Model-View-Controller

MVC Design Pattern

MVC (Model-View-Controller)

Model
Defines data structure
e.g. updates application to reflect added item

View
Defines display (UI)
e.g. user clicks 'add to cart'

Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

Updates
e.g. list item to show added item

Manipulates

Sends input from user

Sometimes updates directly
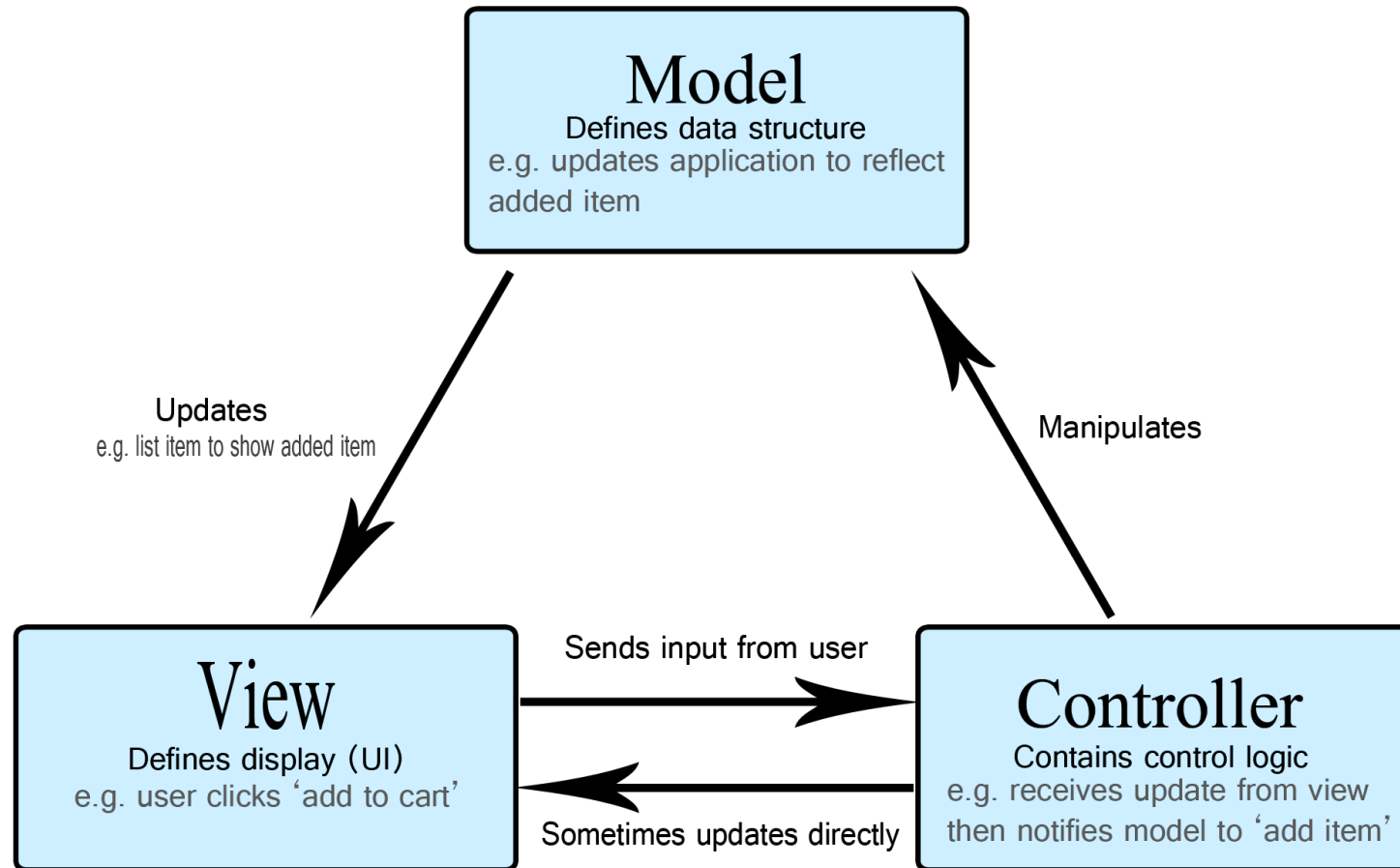
Model

- Manages data and data related logic
- It is known as the lowest level which means it is responsible for maintaining data.
- Handle data logically so it basically deals with data.
- The model is actually connected to the database so anything you do with data, adding or retrieving data is done in the model component.
- It responds to the controller requests because the controller never talks to the database by itself.
- The model talks to the database back and forth and then it gives the needed data to the controller.
  - Note: the model never communicated with the view directly.

View

- Handles layout and display.

- Data representation is done by the view component. It actually generates UI or user interface for the user.

- So at web applications when you think of the view component just think the Html/CSS part.

- Views are created by the data which is collected by the model component but these data aren't taken directly but through the controller, so the view only speaks to the controller.

Controller

- Routes commands to the model and view parts

- It's known as the main man because the controller is the component that enables the interconnection between the views and the model so it acts as an intermediary.

- The controller doesn't have to worry about handling data logic, it just tells the model what to do.

- After receiving data from the model it processes it and then it takes all that information it sends it to the view and explains how to represent to the user.
  - Note: Views and models can not talk directly.

Advantages of MVC

- MVC architecture will separate the user interface from business logic and business logic

- Components are reusable.

- Easy to maintain.

- Different components of the application in MVC can be independently deployed and maintained.

- This architecture helps to test components independently.

Disadvantages of MVC

- The complexity is high.
- Not suitable for small applications.
- The inefficiency of data access in view.

Summary

- MVC is an architectural pattern consisting of three parts: Model, View, Controller.
  - Model: Handles data logic.
  - View: It displays the information from the model to the user.
  - Controller: It controls the data flow into a model object and updates the view whenever data changes.
- It is invented by Trygve Reenskau.
- Even though it's very popular and it does have some disadvantages, the main one is complexity.