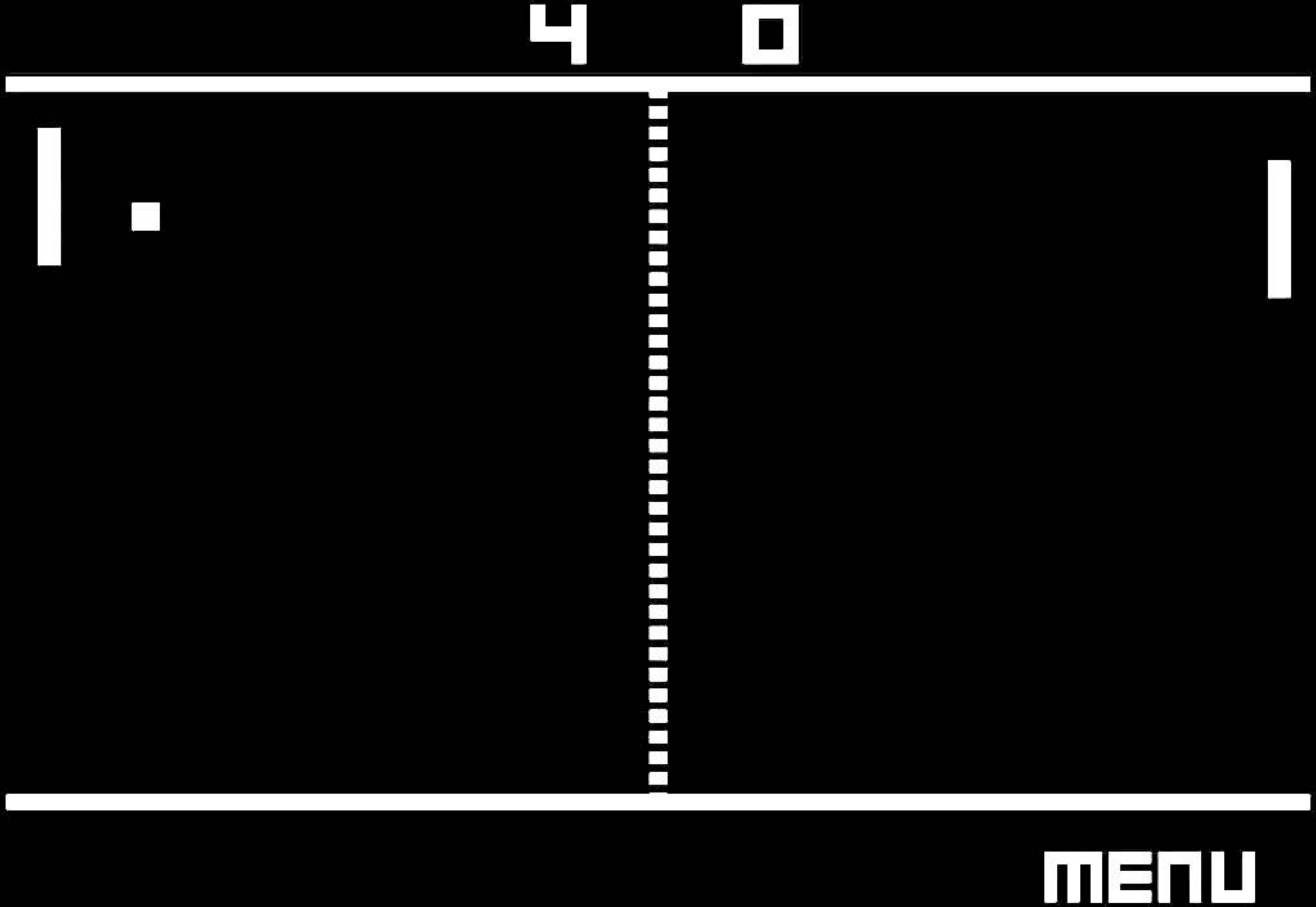
