Technology Review

November 10 2016

Abstract

The Forge VR Explorer branches from an Autodesk prototype project called Vrok-It, which is a simple web-based 3D model viewer and mobile virtual reality (VR) explorer. The project will expand upon its ability to display uploaded 3D models in browser and in VR, and improve its accessibility. Conventionally, viewing 3D models in VR is a challenge if you have model files on many devices, or have a headset that only works in conjunction with a smart-phone. The Forge VR Explorer aims to do this by utilizing a web-based software that uses the features of the Autodesk Forge API. The project will also be expanded with new ideas and stretch goals as the project is developed.

TABLE OF CONTENTS

1 Introduction

2 Technologies

1) File Uploading

   a) Options

      - Use a plug-in: Using a plug-in that would handle file uploading would be very convenient as it takes a lot of the work out of it. One of the biggest benefits would be that this could save us a lot of time. If we were to use the plug-in we would not have to worry about the effort that would normally go into creating our own, we could simply just find one that would fit what we needed and the put it into our website. However this does also lead to the problem that we may not be able to find one that exactly fits what we want. Many of the plug-ins I have found so far simply upload the file to a server and then it is done. In our case we need the file to be usable by the website once it is uploaded.
      - Create our own file upload feature: Creating our own uploading feature would allow us to make the feature work exactly the way that we would want to. Their a many different ways to do this though. One common way I have found is to first create a web form using HTML to make the upload button.Then you could use PHP to submit the file through a POST. Using AJAX and JQuery you could also add features that show the progress of the upload and verification such as verifying correct file types are being used. This could take a lot more time and effort to complete though depending on how skilled the person creating the upload feature is and how fancy they want the feature to be.
      - Create our own file uploading feature using new File API added to HTML5: This is similar to the previous option but seems like it might be easier to learn and understand. Many of the features that we wanted are built into the API making it easier for us to create the upload file file feature the way we want it. Though there will be some time for learning how to use the API and some time to completely create the feature, there is a lot of good documentation on this API that should make the process go smoothly.

   b) Criteria being evaluated

      i) How much time will it take compared to how important it is to the overall project.
      ii) How much effort will have to go into learning how to complete the task.
      iii) Even if one options isn't exactly what you want does the time benefit outweigh making it exactly the way you wanted.
      iv) Which option will offer the most secure way to upload the file.

c) Discussion: The plug-in option will definitely save the most time in terms of learning it and getting it to work in on the website, however it will likely give you little in the way of customizing and making the file upload feature work exactly how you would want it to. Both of the options to make you own file upload feature will result in a longer time to learn how to do this as well as a longer time to complete the task. This does allow you to make sure the the file upload will do exactly what you are wanting it to do. the third option will is also using HTML5 this means that is will work in all the new browsers.

d) Select Option: I believe that the extra time that it takes to create our own custom file upload feature will outweigh the extra amount of time that it will take to do this. I also think that the use of the file API is better since it is well documented, easy to find examples for, and has many of the features that we wanted built into it.

2) File conversion

a) Option

- accusoft API: This is an API that allows users to convert their CAD files in to raster files such as SVG (scalable vector graphic) files that could to view in the model viewer. This API does cost money and depending on how many conversions the user is planning on doing this option could turn out to be relatively expensive. There is a lot of documentation on how this API can be integrated into an application but it seems as though the integration process could be harder if you are not using a certain types of applications.
- Forge Model derivative API: This API has the ability to take users CAD source file and convert them into OBJ and STL files. This API can also convert the files directly into SVF files the same type of file that the model viewer API we will be using on this project uses. This API would also not cost us anything to use while we are working on the project since the company that created the API is the same company we will be making the project for. There might be a little bit of a learning curve when it comes to integrating it into the website but the API has a lot of tutorials and documentation on how to use the API in an application. This API also uses a token based authentication system that provides security when converting the users files.
- cloudconvert API: This API gives you the ability to convert from CAD source files to a SVG file but it also give you the ability to convert the other way was well. This API also has well documented instruction on how to integrate this API into an application. That would make it easier to learn and take less time to complete the task. This API would cost money to use depending on how much time is spent doing the conversions. this one also requires that a

key be used when doing the to be created which make this more secure to do file conversions on since it will be able to verify that it is an actual users trying to do conversions.

