EGR 401 – Capstone Design

Deliverable 4: Concept Generation and Selection

Team Name: Blueprints

Team Members: Jordan Ziegler

  Jeremy Fischer

  Justus Karenzi

  Sudi Nsengiyumva

  Dylan Shanahan

Client’s Name: Dr. Mark Gordon

Advisor’s Name: Professor Thomas Renck

Version: 3

Date: April 9, 2015

**4.1 Concept Generation**

|  |  |  |  |
| --- | --- | --- | --- |
| Features | Option 1 | Option 2 | Option 3 |
| Operating System | Android | iOS | - |
| SDK | Vuforia | Metaio | D’Fusion |
| Methodology |  |  |  |
| Image Processing | Scanning | Embedded | - |
| Identification | QR Code | Text | jpeg Images |
| Mapping | Cloud-based | Included | - |

Table 4.1.1 Option Identification

Table 4.1.1 is a table which identifies each option available to the developing team. The features we intend to include are things like operating system and software development kit (SDK). Image processing systems, identification systems, and mapping systems are methods we are using in order to accomplish the task. The different options are described in greater detail in the subsequent sections under the 4.1.X headings.

**4.1.1 Operating Systems**

**Android**

Pros

* Most software development kits are free to use and open source.

Cons

* Members of the group do not own Android phones to run/test the software.

**iOS**

Pros

* Members of the group have mobile Apple devices to run/test the software.

Cons

* Most of the software developments kits are expensive and do not run on a PC.
* Members of the group do not have Mac computers to use the SDK.

**4.1.2 SDKs**

**Vuforia**

|  |  |
| --- | --- |
| Type | Free + Commercial SDK option |
| Platforms |  |
| iOS | **Yes** |
| Android | **Yes** |
| Windows Mobile |  |
| Web | **No** |
| PC/Mac/Linux |  |
| Features |  |
| 3D Object Tracking | **Yes** |
| Natural Feature | **Yes** |
| GPS | **No** |
| IMU Sensors | **Yes** |
| Marker | **Yes** |
| VisualSearch | **Yes** |
| FaceTracking | **No** |
| ContentAPI | **Yes** With Vuforia Cloud |
| Plugin Compatibility |  |
| Unity (3D) | **Yes** |
| Website | [qualcomm.com/...](http://developer.qualcomm.com/dev/augmented-reality) |

**Metaio**

|  |  |
| --- | --- |
| **Type** | **Free + Commercial SDK option** |
| **Platforms** |  |
| **iOS** | **Yes** |
| **Android** | **Yes** |
| **Windows Mobile** |  |
| **Web** | **Yes** |
| **PC/Mac/Linux** | **PC/Mac** |
| **Features** |  |
| **3D Object Tracking** | **Yes** |
| **NaturalFeature** | **Yes** |
| **GPS** | **Yes** |
| **IMU Sensors** | **Yes** |
| **Marker** | **Yes** |
| **VisualSearch** | **Yes Client-based +100 unique objects, cloud-based continuous visual search engine** |
| **FaceTracking** | **Yes** |
| **ContentAPI** | **Yes OpenGL support, in-house 3-D renderer** |
| **Plugin Compatibility** |  |
| **Unity (3D)** | **Yes** |
| **Website** | [**metaio.com**](http://www.metaio.com/) |

**D’Fusion**

|  |  |
| --- | --- |
| Type | Free + Commercial SDK option |
| Platforms |  |
| iOS | **Yes** |
| Android | **Yes** |
| Windows Mobile |  |
| Web | **Yes** Flash |
| PC/Mac/Linux |  |
| Features |  |
| 3D Object Tracking |  |
| NaturalFeature | **Yes** |
| GPS | **Yes** |
| IMU Sensors | **Yes** |
| Marker | **Yes** |
| VisualSearch | **Yes** |
| FaceTracking | **Yes** |
| ContentAPI |  |
| Plugin Compatibility |  |
| Unity (3D) | **Yes** |
| Website | [t-immersion.com](http://www.t-immersion.com/) |

[15] \*Reference Page

**4.1.3 Image Processing**

**Scanning (QR codes)**

Pros

* Very quick transfer of information. Since the advent high speed internet, people have got very used to getting information very quickly.

Cons

* Scanning can be a long process: The actual process of scanning a code can be a pain. Users have to get their phone out, fire up the code reader, before scanning and waiting for the landing page.
* Security issues. The transfer of data between two devices can always lead to security issues. Also, before scanning a code, the scanner can never really know where the code is going to lead them.
* Scanning would let a user insert a scan into a database for use later.

**Embedded**

Pros

* Developers are able to include only what is needed for the project.
* Users have no control over the database files.

Cons

* Embedded systems can take up memory due to the inclusion of unused variables for a specific user.

