Digital storytelling: building the holodeck

I’ve been in and out virtual worlds since I was an undergraduate in the mid '90s and man how have things changed between now and then. This series of essays outline some of the historical context for some ideas for augmented additions to the real world and fully immersive virtual environment.

In 1987 we were introduced to Star Trek: The Next Generation and the multiple new technologies it introduced, the holodeck chief among them.

> A Holographic Environment Simulator, or holodeck for short, was a form of holotechnology designed and used by Starfleet. They were installed aboard starships, space stations, and at Starfleet institutions for entertainment, training, and investigative purposes. A typical holodeck consisted of a room equipped with a hologrid containing omnidirectional holographic diodes, enabling holographic projections and creation of holodeck matter through the manipulation of photons contained within force fields.
>
> The most obvious function of a holodeck is to provide entertainment and diversion for the crew. (TNG: “11001001”, “The Big Goodbye”, “Elementary, Dear Data”, “A Fistful of Datas”; VOY: “The Cloud”, “Homestead”)
>
> A holodeck can be used to create training simulations and exercise environments not otherwise available or safe, including star ship battle simulations and the Bridge Officer’s Test. (TNG: “Code of Honor”, “Where Silence Has Lease”, “The Emissary”, “New Ground”, “Firstborn”; VOY: “Learning Curve”, “Extreme Risk”, “The Fight”)
>
> The holodeck can be used as a laboratory to aid in analysis, such as recreating the scene of a crime or accident to aid in forensic investigations. (TNG: “A Matter of Perspective”) They can be used to visualize a 3D scene from alternate data sources for analysis (TNG: “Identity Crisis”, “Phantasms”; VOY:

> “Distant Origin”) or used as a brainstorming tool. (TNG: “Schisms”, “Booby Trap”)
>
> — Memory Alpha

Holodecks can create both open ended worlds (like the worlds Jake Sisko ans his father shared in DS9’s Emissary) and specific scenarios (like the Dixon Hill Series of pulp novels that Captain Picard likes in TNG’s “The Big Goodbye”, “Manhunt”, “Clues”) or more open ended scenarios like the opening of Deep Spance Nine’s Emmisary or the Victorian novels of Voyager’s captain Janeway.

While we don’t have the holodeck quite like the one in the series but in the last 4 years or so virtual technologies have matured tto the point where an early version of the holodeck may be possible… nothing like Star Trek or the Full Dive Gear from Sword Art Online but immersive enough to trick our brains into thinking we’re already there.

In these essays we’ll explore Beacons as storytelling devices, Augmented Reality gear to tell stories and fully immersive environments and the stories we can tell in and with these 3D virtual worlds.

Digital worlds then and now

The progress to a fully 3D immersive holodeck is not a new endeavor or something that has appeared in a single discipline. In order to better understand what we want to do in the VR world it pays to get an idea of where we’ve already been. I’ll break the old virtual worlds into three categories:

- Text-based virtual environments
- SIMNET and military simulations
- Early attempts at virtual reality: Lucasfilms Habitat

Text-based virtual environments

The text-based environments I refer to the in this section are the TinyMUD created by James Aspnes in 1989 as the first in a line of successors to the original MUD created by Roy Trubshaw and Richard Bartle in 1978.

What differentiated TinyMUD and its children was a move away from combat as emphasized in the original MUD into a more social and communication environment. It also allowed al players to create objects and spaces in the world and then link it to existing content, whether player created or not.

I still remember my time as an anthropology undergrad (1994 - 98) and how I became involved in the early forms of immersive storytelling environments. MUDs (Multi User Dungeons) and their derivatives provided free-form interactions in a fully immersive textual experience.

For me it was Lord of The Rings and Battlestar Galactica. These style of worlds known as MUSH (Multi User Shared Hallucination) provides for a fluid style of writing that describes both the users' speech and their interactions.

The example below, taken from Elendor MUSH illustrates the type of interaction players write.

> Below is a snippet from an RP scene in Gondor recently played. The two players involved are Tathar and Turlach Nimothan, aunt and nephew.
>
> There is not much to be seen here, for the weather is dark and rainy and the bookshelves unlit. But at a table is set a candle, its wick nearly drowned in melted wax, and beside it, Turlach's dark head rests upon an open book. His eyes are closed.
>
> The door swings open quietly, and the wavering light of another candle shedding its soft golden glow into the room. Tathar comes in, stopping as she sees Turlach, then moving to the bookshelves, where she lights a lamp, and pulls out one of the books, taking it and her candle to a chair nearby.
>
> [fast-forward to a bit of their conversation.]
>
> "It means ..." Turlach looks up cautiously, eyeing the slightly-ajar door. "She may have been taken by the Black Company. But why, I wonder?"
>
> The door is pushed further open as an elderly butler comes in. "Yes, m'lady?"
>
> "Something hot, please," Tathar instructs him. "Turlach

> and I are growing chilled, doing nothing but sitting and reading. A hot drink would be very welcome.” Beldin nods and vanishes again, and Tathar turns her attention back to her nephew. She is frowning.
>
> “Taken…? But of what avail would that be to them? The Ephalkhir have nothing to do with Mormegil or its estates.”

The idea that attracted a large number of people (myself included) to MUDs was the chance to recreate the worlds of our favorite novels or TV series and push them beyond their conclusion. It wasn’t enough to know how the last episode of the single season of Battlestar Galactica ended, we wanted to build the story of what happened afterwards. Did the Galactica find Earth? How did it happen?

MUD creators provided us with very detailed worlds built on text and the possibilities to interact with the environment and with objects created by the system operators or by other players. We recorded the stories we played in to have a record of our collective performances for posterity.

The experiences were not always positive. Some of the first conversations about online communities, punishment and enforcement originated from less than pleasant, some may say terrifying, events that happened in these early text-baed communities.

Wherever you see an MMORPG you’re seeing the children of MUD and when looking at AR and VR Stories it is no different. We are all children of MUD.