b) Criteria being evaluated

   i) How much time will it take compared to how important it is to the overall project.
   ii) How much effort will have to go into learning how to complete the task.
   iii) How secure is the file conversion.

c) Table?

d) Discussion: These three options all offer similar conversions. They all convert the CAD source files into a usable file for the model viewer. The Forge API will however convert the file into an SVF file that can then be directly loaded into the viewer. This could save time and make it easier to complete the task. The Forge API is also free for us to use during the duration of the project so that would also make a big difference as well. Finally the Forge API offers the most secure option when it comes to converting the users files since it requires a token authentication to even do the conversion. This token is created as soon as the user visits the website and last for the duration that the user is on the website. Also with lots of documentation as well as tutorials on how to integrate the API into our application I feel like this option will take the least amount of time to complete the task.

e) Select Option: I Think that we should use the Forge model derivative API since it seems the most straight forward on how to integrate it into our application. It won't cost us any money to use and has a more secure way of converting the users files.

3) Connecting the users device

   a) Options:

      - Use a QR code plug-in: jquery.qrcode.js is a jquery plug-in that generates a QR code which can be scanned by a user to using a QR scanning application on their device to connect their device to the current web session. Using this would be relatively easy since we would just need to add the library to our project. That means that there wouldn't be a lot of learning time involved in using this option.
      - QRickit API: This API can dynamically generate QR codes to be used in a web/mobile application. The API is free to use for personal and small business applications so there would be no cost for us to use it with our project. There is also documentation on how to setup and use the API so this would cut down on the time that it take to learn how to integrate it into our project.

- QR Code Generator: This is another API that will that will generate a QR code for the us. It can easily be embedded into the web page by putting the source into an image tag. This API also allow for you to specify how big you want it and allows for different backgrounds to be used. The API also Has all the documentation that is needed to set up and use so the time to integrate this API into the project would be much.

b) Criteria being evaluated

  i) How much time will it take to integrate into the project: I think that all of the options can pretty easily be integrated into the project. all have documentation that clearly describes how set-up and use them. Overall I would say that each of them are about the same when it comes to the amount of time that it will take to integrate them into our project.

  ii) Will there be a lot of learning involved in using the option: I also feel like that since each of these options has documentation describing how to use each that the learning involved in each is pretty minimal.

  iii) Will it connect the users device quickly: each of the should have about the same time to connect the device. The only reason that there would be a slow down in connecting the users device is if they had a slow Internet connection or if there was a lot of traffic on the website itself.

c) Table?

d) Discussion: Overall I think that all the options are pretty close to one another. All three of the options are pretty straight forward in terms of integrating them into the project. All three should also connect the users device quickly. The two API options add the ability to customize the QR code, but that really isn't needed for our project since we just need the QR code to be there so the user can connect their device.

e) Select Option: I would say that the plug-in is the best way to go since we can integrate the library directly into our project and not have to worry about making any calls to an outside API.

4) Technology 4

  a) Option
  b) Goals
  c) Criteria being evaluated
  d) Table?
  e) Discussion
  f) Select Option

5) Technology 5

    a) Option
    b) Goals
    c) Criteria being evaluated
    d) Table?
    e) Discussion
    f) Select Option

6) Technology 6

    a) Option
    b) Goals
    c) Criteria being evaluated
    d) Table?
    e) Discussion
    f) Select Option

7) Technology 7

    a) Option
    b) Goals
    c) Criteria being evaluated
    d) Table?
    e) Discussion
    f) Select Option

8) Technology 8

    a) Option
    b) Goals
    c) Criteria being evaluated
    d) Table?
    e) Discussion
    f) Select Option

9) Technology 9

    a) Option
    b) Goals
    c) Criteria being evaluated
    d) Table?

e) Discussion

f) Select Option

3 CONCLUSION

4 BIBLIOGRAPHY