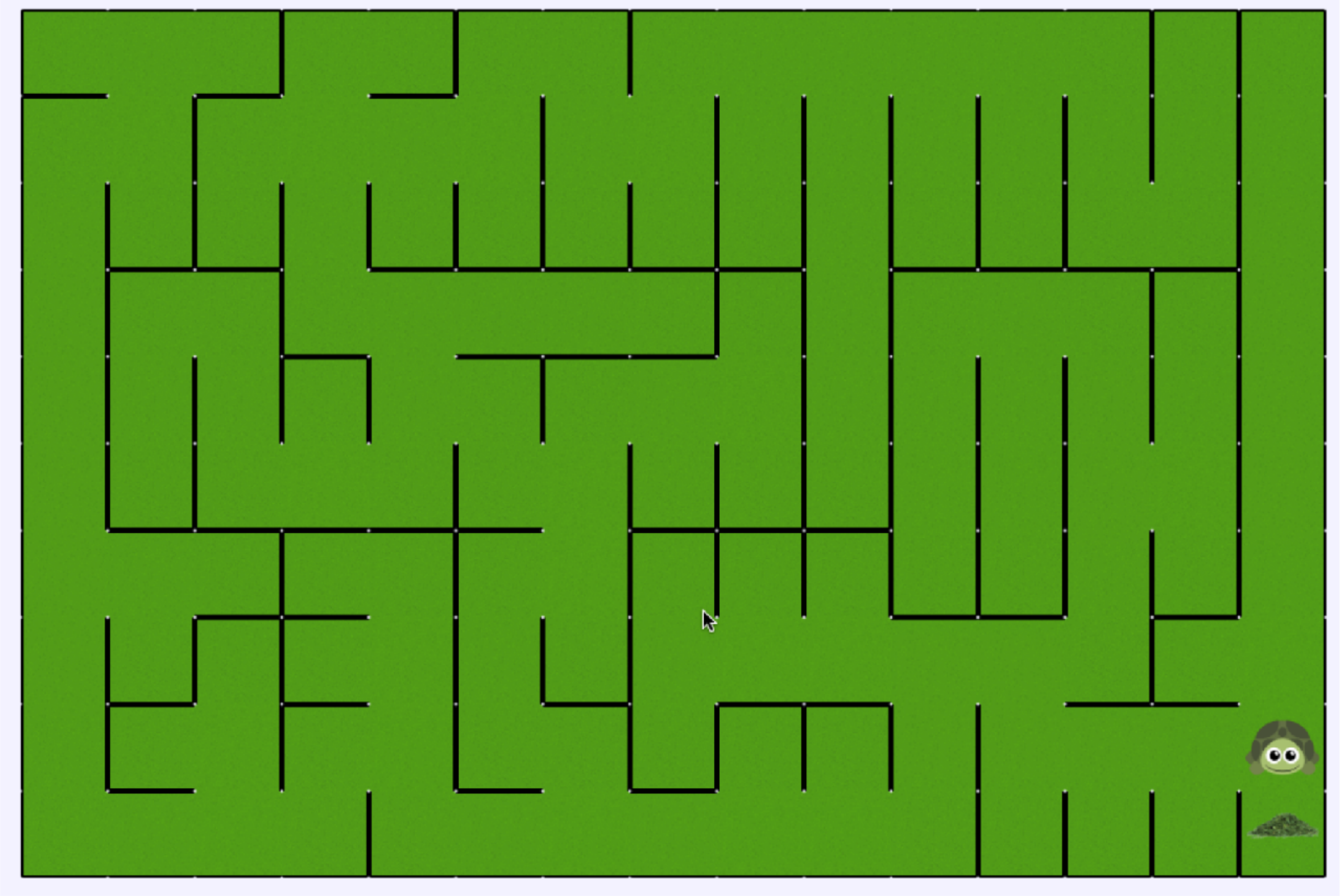
REDUX

