Simulation – Model, View, Controller

Learning Goals

- Represent the state of a system in a model by identifying components and rules
- Visualize a model using graphics
- Update a model over time based on rules
- Update a model after events (mouse-based and keyboard-based) based on rules

Simulations and Models

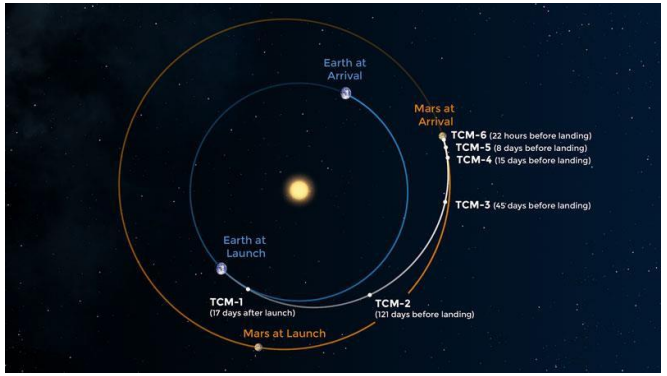
Simulations are Imitations of Real Life

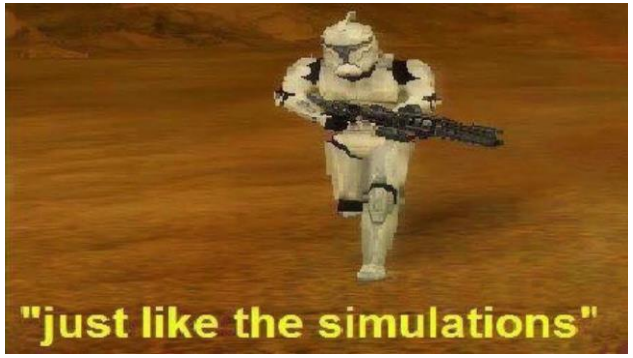
A simulation is an automated imitation of a real-world event.

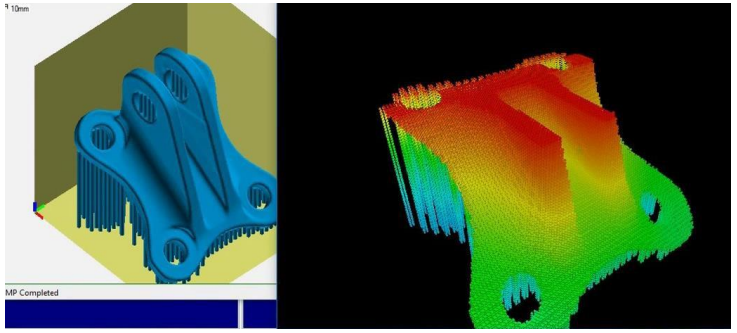
By running simulations on different starting inputs, and by interacting with them while they run, we can test how the event will change under different circumstances.

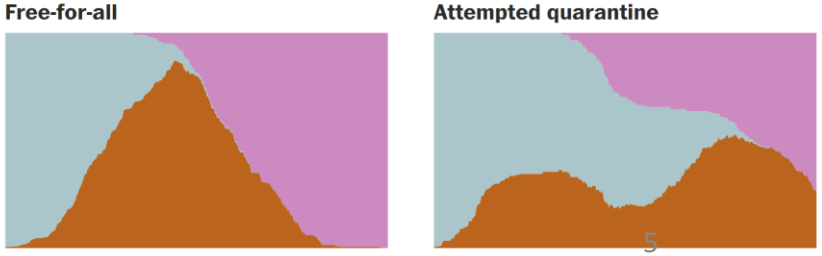
Examples of Simulations

Simulation is used across many different fields, including training people, testing designs, and making predictions (like whether a flight plan will work, or how a pandemic will evolve over time).









Simulations vs. Real-world Experiments

Simulations share a lot in common with real world experiments. Major differences include:

- Experiments run in **real time**; simulations can be **sped up, slowed down, or paused**.
- Experiments can be **expensive**; simulations are fairly **cheap**.
- Experiments include **all possible factors**; simulations only include **factors we program in**.

Example Simulations

You can explore simulations across a variety of fields on the site NetLogo.

- Ant colony movements
- Flocking behavior
- Gravitational forces
- Climate change
- Fire spreading
- Rumor mills

Simulations Run on Models

How do we program a simulation? You need to design a good **model**, which will mimic the part of the real world you want to study. The simulation represents how the system represented by the model changes **over time**, or how it changes **based on events**.

Models are composed of two parts:

- The **components** of the system (information that describes the world at an exact moment).
- The **rules** of the system (how the components should change as time passes/events occur).

Components are like variables, and rules are like functions!

Example Model

Problem: how will increasing the price of bread over the course of a few months affect how many people buy bread?

