***Component Technologies V1.3***

***Skel.js***

**Skel.js** is a miniature front end JavaScript framework deployed within this project that significantly **streamlines construction of responsive multi device layouts.** It is comprised of a CSS3 grid system of 12 columns with the ability to manage different types of intricate layouts by defining and dealing with breakpoints in the main skel.js function, thereby allowing straight forward CSS media query management.

It provides shortcuts to modify of the global model box and browser reset. It is driven by JavaScript, has no dependencies and functions seamlessly with other JavaScript utilities. It can be downloaded from <http://skeljs.org/>

Reference - <http://softstribe.com/webdesign/skel-js-framework-to-build-responsive-layouts>

***Angular.js***

Angular.js is an organizational framework designed for use with for dynamic web applications. It views declarative HTML static pages as a basic template that can be accentuated and extended allowing simplified component manipulation achieved in a less verbose manner through “directives” like data-binding, coding around DOM elements, and combining HTML elements to create components that can be reused. It is processed through the browser and work well with server based technologies.

Commonly the static-dynamic divide is dealt with by traditional libraries like jQuery and frameworks like Ember, however Angular differs in that it provides HTML constructs that alters the browsers perception of what it is processing. It provides the ability to encapsulate DOM and AJAX code in a precise well organized less verbose structure, when considering how data is dealt with, the cumbersome coding required when registering call backs, and low level manipulation of the DOM. Therefore it has a defined methodology for how to construct CRUD applications. It also provides comprehensive testing.

The level and intricacies of the DOM manipulation involved in Games are distinct from CRUD applications and therefore in this instance jQuery may be a better use case than Angularjs. The core tenet of Angular is that declarative code should be used to build User Interfaces and to wire components together, whilst imperative code is preferred for business logic. This promotes decoupling, which in turn promotes testability. It also allows a uniform approach, which works well in a team environment. Angular.js can be downloaded from <http://angularjs.org/>

Reference - <http://docs.angularjs.org/guide/introduction>