Distributed Training Simulations: SIMNET

Figure 1: Simnet Simulators

> SIMNET was the first successful implementation of large-scale, real-time, man-in-the-loop simulator networking for team training and mission rehearsal in military operations.
>
> — SIMNET: The Advent of Simulator Networking

> SIMNET stands for SIMulator NETworking. Initiated in 1983, it was the first “shared virtual reality” distributed simulation system, which continues to have significant influences. It was sponsored by DARPA, the Defense Advanced Research Projects Agency, the Department of Defense’s principal high-risk, high-payoff research and development organization, established in 1958. […]
>
> — SIMNET and Beyond: A History of the Development of Distributed Simulation

SIMNET was created in the early 1990s to provide training simulations for large scale forces without the cost associated with moving such large forces to training sites like The National Training Center in Fort Irwin California.

Figure 2: SIMNET Simulators

There were two revolutionary aspects of SIMNET: The real time long distance networking infrastructure and the simulator pods built to participate in the simulations.

SIMNET Networking

SIMNET provided a fully decentralized architecture for each of the simulator pods representing anything from an M1 Abrahams tank to an AH-64 Apache attack helicopter.

The biggest improvement was the lack of a central coordinating unit. Each simulator pod was responsible for tracking its status, the status of any unit it interacted with directly and any result of interaction (did it hit another object such as a mine, did a gun fired hit something or was the unit hit by an object fired by another unit) and of reporting its state to units it interacts with in the network. This was defined as each unit being an “autonomous node”.

This made SIMNET scalable to several hundred units spread around military bases in the United States (Fort Benning, GA, Fort Rucker, AL, Fort Knox, KY, and Fort Leavenworth, KS) and Grafenwoehr, Germany. To accommodate for the increased bandwidth requirements and potential latency, SIMNET adopted two additional principles:

First, a simulation node would send updates only when its state changed. For example, a stationary entity would not send state updates (except for minimum “heart-beat” updates that assured other nodes that network communications had not been lost).

Second, an entity moving in a straight line at a steady speed would send state updates only when it changed its speed or direction.

SIMNET Simulator Pods

The simulator pods each represented a fully crewed vehicle (a 4-man tank or a 2-man helicopter). It would have been very difficult to fully reproduce a combat vehicle in the simulation environment SIMNET adoped a “selective fidelity” (also known as “the 60% solution”) where only parts relevant to the functioning of the vehicle in the simulation were kept. Commanders and other mission supervisors could decide what systems and aspects of the 60% were carried over. An example of this was the decision to induce “jiggle” in the vision block views and a “rumble” in the crew’s seats to give them clues as to how fast they were moving and what kind of terrain they were traversing.

Some of the vehicle simulator pods (taken from SIMNET and Beyond):

M1 Abrams Main Battle Tank. The M1 has been the Army’s primary heavy combat vehicle since entering service in 1980. It carries a crew of four: the driver, who reclines in a very tight space inside the forward hull compartment, plus the tank commander, the loader, and the gunner, who sit inside the turret compartment. The driver’s orientation is fixed, but the turret can rotate 360 degrees (quite rapidly), so the turret’s orientation can be independent of which direction the tank is moving. The tank commander’s role is to control and coordinate the tank’s overall operation, as well as to coordinate tactically with other commanders. He tells the driver where to go, designates targets for the gunner, and tells the loader what type of ammunition to pull from the rack behind the blastproof doors and load into the breech of the 105 mm or 120 mm cannon. The crew communicates with each other over a local intercom, and the commander communicates over a multi-channel radio.

Similar to the real vehicle, the M1 simulator has two separate compartments: one for the driver and one for the other three crew members, each with the most essential controls and displays (chosen in accordance with the “selective fidelity” principles previously noted). The commander sits above the gunner, with the loader located to their left between the ammunition rack and the gun breech. A full description of the simulator crew compartments and controls from the SIMNET era can be found in (Chung, 1988). The driver has three “vision block” displays, the gunner has a stabilized gunsight with high/low magnification, the commander has three vision blocks plus an auxiliary sight that duplicates the gunner’s sight, and the loader has a periscope display. Both the commander’s and loader’s cupolas can rotate to simulate additional vision blocks in a closed-hatch M1.

Rotary-Wing Aircraft (RWA). Generic attack/scout helicopter, approximating the AH-64 or OH-58. Includes pilot and copilot/gunner stations with flight controls and displays. Includes a slewable electrical-optical sensor for use in target acquisition and engagement. It is armed with HELLFIRE laser-guided anti-tank missiles and air-to-air STINGER missiles, in addition to a 30-mm chain gun. The pilot and co-pilot share visual access to eight out-the-window displays, arranged in a “three over five” configuration, plus a single sensor channel, switchable between a thermal image and a daylight TV image. The RWA uses typical helicopter cyclic and collective controls, along with a throttle, for flight maneuvers. (Harrison, 1992)

SIMNET Impact and Legacy

The simulators were not a marvel of high technology but they did their job well enough that SIMNET evolved from a training system into mission planning and, eventually, technology development simulator. The technologies pioneered in SIMNET are now part of a family of Distributed Interactive Simulation standards that have grown past their origins in wargaming and war fighting simulations; space exploration and medical fields have also embraced simulation as a lower cost, lower risk alternative to live activities.

The events are reproducible. The Battle of 73 Easting was fully recreated in collaboration with the member of the unit involved and a very thorough analysis. They captured GPS coordinates from the American vehicles involved in the battle and reconstructed the event as accurately as possible. People can now review the simulated battle and see the action as it happened more than 20 years ago. Simnet became a study (and marketing) tool for the Army and the basis for a new family of training environments, the latest of which is the Synthetic Training Environment.

