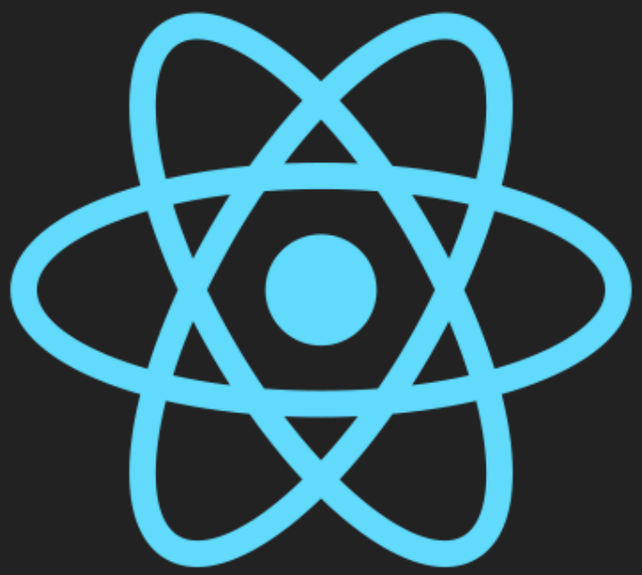
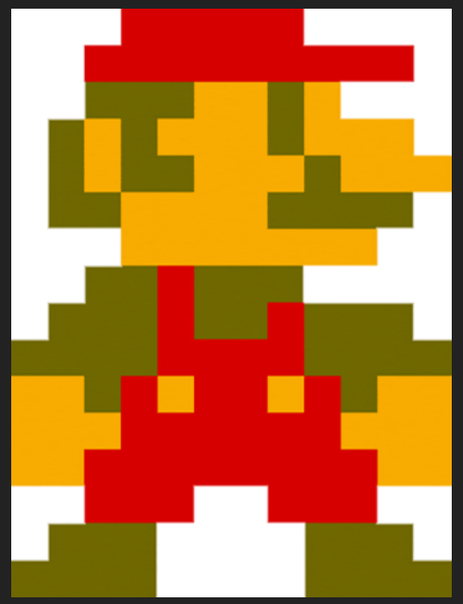
REACT AND GAMES

An experiment about making games

@BOBYLITO

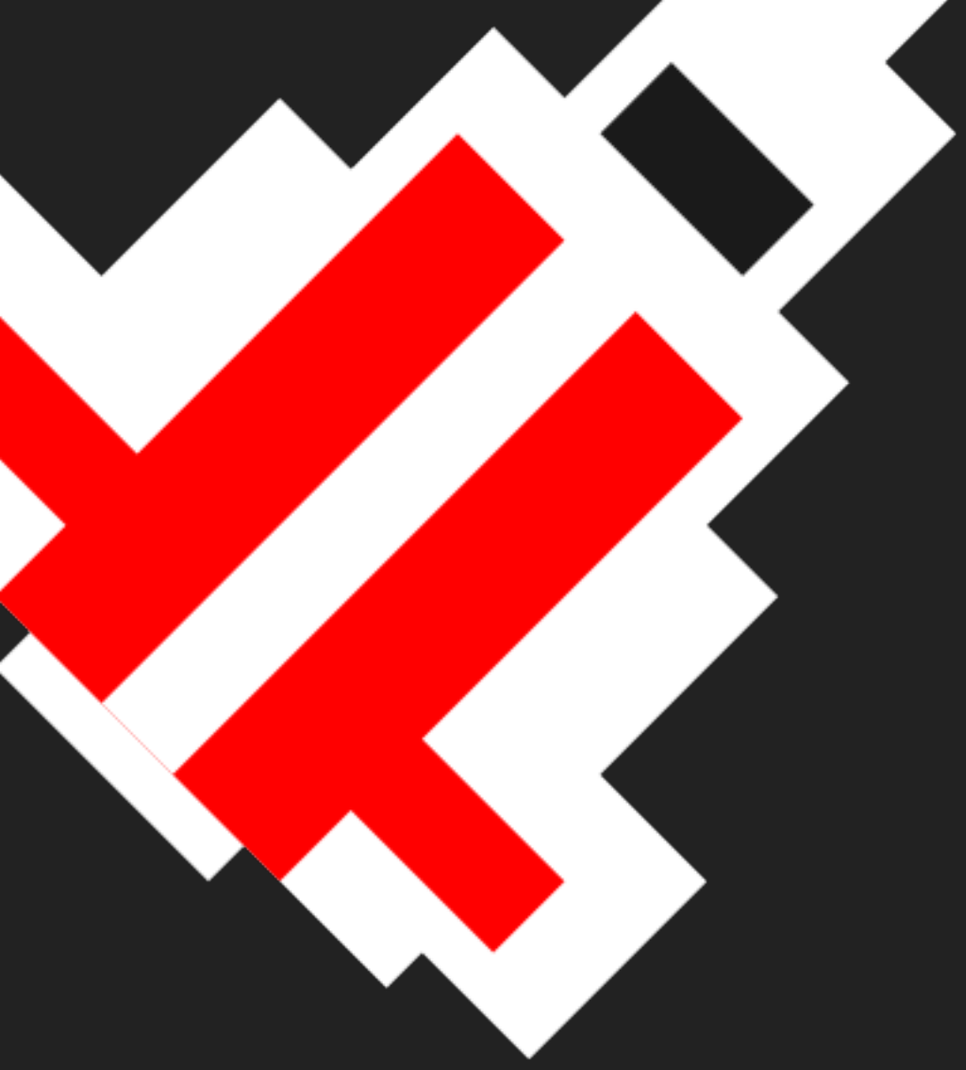


+



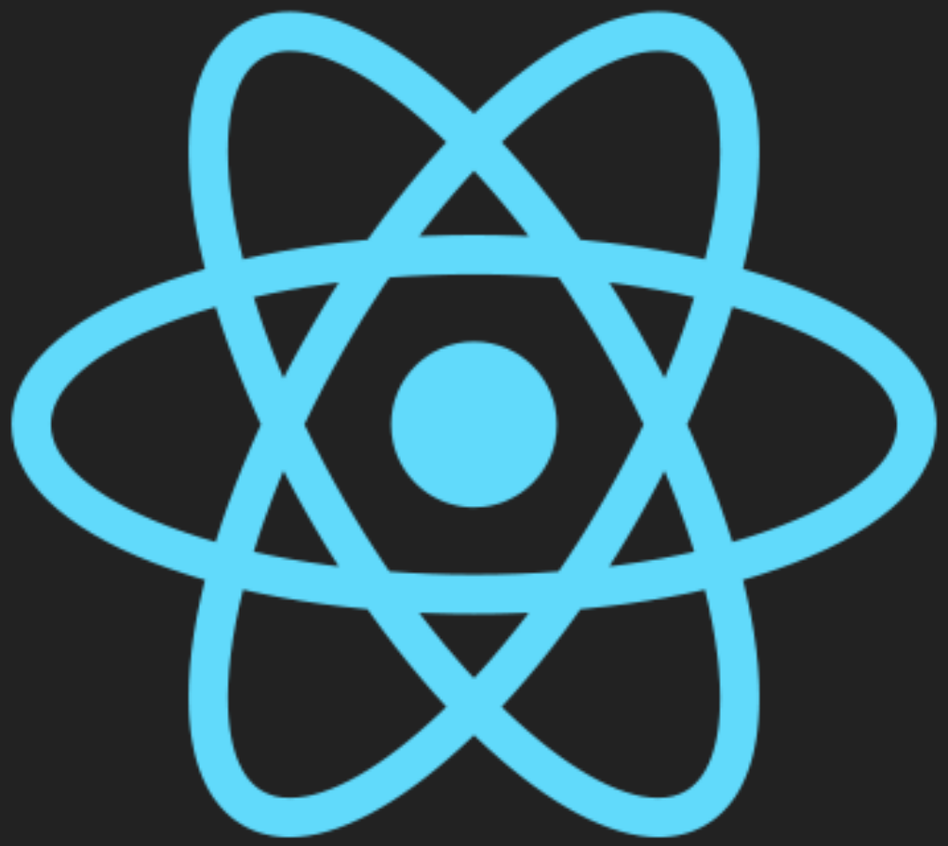
=

?



Little Shooter v2

1. (re)Introduction to React
2. Game dev patterns with React
3. Demo
4. Next



Rendering

```
1 React.render( <div>Hello World</div>, document.body );
```

Custom components

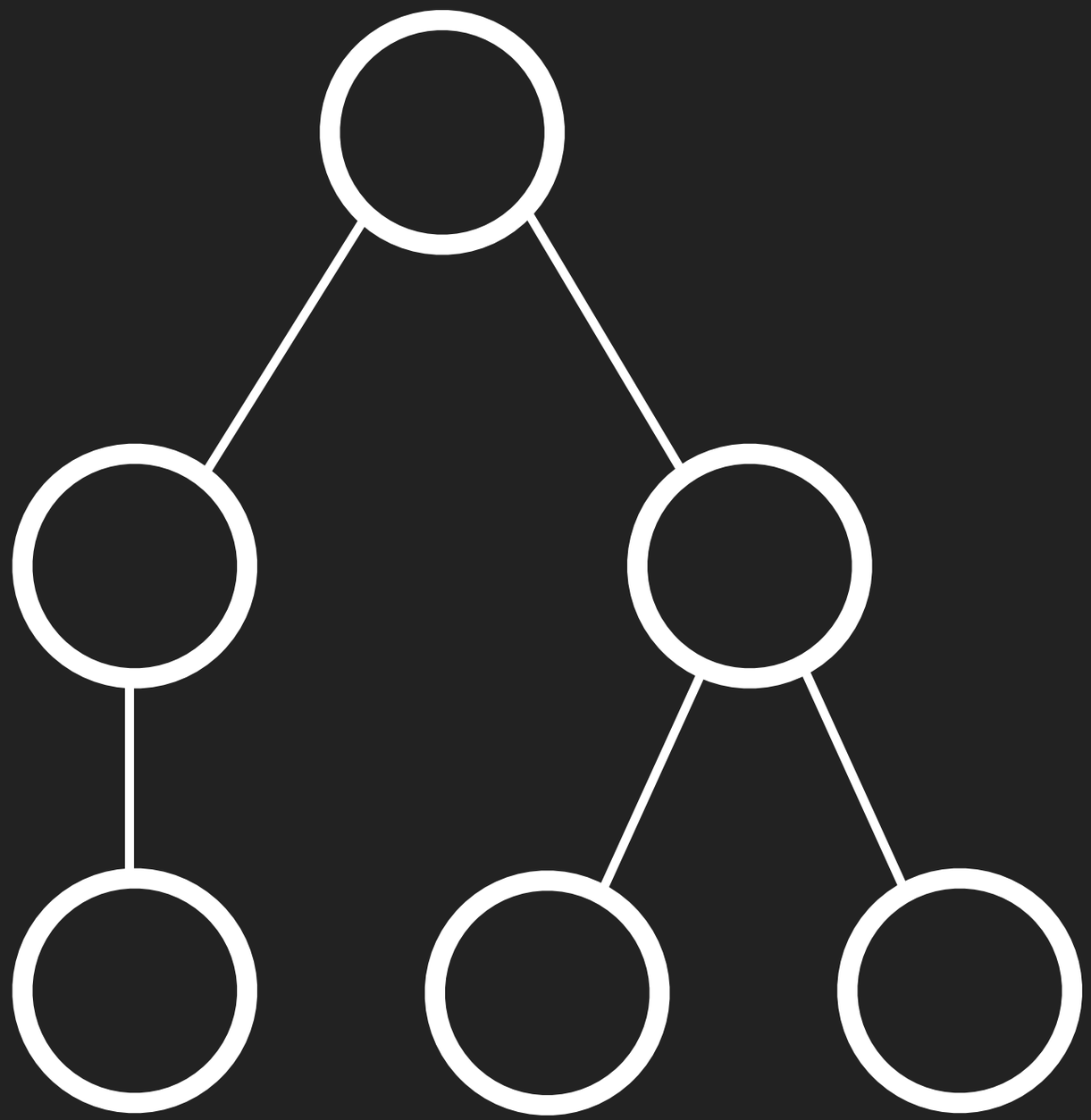
```
1 var MyComponent = React.createClass({
2   render: function(){
3     return <div><MyOtherComponent prop1="value"/></div>
4   }
5 });
6 React.render( <MyComponent>, document.body );
```

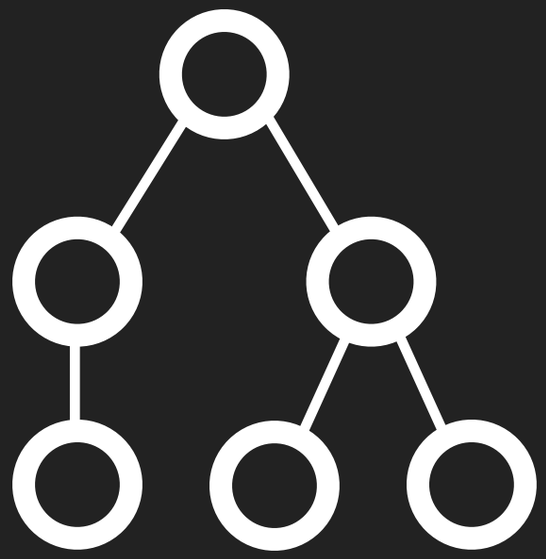
Component state

```
1 var MyComponent = React.createClass({
2   //state: {}
3   getInitialState: function(){ return {foo:"bar"} },
4   render: function(){
5     var valueFromState = this.state.foo;
6     return <div><MyOtherComponent prop1="value"/></div>
7   }
8   //setState: function( newState ){ /*set the new state*/ }
9 });
```

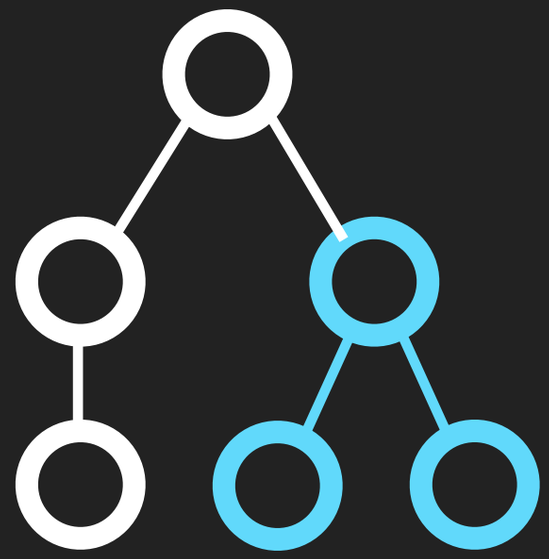
Lifecycle and hooks

```
1 var MyComponent = React.createClass({
2   render: function(){
3     return <div><MyOtherComponent prop1="value"/></div>
4   },
5   componentWillMount: function(){},
6   componentWillReceiveProps: function( props ){ /* new values
handling, modify state ;) */}
7 });
```





T-0



T-1

Diff T-0 T-1

LET'S MAKE A GAME!