**4.1.4 Identification**

**QR Code**

Pros

* Ease of use: QR codes can be added to just about anything, from cereal packets to adverts on the Underground, and this versatility can be very useful
* Range of uses: There are hundreds of potential uses of QR codes. They can be used to extend the user experience in schools, restaurants, museums and more.
* QR codes can be cost effective: Creating the QR code itself doesn't have to cost anything

Cons

* Users need to download a QR code reader: This is the big drawback for many. Mobile users have to download a (normally free) QR reader app before they can even begin to use them, which limits the audience.
* Scanning can be a long process: The actual process of scanning a code can be a pain. Users have to get their phone out, fire up the code reader, before scanning and waiting for the landing page.

**Text**

Pros

* Has the ability to read different types of text-based signs at various buildings/universities.
* Will take up less memory than recognizing picture files.

Cons

* The text is different in every building and there is no universal font used for door signs.

**jpeg Images**

Pros

* Colors Supported: Supports millions of colors (24-bit).
* Browser Support: Fully supported for use as inline images in version 2.0 and higher of Netscape Navigator and Internet Explorer, as well as in most other current browsers.

Cons

* Compression: Some image data is discarded when it is compressed, reducing the quality of the final file
* Browser Support: Fully supported for use as inline images in version 2.0 and higher of Netscape Navigator and Internet Explorer, as well as in most other current browsers.

**4.1.5 Mapping**

**Cloud-based**

Pros

* Takes up less memory space on the users phone.
* User can quickly switch between the maps of different buildings.

Cons

* Must have an internet connection in order to use the app.
* A server must be maintained off-site, which will cost money.

**Included**

Pros

* Can use the app offline.
* All of the building information is stored on the users phone.

Cons

* Has the possibility of using a large amount of memory space on the users phone.
* User must individually download the plans for each building.

**4.1.6 Concept Generation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Selection |  |  |  |  |  |
|  | OS | SDK | Image Processing | Identification | Mapping |
| Concept A | Android | Vuforia | Scanning | QR | Included |
| Concept B | OS | Metaio | Embedded | Text | Cloud-based |
| Concept C | Android | D’Fusion | Embedded | JPeg | Cloud-based |
| Concept D | OS | Vuforia | Scanning | Text | Included |

Table 4.1.2 Concept determination

Table 4.1.1 determines what options are associated with each concept. The table is creating the concepts we compare in the subsequent sections in order to determine the most efficient and effective method of creation. Each concept is designed by choosing arbitrarily different options from the above table 4.1.1, which shows each option available.

**4.2 Concept Selection**

**4.2.2 Operating System**

|  |  |  |
| --- | --- | --- |
| OS Selection |  |  |
|  | iOS | Android |
| Criteria |  |  |
| Functionality (.3) | 0 | 0 |
| Usability (.3) | 0 | 0 |
| Complexity (.2) | -1.0 | 1.0 |
| Cost (.1) | -1.0 | 0 |
| Marketability (.1) | 0 | 0 |
| Sum | -2.0 | 1.0 |

Table 4.2.2

Table 4.2.2 compares the operating systems based on the criteria the team decided on. According to the table Android is more effective for the project due to the complexity and cost associated with iOS. Android also gains an advantage in complexity due to it being written in Java. The team has experience in C++ which is similar to Java so the transition will be a smaller learning curve than transitioning to iOS which uses Objective C.

**4.2.3 Software Development Kit (SDK)**

|  |  |  |  |
| --- | --- | --- | --- |
| SDK Selection |  |  |  |
|  | Vuforia | Metaio | D’Fusion |
| Criteria |  |  |  |
| Functionality (.3) | 1.0 | 0 | 0 |
| Usability (.3) | 0 | 0 | -1.0 |
| Complexity (.2) | 0 | 0 | -1.0 |
| Cost (.1) | 0 | 0 | 0 |
| Marketability (.1) | 0 | 0 | 0 |
| Sum | 1.0 | 0.0 | -2.0 |

Table 4.2.3

Table 4.2.3 determines the Software Development Kit (SDK) that will be used by the team. According to the Table Vuforia is the best choice based on functionality whereas D’Fusion is the worst choice based on usability and complexity. Metaio is a neutral SDK for the Team.

**4.2.4 Image Processing**

|  |  |  |
| --- | --- | --- |
| Image Processing Selection |  |  |
|  | Scanning | Embedded |
| Criteria |  |  |
| Functionality (.3) | 0 | 1.0 |
| Usability (.3) | -1.0 | 0 |
| Complexity (.2) | 0 | 0 |
| Cost (.1) | 0 | 0 |
| Marketability (.1) | 0 | 0 |
| Sum | -1.0 | 1.0 |

Table 4.2.4 Image processing

The table above (Table 4.2.4) shows the different styles of addressing image processing. The styles are ranked according to the objectives desired from our objectives tree. The table shows us that an embedded approach to image processing is more desirable than a scanning system. The embedded approach includes a greater functionality and usability. Embedded image processing is has more functionality and usability than the other system because it does not require as much activity from the user.