Several members of the original teams that worked in SIMNET moved on to gaming and distributed networking endeavors after SIMNET was over. Some of these startups include:

- RTIME, Inc. (Rolland Waters) game network engines
- MetaVR, Inc (W. Garth Smith), simulation and training, GIS systems
- MaK Technologies (Warren Katz and John Morrison), simulation software
- Reality by Design, Inc (Joanne West Metzger and Paul Metzger), simulation and training systems
- Zipper Interactive (Brian Soderberg), game developer (SOCOM game series)
- Wiz!Bang (Drew Johnston), game developer

Early attempts at graphic VR: Habitat

Figure 3: Lucasfilm's Habitat

One of the first attempts at creating a graphical online virtual world was Habitat. Created by F. Randall Farmer and Chip Morningstar for Quantum Link an early Internet Service Provider targeted to Commodore 64 and Commodore 128 systems using dial up connections.

Figure 4: An Habitat scene.

The lead creators of Habitat, Chip Morningstar and F. Randall Farmer have written extensively about what Habitat is and what lessons they have learned in the process.

> The essential lesson that we have abstracted from our

> experiences with Habitat is that a cyberspace is defined more by the interactions among the actors within it than by the technology with which it is implemented. While we find much of the work presently being done on elaborate interface technologies – DataGloves, head-mounted displays, special-purpose rendering engines, and so on – both exciting and promising, the almost mystical euphoria that currently seems to surround all this hardware is, in our opinion, both excessive and somewhat misplaced. We can't help having a nagging sense that it's all a bit of a distraction from the really pressing issues. At the core of our vision is the idea that cyberspace is necessarily a multiple-participant environment. It seems to us that the things that are important to the inhabitants of such an environment are the capabilities available to them, the characteristics of the other people they encounter there, and the ways these various participants can affect one another. Beyond a foundation set of communications capabilities, the details of the technology used to present this environment to its participants, while sexy and interesting, are of relatively peripheral concern.
>
> — The Lessons from Lucasfilms Habitat

It is interesting to see the parallels between Habitat and its modern descendants like Destiny and World of Warcraft. The characters and the backgrounds and the interactions are different but how you move and how you interact with people hasn't really changed significantly since the early days of online multi player games.

Some things the authors warn us that central planning is impossible. In a fully interactive multi player environment makes it impossible to plan every little contingency. We'll discuss this when we talk about storytelling and planning versus spontaneity.

The emphasis has always been in community building. Habitat was immersive not because of its technology, the graphics don't hold a candle to today's massive 2D and 3D game environments, textures and animations, but because of the community and the interactions between people who were in different geographical locations yet were able to live and interact together in a community that was free of trolls and free of the harassment that permeates our modern online communities.

They also argue for a world that is consistent with itself. If something happens handle in the world. Don't step out of the world to handle it. The two examples they give very strongly illustrate the point.

In the first instance it would have been easy to just take the money away from the 'cheaters' and restore the economy to what it should be. They were lucky that the avatars were good and used their resources for the betterment of the community.

> In order to make this automated economy a little more interesting, each Vendroid had its own prices for the items in it. This was so that we could have local price variation (i.e., a widget would cost a little less if you bought it at Jack's Place instead of The Emporium). It turned out that in two Vendroids across town from each other were two items for sale whose prices we had inadvertently set lower than what a Pawn Machine would buy them back for: Dolls (for sale at 75T, hock for 100T) and Crystal Balls (for sale at 18,000T, hock at 30,000T!). Naturally, a couple of people discovered this. One night they took all their money, walked to the Doll Vendroid, bought as many Dolls as they could, then took them across town and pawned them. By shuttling back and forth between the Doll Vendroid and the Pawn Shop for hours, they amassed sufficient funds to buy a Crystal Ball , whereupon they continued the process with Crystal Balls and a couple orders of magnitude higher cash flow. The final result was at least three Avatars with hundreds of thousands of Tokens each. We only discovered this the next morning when our daily database status report said that the money supply had quintupled overnight.
>
> We assumed that the precipitous increase in "T1" was due to some sort of bug in the software. We were puzzled that no bug report had been submitted. By poking around a bit we discovered that a few people had suddenly acquired enormous bank balances. We sent Habitat mail to the two richest, inquiring as to where they had gotten all that money overnight. Their reply was, "We got it fair and square! And we're not going to tell you how!" After much abject pleading on our part they eventually did tell us, and we fixed the

> erroneous pricing. Fortunately, the whole scam turned out well, as the nouveau riche Avatars used their bulging bankrolls to underwrite a series of treasure hunt games which they conducted on their own initiative, much to the enjoyment of many other players on the system.

The second example is of how to handle incidents “in world” as opposed to handling it as a technical exercise and frustrate users when we violate the contract between the users and the fantasy in their world.

> One of the more popular events in Habitat took place late in the test, the brainchild of one of the more active players who had recently become a QuantumLink employee. It was called the “Dungeon of Death”. For weeks, ads appeared in Habitat’s newspaper, The Rant, announcing that that Duo of Dread, DEATH and THE SHADOW, were challenging all comers to enter their lair. Soon, on the outskirts of town, the entrance to a dungeon appeared. Out front was a sign reading, “Danger! Enter at your own risk!” Two system operators were logged in as DEATH and THE SHADOW, armed with specially concocted guns that could kill in one shot, rather than the usual twelve. These two characters roamed the dungeon blasting away at anyone they encountered. They were also equipped with special magic wands that cured any damage done to them by other Avatars, so that they wouldn’t themselves be killed. To make things worse, the place was littered with cul-de-sacs, pathological connections between regions, and various other nasty and usually fatal features. It was clear that any explorer had better be prepared to “die” several times before mastering the dungeon. The rewards were pretty good: 1000 Tokens minimum and access to a special Vendroid that sold magic teleportation wands. Furthermore, given clear notice, players took the precaution of emptying their pockets before entering, so that the actual cost of getting “killed” was minimal.
>
> One evening, one of us was given the chance to play the role of DEATH. When we logged in, we found him in one of the dead ends with four other Avatars who were trapped there. We started shooting, as did they. However, the last operator to run

