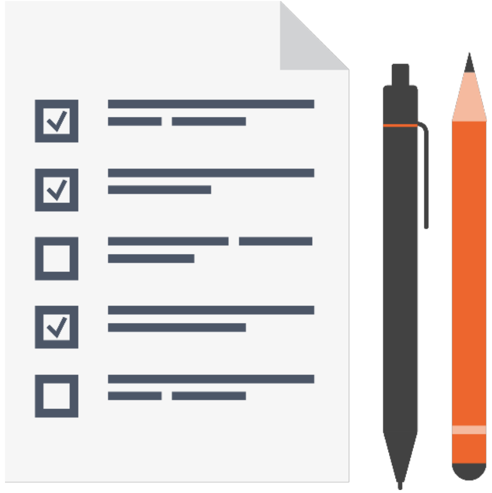
Completing the SPA Web User Interface with Knockout

Jason Roberts

@robertsjason | dontcodetired.com

Overview



Overview of knockout.js

Creating the JavaScript viewmodels

Allowing a player to join the game

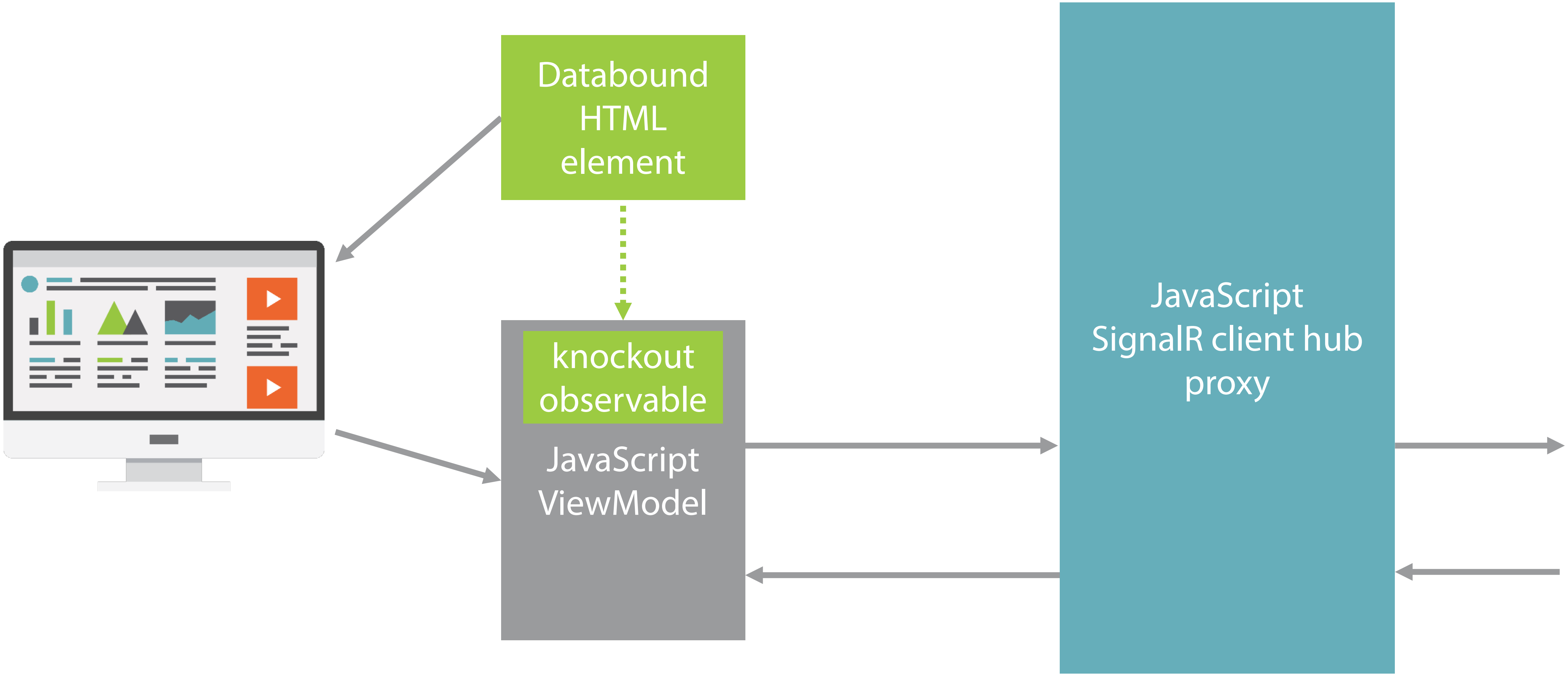
Reacting to other players joining

Implementing the attack behaviour

Loss of game state

Dynamic JavaScript user interfaces using the MVVM pattern and databound HTML elements

Overview of Knockout.js



Data Binding Quickstart

```
<span data-bind="text: thisPlayer().playerName">
<span data-bind="text: thisPlayer().health">
<div data-bind="visible: isJoined">
<h3 data-bind="visible: !thisPlayer().isAlive()">
<form data-bind="submit: joinGame">
<div data-bind="foreach: otherPlayers">
"visible: isAlive() && $parent.thisPlayer().isAlive()"
```

ViewModel Quickstart

```
var GameViewModel = function () {
    var self = this;
    self.newPlayerName = ko.observable('');
    self.isJoined = ko.observable(false);
    self.thisPlayer = ko.observable(new PlayerViewModel());
    self.otherPlayers = ko.observableArray();
    self.joinGame = function () {…}
}
```

“Essential Knockout and JavaScript Tips”

by John Papa

Installing Knockout.js

Install from NuGet

Update bundle config

Update layout to include bundle

Creating the Player JavaScript ViewModel

PlayerViewModel

playerName observable

isAlive observable

health observable

changeHealth function

attack function

Creating the Game ViewModel

GameViewModel

newPlayerName observable

isJoined observable

thisPlayer observable

otherPlayers observable array

joinGame function

var vm = new GameViewModel();

ko.applyBindings(vm);

Completing the SignalR Client Code

Completing the playerJoined function

Completing the updatePlayerHealth function

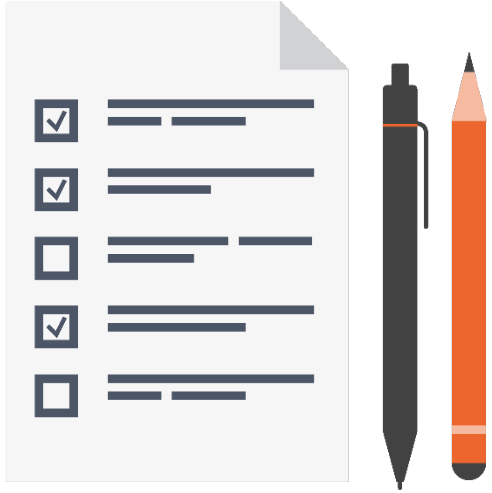
Creating Knockout Bindings

Fixing Observable References and Loss of Game State

```
return playerName === player.playerName;
```

Game state stored in IIS application

Summary



Overview of knockout.js

self.playerName = ko.observable(playerName);

<span data-bind="text: thisPlayer().playerName">

Installed knockout.js from NuGet

PlayerViewModel & GameViewModel

playerJoined() & updatePlayerHealth()

data-bind="foreach: otherPlayers“

Loss of game state

Next:

Hosting Game State in a Windows Service