Game Loop

Game loop

Game loop

Repeat

Game loop

Render

Repeat

Game loop

Update game state

Render

Repeat

Game loop

Get input

Update game state

Render

Repeat

```
var GameApp = React.createClass({
  getInitialState: function(){
    return {
      input : {
        time : (Date.now()) } };
  },
  render : function(){
    return <Game input={ this.state.input }/>
  },
  tick : function(){
    var t = Date.now();
    requestAnimationFrame(this.tick);
    this.setState({
      input:{
        time : t
      }
    });
  },
  componentWillMount : function(){
    requestAnimationFrame( this.tick );
  },
});

var output = d.getElementById("main");
React.renderComponent( <GameApp width="500" height="500" />, output);
```

Inputs

- Time
- Keys

Time

- Date.now() at the tick
- Store the time in the state
- Propagate to sub components
- Will trigger the rendering even if nothing else happen from the real world

Keys

- Bind event listeners to the main component
- Properly handle auto fire

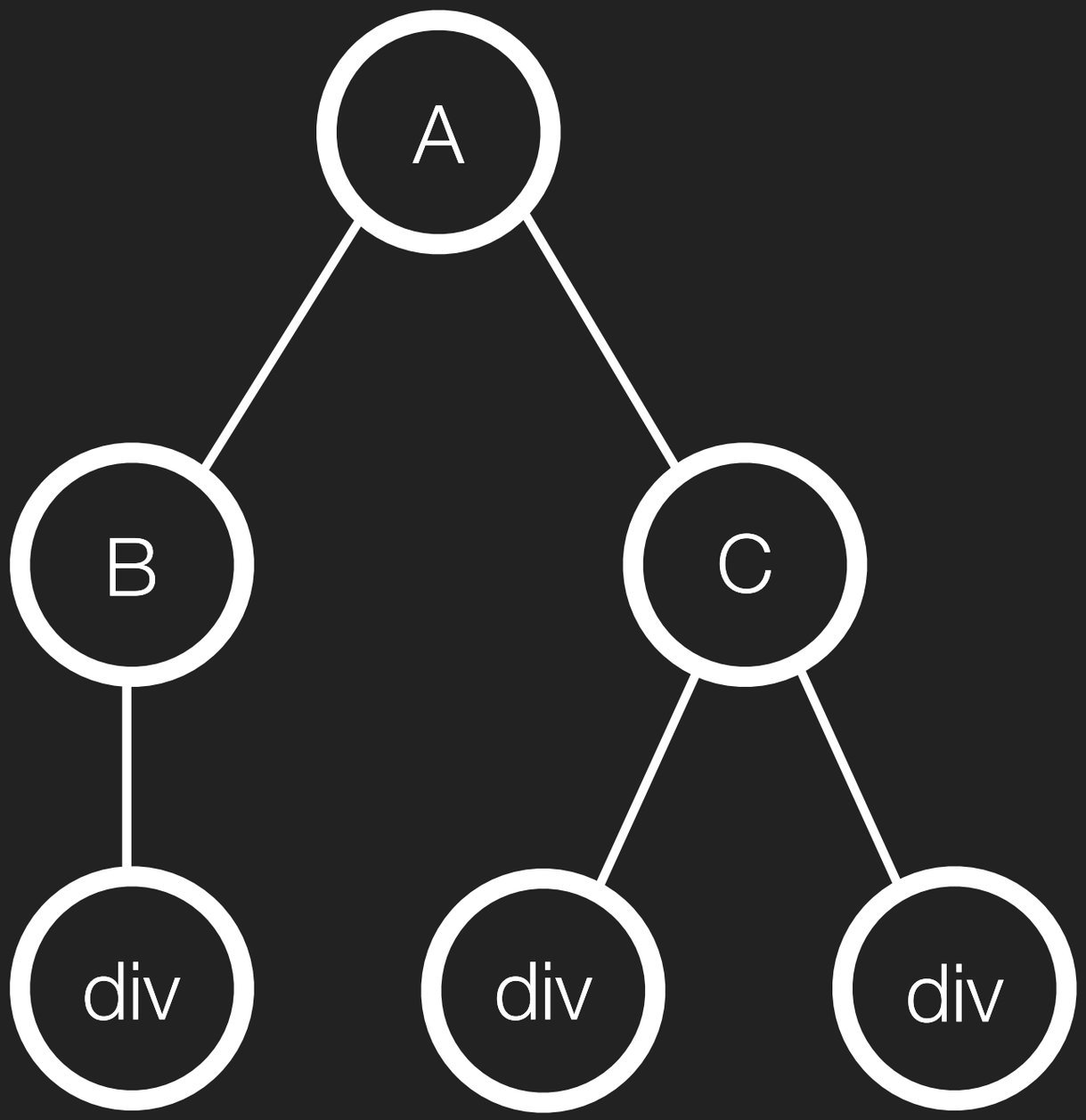
```
...
  render : function(){
    return <div className="game"
      onKeyDown = { this.keyHandler.bind(this, true) }
      onKeyUp   = { this.keyHandler.bind(this, false) } tabIndex="1"></div>;
  },
  keyHandler : function(valueToSet, e){
    var newKeys = {
      left  : this.state.input.keys.left,
      right : this.state.input.keys.right,
      up    : this.state.input.keys.up,
      down  : this.state.input.keys.down,
      space : this.state.input.keys.space
    };

    if(e.keyCode === 37) newKeys.left  = valueToSet;
    if(e.keyCode === 38) newKeys.up    = valueToSet;
    if(e.keyCode === 39) newKeys.right = valueToSet;
    if(e.keyCode === 40) newKeys.down  = valueToSet;
    if(e.keyCode === 32) newKeys.space = valueToSet;

    this.setState({
      input:{
        time : this.state.input.time,
        keys : newKeys } });
  }
...
```

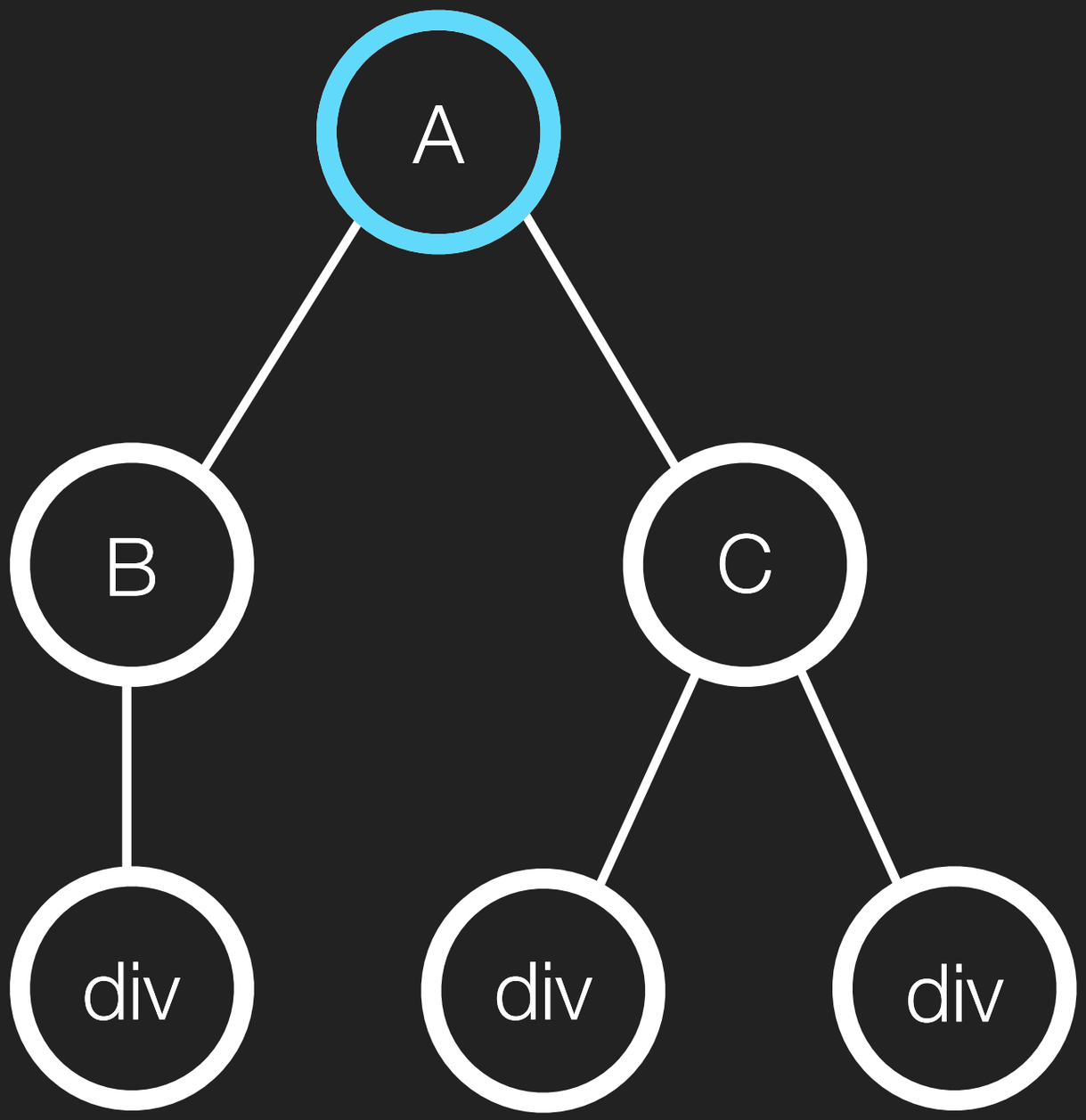
Game screens

React.render(<A />)