> DEATH had not bothered to use his special wand to heal any accumulated damage, so the character of DEATH was suddenly and unexpectedly “killed” in the encounter. As we mentioned earlier, when an Avatar is killed, any object in his hands is dropped on the ground. In this case, said object was the special kill-in-one- shot gun, which was immediately picked up by one of the regular players who then made off with it. This gun was not something that regular players were supposed to have. What should we do?
>
> It turned out that this was not the first time this had happened. During the previous night’s mayhem the special gun was similarly absconded with. In this case, the person playing DEATH was one of the regular system operators, who, accustomed to operating the regular Q-Link service, had simply ordered the player to give the gun back. The player considered that he had obtained the weapon as part of the normal course of the game and balked at this, whereupon the operator threatened to cancel the player’s account and kick him off the system if he did not comply. The player gave the gun back, but was quite upset about the whole affair, as were many of his friends and associates on the system. Their world model had been painfully violated.
>
> When it happened to us, we played the whole incident within the role of DEATH. We sent a message to the Avatar who had the gun, threatening to come and kill her if she didn’t give it back. She replied that all she had to do was stay in town and DEATH couldn’t touch her (which was true, if we stayed within the system). OK, we figured, she’s smart. We negotiated a deal whereby DEATH would ransom the gun for 10,000 Tokens. An elaborate arrangement was made to meet in the center of town to make the exchange, with a neutral third Avatar acting as an intermediary to ensure that neither party cheated. Of course, word got around and by the time of the exchange there were numerous spectators. We played the role of DEATH to the hilt, with lots of hokey melodramatic shtick. The event was a sensation. It was written up in the newspaper the next morning and was the talk of the town for days. The Avatar involved was left with a wonderful story about having cheated DEATH, we got the gun back, and everybody went away happy.

> These two very different responses to an ordinary operational problem illustrate our point. Operating within the participants’ world model produced a very satisfactory result. On the other hand, taking what seemed like the expedient course, which involved violating the world-model, provoked upset and dismay. Working within the system was clearly the preferred course in this case.

Some of these experiences continue to shape our technological communities, how to balance freedom and creativity with keeping people safe. Others have to do with community rules, user types and managing behavior and expectation.

Single user stories, like those I’m most interested in crafting, don’t have these types of problems but, as soon as you bring more than two people together and one or more are women, online you have to deal with the potential for attacks, trolling and sexist remarks.

Places-based stories

One of the things that has really caught my attention is place-based story telling. How can a place tell a story that is different depending on the occasion or the type of story we want to tell. The two biggest technologies for this type of story telling are augmented reality and beacons/Physical web.

Using beacons to tell a story

Beacons, iBeacons or the Physical Web all refer to the same group of technologies that use Bluetooth Low Energy to broadcast information to devices and apps that are ready to use them. In this section I will concentrate in the Physical Web technologies because they are what I'm most familiar with. Other beacon technologies should work in a similar fashion.

In "Coffee With a Googler" Laurence Moroney discusses the Physical Web with Scott Jenson. This is the best introduction to the technology for non-technical audiences.

The idea behind the physical web is that we can incorporate beacons to everyday objects and places and give these objects and places an online presence and interaction capability. I first saw Physical Web devices demonstrated at the Chrome Developer Summit in 2015 with a vending machine taking input from your mobile phone to "sell" you candy.

We will use the URL capabilities of the beacons to build experiences.

We can place different pieces of the story in multiple beacons and leave it up to the player to fully construct his version of the story, realizing that the reader may not work on the story linearly or complete it altogether at which point it'll be up to the reader to complete the story to their satisfaction.

AR: Augmented Reality Worlds

> Augmented reality (AR) is a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. As a result, the technology functions by enhancing one's current perception of reality.[1] By contrast, virtual reality replaces the real world with a simulated one.[2][3] Augmentation is conventionally in real time and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulable. Information about the environment and its

> objects is overlaid on the real world. This information can be virtual[4][5][6][7][8] or real, e.g. seeing other real sensed or measured information such as electromagnetic radio waves overlaid in exact alignment with where they actually are in space.[9][10] Augmented reality brings out the components of the digital world into a person's perceived real world. One example is an AR Helmet for construction workers which displays information about the construction sites. The first functional AR systems that provided immersive mixed reality experiences for users were invented in the early 1990s, starting with the Virtual Fixtures system developed at the U.S. Air Force's Armstrong Labs in 1992.[11][12][13]
>
> — Wikipedia

If you've seen Niantic's AR games Ingress and Pokemon Go you have played an Augmented Reality game. In the images below different Pokemon from the first generation of the Pokemon Go game appear around Embarcadero in San Francisco

Figure 5: Different images from Pokemon Go in San Francisco, CA. Images from http://www.pokemongo.com/en-us/photos/)

Sadly when Google terminated their Google Glass Explorer program it also terminated any posibility of development outside Google Glass at Work they left Microsoft Hololens as the main commercially available augmented reality development tool set (both hardware and software).

Some of the players in this market

the list keeps growing larger so I probably won't update it often:

Microsoft Hololens

- Untethered
- Full PC running Windows 10, can take advantage of desktop applications as well as AR specific
- Requires Unity to develop
- Windows Only, will use Mac or Linux as a development platform

- Expensive (3 to 5 thousand dollars per unit, current developer price)

Meta 2

- Available for presale ($950)
- Tethered

Vuzik M3000 Smart Glasses

- Look like the successor to Google Glass
- Android based
- Geared towards remote applications

Sonny SmartEyeglass

- Requires Android host device (Android 4.4 and newer, see compatibility table)
- Geared towards business

Recon Jet

- Sport and work (pro model) emphasis
- Android based

Magic Leap

- No one knows what the device or the technology looks like
- Adaptive focus
- Cheaper than Hololens but still in the 1 thousand dollar range

CastAR

- Under Development

AR Toolkit

- Toolkit to create AR without devices
- Uses device’s built in camera and requires WebGL and WebRTC to work
- Doesn’t work in iOS yet (no WebRTC)
- Open source Github project

Since I started writing this the Meta 2 went on sale for developers and I’ve discovered many AR devices targeted to business and sports (Recon Jet) markets. The problem with having so many vendors in the market is that you need to decide what devices to target and how many versions of your experience you want to create.

