Montana Tech

Science Mine

Digital Signage Project

Finite State Machine Description

Author:
Phillip J. Curtiss

Principal Investigator:
Fred Hartline

January 17, 2017

1 Introduction

The Science Mine Digital Signage project provides a self-guided signage system for museum exhibits whereby visitors may interact with a variety of input devices to allow for the exploration of the content within the signage system, in a similar way that a visitor to the exhibit is encouraged to learn by exploring through interacting with the exhibit. The variety of different input devices allows visitors of different ages and different knowledge levels to interact with a single signage platform that adapts to provide information that is relevant to what is of interest to the visitor.

To accomplish this task, the signage system is comprised of the following components:

Input Devices: The current set of input devices consists of a bluetooth trackpad and a bluetooth array of buttons driven by an Arduino-based microcontroller. In the future, an additional touch screen is likely to be added. In general, however, any input device that is bluetooth HID compliant should be able to be integrated into the system without much effort.

Compute Platform: The compute platform selected is an extremely small form-factor full-function Linux-based computer made by Intel known as the Stick. In particular, we are using model STK2MV64CC which is quite a capable compute device in a small form-factor that can be connected to a high-definition projector and out of the way of visitors to the exhibit.

Projector/Display: The initial deployment will be a projector to provide a large enough display surface to show high-quality video content, and also provide high-quality thumbnails of next-level video content that may be selected by the visitor. In the future, additional touchscreen monitors might be used, or added, to provide a different presentation format and means of interaction.

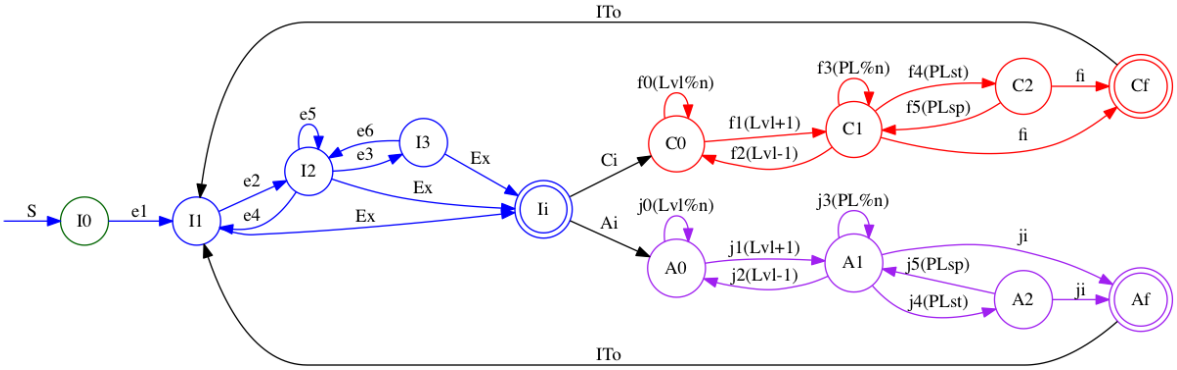
Video Content: The video content is high-quality custom produced content highlighting the use of the exhibit associated with the signage. The video content collection is organized in a graph structure that matches a given organizational structure (learning paths) through the video content collection. In this way, a single video may be included in different organizational unit structures represented within the graph - i.e. be part of multiple learning paths through the collection.

Signage Software: Custom written software that consists of the following components:

- **Single Page Web Application:** Based on the Ember application framework, the single page web application is resident in the client browser running on the Intel Compute Stick. The web application is responsible for implementing the Finite State Machine (FSM) described within this document by mapping different bluetooth HID input (from a variety of devices) to transitions within the FSM. The web application will also enter a default *idle behavior* when the is no interaction from a visitor.
- **Exhibit Object Model (EOM):** The Exhibit Object Model (EOM) is a JSON formatted description of the signage content for a given specific exhibit. This EOM is used by the single web page application to *drive* the signage. The EOM contains references to the entire video content collection and provides the organizational structure of this content. In addition, the meta-information for each video within the collection is stored within the EOM to implement textual overlays to provide descriptive text associated with the video, attribution for the video, and the like.
- **HTML:** The web application is a simple web page written using HTML5. The base-page merely has placeholders for the main background video content, a menu of all the video content for the given level, a region that overlays the background video, and zero or more regions in which thumbnail video content will play, each one representing a topic at the current level that may be explored.
- **CSS:** The cascading style sheet (CSS) is used to provide layout functionality and look and feel behaviors for the html content elements created by the web application.
- **JS:** JavaScript content is used to script and provide functional behavior of the web application.

All of these components remain the same an unchanged no matter for which exhibit they are providing signage, with the exception of the EOM. There is a one-to-one relationship between a given exhibit that has been paired with a digital signage system, and its corresponding EOM that fully describes the signage with be used with the exhibit.

Figure 1: Science Mine Digital Signage Finite State Machine



Finite State Machine (FSM) for Science Mine Digital Signage
I-Nodes Idle Cluster; C-Nodes Children Cluster; A-Nodes Adult Cluster

2 Finite State Machine

The finite state machine (FSM) for the digital signage system consists of three (3) subgraphs, I, C, A, where I describes the states and transitions associated with the system being in an idle state, C describes the states and transitions associated with the system being in a mode of providing signage-content for children, and A describes the states and transitions associated with the system being in a mode of providing signage-content for adults. All of these subgraphs are related into a single FSM in Figure **??** on page **??**.

More formally:

Let I be the graph such that, $I =< I_s, I_v >$ $\quad |I_s = \{I_n \text{ for } n \in [0..3]\} \cup \{I_i\}$
$\quad |I_v = \{e_m \text{ for } m \in [1..6]\} \cup \{E_x\}$

Let C be the graph such that, $C =< C_s, C_v >$ $|C_s = \{C_n \text{ for } n \in [0..2]\} \cup \{C_f\}$
$\quad |C_v = \{f_m \text{ for } m \in [0..5]\} \cup \{f_i\}$

Let A by the graph such that, $A =< A_s, A_v >$ $|A_s = \{A_n \text{ for } n \in [0..2]\} \cup \{A_f\}$
$\quad |A_V = \{j_m \text{ for } m \in [0..5]\} \cup \{j_i\}$

Let I_0 be the start state

Let C_0, A_0 be the respective start states for the subgraphs defined by C and A respectively.

2.1 System Startup Behavior $\delta(I_0, e_1) \mapsto I_1$

2.2 Idle Mode Startup Behavior $\delta(I_1, e_2) \mapsto I_2$