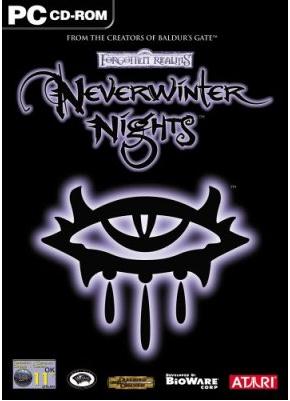
# WorldWizards: A modern world builder for instruction, experimentation and artistic expression



Jeffrey P. Kesselman  
Lecturer Interactive Media and Game Development  
Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA 01609-2280  
  
508-831-6369  
724-484-3728

# Introduction

The Bioware RPG toolkit Neverwinter Nights and its Aurora Engine, published in 2002, has been one of the most successful user generated content (UGC) based games ever shipped. A best seller when it was released, it was in continuous publication for more then 10 years making it one of the longest lived videogames of all time.

Its success was predicated on how it made it simple for people with no art or programming talent to build interactive immersive environments. A lego-like 3D tile system enabled non-artists to put pre-created pieces of environment together to create original spaces. These could then be populated with pre-built objects and creatures who also had pre-coded behavior mechanisms

An open data format allowed those with art and/or programming skills to further modify the experiences as well as creating pluggable world components for other users.

This toolkit was not only used by consumers to make and play each others games, but became a key teaching tool at schools such as WPI[[1]](#footnote-1), Becker[[2]](#footnote-2), and SMU Guildhall[[3]](#footnote-3)

Unfortunately, all software ages, and limitations built into the Aurora Engine eventually led to it losing relevance to modern game development. Since the application itself was closed source and not updatable by its users, it fell into over-all disuse. Chief among these limitations were:

1. **Low Polygon models**. Necessary for the time but giving the game a dated look as game hardware continued to advance.
2. **A dated render engine.** Because the engine was custom code, it did not advance to keep up with the state of the art in game technology.
3. **A 2.5D play space.** Although the world was 3D, players were limited to a single play surface and could not fly or jump over obstacles
4. **Very limited multiplayer ability.** The number of players that could play together on the same host computer topped out at about a dozen and had to all be on the same “map” or “zone.”
5. **An in memory execution model.** All the data for the current zone had to be loaded into main memory.
6. **Very limited player activity tracking.** Both this and the limited user count made it infeasible to collect large data sets for sociological research.
7. **No VR Support.** A key usage for a world builder today would be to encourage and enable experimentation with the virtual environments enabled by today’s Virtual Reality hardware,

# Proposal

The proposed project is to build a new world building kit that overcomes all of these limitations. It will be built a modern 3D Game engine (Unity3D) both lowering the cost of creation as well as leveraging the ongoing development of the Unity3D team to keep its visual content up to date.[[4]](#footnote-4)

This project would be built in stages as a series of sub projects:

1. **World Builder**  
   Goal is a functional true 3D tile based world assembly system built for VR environments. Tiles would be organized in matching tile-sets that are run-time loaded and arranged on a 3D grid. Level change tiles (stairs, pits, elevators, etc) would be implemented as filling multiple grid spaces. Tiles would have enough connectivity information to allow for auto-assembly (just like most 2D grid systems), and carry their own collision and navigation information that can be stitched together at run-time.  
     
   Objects will be place-able and movable on tiles at run-time. Doors would be listed in separate “door sets” and door anchor points baked into the tiles. Door motion will be controllable through scripting hooks.
2. **Creature Interface**Goal is to enable both humanoid and non-humanoid creatures to be placed in the environment. These creatures will have a combination of a standard action interface (move to, pick up, mele attack, etc) and additional creature specific commands that they will advertise into the editing environment.
3. **PC Creator**Goal is a flexible modern parameterized character avatar creator that can be used to create customized humanoid characters wearing changeable clothes. PCs will support the same creature interface as NPCs do, with a PC control layer ontop.
4. **Inventory Manager**Goal is the ability for PCs and NPCs to pick up, drop and chose items to use or store. PCs will get an inventory UI, NPCS will be code driven by eventual AI.
5. **Default Combat System**A game combat system will be chosen and implemented in a plug-replaceable fashion.
6. **Server System**A remote server will be implemented capable of supporting large numbers of players with some reasonable modicum of game state security.

From here many individual projects open up using the platform as a base

.

# Deployment and Future Work

Just like Neverwinter Nights I see this becoming both a platform for teaching and research. The codebase and assets will be open—sourced to encourage third party development participation.

Once the base system is completed there is an almost unlimited scope of experimental world simulation systems that could be built on top. Some areas I see of potential future work include AI/artificial life experimentation, immersive user interface and control research, and experimental game rule systems.

1. Dis: A Neverwinter Nights Mod, Corliss Poliquin and Urko 2009,  
    https://web.wpi.edu/Pubs/E-project/Available/E-project-030509-232047/unrestricted/MQP-DMO-4012.pdf [↑](#footnote-ref-1)
2. http://www.becker.edu/academics/departments-programs/design/interactive-media/faqs [↑](#footnote-ref-2)
3. . [↑](#footnote-ref-3)
4. https://unity3d.com/ [↑](#footnote-ref-4)