I’m afraid that it won’t be long before we go back to the bad old web days where we had to develop content that would work well on each different browser and not at all in others. Even when using a common development platform (Unity code written in Microsoft’s C# language) the code will still be different across platform to make a common development platform still a dream.

For this exploration I’ve chosen to plan a reading experience for Microsoft Hololens. The device is the best for the way I want to tell stories and for the type of stories that I want to tell.

Building AR experiences with the Hololens presents different challenges than just placing beacons in a place. It needs at least one desktop/laptop machine to create the content, one device to build the world and a way to share the experience with other people wearing the device.

The big advantage of AR over beacons or VR is that we can leverage the world around us to place our content. Augmented Reality or Mixed reality as it’s also known allows creators to superimpose virtual object into the real world for users to interact with.

One of the disadvantages that I see for AR storytelling is the size of the device and the adverse public reaction people have had to public use of earlier AR devices like Google Glass and how people saw them as a threat to privacy and as a way to break the law without people realizing it. We’ll talk more about this when we talk about if we are ready for the technology.

Full, Immersive VR

We’ve looked at using beacons and we’ve began our discussion of digital storytelling using Augmented Reality. Now we’ll look at a fully immersive experience using Virtual Reality Gear. This section is less about code and more about the possibilities inherent in telling stories in this medium using a small set of technologies:

- Leap Motion to provide a set of interaction gestures
- WebGL 3D environments to build the world we’ll be interacting in
    - A-Frame
- Possible voice interfaces to add further interfaces to the stories we tell
    - Speech Synthesis API
- Unity 3D environment

As we explore these tools we’ll also look at:

- Navigation and travel through digital worlds
- Bots and other clues for the virtual tourist

We’ll start working with A-Frame, from Mozilla. It does an excelent job of abstracting the underlying Three.js and WebGL code into a tag-based language, similar to HTML. At its simplest an A-Frame scene looks like the code below. We link to the A-Frame library and then write tags to create the content we want:

- A scene to hold our content
- A box
- A sphere
- A cylinder
- A plane to put the content in
- A sky element with a set color

A-Frame does it all in less than 10 lines of HTML-like tags. The creator doesn’t need to know what goes on “under the hood” to see the results of the code.

```
<html>
  <head><head>title>Hello, WebVR! - A-Frame</title>
    <meta na</title>ription" content="Hello, WebVR! - A-Frame">
    <script src="https://aframe.io/releases/0.5.0/aframe.min.js"><script
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-bo
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sph
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#F
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" c
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```

We can get as complex as we want to. The example below shows how to do animation in A-Frame. The example below from the A-Frame site provides a much richer experience without increasing the lines of code.

> The example introduces additional elements like entities, light and camera.
>
> The full working example can be found in the A-Frame Examples library

```
<!DOCTYPE html>
<html>
  <head><html>title>lazerGlazer_Viz</title>
    <style></title>
      html, body { margin: 0; padding: 0; overflow: hidden; }

    <script src="https://aframe.io/releases/0.4.0/aframe.min.js"></script
    <script src="https://rawgit.com/ngokevin/aframe-animation-component/ma
  </head>
  <body>
    <a-scene>
      <a-camera position="0 0 0"></a-camera>
      <a-sky></a-sky>
      <a-entity id="object-container" position="0 1.6 -2" scale=".4 .4 .4"
      <a-light type="ambient" intensity="0.7" color="#ffffff"></a-light>
      <a-light intensity="3" color="#00ffff" position="-5.72 6.65 0.80"
          animation__color="pr</script>lor; dir:alternate; dur:2000; easin
          alongpath="path:10,10,-10 -20,10,-10 10,0,-10; closed:true; dur
```

```
   <a-light intensity="5" color="#ff0000" position="8.60 6.65 0.80"
     animation__color="property:color; dir:alternate; dur:2000; easi
     alongpath="path:-2,-2,5 2,-1,5 0,-1,5; closed:true; dur:3000;">
  </a-scene>
 <script type="text/javascript" src="scripts/app.js"></script</a-light><
 </body>
</html>
```

Starting with code like this we can build any type of structure that we can in Three.js. We can also use lower level constructs like WebGL shaders, the low level scripts that make WebGL elements change and do things beyond what stock WebGL allows for.

A-Frame's discussion of materials (one of the two components for objects in WebGL) discuses how to add custom Shaders to enhance the materials we use in our content.

We can also drop directly to Three.js code and use it within the A-Frame code. The A-Frame project also provides instructions on using Three.js from within A-Frame.

A-Frame gives you a declarative way to create 3D content and scenes without loosing the flexibility of Three.js and Shaders when we need to use them. We'll see whether using declarative markup or procedural code works better.

Creating these environments presents another question. ***How do we navigate from one scene to another***. For example, if our story takes place in a house, how do we navigate from one room to another and how do we restrict what places we can and cannot go to?

In an ideal world we'd load all the assets we need as a large application and then transition seamlessly between areas that works but puts a lot of assets in the user's computer that might or might not be neecessaary.

The simplest possibility is to tie the movement across sectors or boundaries. For example If we set the story in a house we can trigger movement from the house to a different location by loading the destination's assets when the user touches the door or waslk near it; similar to the way in which World of Warcraft and other MMORPGs load assets for a new section of the world.

Another way to do load assets is to load an initial set of assets, where the user begins the experience, and then stream new assets to the user's machine as needed and with a large splash screen to separate major areas of the world.

The final example I want to show is what a Leap Motion integration looks like in an A-Frame application. I've take the example from the aframe-leap-hands Github repository. Technologies like Leap, Oculus Touch and the Valve controllers help give a fuller virtual experience by allowing hand getstures and other haptic feed back for the user.

The new entities create a right and left hands for the Leap Motion controller and, once again, hides all the complexity of setting up the Three.js environment