A Predictable State Container for JavaScript Applications

STATE


4
0
MENU

The Adventures of Turt McSquirt



YouTube
Upload
Subscribe
385
Add to
Share
More
11,255
144
Up next
Autoplay
vevo
6:39
50+
VIDEOS

kendrick lamar colbert report

Kendrick Lamar Performs On Late Show with Stephen Colbert

S Sound Flaw

1,342,357 views

Published on Sep 9, 2015

Kendrick Lamar On Late Show with Stephen Colbert:

Kendrick Lamar - These Walls (Live on Ellen) ft. Bilal, Anna
KendrickLamarVEVO
3,831,319 views

Mix - Kendrick Lamar Performs On Late Show with Stephen
YouTube

FULLSTACK ACADEMY
Mail
COMPOSE
Inbox (1,207)
Starred
Sent Mail
Drafts
More
Search people...
Mogan Subraman…
David Yang
a
academics
Alex Bloomer
Alex Castillo
More
1–50 of 1,997

The conversation has been moved to the Trash. Learn more Undo

□ ☆	Wufoo	Workshop Feedback Form [#1447] - User Mongo ID 57ed47cf549c9d03001cab7d Your Name (Optional) Jacob Brodsky You	11:37 am
□ ☆	Gabriela Medina	Screencast Info - Hi Seniors, Here is an overview of how to complete a screencast. The main benefit of creating a screencas	11:32 am
□ ☆	Kate, Gabriel, David (4)	Fwd: Camtasia Inquiry - Yes. On Mon, Dec 5, 2016 at 8:25 AM, Gabriel Lebec <gabriel@fullstackacademy.com> wrote: Hi D	11:26 am
□ ☆	David Yang (Google Docs)	» React-ing to the Market - David Yang resolved comments in React-ing to the Market Resolved 12 c Open Discussion	11:08 am
□ ☆	**Slack**	» **[Slack] Grace Hopper Academy updates for the week of November 27th, 2016** - Grace Hopper Academy's Weekly Sum	**10:01 am**
□ ☆	**david**	**Playbook Status - 12/5/2016** - Charts	**8:23 am**
□ ☆	**Asana**	» **You have 5 tasks due: Lead Retrospective... and 4 more - fullstackacademy.com** - Hi Joe, You have some fullstackacad	**4:16 am**
□ ☆	Wufoo	Workshop Feedback Form [#1446] - User Mongo ID 57c4887f79058603002025e7 Your Name (Optional) Ian Munro Your En	Dec 4
□ ☆	Wufoo	Workshop Feedback Form [#1445] - User Mongo ID 57c4887f79058603002025e7 Your Name (Optional) Ian Munro Your En	Dec 4
□ ☆	Wufoo	Workshop Feedback Form [#1444] - User Mongo ID 5842519643adc80004d4fa2f Your Name (Optional) Evan Williams Your	Dec 4

