Challenges of VR Application Distribution

David J. Zielinski
Smith Media Labs Technology Specialist
VR/AR Software Developer
http://people.duke.edu/~djzielin/
How did I get here?


Research and development engineer for the DiVE virtual reality lab. (2004-2018)

Smith Media Labs Technology Specialist. (2018-present)
VR Hardware Systems

Era of Expensive and Bespoke Systems.
Unique Libraries / API’s per device.
Lack of standardization (HW + SW).
Difficult to share / distribute content.

Era of “Low Cost” VR Hardware.
Made possible via smartphone arms race
Some standardization. App stores.
Easier to share / distribute content.
VR Hardware + Software Obsolescence

Q: Does the built application still run on current computers? (non-standalone systems)

Q: Do we still have access to the hardware? is it working?

Q: Do the software tools/libraries used to create the applications support modern hardware?

Q: Do the software tools used to create the application still exist?
Q: Does the built application still run on current computers? (non-standalone systems)
Oculus DK2 (released 2014) application not working with current operating systems. (2019)

Q: Do we still have access to the hardware? is it working?
Duke DiVE planned to be decommissioned. (2019)

Q: Do the software tools/libraries used to create the applications support modern hardware?
Open-source Syzygy library no longer maintained. Doesn’t support modern HMD’s. (2016)

Q: Do the software tools used to create the application still exist?
Virtools commercial game engine discontinued. (2009)
## Modern VR Content Creation

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>360 Videos</td>
<td>Easy to distribute</td>
<td>No interaction</td>
</tr>
<tr>
<td></td>
<td>Film type workflow (no coding)</td>
<td></td>
</tr>
<tr>
<td>Webpage (WebVR)</td>
<td>Easy to distribute / Self publish (put up on a webserver!)</td>
<td>Lower performance (lower quality visuals)</td>
</tr>
<tr>
<td></td>
<td>Works on most platforms!</td>
<td>Messy coding for interactions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Web is changing quickly!</td>
</tr>
<tr>
<td>Binaries</td>
<td>High performance</td>
<td>Separate build for each platform.</td>
</tr>
<tr>
<td></td>
<td>High visual quality</td>
<td>Distribution can be complicated</td>
</tr>
<tr>
<td></td>
<td>Flexible interaction programming.</td>
<td>Tools can go away or change monetization</td>
</tr>
<tr>
<td>“Easy VR Authoring Tools”</td>
<td>No coding</td>
<td>$$$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specific vs generalizable tools?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Support for platform we need?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tools can go away or change monetization</td>
</tr>
</tbody>
</table>
Modern VR Content Creation

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>360 Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webpage</td>
<td>Webpage (WebVR)</td>
</tr>
<tr>
<td>Binaries</td>
<td>Binaries</td>
</tr>
<tr>
<td>“Easy VR Authoring Tools”</td>
<td>“Easy VR Authoring Tools”</td>
</tr>
</tbody>
</table>

**Insta360 One X**
- Price: $400
- Resolution: 5.7k
- Format: .mp4 file

**Vuze XR**
- Price: $430
- Resolution: 360° / 180° stereo
- Format: .mp4 file

**Raw files on SD Card**
- Insta360 Studio / Insta360 Stitcher

**Desktop Computer**

**“Sideload” directly onto device**

**YouTube**
- “Install app from Oculus Go app store”

**YouTube VR**
Modern VR Content Creation


Deliverable

360 Videos

Webpage (WebVR)

Binaries

“Easy VR Authoring Tools”
Modern VR Content Creation

Deliverable

360 Videos

Webpage (WebVR)

Binaries

“Easy VR Authoring Tools”

Typically going to use a game engine:

How to get Unity to support VR?
- Libraries to abstract out HMD: VRTK, MiddleVR.
- Unity is working on XR API.
- Vendor specific library.

Build generates a directory of files, or for Oculus Go, a single APK file.

This can be sideloaded directly onto device (via micro USB) or via app stores.
Modern VR Content Creation

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>360 Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Webpage</td>
</tr>
<tr>
<td></td>
<td>(WebVR)</td>
</tr>
<tr>
<td></td>
<td>Binaries</td>
</tr>
<tr>
<td></td>
<td>“Easy VR Authoring Tools”</td>
</tr>
</tbody>
</table>

**Revit Live** lets architects VRitize Revit models.

**VRDirect** allows home sellers to create 360 video tours. Jump from hotspot to hotspot.

**Google Expeditions** allows educators to create and use annotated 360° tours.
App Stores

Advantages:

- Easy for users to install your (binary) app.

Disadvantages:

- Steam requires $100 application fee per application.
- Oculus has many technical requirements (high frame rate, run for 45 minutes with no thermal issues, loading indicator, ...).
- App Stores may limit types of content (violence, political / hate speech, adult). Steam is now looking to change to a more “anything goes” policy.

* Note: I haven’t published anything to an app store yet...
What about distributing source code?

Git is a recent standard for “publishing” code in an open-source way.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Potential Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unity projects can be large. Over the 1G GitHub limit.</td>
<td><a href="https://gitlab.oit.duke.edu">https://gitlab.oit.duke.edu</a></td>
</tr>
<tr>
<td>Utilizing paid Unity assets in a public repository violates TOS.</td>
<td>Avoid all non-free assets? Keep repository private?</td>
</tr>
<tr>
<td>Some “free” assets are monetized by number of monthly users</td>
<td>Have to pay if app gets too popular...</td>
</tr>
<tr>
<td>Ideal Solution: A Git submodule solution that pulls directly from asset store?</td>
<td></td>
</tr>
</tbody>
</table>
Colossi of Sesostris
(Hdt. II,110; Diod. I,57,5)

Moreover alone of all the Egyptian kings had rule over Ethiopia; and he left as memorials of himself in front of the temple of Hephaistos (i.e. Ptah) two stone statues of thirty cubits each, representing himself and his wife, and others of twenty cubits each representing his four sons.
Class to Consider:

Introduction to Programming and User Interface Design in Unity3D. ISS 320, VMS 326. Fall 2019

2016 Final Projects

2017 Final Projects

2018 Final Projects