```
<html>
  <head><head>meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta http-equiv="con<meta http-equiv="content-type" content="text/htm
    <title>Examples • Leap Hands</title>
    <script src="../budo.js"><title>t>
  </head>
  <body>
    <a-scene>
      <!-- Pl</head><!-- Player -->ty camera="near: 0.01" look-controls>
        <a-entity leap-hand="hand: left"></a-entity>
        <a-entity leap-hand="hand: right"></a-entity>
      </a-entity>

      <!-- Terrain -<a-entity leap-hand="hand: left"><!-- Terrain -->->
      <a-light type="ambient" color="#ccc"></a-light>
      <a-light color="#ddf" distance="100" intensity="0.4" type="point"></
      <a-light color="#ddf" position="3 10 -10" distance="50" intensity="0
    </a-scene>
  </body>
</html>
<a-light type="ambient" color="#ccc">
```

A-Frame has an audio component that is affected by the position of the object it's attached to. Combined with events this opens a wide variety of additional possibilities for the stories we tell… More research is needed.

Some of the devices I want to explore with require Unity as the development engine and provide additional code and libraries to make the experience easier. At first I was a little concerned on the requirement, yet another language to learn for uncertain return, but as I've started looking at the code it's become less scary than I expected.

I'm still trying to figure out C Sharp and how to best learn it and, because I'm lazy, how much of the code is actually hidden by the Unity Editor and how much code I have to write.

Below is an example video taken from Unity's Roll A Ball Tutorial

The code described in the video appears below:

```
using UnityEngine;
using System.Collections;

public class CameraController : MonoBehaviour {

    public GameObject player;

    private Vector3 offset;

    void Start () {
        offset = transform.position - player.transform.position;
    }

    void LateUpdate ()
    {
        transform.position = player.transform.position + offset;
    }
}
```

The final piece of software I want to discuss is AR.js by Jerome Etienne and team. It uses existing open source software and mobile phones to create AR models.

The example in the video (code below) extends A-Frame with AR specific information. The AR tools for AR.js extend AR Toolkit an open source AR tracking toolkit.

The idea is that we create the code to go with each marker and then when a mobile camera, like the one on a mobile phone, looks at the marker the corresponding AR model (in this case a rotating sphere of the world) will show up and rotate as an animation on your phone.

The code to display an image over the marker looks like this:

```
<script
  src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
```

```
<script
  src="https://rawgit.com/jeromeetienne/ar.js</script>frame/build/aframe-a
<script></script>
  THREEx.ArToolkitContext.baseURL = 'https://rawgit.com/jeromeetienne/ar.j
</script>
<body style='margin : 0px; overflow: hidden;'>
    <a-scene embedded artoolkit='sourceType: webcam;'>
    <a-sphere
    src="https://raw.githubusercontent.com/aframevr/sample-assets/master/a
      radius="0.5" position="0 0.5 0" segments-height="53">
         <a-animation attribute="scale"
              dur="1000"
              from= "1 1 1"
              to="2 2 2"
              direction='alternate-reverse'
              easing= "ease-in-out-circ"
              repeat="indefinite"></a-animation>
     </a-sphere>
        <a-marker-camera preset='kanji'></a-marker-camera>
    </a-scene>
</body>
```

Creating Stories For The Metaverse

We've looked at three events I consider ancestors to our modern AR and VR efforts, we've looked at some of the technologies, both hardware and software, we can use to create AR and VR experiences.

Using beacons is easy. Telling stories with beacons, or any sort of AR and VR, gets more complicated. I've chosen to play with interactive story telling as a way to tell stories in the metaverse. Using tools like these allows me to build both the main stories and the branches for it… how we translate the stories is the real challenge.

The more I think about it the more I come back to the tools of the story teller and, specifically, the tools of the LARP story teller. Whether it's Nordic LARP or a modern set of rules like By Night Studio's Vampire: The Masquerade and Werewolf: The Apocalypse they provide a complete set of mechanics for you to use as-is or adapt to a virtual environment… We'll explore how the stories change as we move further into the VR field.

Figure 6: Example of costuming for a Vampire The Masquerade LARP

Working with beacons presents a straightforward challenge. How do we segment the story we want to tell while, at the same time, preserving the reader/participant's interest through a potentially long sequence of beacons.

It's also important to remember that beacons only provide an access point. In and of themselves they will not give content to the user. This prompts the question: why use them?

Beacons are the perfect advertisement and storytelling system. They make things we wouldn't normally associate with the web or the Internet into entry points to interacting with online content… as simple as a vending machine transaction or as complex as your imagination can make them. We can also have as many beacons as we want in a building or groups of buildings.

Telling stories in AR is a little more complicated. Wherever we choose to tell the story we would seed the space with AR objects to help us tell stories. That introduces additional levels of complexity to our story telling: unless we control where a story is told we have no way to know what, if any, objects the actor/reader will interact with and in what order they will do so. Each of the elements in our story should stand on its own and help bring other important elements into the reader's view.

Microsoft provides a good set of guidelines for users on how to create shared holographic experiences that makes a good starting point

We also need to be aware of how the devices themselves will affect the way people react and the ways and places where we tell the stories.

Full VR is the most complex medium for telling stories but it's also the most rewarding. We can't take advantage of existing environments like we can with AR stories but we have to build the rooms, building and elements that we want our story to use. We also have to create bots and other interactions for the users.

Single versus Multi User Stories

AR stories also need some consideration regarding individual versus community (multiplayer) styles.

Creating single reader stories is as simple as placing the content in the environment and providing ways for other devices to access the anchored elements and stories. Because we're placing the stories in physical places we also have to account for the usage of the place where we're telling the stories. A park is different than a coffee shop or a bar as a place to tell a story.

When/if we decide to build a multi-user experiences we need to start considering how will people appear on each other's experiences like, if they will have the option of doing so and how will interactions change individual stories. Furthermore we need to consider if we'll need a server to handle player interactions and how will these interactions affect the individual stories.

How does the narrative change

when you're fully immersed in the story?

Writing a story for print or even web publication is one thing. When we put ourselves in the story by creating an avatar that will represent the reader and interact with the content of the world is something different. We need to consider what consequences will the body have in the virtual world.

In a digital environment we can engage more than just your brain and your hands. Depending on how the environment is configured we may have a full technologically mediated presence exercise

The International Society of Presence Research defined presence as:

> … [A] psychological state or subjective perception in which even though part or all of an individual's current experience is generated by and/or filtered through human-made technology, part or all of the individual's perception fails to accurately acknowledge the role of the technology in the experience. Except in the most extreme cases, the individual can indicate correctly that s/he is using the technology, but at *some level* and to *some degree*, her/his perceptions overlook that knowledge and objects, events, entities, and environments are perceived as if the technology was not involved in the experience. Experience is defined as a person's observation of and/or interaction with objects, entities, and/or events in her/his environment; perception, the result of perceiving, is defined as a meaningful interpretation of experience.
>
> International Society for Presence Research. (2000). The Concept of Presence: Explication Statement. Retrieved April 30, 2017 from https://ispr.info/

How much do we want to push the presence that we create in our stories? How much do we want to "fool" our users into thinking the objects and interactions we provide are real, or at least, that they are real enough to warrant suspension of disbelief for as long as they are engaged with our content?

How do we represent the character's physical interactions? It would be tempting to attempt to make a full body avatar of the user and provide full body tracking to translate the player’s movement into the Avatar’s actions and how he/she interacts with the content we create for them.

We also need to be mindful to represent the user as close as he/she is. Unless we can create avatars that directly represent the player or provide generic enough models of the user’s presence in the world we may be better off with providing less presence or doing so in such a way where only parts of the body are visible.

How far do we want to push the technology(ies)? It is becoming possible to do full body tracking and providing haptic feedback in Virtual environments. Before we jump too deep, let’s define what we mean by Haptic Feedback, Haptics or kinesthetic communications:

> Haptic or kinesthetic communication recreates the sense of touch by applying forces, vibrations, or motions to the user.[1] This mechanical stimulation can be used to assist in the creation of virtual objects in a computer simulation, to control such virtual objects, and to enhance the remote control of machines and devices (telerobotics).
>
> Wikipedia

Technologies like the Dexta Robotic’s Dexmo Haptic Exoskeleton and Cloud Gate Studio’s custom full-body tracking system for VR experiences would allow us to use more than our hands and heads to build experiences moving the problem back to creating a custom avatar for the user.

Scale considerations

> Room-scale (sometimes written without the dash) is a design paradigm for virtual reality (VR) experiences which allows users to freely walk around a play area, with their real-life motion reflected in the VR environment. Using 360 degree tracking equipment such as infrared sensors, the VR system monitors the user’s movement in all directions, and translates this into the virtual world in real-time. This allows the player to

> perform tasks, such as walking across a room and picking up a key from a table, using natural movements. In contrast, a stationary VR experience might have the player navigate across the room using a joystick or other input device.
>
> Wikipedia

One of the decisions to make is the scale we want to tell our story in. Until fairly recently we could only create virtual environments that gave us synthetic spaces that we moved though using keyboard, controllers or VR/AR specific controllers and devices.

New technologies allow for larger room-scale models where the user places sensors on different locations in the space and those provide tracking capabilities and the ability for the player to actually move their physical bodies the same way the bodies would move in the virtual world, increasing the level of presence we can provide.