GLOBAL VS. LOCAL STATE

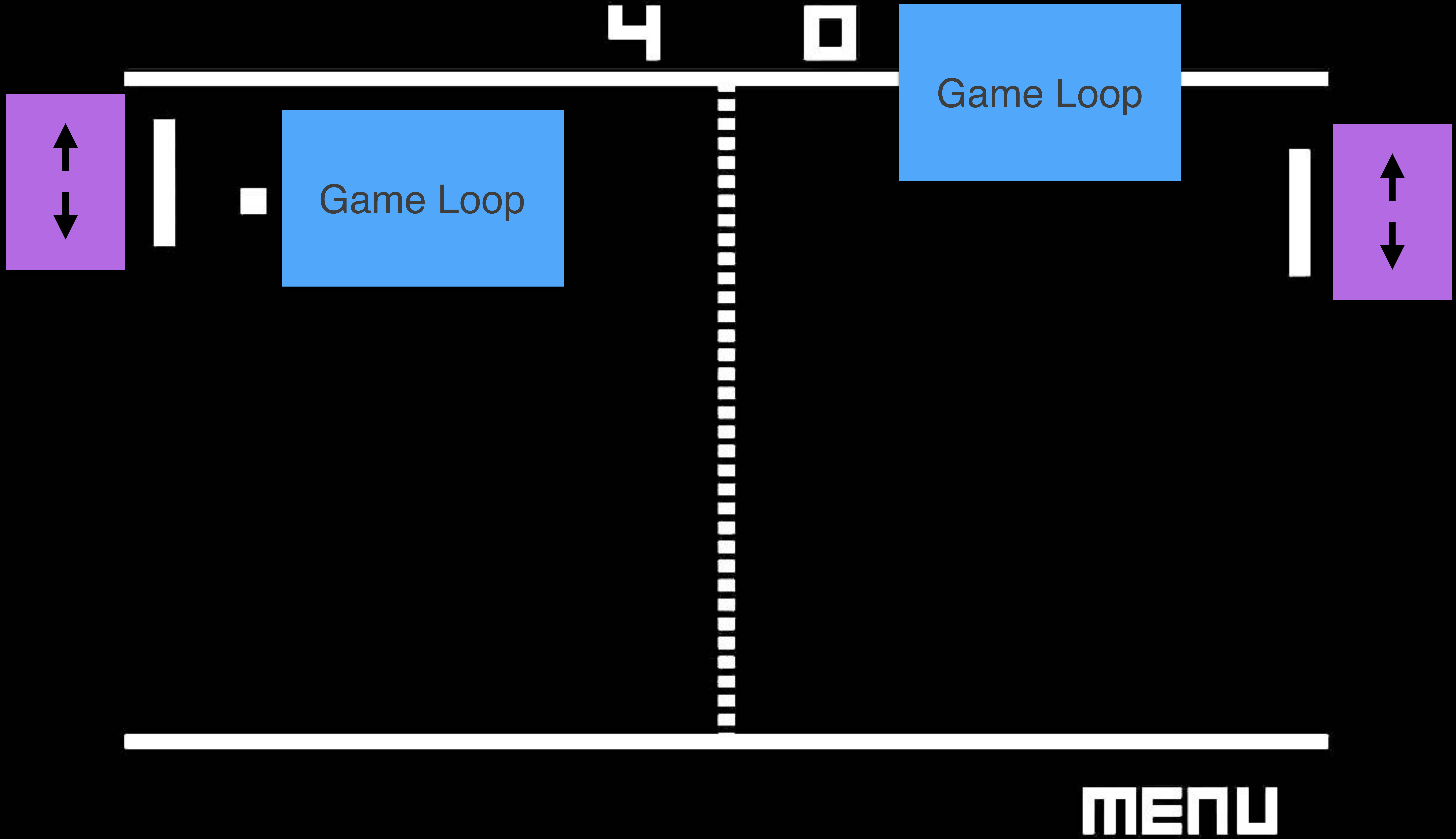
- **Some dynamic information is sometimes generated and only needed by a component**
- **Other information is needed by the application as a whole**
- **Game <Countdown />**
  - The remaining time may be local state
  - When remaining time hits 0, it should set a global state gameOver: true
- **<input />**
  - onChange: set to local state to hold and display value
  - onSubmit: send value to global state for processing

HOW STATE IS AFFECTED

FULLSTACK ACADEMY
Mail
COMPOSE
Inbox (1,207)
Starred
Sent Mail
Drafts
More
Search people...
Mogan Subraman…
David Yang
a
academics
Alex Bloomer
Alex Castillo
More
1–50 of 1,997
Open Discussion