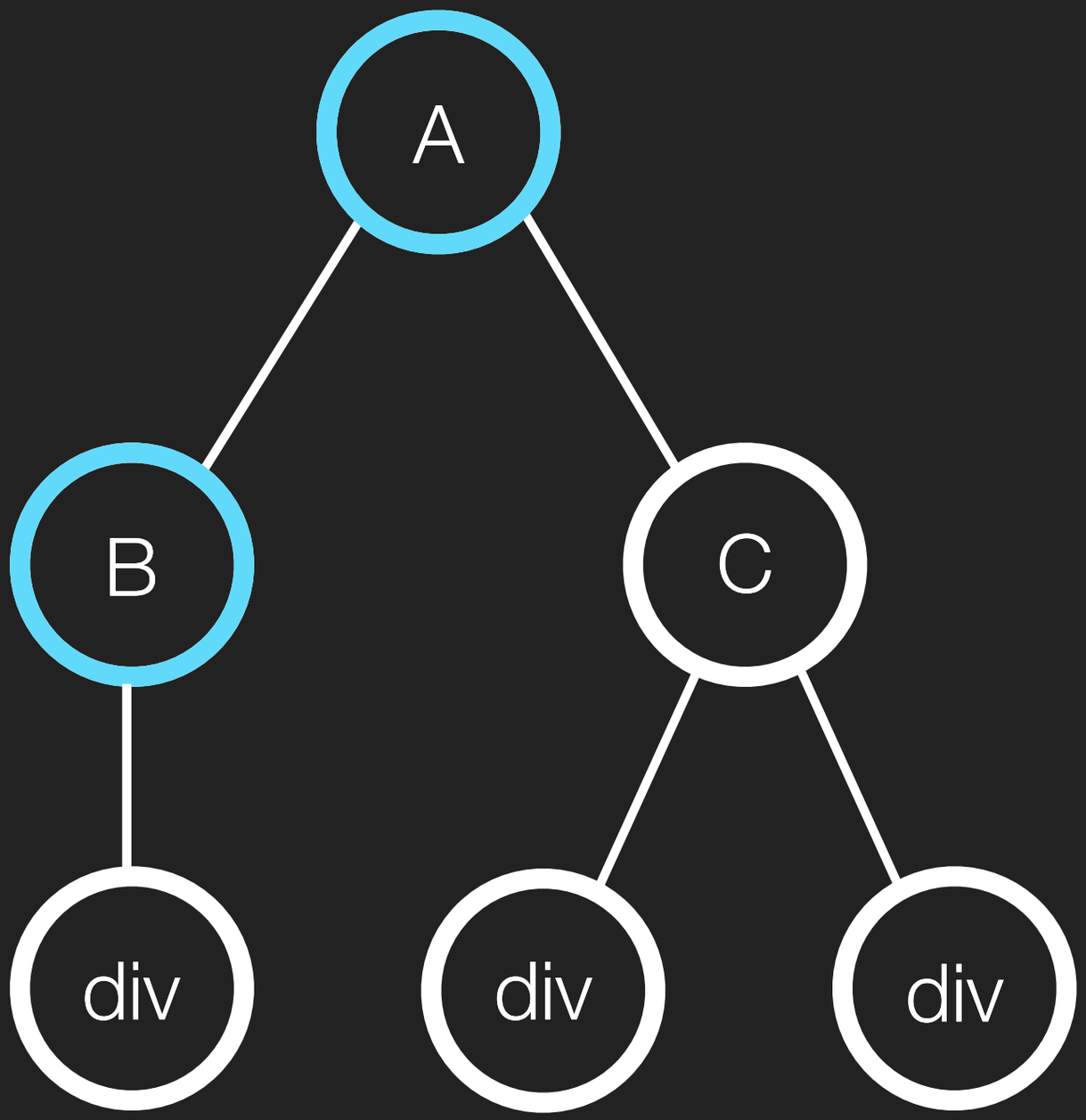

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



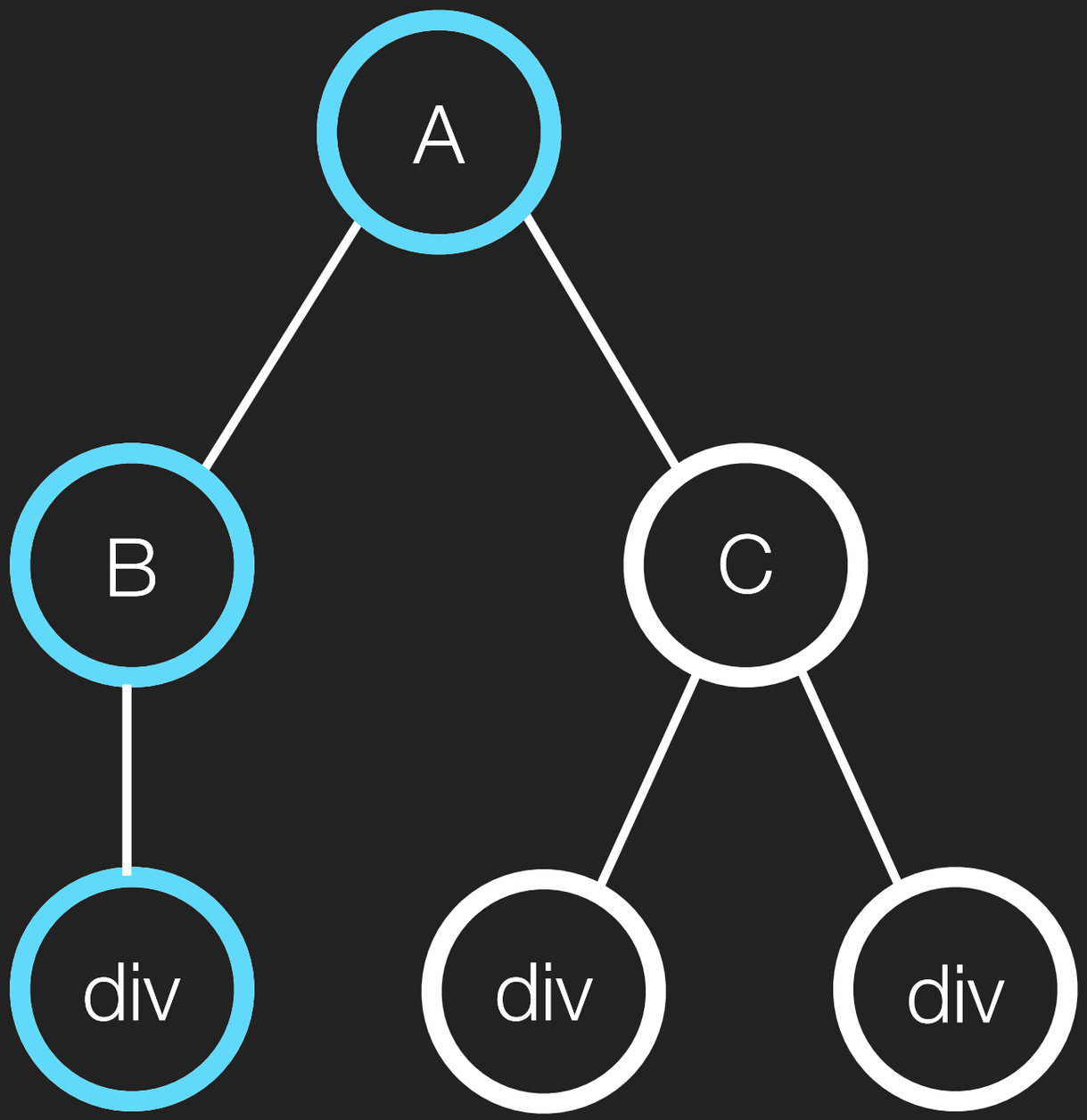

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



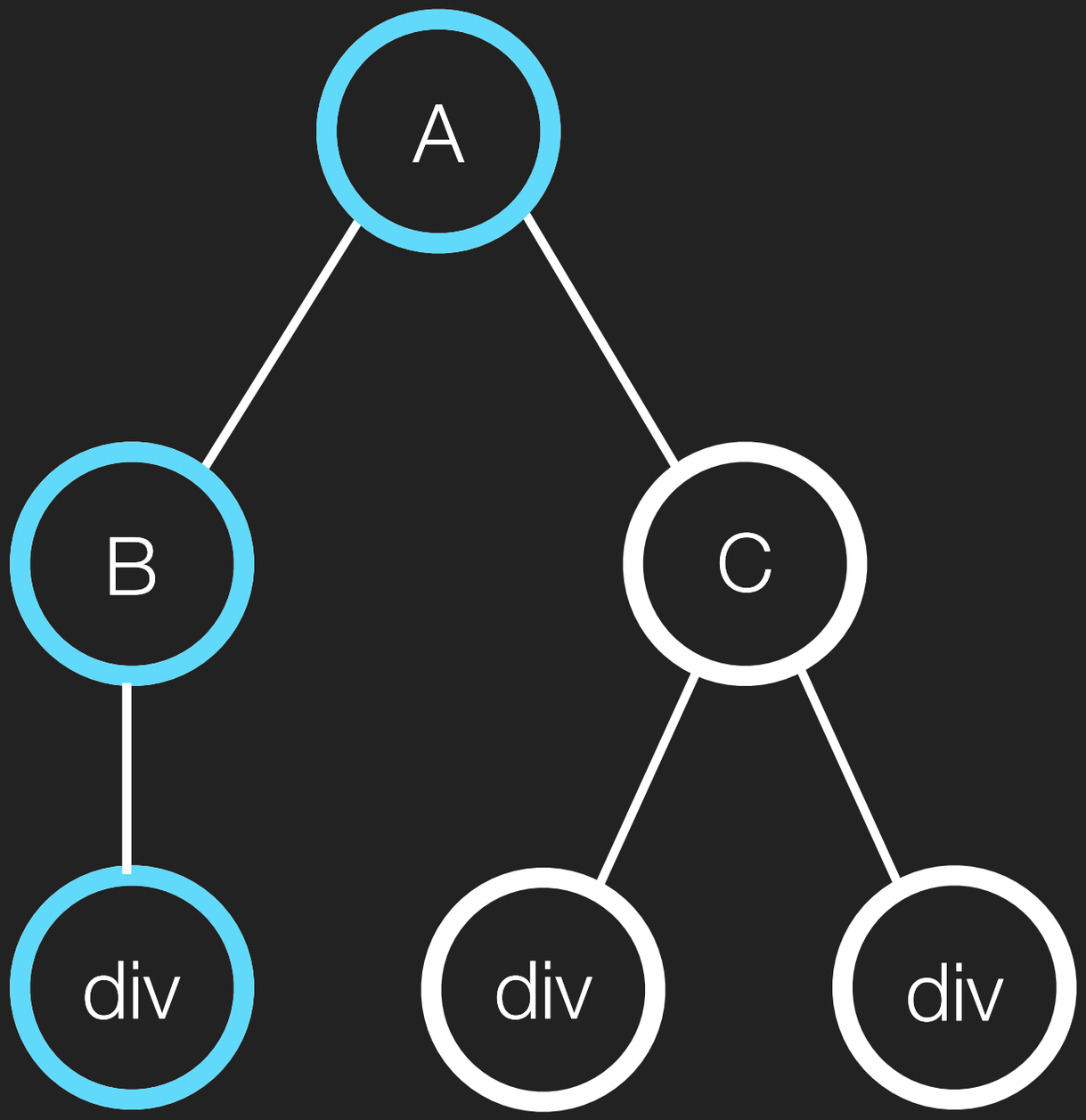

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



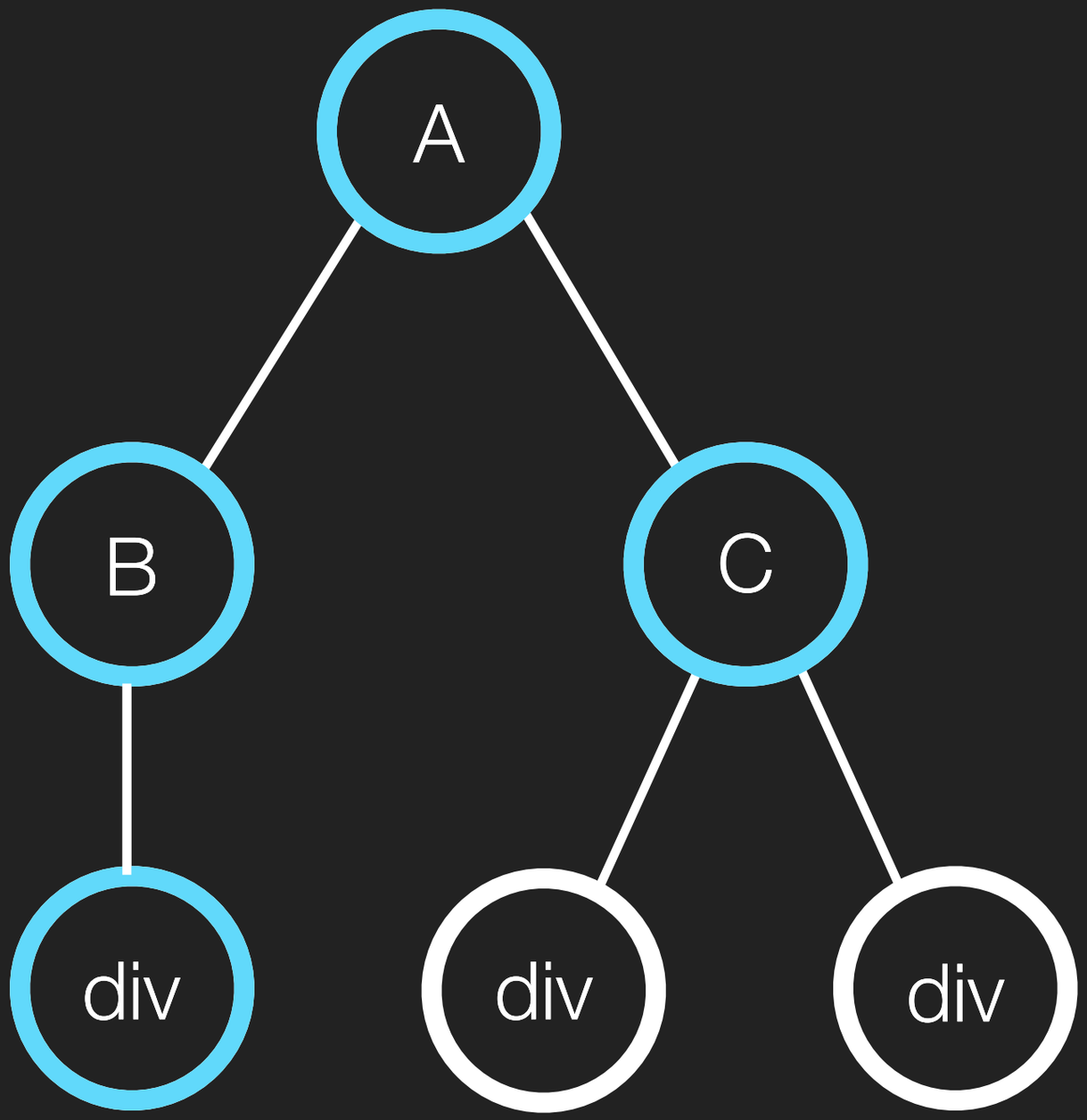

