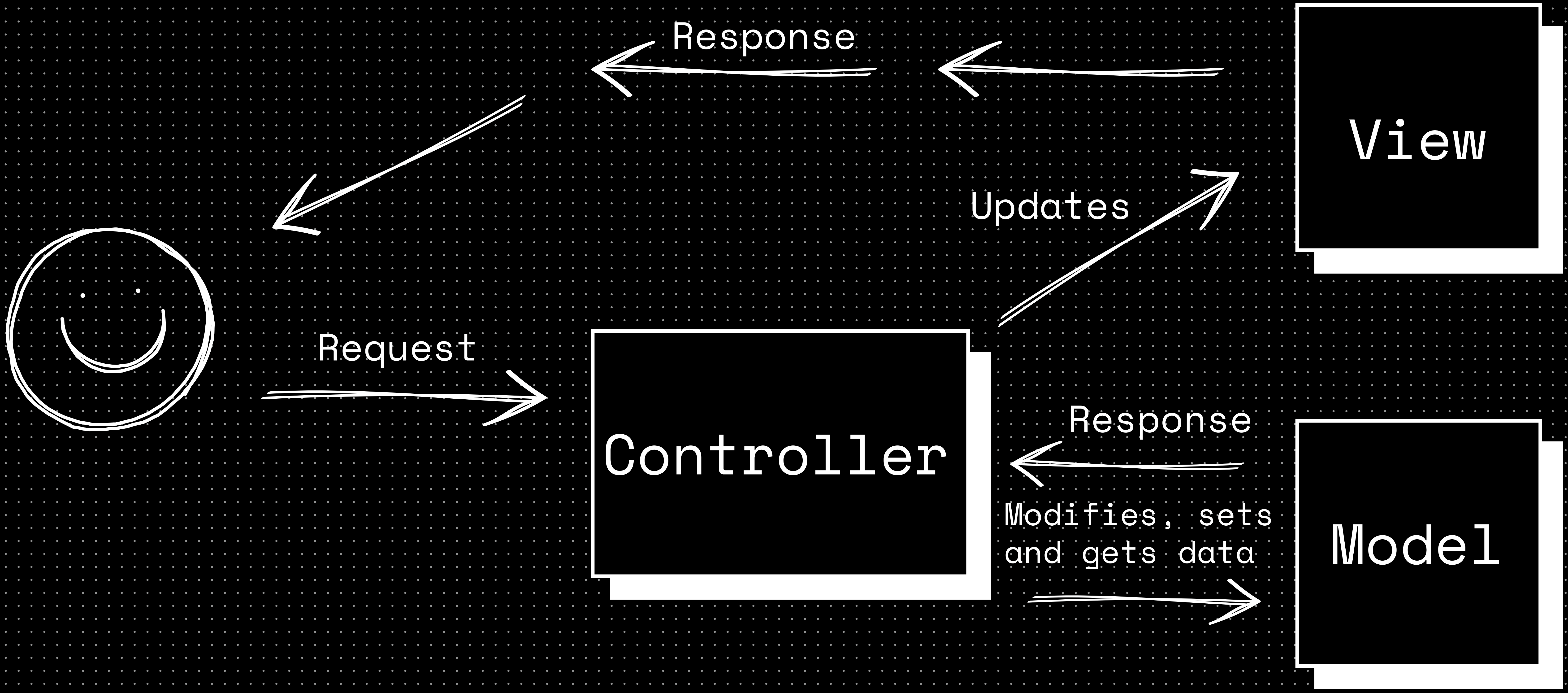
View

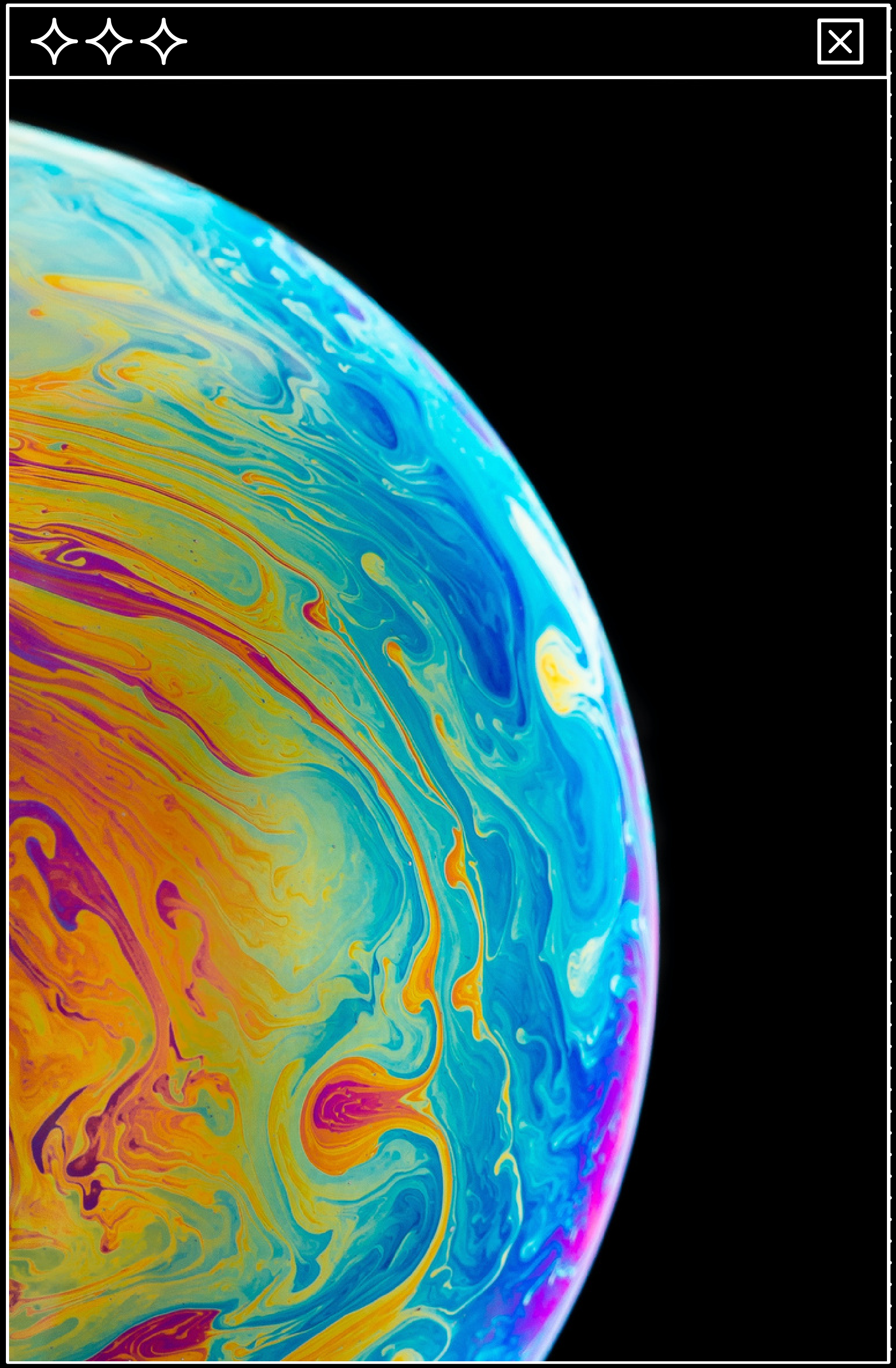
The presentation of data.

Controller

Links the user, model and view.

Example


Response
View
Updates
Request
Controller
Response
Modifies, sets
and gets data
Model



Benefits of MVC

MVC is a helpful design pattern for organising and reusing code within a large application.

The seperation of concerns means that applications using the MVC pattern are easily modifiable and can support many user interfaces.

Drawbacks of MVC

In practice, the controller is often obfuscated within the view, and the model may interact with the view.

This can create overly confusing and complex code bases if strict rules are not adhered to.