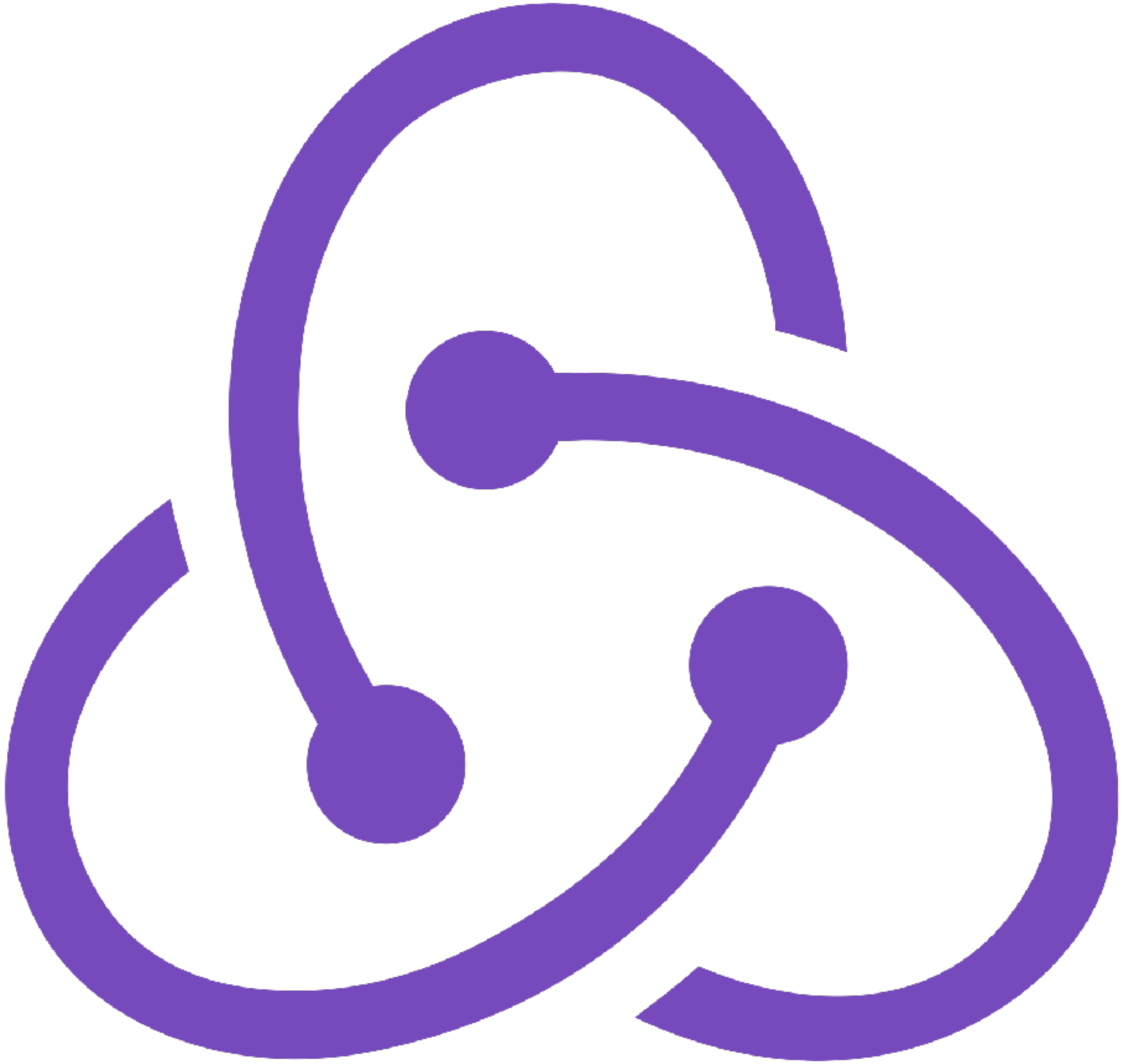
The conversation has been moved to the Trash. Learn more Undo

Wufoo	Workshop Feedback Form [#1447] - User Mongo ID 57ed47cf549c9d03001cab7d Your Name (Optional) Jacob Brodsky You	11:37 am
Gabriela Medina	Screencast Info - Hi Seniors, Here is an overview of how to complete a screencast. The main benefit of creating a screencas	11:32 am
Kate, Gabriel, David (4)	Fwd: Camtasia Inquiry - Yes. On Mon, Dec 5, 2016 at 8:25 AM, Gabriel Lebec <gabriel@fullstackacademy.com> wrote: Hi D	11:26 am
David Yang (Google Docs)	» React-ing to the Market - David Yang resolved comments in React-ing to the Market Resolved 12 c	11:08 am
Slack	» **[Slack] Grace Hopper Academy updates for the week of November 27th, 2016** - Grace Hopper Academy's Weekly Sum	**10:01 am**
david	**Playbook Status - 12/5/2016** - Charts	**8:23 am**
Asana	» **You have 5 tasks due: Lead Retrospective... and 4 more - fullstackacademy.com** - Hi Joe, You have some fullstackaca	**4:16 am**
Wufoo	Workshop Feedback Form [#1446] - User Mongo ID 57c4887f79058603002025e7 Your Name (Optional) Ian Munro Your En	Dec 4
Wufoo	Workshop Feedback Form [#1445] - User Mongo ID 57c4887f79058603002025e7 Your Name (Optional) Ian Munro Your En	Dec 4
Wufoo	Workshop Feedback Form [#1444] - User Mongo ID 5842519643adc80004d4fa2f Your Name (Optional) Evan Williams Your	Dec 4