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



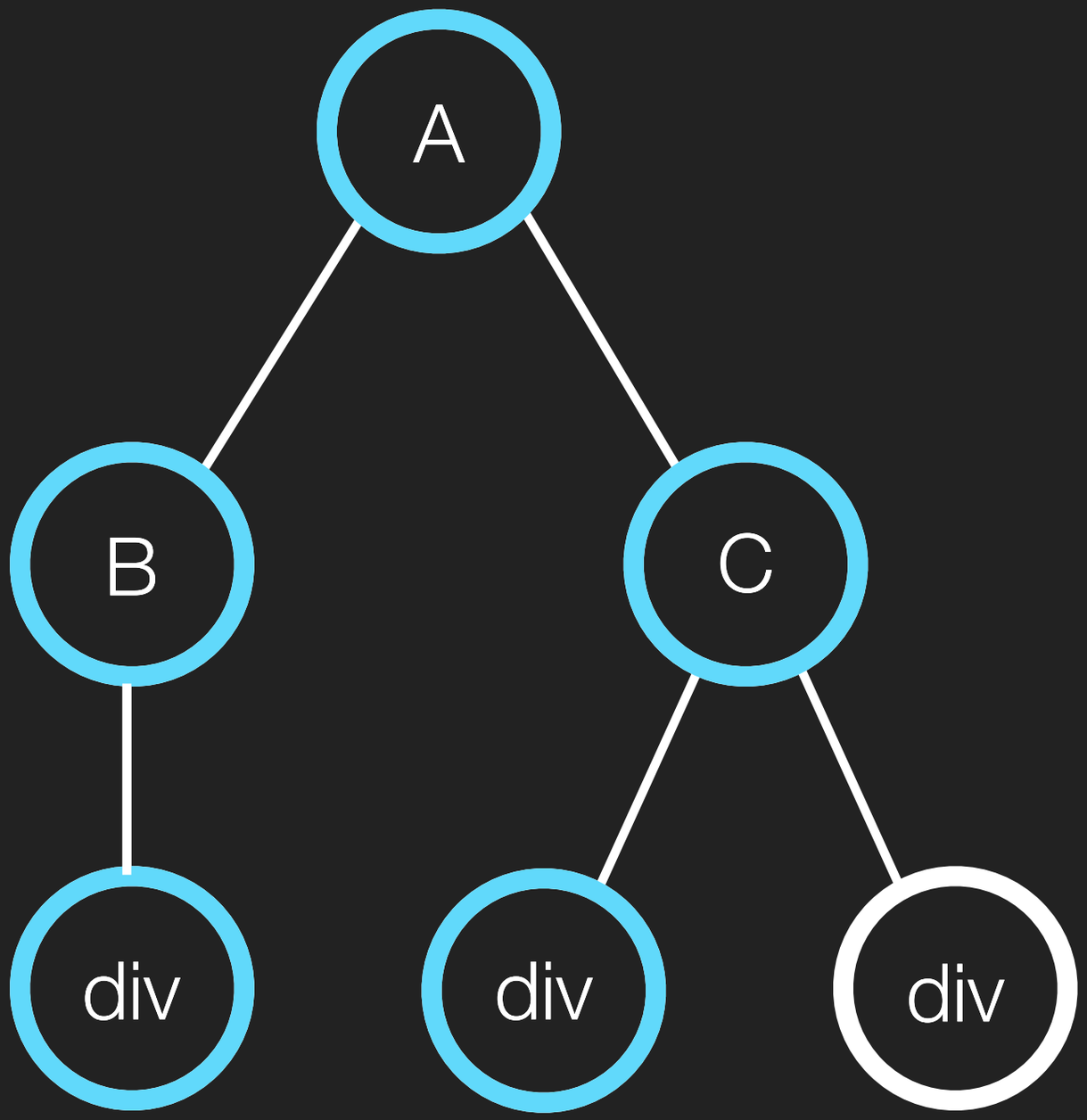

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



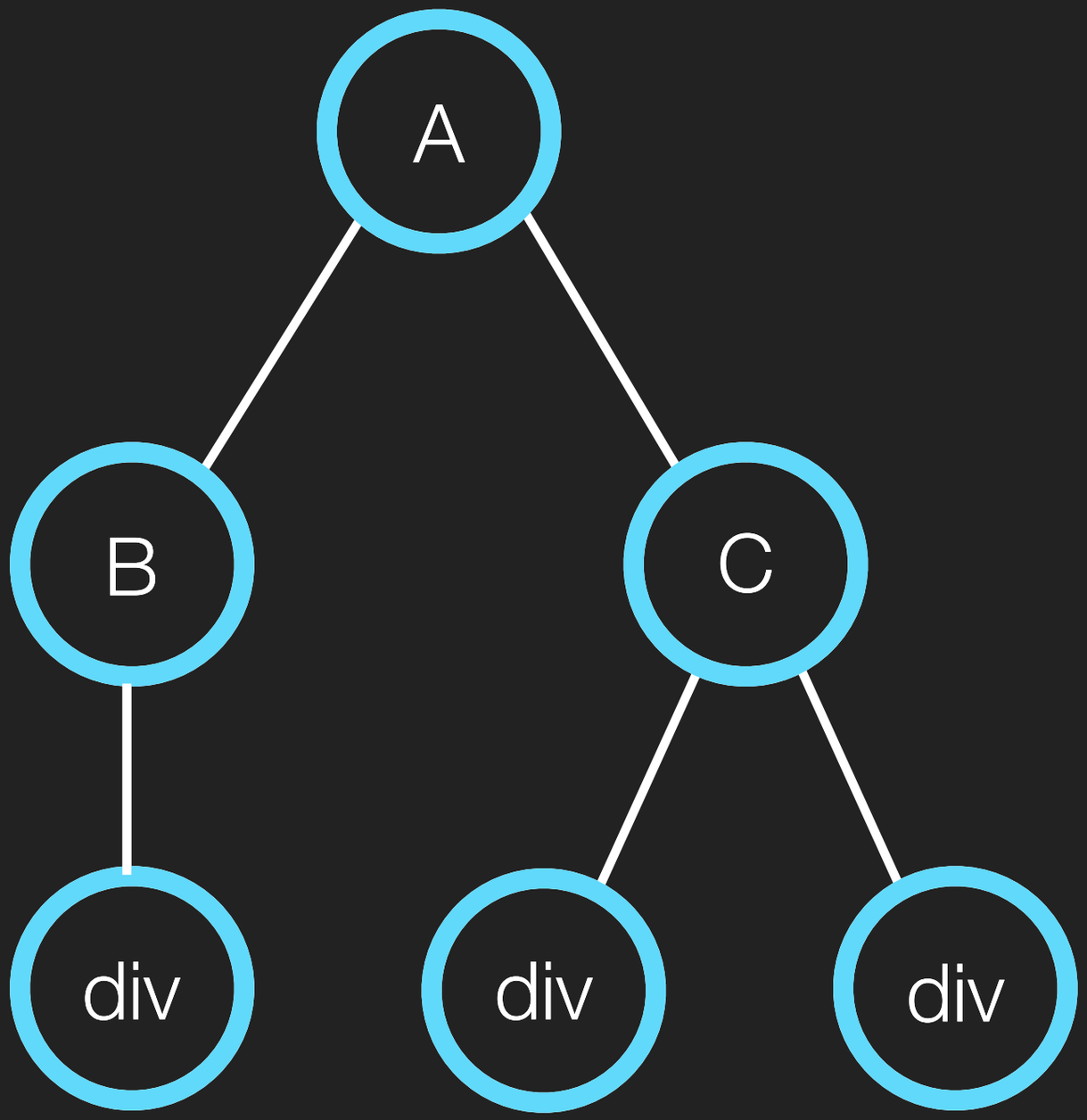

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



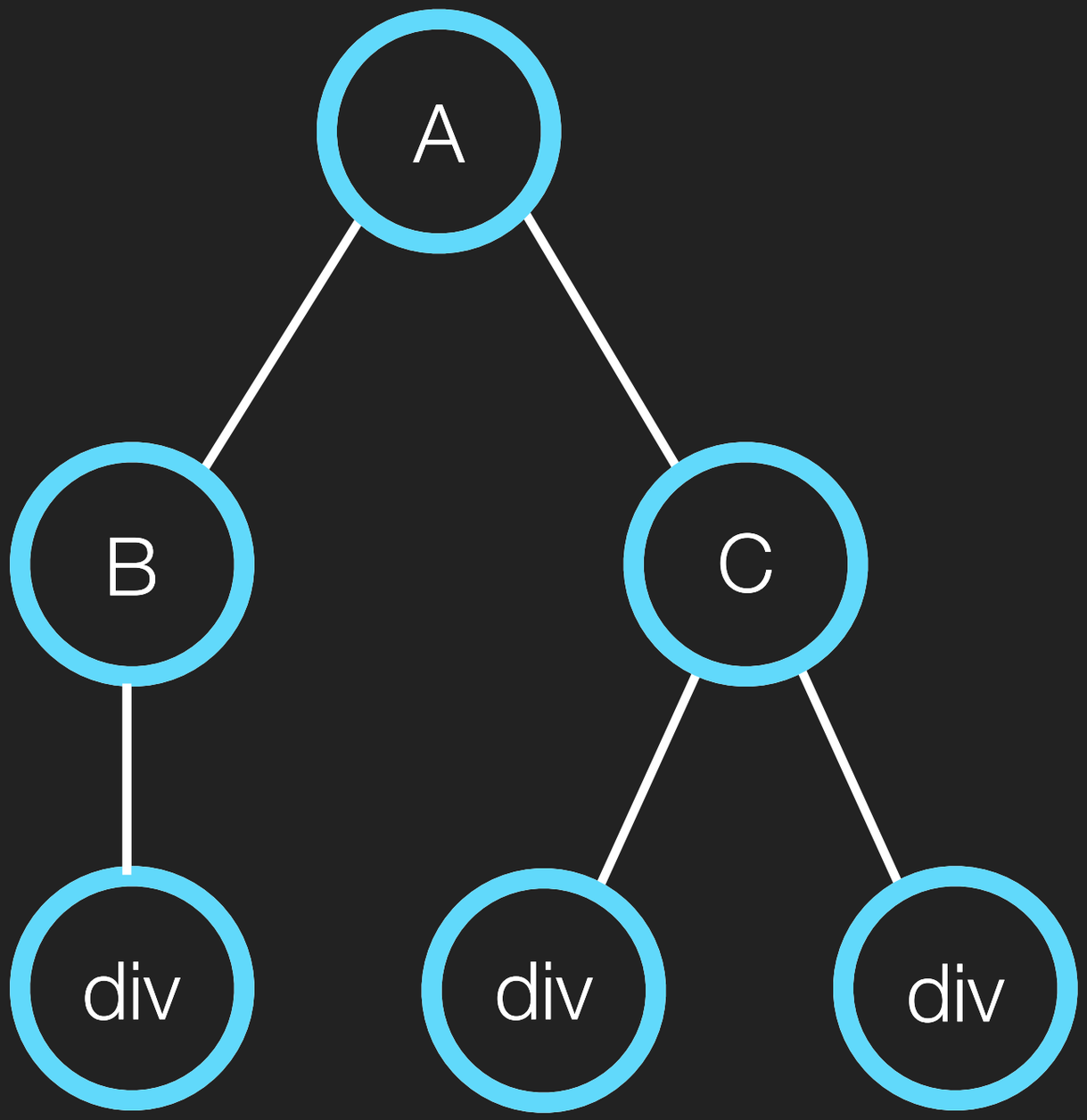

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)