**4.2.5 Identification**

|  |  |  |  |
| --- | --- | --- | --- |
| Identification Selection |  |  |  |
|  | QR | Text | JPeg |
| Criteria |  |  |  |
| Functionality (.3) | 0 | 0 | 0 |
| Usability (.3) | 0 | 1.0 | -1.0 |
| Complexity (.2) | 0 | 0 | -1.0 |
| Cost (.1) | 0 | 0 | 0 |
| Marketability (.1) | -1.0 | 1.0 | 0 |
| Sum | -1.0 | 2.0 | -2.0 |

Table 4.2.5

Table 4.2.5 determines the best image identification to use. The SDK’s are able to recognize and process QR codes, Text, and JPeg so these were the options limited to the image identification. The QR code is lacking in marketability due to the fact that a QR code must be generated, printed off, and placed so the user may be able to scan it. The JPeg is at a disadvantage when it comes to usability and complexity. JPeg’s require more memory for storing the image and processing to recognize what the image is; this can result in longer processing time and more memory being taken up by the application. Text is the clear winner due to its usability and marketability. It requires less space and is easier to implement.

**4.2.6 Mapping**

|  |  |  |
| --- | --- | --- |
| Mapping Selection |  |  |
|  | Cloud-based | Included |
| Criteria |  |  |
| Functionality (.3) | 0 | 0 |
| Usability (.3) | 0 | 0 |
| Complexity (.2) | -1.0 | 1.0 |
| Cost (.1) | -1.0 | 0 |
| Marketability (.1) | 0 | 0 |
| Sum | -2.0 | 1.0 |

Table 4.2.6 Mapping Selection

Table 4.2.6 shows the value of selecting an included mapping selection as opposed to a cloud based system. The cloud system is more complex and costly than the included system. The included system is more efficient from a coding perspective and also is more cost effective. The cloud based system requires an interface with a cloud, and also requires internet access.

**4.2.7 Concept Collaboration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Concept A | Concept B | Concept C | Concept D |
| Criteria |  |  |  |  |
| Functionality (.3) | 1 | 1 | 1 | 1 |
| Usability (.3) | -1 | 1 | -2 | 0 |
| Complexity (.2) | 2 | -2 | -2 | 0 |
| Cost (.1) | 0 | -2 | -1 | -1 |
| Marketability (.1) | -1 | 1 | 0 | 1 |
| Sum | 1 | -1 | -4 | 1 |

Table 4.2.7 Concept Selection

Table 4.2.7 is designed to showcase the values of each concept according to the development of our critical values determined by the objectives tree. The table shows how each concept compares on each of these options. The table compares each of them according to functionality, usability, complexity, cost and marketability. The values were generated by taking the concepts value and scaling it according to the weight of the criteria. The values are then added and shown at the bottom of the table. This weighted sum tells us what concepts are most desirable and it is clearly evident that concept A and concept D are the most desirable with equal values of 1.

**4.2.8 Final Concept**

|  |  |  |  |
| --- | --- | --- | --- |
| Final Concept Generation |  |  |  |
|  | Concept A | Concept D | Final Concept |
| OS | Android | IOS | Android |
| SDK | Vuforia | Vuforia | Vuforia |
| Image Processing | Scanning | Scanning | Embedded |
| Identification | QR | Text | Text |
| Mapping | Included | Included | Included |

Table 4.2.8

Table 4.2.8 shows the final concept for the application. The Final Concept is a combination of Concept A and Concept D along with any advantages from the previous concept selections. For Image Processing it was decided by the Team to use Embedded because it held the advantage over Scanning even though Scanning was included within Concepts A and D.

**Updated Final Concept**

|  |  |
| --- | --- |
|  | Final Concept |
| OS | Android |
| SDK | Vuforia using Unity |
| Image Processing | Embedded |
| Identification | Image Processing |
| Mapping | Included |

Table 4.2.9

After working on this project our group has had many difficulties working with the Android Studios IDE and Vuforia SDK. As such we’ve found that we have to use Unity (a Game Engine) in order to optimize the SDK. Once Unity is able to run the portion of the programming we need we can import it into Android Studios to sync it with the final application. Also, though our group would have preferred to use text recognition due to the constraints of Unity along with the amount of tutorials available we thought it best to show proof of concept with image recognition processing.

**4.3 Statement of Work**

Our statement of work goes through what the team is currently working on, and the current standpoint on the project’s completion.

1. Currently working on Deliverable 4
   1. Generating concepts
   2. Comparing concepts to find out which approach would be best for the app
   3. Selecting our final planned design to implement in the app
2. Performing background research to solidify our understanding of the SDK and Android programming.
3. Researching relevant technology to find out what types of features are applicable to the app, and choose which types of features we plan on implementing.
4. Learning the Javascript programming language that is associated with the Vuforia SDK.