4
0
Game Loop
Game Loop
MENU

FINITE POSSIBLE ACTIONS

- Paddle 1 move up
- Paddle 1 move down
- Paddle 2 move up
- Paddle 2 move down
- Ball move
- Player1 gain point
- Player2 gain point



WHAT REDUX IS

- A small tool for containing, accessing and affecting a set of information — often called “state”

- An effective player in architecture of an application using React

- A good gateway drug to strict functional programming

WHAT REDUX IS NOT

- A React-specific tool
- react-redux
- Tremendously valuable in a small-scale application
- The only game in town

THE STORE

- The singular holder of information
- Can receive dispatched signals (actions) meant to affect state
- Provides an interface to access state as well as listen for state changes

THE REDUCER

- Component of the store that decides: based on this signal (action), the new state should be this
- Creates new states per action, rather than modifying previous states
- Can be replaced and/or chunked (explored later)

ARRAY.PROTOTYPE.REDUCE

```
function reduce (array, callback) {
  return reducedValue;
}
function callback (memo, nextValue) {
  return memo;
}
```

ARRAY SUM

```
[1, 2, 3].reduce((memo, next) => {
  return memo + next;
});
```

REDUX

```
let actions = [incrementCounter, updateUserName]

actions.reduce((state, action) => {
  return resultOfUpdatingOldStateWithAction
}, oldState);
```

OTHER ANCILLIARY FEATURES

- **combineReducers**
  - Combine many reducers into a composite reducer
  - Useful for categorizing per property or grouping on your state

- **Middleware**
  - Components that act as a channel between your dispatch and your reducer
  - Useful for logging, asynchronous operations
  - Most of redux third-party ecosystem comprised of these

IMMUTABILITY AND PURITY

NEW STATE

- **Imperative that your reducer produces a new version of your state, rather than mutating the previous state**
  - Unintended side effects
  - Maintaining history and time traveling
- **Think of it like a “git commit”**
- **Operations that will mutate**
  - Adding/removing/modifying properties
  - Array.prototype.push/shift/unshift/pop/reverse/splice
  - Object.assign (if used the wrong way)

STAYING IMMUTABLE

- **Object.assign({}, previousState, newProperties)**
  - ES6 feature, merges objects in order
  - Start with an empty object literal, which will receive previousState properties first, then newProperties in turn
- **listOfItems.concat([newItem])**
  - An immutable alternative to .push
- **[…listOfItems, newItem]**
  - ES6 feature called “spread”, spreads elements of an array into another array
  - Often used for argument lists as well (replaces use case of Function.prototype.apply)

FUNCTIONAL PURITY

- **More formally called “referential transparency”**
- **The same function called with same arguments will always produce the same output**
  - (x, y) => x + y
  - NOT: (x, y) => x + y + Math.random()
- **No side-effects**
  - (x, y) => x + y
  - NOT: (x, y) => { z = y; return x + y; }

YOUR REDUCER SHOULD BE A PURE FUNCTION

COMMON IMPURITIES

- **AJAX calls or socket messages**
  - Considered side effects
  - These belong in action creators

- **Any randomization**
  - Randomize the value that will be sent in the action

- **Usage of dynamic closure information**
  - Once again, compute in action creator