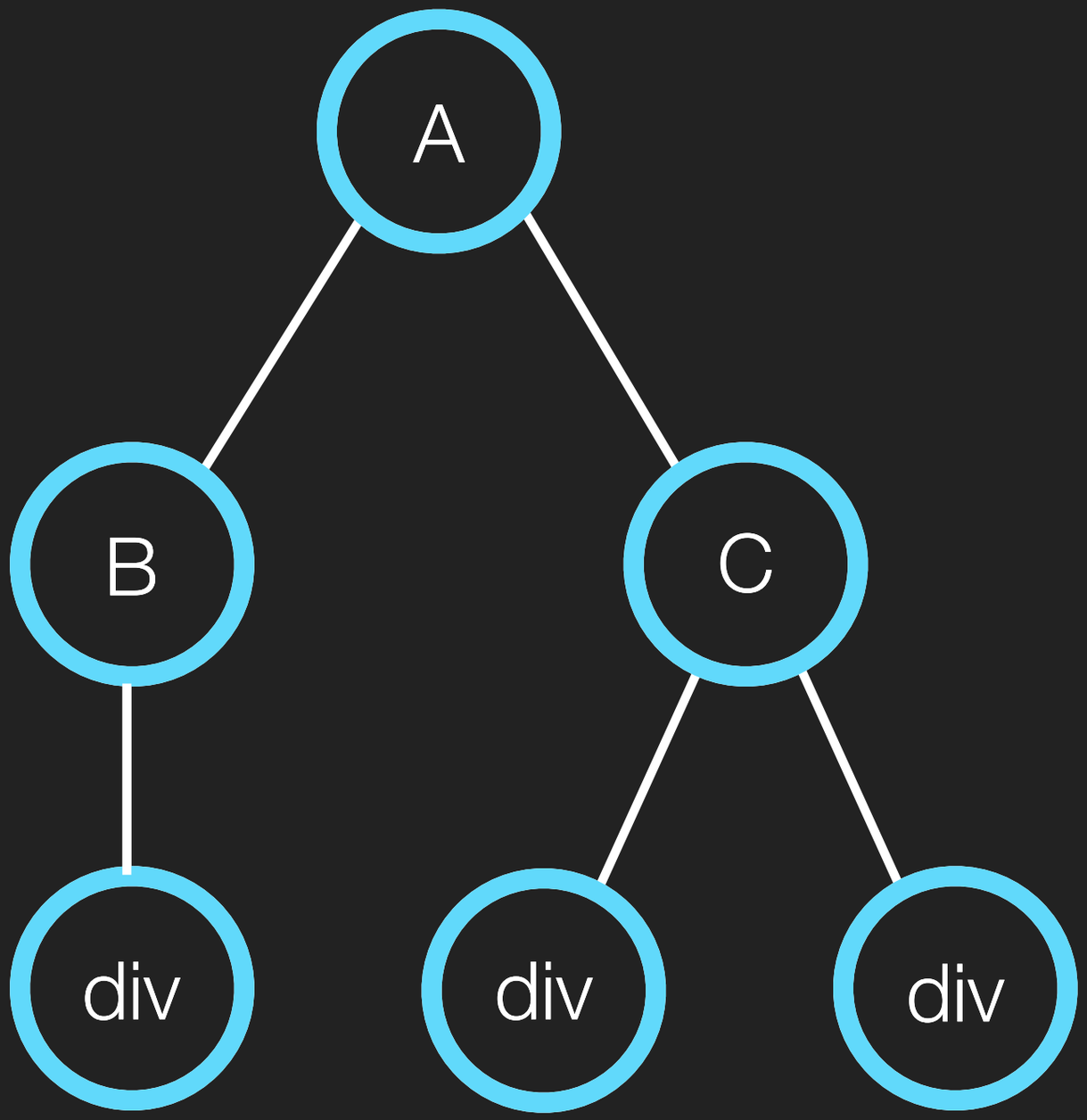

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

React.render(<A />)

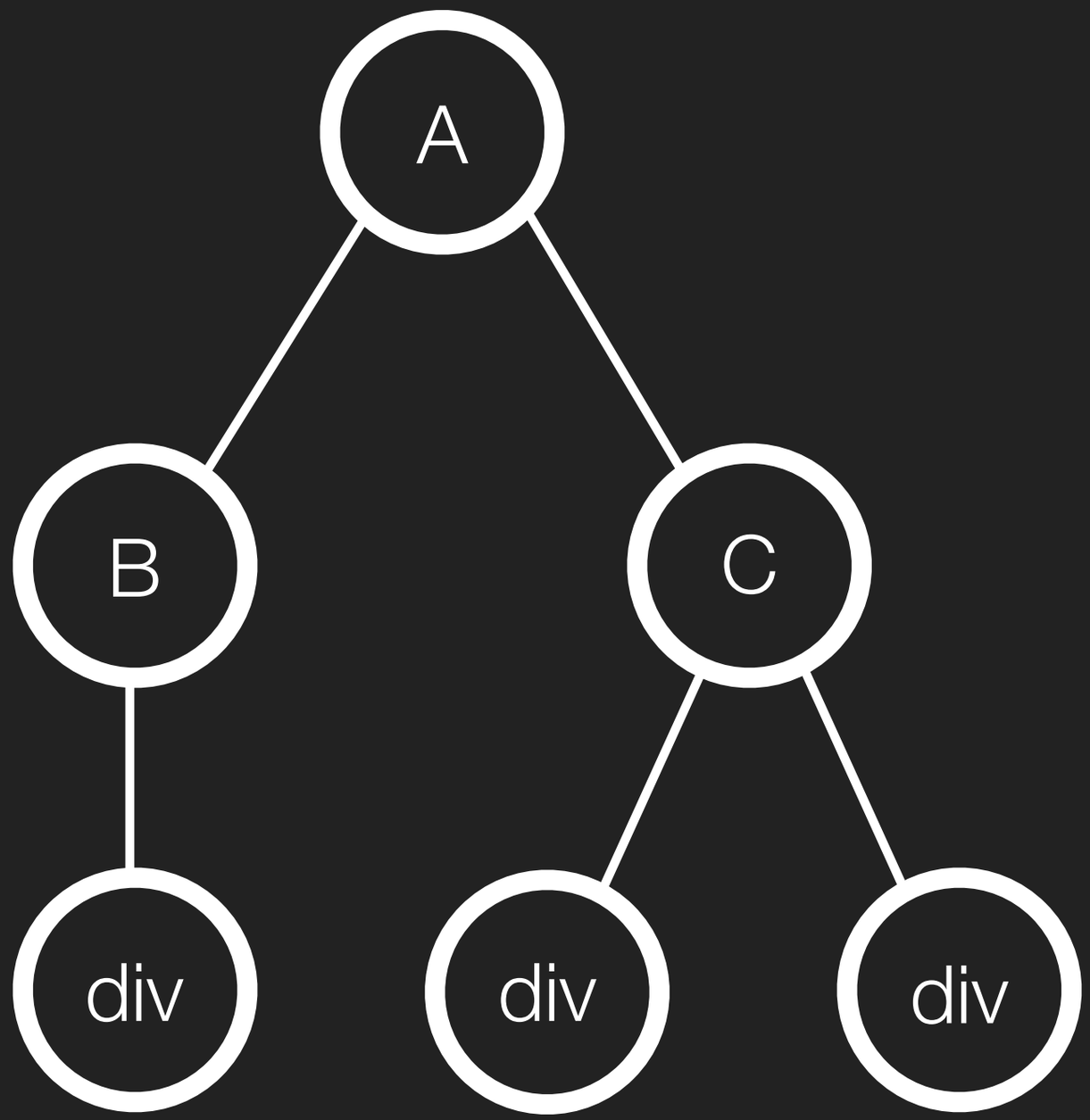


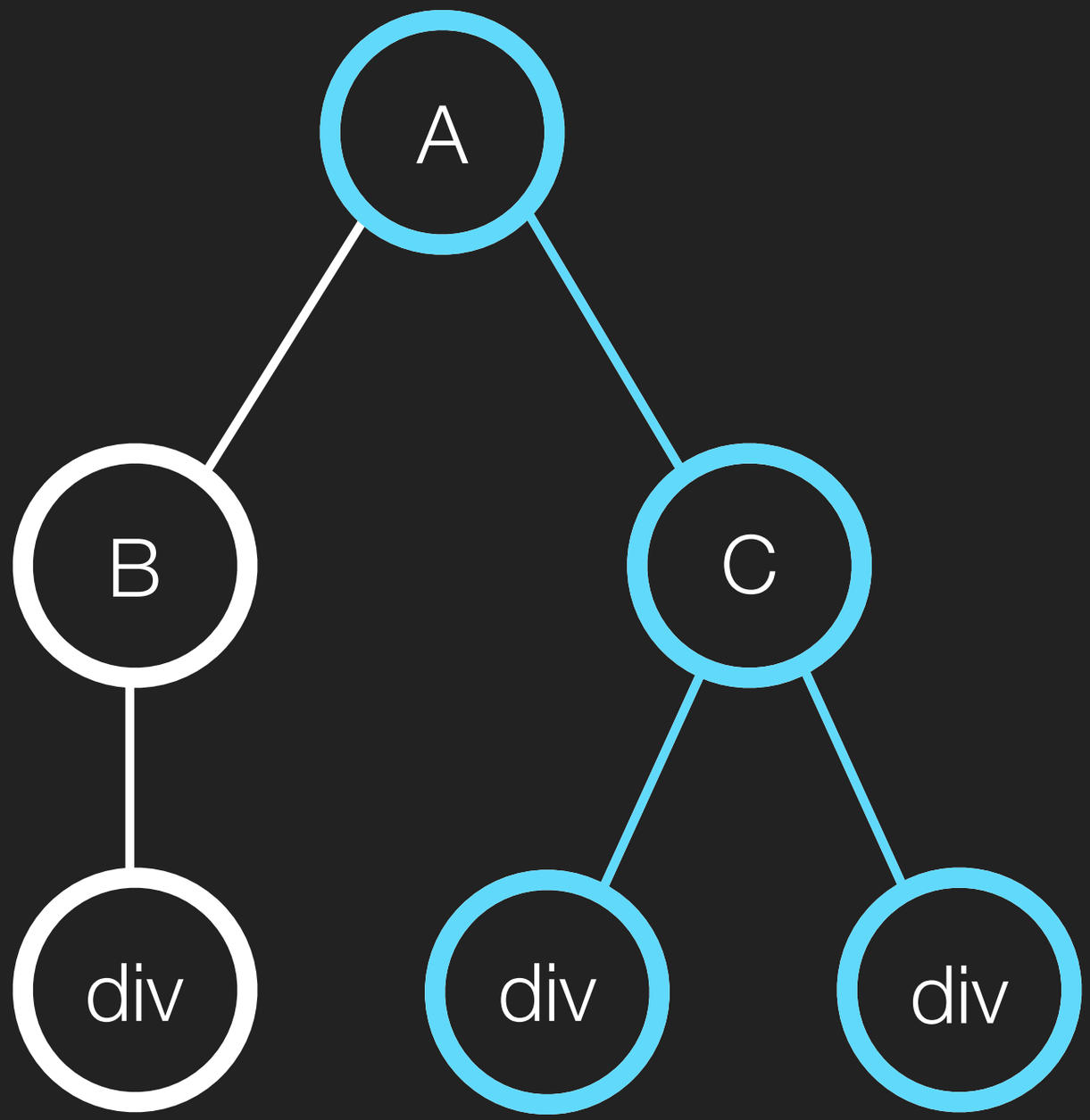

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

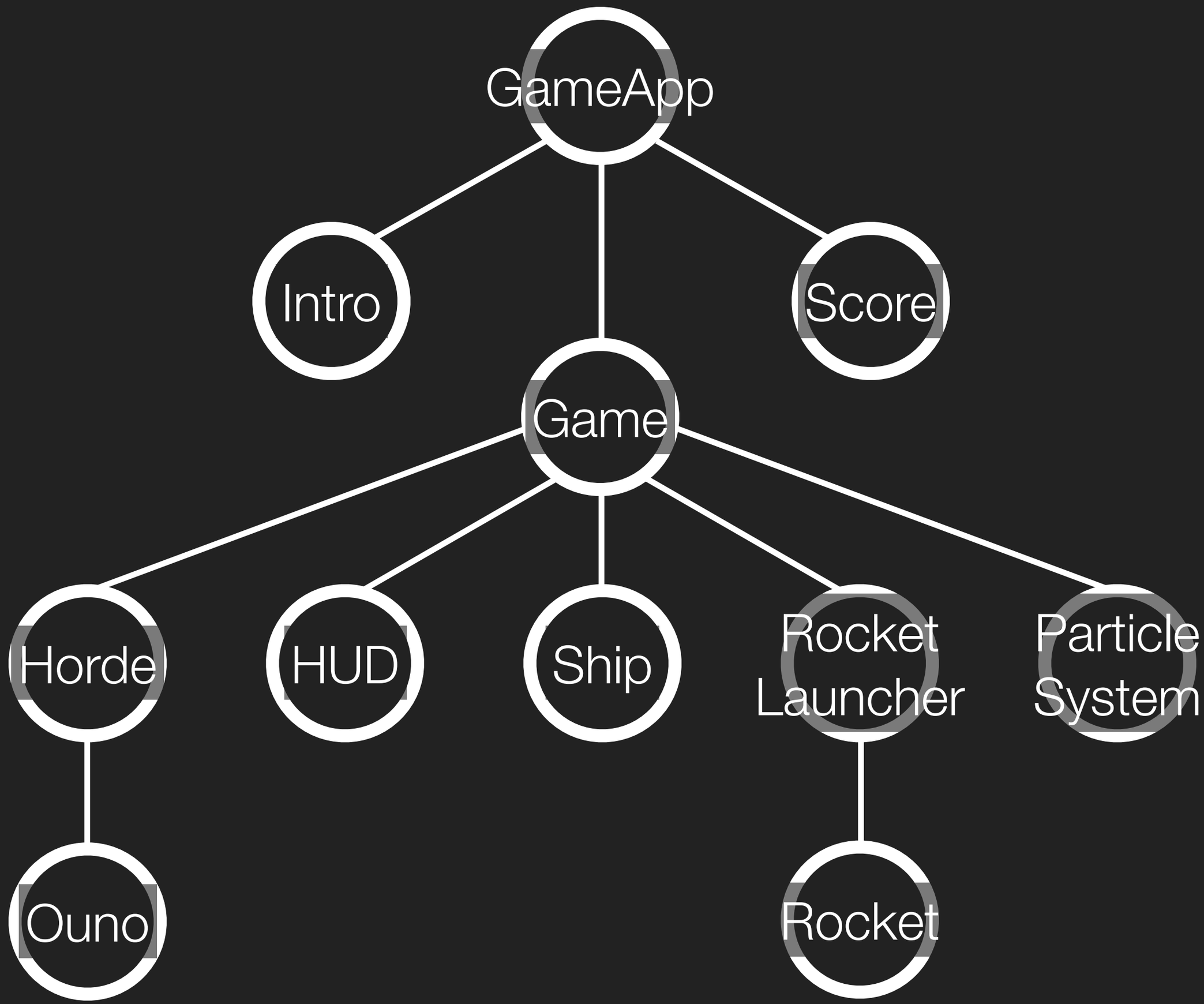
React.render(<A />)