Model Components: current price; delta change in price; overall consumer count; distribution of consumer incomes

Model Rules: supply/demand relationship for bread; relationship between income and max amount willing to pay

Activity: Design a Model

Problem: say we want to track how many birds are in a local area over time.

You do: What are the components of this model? What are the rules?

Coding a Simulation

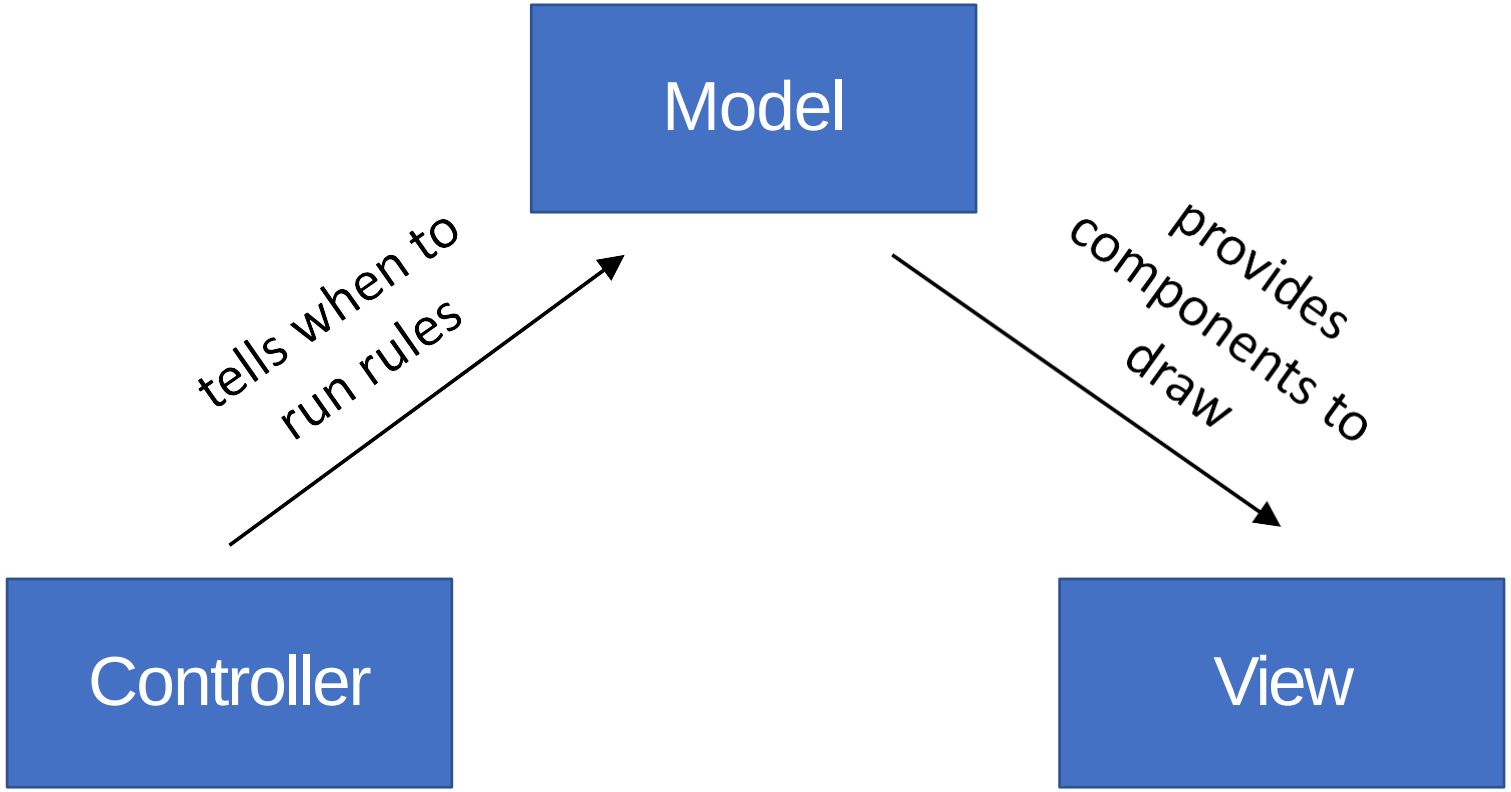
Simulation Parts in Code

We'll implement simulations in this class **graphically**.

Our simulation code will be composed of three parts:

- A **model** which stores the core components in a shared data structure and implements core rules in functions
- Time and event **controllers** which tell the model when to run rules that update the components
- A graphical **view** which repeatedly displays the current state of the model

Model, View, Controller



Simple Example – Roll the Dice

Let's start with a simple simulation. Saywe want to roll the dice.

The **components** should hold any values that might change. In this case, that's the value of the **dice**.

The **rules** should describe how the model changes over time. In this case, we **click the button** in the window with a call to Roll.

The **view** should draw the dice in the middle of the window and set its value.