The downside is that we need a physical space that is large enough to match the space we provide our users in the virtual world. Still it may present interesting avenues for storytelling.

Navigation and travel through the Metaverse

An issue related to scale is navigation. Whether the world is room-scale or a synthetic world we need to consider how our users will navigate around our world. Whether it’s a synthetic world that is no larger than a house or a large expanse of terrain where the user must either run, find a means of transportation (car or horse) or a teleportation device.

If the space is small or if we don’t have large spaces available then providing keyboard or controller navigation is the only way, particularly when working with tethered devices like the Rift.

Bots and other clues for the virtual tourist

Creating interaction in augmented or virtual spaces is another challenge worth considering. Do we want all interaction to be between players and objects or do we want to provide avatars and bots with particular interactions canned to respond?

I’m not implying that using objects to drive the story forward is necessarily bad but it makes the world a lonely place.

An idea worth exploring is whether we can use speech recognition to capture keywords and have the both react based on a pre-determined set of words to trigger responses that will move the story forward. Would something like the web speech API work in situations like this where the content is not served through a browser? Are there similar ways to work with this kind of technology in VR / AR space?

Keeping the world coherent

As we talked about when we discussed Habitat we need to keep the world coherent with itself. We must keep things consistent with the story we tell in the world we choose to tell them in.

This also means that we need to make sure that what we do throughout our story remains consistent, that whatever mechanics we choose to implement we do them the same way every time we have to use it.

Avoiding Sensory Overload

It’s easy to go overboard and provide everything for the user and build a large world for the user(s) to interact with. We can now create large, room scale VR experiences where we place sensors in multiple locations within a space and allow the user to move around the simulated space.

AR devices do something similar by allowing you to mix the real world with virtual objects so we need to be careful with the number and function of the virtual objects and their interaction with the real world.

Hamlet on the Holodeck

Figure 7: A holodeck from Star Trek: The Next Generation

The Holodeck is a lot more complex than I thought it was when I first started working on putting these ideas together. The stories we tell, how we tell them and how much they become the reader/participant’s stories is still being fleshed out and a lot of the work will fall to designers and developers.

Story Ideas

We’ve talked a lot about the how and what technologies to use when working in AR and VR storytelling. The following section will work on the what, ideas of types of stories that because of structure or content I believe lend themselves well to an AR or VR environment and, possibly, to a transmedia storytelling mesh.

Daemon and Freedom^{tm}

In these two novels, author Daniel Suarez chronicles the birth of the Daemon a self sustaining network “entity” capabable of interacting with other networks and real people, and the building of the Darknet, a community based on different principles than the corporate overlords trying to take over the world.

Freedom^{tm} is the more approachable story of the two and one where we could change to an augmented reality story. The Daemon’s Darknet, after all, already uses HUD glasses to communicate and provide information to its members… It shouldn’t be hard to convert that paradigm into a story.

Furthermore, if we constraint the story to a given geographical location it shouldn’t be too hard to augment real places in this location with story elements and way for readers to interact with them. The novel tells us how such an interaction of the real and the virtual happens.

Rayuela / Hopscotch

Hopscotch (Rayuela in Spanish), by Argentinian writer Julio Cortazar tells multiple stories about a group of expatriates first living in Paris and the further adventures of Horacio Oliveira, one of these expatriates upon his return to Argentina.

What attracted me to this story is that it’s multiple stories. You can read the first 73 chapters sequentially and be done. After chapter 73 the story ends and you move along. The other option is to read it using a table of directions that tell you what order to read the 155 total chapters.

What attracts me from this story is the multiple stories contained within. If you read it sequentially it’s one story but if you use the table of directions you discover more layers of the story as you move along.

This may be a good way to tell stories in VR and AR environments. Tell one core story and let the user build as many additional layers to the story as they want to.

Sword Art Online

Sword Art Online is a multimedia franchise (novels, anime series and movies) that tells the story of how people deal with fully immersive technologies and the stories they live in these virtual worlds.

The stories make use of fully immersive (full dive in the novels) technologies and also explores some of the implications of full dive technologies when missused or when applied to people with terminal medical conditions.

There is also a movie, Ordinal Scale, where the technology is essentially AR. The game is played in the real world and people and how they’d interact in an augmented reality environment.

Conclusion: Are we ready for the change?

Does anyone remember Google Glass and their deployment nightmares? Stories like the Seattle restaurant that banned Google Glass citing privacy concerns presents one of the first challenges to public AR and VR but not the last or, by any means, the only one. For more information see this Huffington Post Summary Post and this Guardian outlining why Google Glass was banned in theaters.

But the changes go beyond physical comfort with the technology and reach deep into what the technologies can do to people's real lives. In A Rape in Cyberspace Julian Dibbell tells the story of a "virtual" rape and the consequences (virtual and real), both for the people involved directly, the two victims of the event, the perpetrator, for the administrator, wizard in MOO parlance, who took it upon himself to enact the ultimate punishment available in the MOO and the writer himself.

The what and the how of the event have never been in question. The perpetrator, a character named Mr. Bungle, used a MOO object programmed to force other users to perform actions he created… in this case text based representations of rape and lesbian sex, all done against the other players' will and without their consent.

The perpetrator's character was deleted from the MOO's database by a Wizard, as administrators in MOOs are known. However the MOO doesn't ban people based on IP or email addresses used during registration so it was easy for Mr. Bungle to create a new account in the MOO.

Perhaps the most important thing, to me, is how real the event was for the participants even when it wasn't real (in the sense that it didn't happen in real life) but the consequences were just as serious as if they happened in the real world.

Bibliography

Text based MUDs

- The dragon ate my homework — Wired Magazine, 1993
- A Rape in Cyberspace
- The Original Internet Abuse Story: Julian Dibbell 20 Years After ‘A Rape in Cyberspace’

Simnet

- SIMNET
- SIMNET Wikipedia Entry
- SIMNET and Beyond
- SIMNET: An Insider’s Perspective

Habitat

- Lessons from Lucasfilm’s Habitat
- Social Dimensions of Habitat’s Citizenry
- The Game Archaeologist moves into Lucasfilm’s Habitat: Part 1
- The Game Archaeologist moves into Lucasfilm’s Habitat: Part 2
- The Game Archaeologist: The return of Habitat
- Bringing Habitat Back to Life
- Neoclassical Habitat Docs
- Neoclassical Habitat Github

Hololens

- Holographic Academy
- Shared Holographic Experiences
- Microsoft Mixed Reality Design

Creating stories for the Metaverse

- Virtual reality will transform cinema in 2016

- Undnerstanding the Daemon
- Joi Ito Review: Daemon

Are we ready for the change?

- Huffington Post Articles on Google Glass Ban
- Pokemon go: get outta here
- Will the real body please stand up??: Boundary Stories About Virtual Cultures (PDF)
- Cyborg Anthropology
- The Second Self: Computers and the Human Spirit
- Life on the Screen: Identity in the Age of the Internet
- Virtuality And Its Discontents
- Always On/Always On-You: The Tethered Self

Links, articles, books and concepts

- The Virtual Community: Homesteading on the electronic fronteer
- The whale and the reactor: a search for limits in an age of high technology
- Technology is making it easier to trust strangers
- CABBIBO
- AR.js

Videos

- Turing’s Cathedral: The Origins of the Digital Universe
- The Most Human Human: What Artificial Intelligence Teaches Us About Being Alive
- Computer History Museum Revolutionaries: George Dyson with Museum CEO John Hollar
- Authors at Google: George Dyson, “Turing’s Cathedral”
- Amber Case: We are all cyborgs now
- Youth and Media - Re:Born Digital, in Video: Identities
- A cyber-magic card trick like no other | Marco Tempest TED
- The magic of truth and lies (and iPods) | Marco Tempest