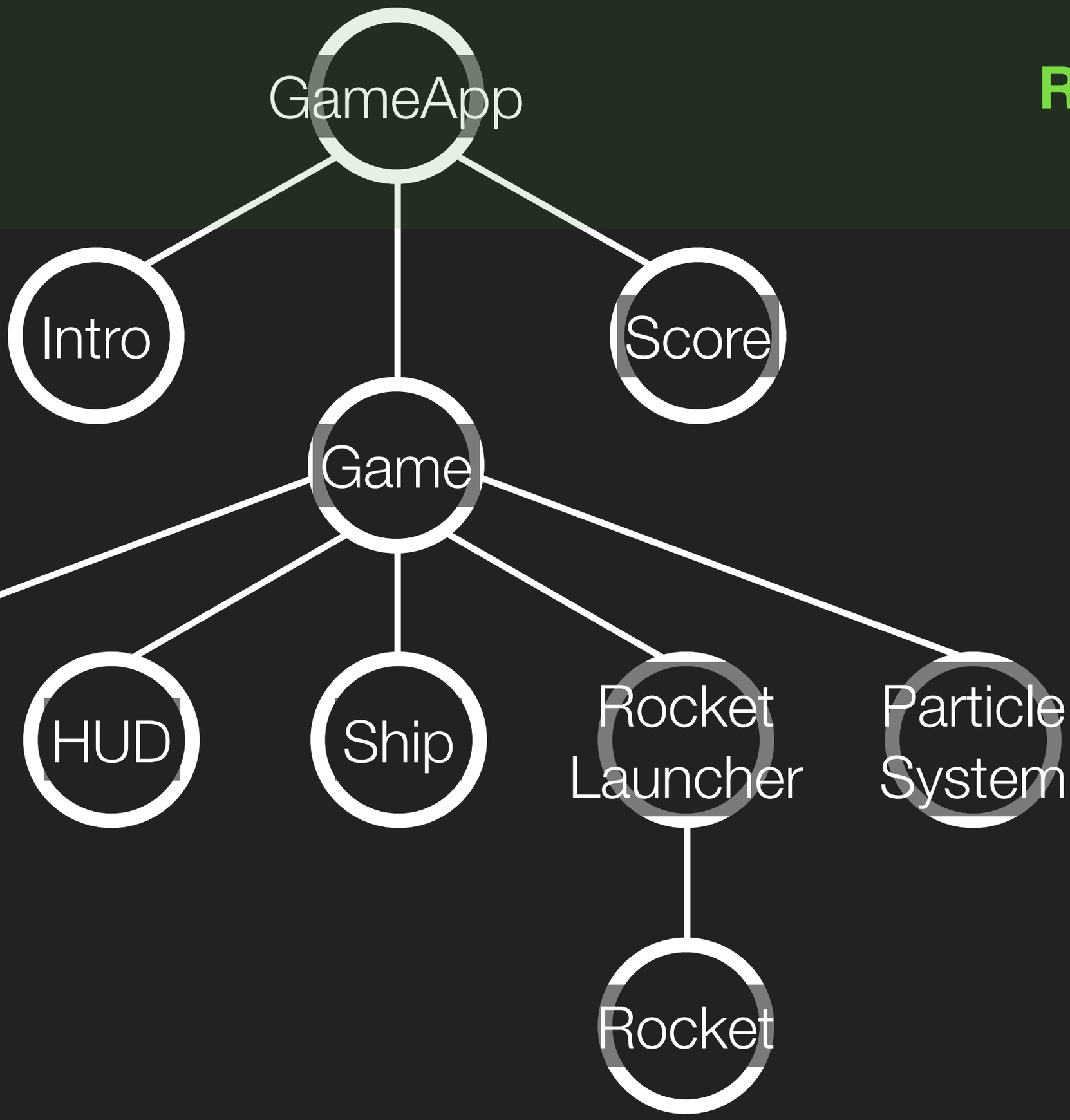

```
<div class="a">
    <div class="b">
        <div/>
    </div>
    <div class="c">
        <div/>
        <div/>
    </div>
</div>
```

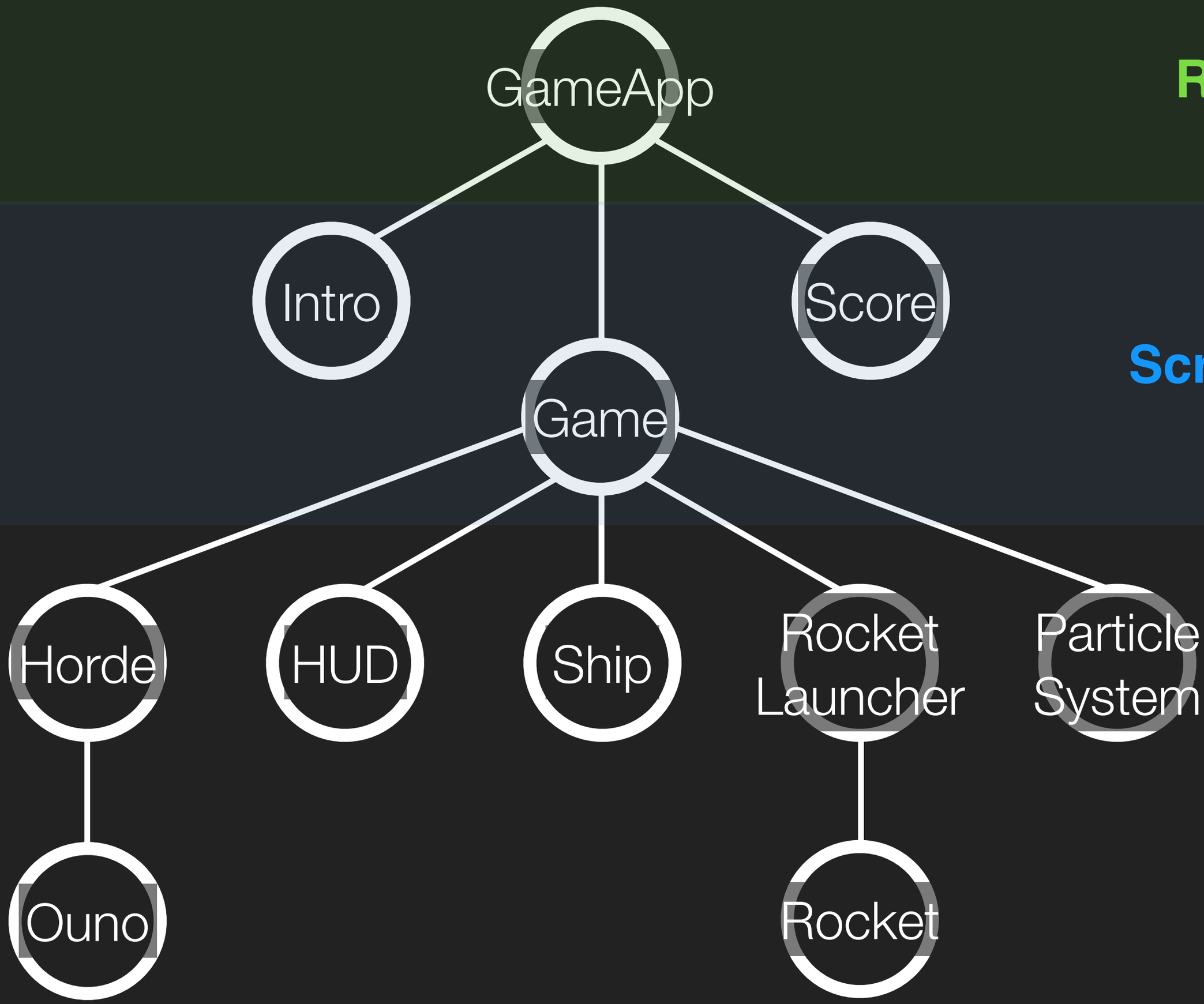

A
B
C
div
div
div

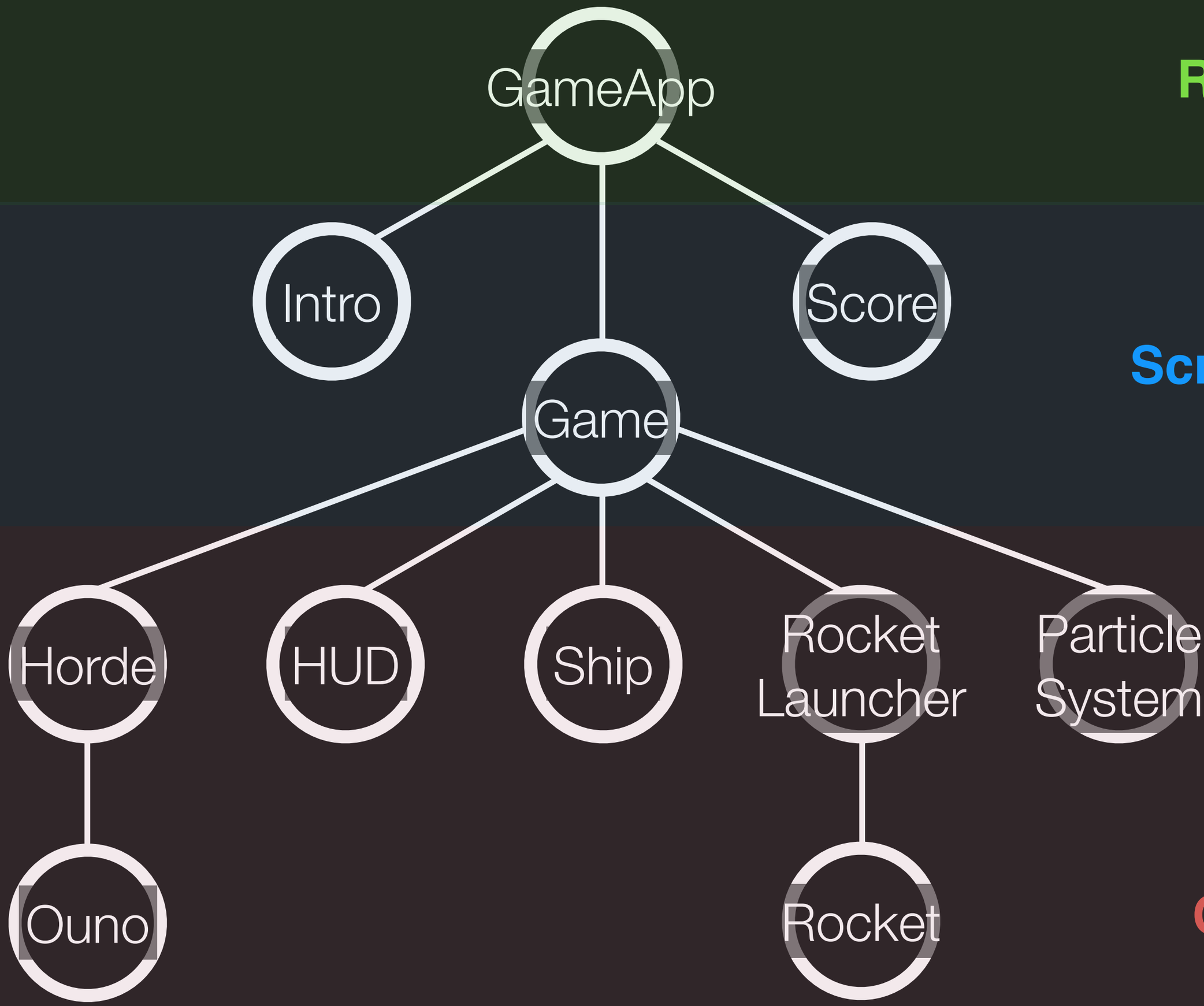

A
B
C
div
div
div


GameApp
Intro
Score
Game
Horde
HUD
Ship
Rocket Launcher
Particle System
Ouno
Rocket

Router


GameApp
Intro
Score
Game
Horde
HUD
Ship
Rocket Launcher
Particle System
Ouno
Rocket

