Porting the Naval Architecture Application from Microsoft XNA to WebGL/HTML5

Nicholas Wrobel  Tien Ho  Andrey Gavel  Blake Langland  Dr. Roger A. Dougal

Introduction

Naval Architecture provides a 3D design environment for the Ship Smart System Design (SSD) where multidisciplinary engineers are able to collaborate on the design and validation of ship structures. This application integrates the 3D visualization with an interactive graphical user interface to enhance comprehension and manipulation of CAD models. The SSD currently maintains the NavalArch tool using XNA in Microsoft Silverlight.

However, Microsoft has ceased further development of Silverlight since 2013 while Chrome and Firefox no longer support plugins for the framework. As a result, different technological alternatives were identified and experimented for the replacement of XNA in NavalArch tool. In light of the multitude and diversity of available 3D frameworks, selection benchmarks consist of:

1. Support for basic 3D features
2. Portability across most modern browsers and devices without plugins using WebGL.
3. A strong base of development and maintenance
4. Ease of interaction between the canvas and the GUI
5. The ability to load and render a great number of CAD models with reasonable frame rates
6. Dynamic mesh generation at runtime

Experiment

The three most popular 3D engines that satisfy the selection benchmarks are Three.js, Babylon.js, and Unity. A stress test was performed to assess the rendering capacity and performance of each engine. The objective was to load as many 24-polygon cubes as possible into the scene of each engine and observe the frame rate change until the web browser crashes due to memory depletion.

Comparison of Rendering Capacity and Performance between Three.js, Babylon.js, and Unity3D on Chrome

Three.js was chosen over Unity and Babylon.js because of its superior rendering capabilities. As an open-source framework, Three.js has a much larger and stronger online community than most other WebGL engines. Other frameworks used for the NavalArch tool include Kendo UI, Knockout.js, JQuery, backbone.js, and Bootstrap.

Result and Application

Three.js was chosen over Unity and Babylon.js because of its superior rendering capabilities. As an open-source framework, Three.js has a much larger and stronger online community than most other WebGL engines.

Other frameworks used for the NavalArch tool include Kendo UI, Knockout.js, JQuery, backbone.js, and Bootstrap.

Progress

- Add multiple cameras and lighting to the scene
- Pan, zoom, and orbit the camera with the mouse
- Display the scene in perspective and orthographic views
- Import CAD models into the scene from the equipment library
- Perform hit testing to select models, and change color on mouseover
- Display a bounding box and transformation widgets for a selected model
- Perform translation, rotation, and scaling on a selected model through its 3D widgets and the GUI
- Render decks and bulkheads to automatically fit the hullform contours
- Detect collisions between models
- Allow a model to expand and fill the area of a compartment between decks, bulkheads, and ship hulls
- Measure the distance between two models
- Determine the centroid of a model using its bounding box
- Allow a model to fall on a specified deck and move to a specified bulkhead without the use of a physics engine
- Communicate with the server to create, access, update, and delete model data
- Determine the volume of a model
- Create a pathfinding algorithm to avoid collisions while routing a cable

Conclusion

Almost all features have been ported over from XNA to Three.js / HTML5. However, there are some current challenges:

- Detecting collisions accurately with reasonable performance
- Maintaining acceptable frame rates with large ship designs
- Implementing intelligent cable routing/pathfinding
- Maintaining cross-browser compatibility with a complex project

Future aspirations for the NavalArch application include:

- Use of holographic technologies such as Microsoft Hololens or Oculus Rift to enrich the user experience
- Use of virtual reality gloves to improve the user interaction when building ships

References