BACK TO REACT

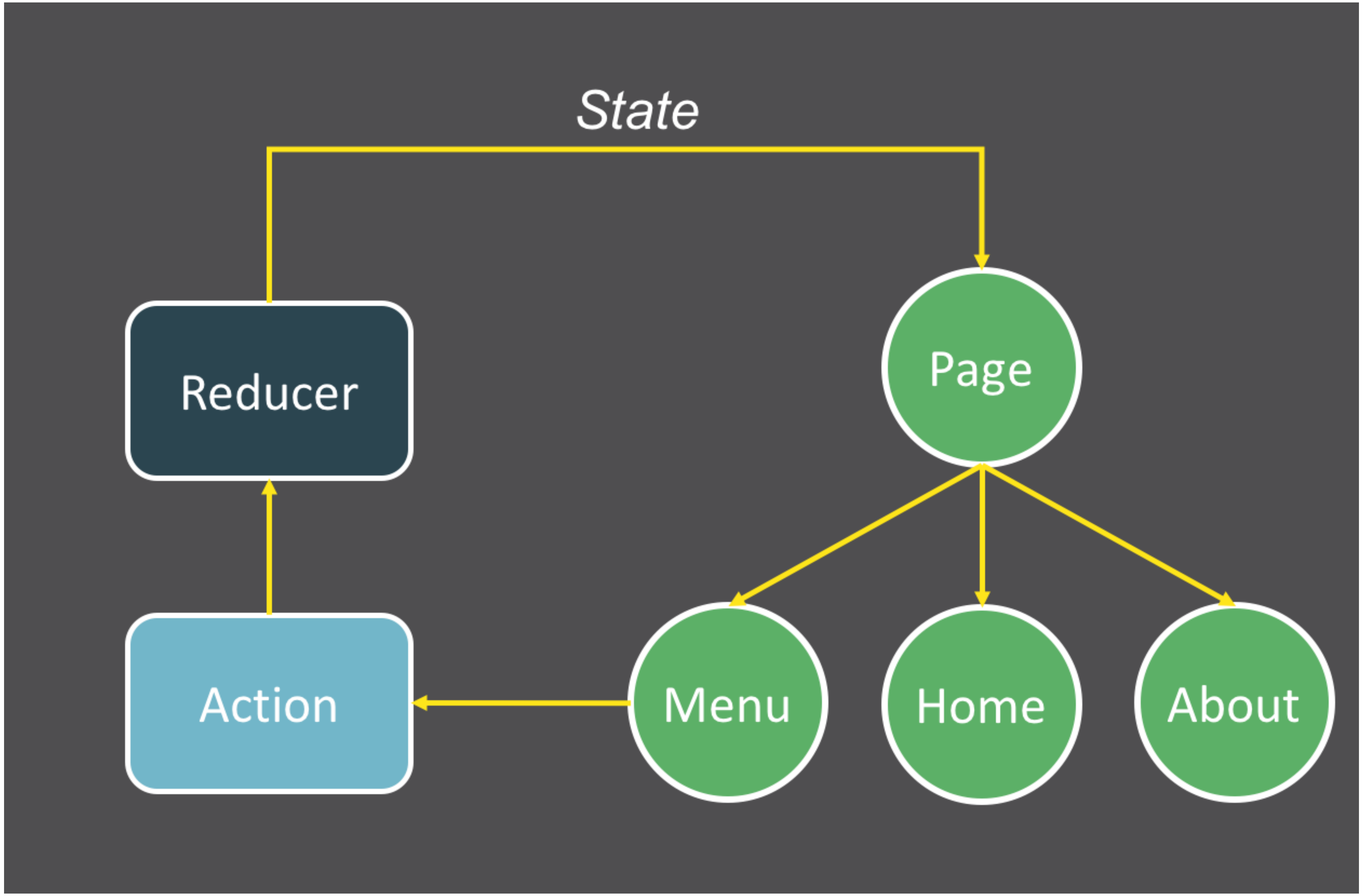
REACT & REDUX

- **React does not try to solve the problem of application state or architecture**

- **Redux is a good fit for a component-system based on unidirectional data flow**
  - “State at the top, passed down as prop”

- **Components trigger signals (actions) to Redux store, new store is produced, smart components react and setState**