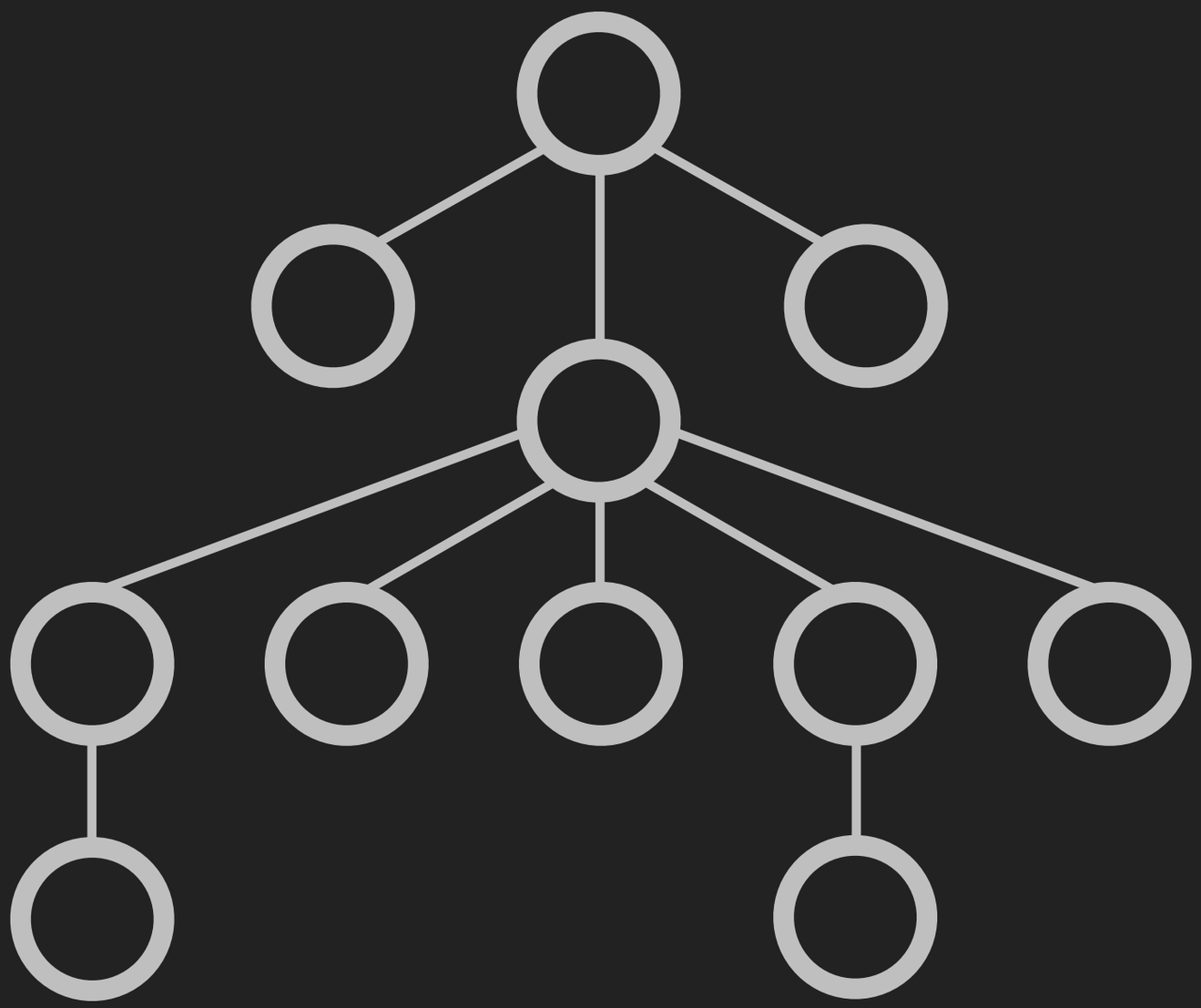

GameApp
Intro
Score
Game
Horde
HUD
Ship
Rocket Launcher
Particle System
Ouno
Rocket
Router
Screens


GameApp
Intro
Score
Game
Horde
HUD
Ship
Rocket Launcher
Particle System
Ouno
Rocket
Router
Screens
Other

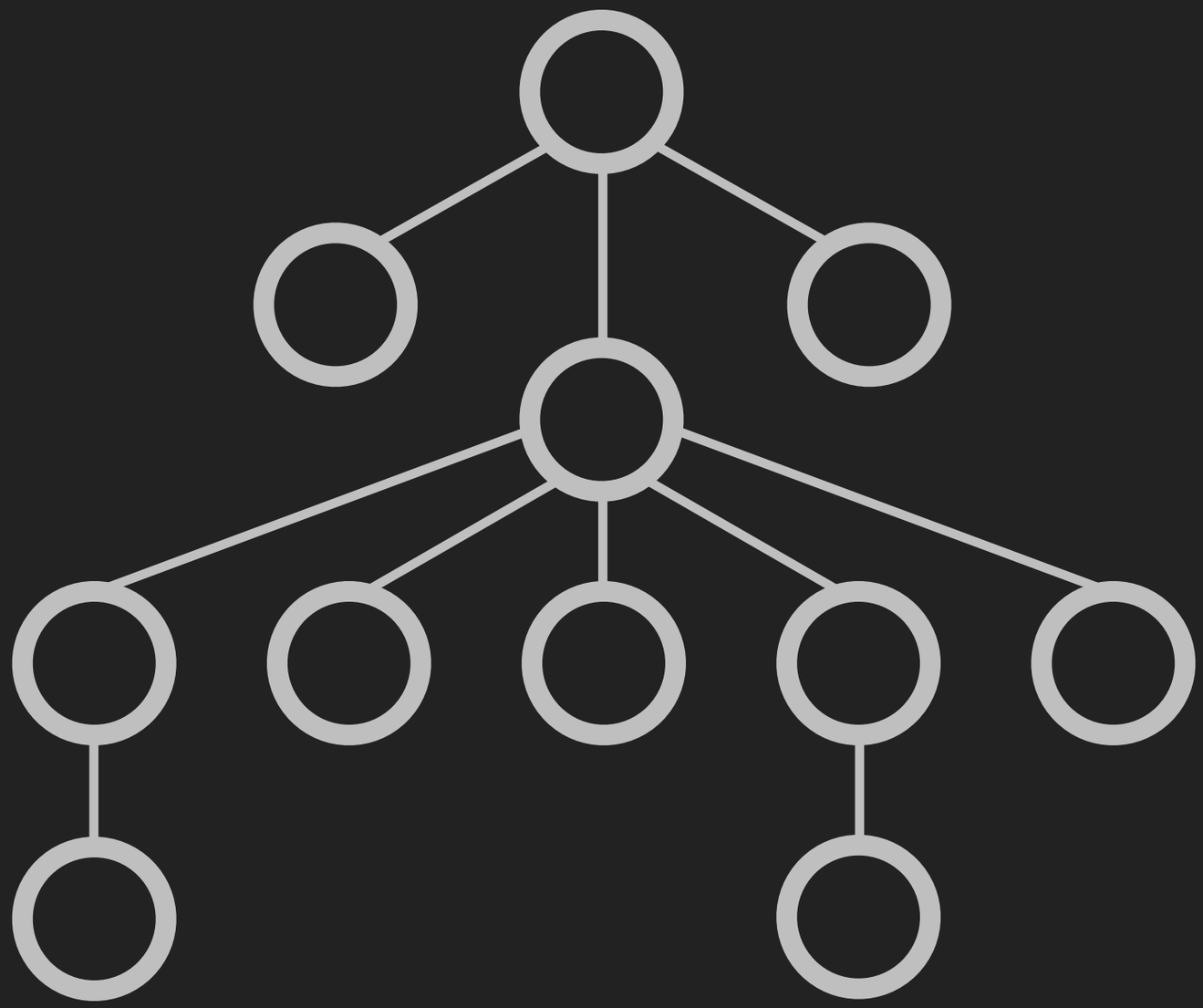
The world

Model

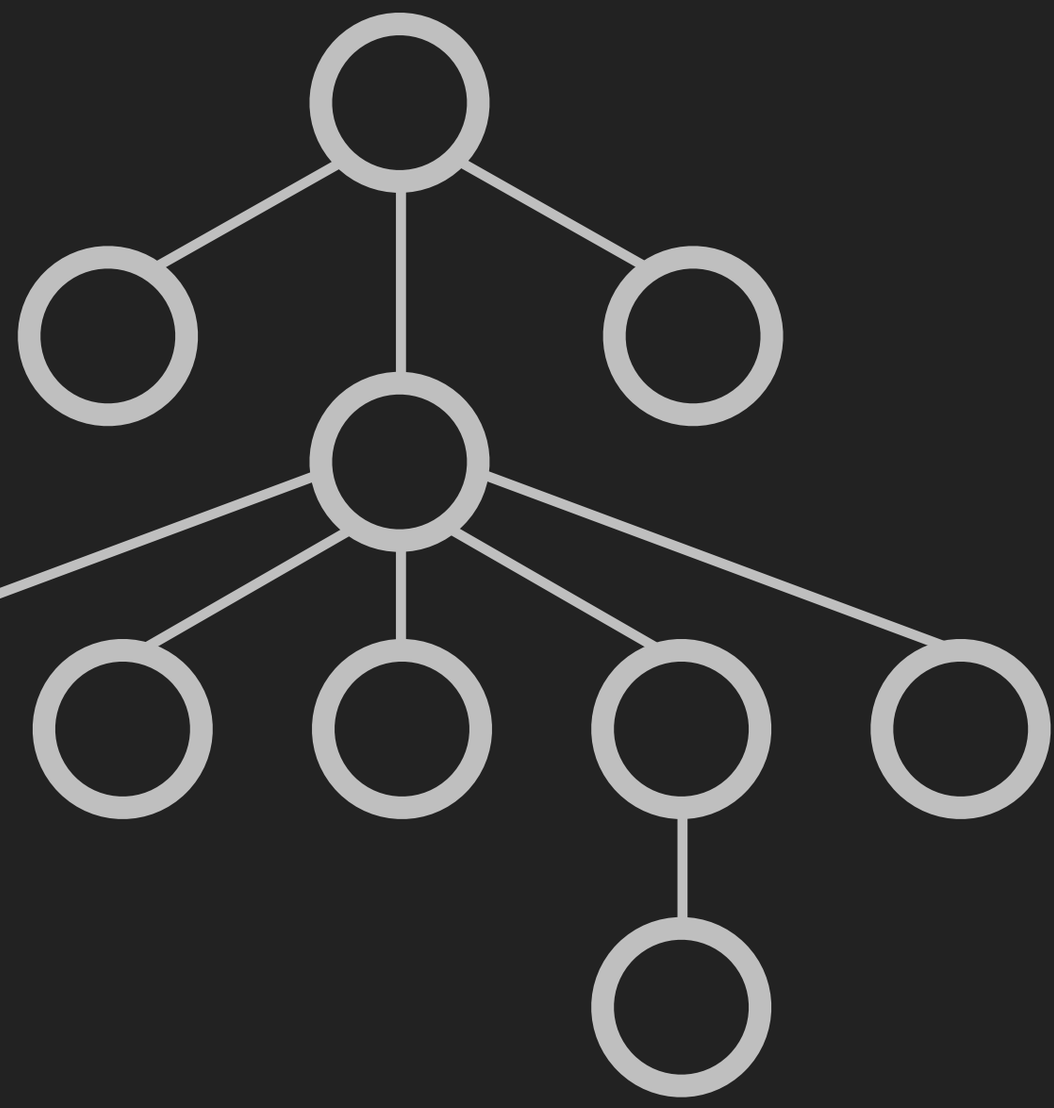
- One centralized model for the game
- Old school prototypal objects
- One function to update the world



