All practice for this chapter is done in the 06.Thinking in React – usePopcorn App

Effects & Data Fetching

# Component (instance) Lifecycle

* There are different phases that a Component can go through at any given time

## Mount / Initial Render

* Component instance is rendered for the first time
* Fresh state and props are created

## Re-render

* State change
* Props Change
* Parent Re-renders
* Context Changes

## Unmount

* The component instance is completely destroyed and removed
* State and Props are destroyed

# useEffect Hook

* This hook is used to give us a place where we can SAFELY write side effects (that are not otherwise allowed into the Render logic (usestate, return) of the APP)
* The useEffect we have used contains a side effect (contact with the REAL WORLD)

### A side effect can be made in

* Event handlers ( onClick, onSubmit,etc)
* Effects (useEffect) – in different phases of the life cycle

# Error Handling

* We can add a Try Catch Block into the Async function and by using a piece of state we can alter the UIA screen shot of a computer program

  Description automatically generated

## When are Effects executed?

1. Mount of the component
2. Commit
3. Browser paint

## Effects are executed after the browser has painted

* 1. Effects may contain long running processes
  2. Effects are asynchronous

1. Unmount

# Cleanup Function of the useEffect

* Cleanup function is a **OPTIONAL**  function that will be called on UNMOUNT
* It’s an optional part of the useEffect hook
* This runs on 2 different occasions
  + Before the effect is executed again
  + After a component has unmounted

## Cleanup the Fetch Requests as soon as a new ONE IS STARTED

* Without this, we are making SOSOOOOSOOOO many network calls to fetch data
* For this we are using the **Abort Controller – native browser APIA screenshot of a computer

  Description automatically generated**
* All the initial **FETCHES**  got **CANCELED** because the cleanup function has been called **BEFORE THE EFFECT IS EXECUTED AGAIN**