ACTION TYPES AND CREATORS

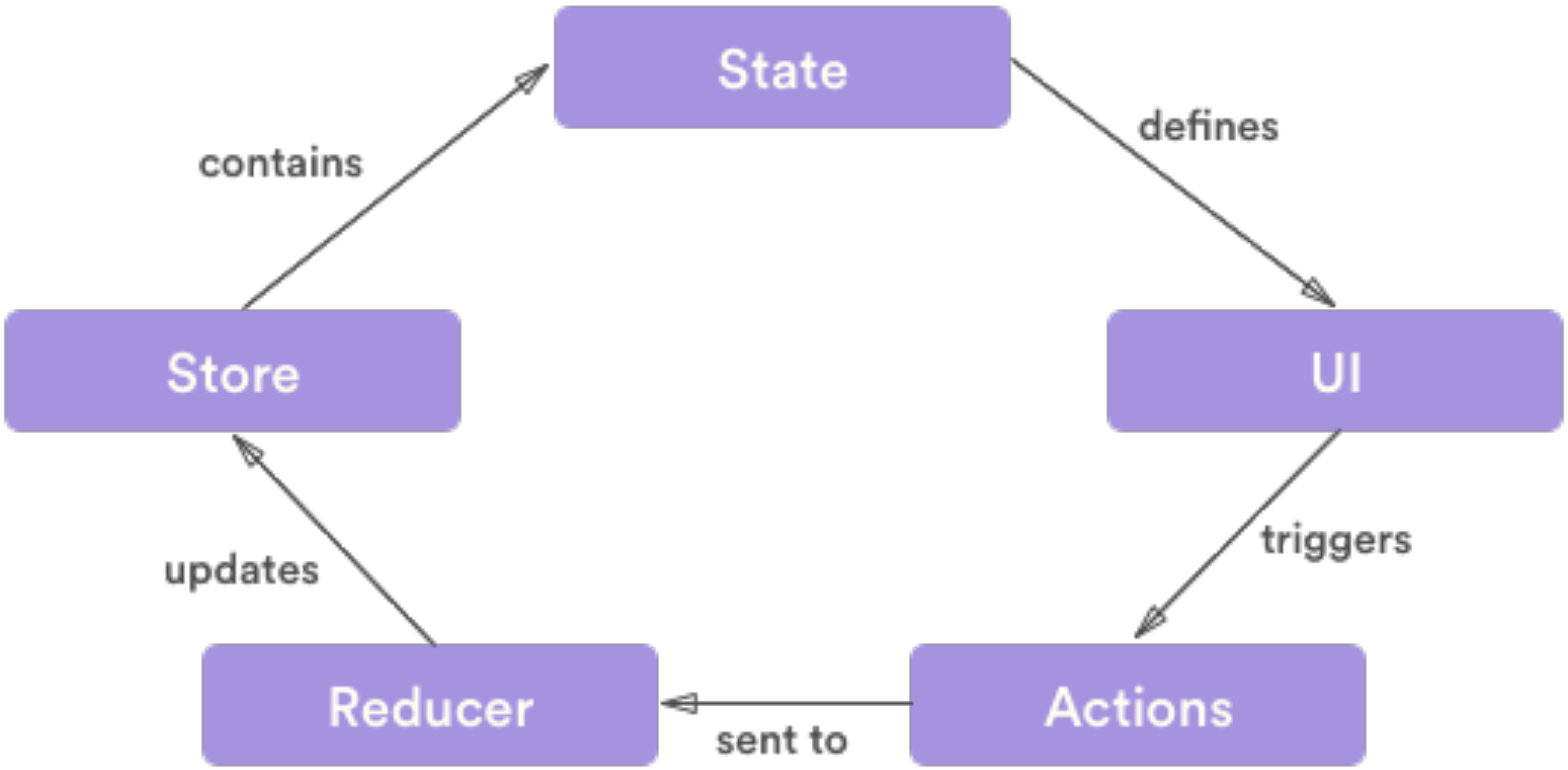


http://www.bebetterdeveloper.com/coding/getting-started-react-redux.html

CONNECTING SMART COMPONENTS

- **store.subscribe(() => this.setState(store.getState()))**
  - componentDidMount
  - When Redux produces a new state, registered listeners will be invoked
  - Very much like event emitter
  - Returns an “unsubscribe” function

- **Unsubscribe (returned from subscribe)**
  - Necessary for preventing memory leaks and errors
  - componentWillUnmount



http://www.theodo.fr/uploads/blog//2016/03/ui_workflow.png

> *“Simplicity is prerequisite for reliability.”*

– EDSGER DIJKSTRA

REDUX IS SIMPLE

- It’s just a couple of functions - you could write it yourself
- Simplicity is powerful
- All signals to change application state go through one channel
- Encourages/enforces programming tactics of immutability and function purity

THREE PRINCIPLES

- **Single source of truth**
- **State is read-only**
- **Changes are made with pure functions**
  - No side effects (AJAX, mutations)
  - Same input => same output, no matter what!