ONE WAY

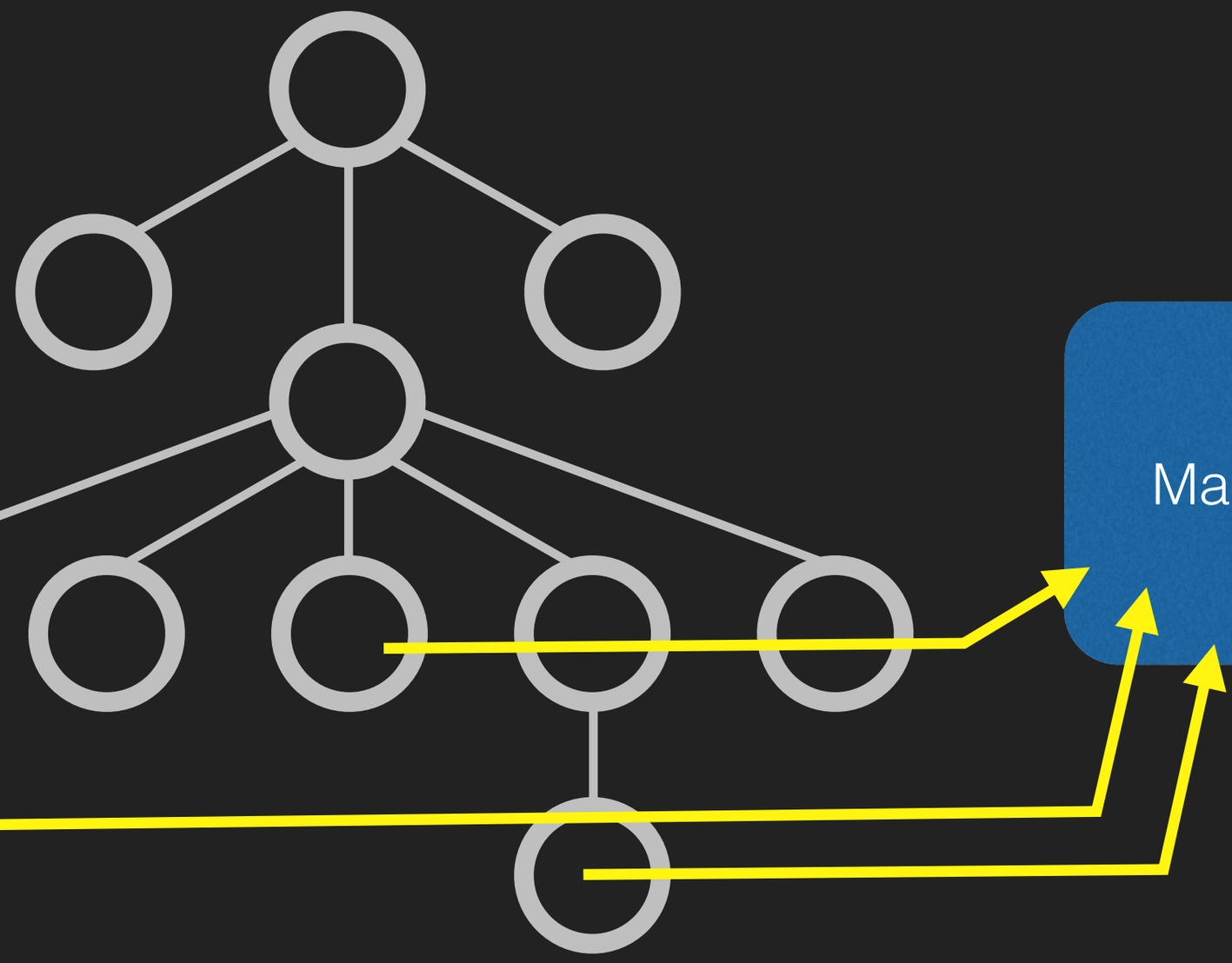


ONE WAY



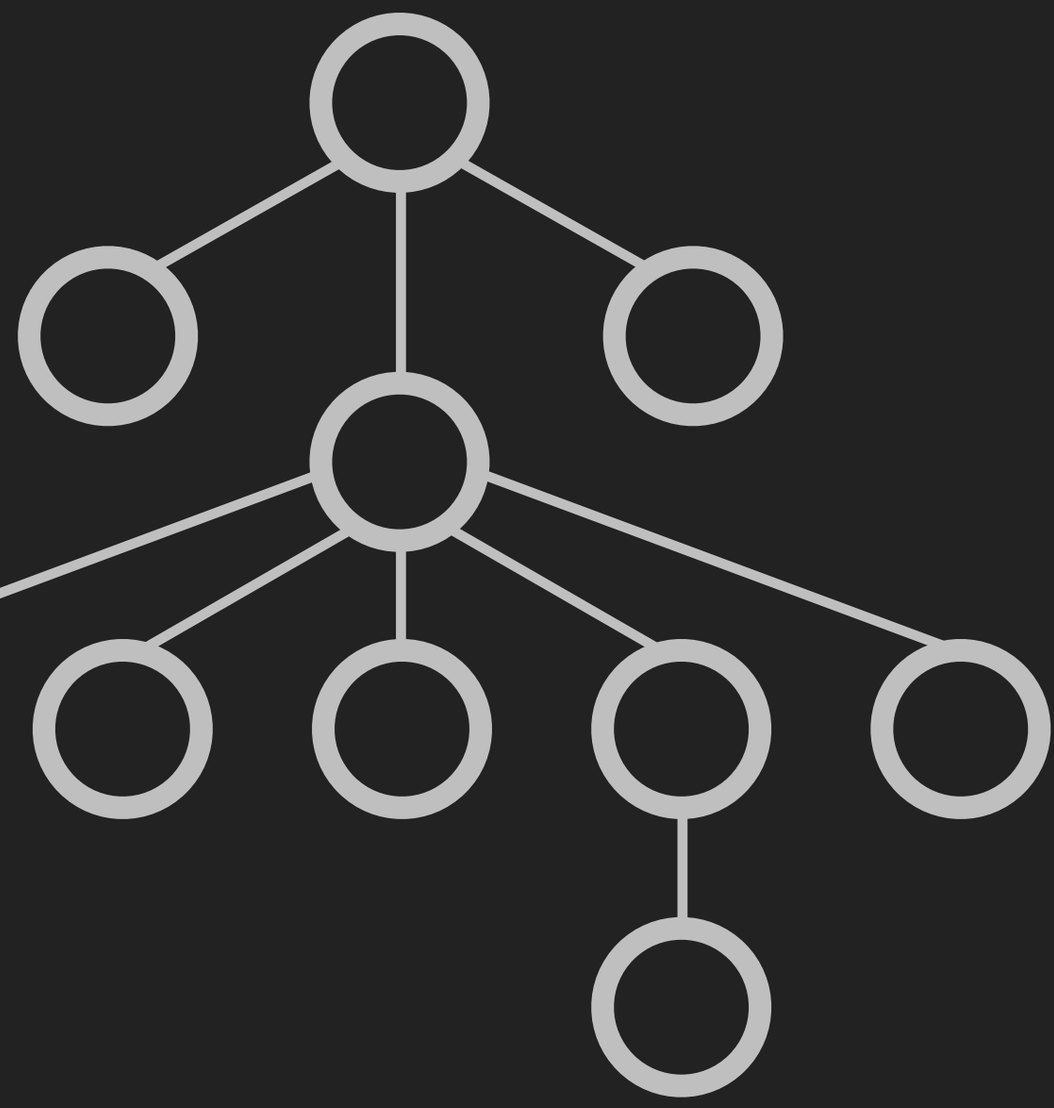
MailBox

ONE WAY



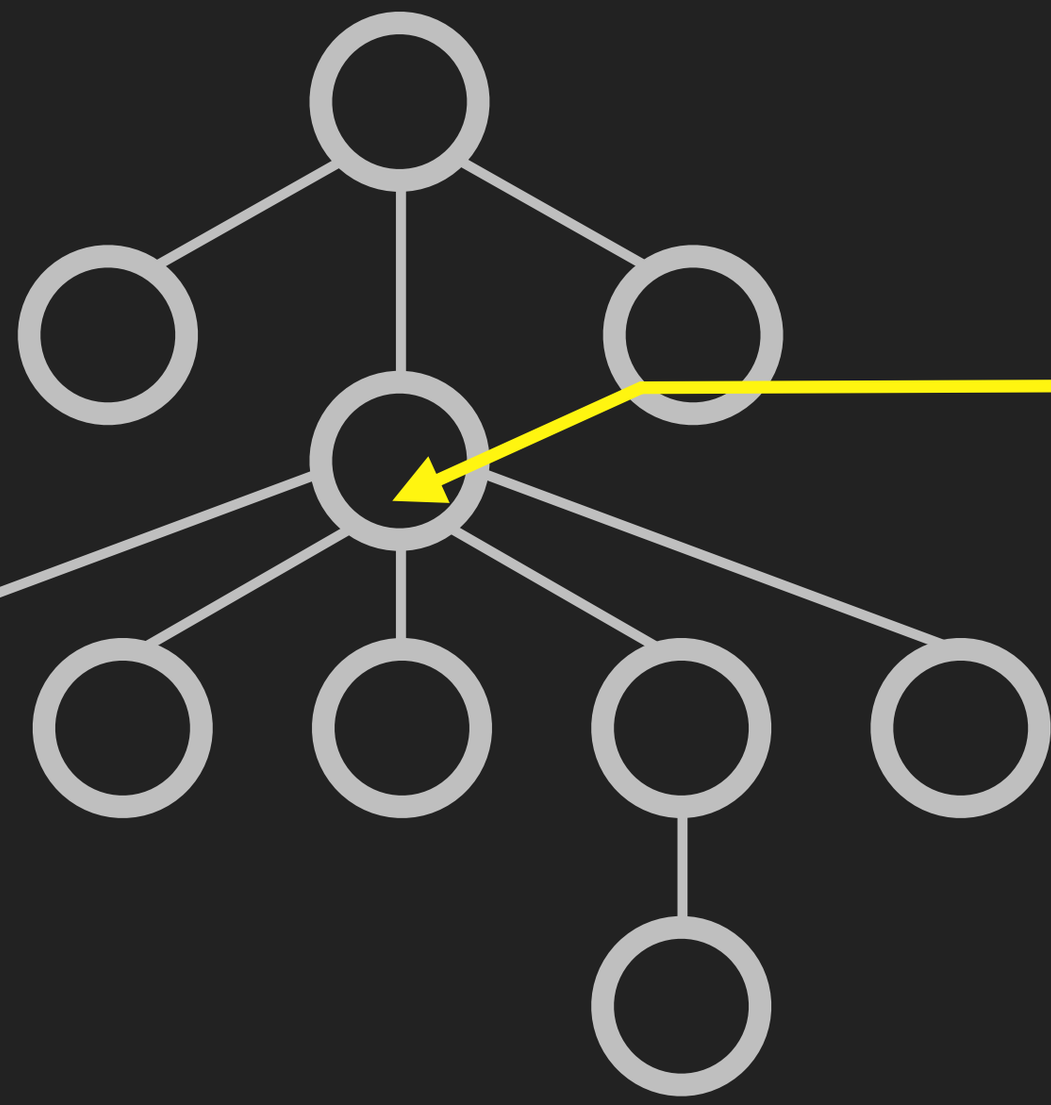
MailBox

ONE WAY



MailBox

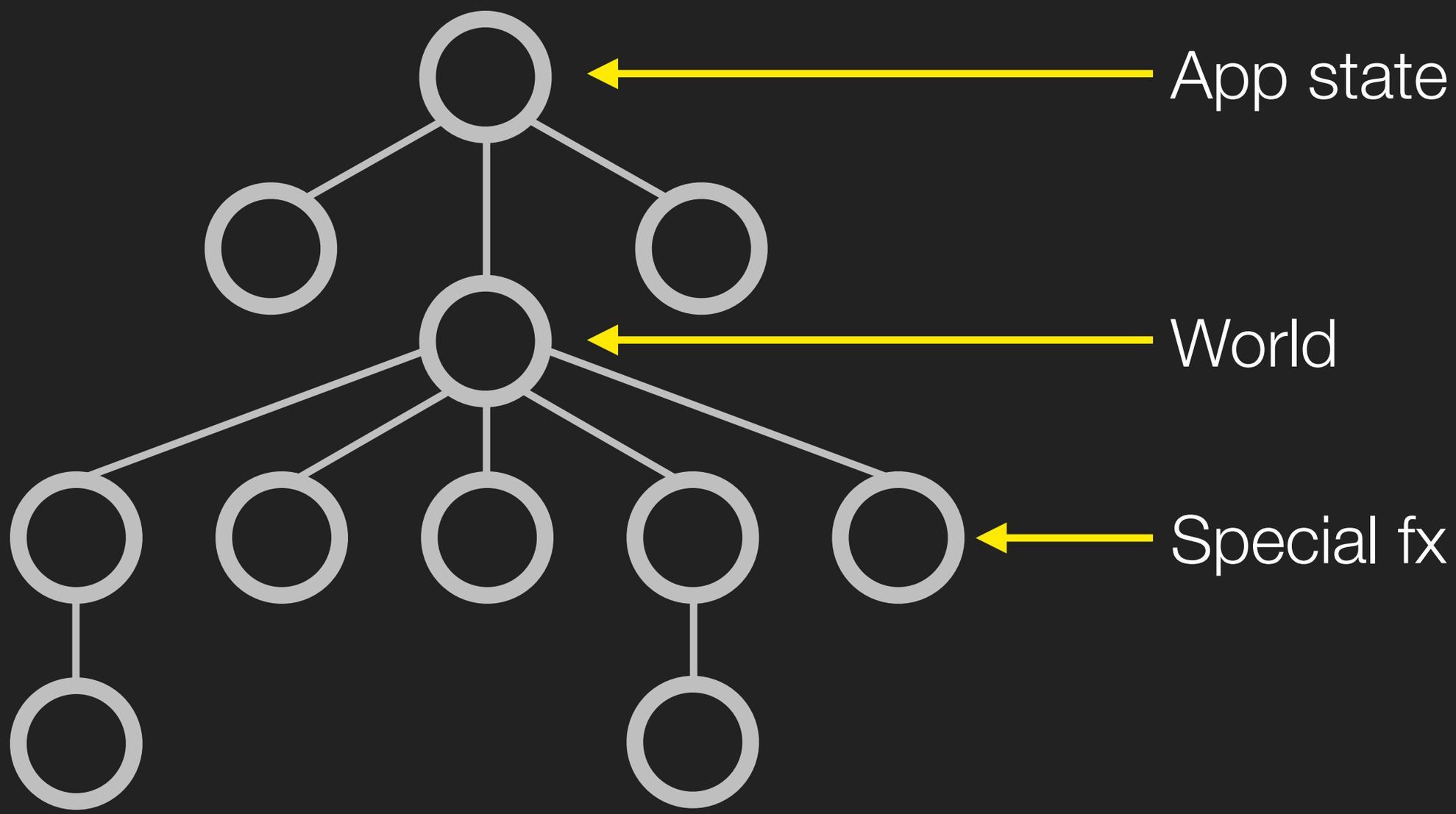
ONE WAY



MailBox

Communication

- Every frame there is an update
- Simple message box system
  - Every component can send events
  - Each tick, the model is updated with the events
  - Message box is emptied once read


App state
World
Special fx

Demo

Future

- Mixins
  - Game loop
  - Game screen
  - Sprites
- Ludum dare

Questions?

https://github.com/bobylito/littleshooter2/

References

- React : http://facebook.github.io/react/
- Other related frameworks based on React for doing similar stuff :
  - Pixi / react : https://github.com/lzzimach/react-pixi
  - React art : https://github.com/reactjs/react-art