

DÜNYA ÜZERİNE YAYILMIŞ ÇOK-KULLANICILI ÇEVİRİM-İÇİ EĞİTSEL BİR BİLGİSAYAR OYUNUNUN TEKNİK OLARAK TAMAMLANMASI

Hakan Tüzün, Öğretim Görevlisi Dr.

Hacettepe Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi

Bölümü, 06532, Beytepe/ANKARA

htuzun@hacettepe.edu.tr

ÖZET

Günümüzde bilgisayar oyunları eğlenmeye ek olarak eğitim faaliyetleri için de kullanılan bir araç haline gelmiştir. Gerek örgün gerekse yaygın eğitim için kullanılan böyle bir eğitsel bilgisayar oyununa örnek Quest Atlantis'tir. Quest Atlantis 4 kıtada 7 ülkede yaklaşık 5000 çocuk tarafından kullanılmaktadır. Bu eğitsel oyunun teknik yapısı bir 3-Boyutlu grafik motorundan, 3-Boyut içerisindeki sanal dünyalar ve yapılardan, bir veritabanı yönetim sisteminden ve program kodlarından meydana gelmektedir. Bu makalede, sözü edilen teknik yapılar ve bunların çalışması, bu teknik yapıların birbirleri ile bütünleştirilmesi ve ortaya çıkan sorunlar ve bu sorunların giderilmesi için izlenen yöntemler ele alınacaktır.

ABSTRACT

In today's world, computer games have been used for education in addition to entertainment. Such an educational computer game, which is used for both in-school and after-school contexts, is Quest Atlantis. Quest Atlantis has been used by 5000 children on 4 continents in 7 countries. The technical infrastructure of this educational game includes a 3-

Dimensinal graphics engine, virtual worlds and structures in 3-Dimensional, a Database Management System, and program codes. In this article, these structures and their operations, integration of these structures, problems, and solutions to these problems will be covered.

Giriş

Quest Atlantis (QA, <http://www.QuestAtlantis.org>) eğitsel faaliyetlerde bulunmak üzere öğrenenleri 3-Boyutlu (3-B) paylaşılan sanal bir ortama çeken eğitsel bir bilgisayar oyunudur. Oyunda oyuncular efsanevi Atlantis şehrini yaklaşan bir felaketten kurtarmaya çalışmaktadırlar. Oyunun senaryosuna göre öğrenenler “Quest” olarak adlandırılan eğitsel faaliyetleri tamamlayarak Atlantis’in bu felaketten kurtarılmasına yardımcı olmaktadır.

Her ne kadar Quest Atlantis’i bir bilgisayar oyunu olarak tanımlasak da onu sadece bilgisayarda kullanılan basit bir yazılım olarak düşünmek uygun değildir. Quest Atlantis daha uygun bir şekilde dünya üzerine yayılmış bir oyuncu topluluğunu ve yerel toplulukları desteklemek üzere tasarlanmış sanal bir çevre olarak tanımlanabilir. Quest Atlantis senaryosu, sanal dünyaları ve kuralları Quest Atlantis meta-oyununu meydana getirir. Quest Atlantis meta-oyun bağlamı eğitsel faaliyetler ve QA katılımcı yapıları için bir şemsiye görevi görür.

Quest Atlantis’in hem İnternet üzerinden kullanılması hem de dünya üzerine yayılmış kullanıcıları ve yerel kullanıcıları desteklemek üzere tasarlanmış oyun içindeki katılımcı yapıları karmaşık bir teknik yapı gerektirmektedir. Bu karmaşık teknik yapı içerisinde yer alan oyunun 3-B grafik motoru, 3-Boyut içerisindeki sanal dünyalar ve yapılar, Perl ile dinamik olarak oluşturulan bilgiler ve MySQL veritabanı üzerinde tutulan

oyun ve öğrenci verileri ahenk içinde çalışarak oyunun teknik iskeletini meydana getirmektedir. Bu teknik iskelet şu ana dek dünya üzerindeki 5000 kadar kullanıcının eğitsel faaliyetlerini başarılı bir şekilde desteklemiştir. Bu makalede, sözü edilen teknik yapılar ve bunların çalışması, bu teknik yapıların birbirleri ile bütünleştirilmesi, ve ortaya çıkan sorunlar ve bu sorunların giderilmesi için izlenen yöntemler ele alınacaktır.

QA 3-B Grafik Motoru

Tasarımcılar Quest Atlantis oyununu tasarlarken projenin başlarında WebQuest türünde, eğitsel etkinliklerin 2-Boyutlu bir Web arayüzü üzerinden yapılacağı bir grafik kullanıcı arabirimi yapısını benimsemişlerdir. Bununla birlikte ortaya konan ilk tasarım prototiplerini kullanan çocuklar oyun ortamının bu 2-Boyutlu yapısından memnun kalmamış, ve 3-B bir oyun ortamının kendileri için daha çekici geleceğini ve böyle bir ortamın eğitsel etkinlikleri yapmak üzere motivasyonlarını daha çok artıracaklarını belirtmişlerdir. Şu andaki literatür de 3-B grafik motorlarını kullanan oyunları oynayarak büyüyen günümüz çocuklarının benzer tercihlerde bulunduklarını göstermektedir. Örneğin günümüz çocuklarına 70'lerin ve 80'lerin en popüler bilgisayar oyunları oynatıldığında bu çocuklar PacMan, Tetris ve Space Invaders gibi klasikleri yerden yere vurmuşlar ve bu oyunları sadece birkaç kere oynamaya dayanabileceklerini belirtmişlerdir.

Günümüzde birçok bilgisayar yazılımı bir 3-B grafik motorunu bünyesinde barındırmaktadır. Bu yazılımların başını ise bilgisayar oyunları çekmektedir. Her ne kadar bilgisayar oyunlarında 3-B ortamları kullanmak için değişik teknikler daha önceden denendiyse de (Atari platformundaki "Battlezone" oyunu gibi) gerçek anlamda 3-B bir grafik motorunu bünyesinde barındıran ilk bilgisayar oyunu "Doom"dur. Doom oyununun 3-B atmosferi bu oyunun popüler olmasında çok etkili olmuş, ve bundan sonra tasarlanan ve

finansal olarak başarı sağlamış birçok Birinci Tekil Şahıs Vurucu (First Person Shooter - FPS) türündeki oyun bir 3-B grafik motorunu standart olarak bünyesinde barındırmıştır (Doom serileri, Heretic serileri, Hexen serileri, Quake serileri, Unreal serileri, Tomb Raider serileri, Half-Life serileri ve Halo serileri gibi)

Bir 3-B grafik motoru bilgisayar ortamında ifade edilecek 3-B bir ortamın ortaya konulması ile ilgili yordamları içerir. 3-B grafik motoru bu yordamlar sayesinde 3-B nesnelerin içeriğinin belirli bir formatta ifade edilmesi, bu nesnelerin 3-B ortama aktarılması, yorumlanması, nesnelerin doku ile kaplanması ve oyun içerisindeki ışık kaynaklarının konumuna göre bu nesnelerin aydınlatılması gibi görevleri yerine getirir.

Günümüzde bir 3-B grafik motoru geliştirilmesi grafik işlemlerinin daha hızlı ve verimli yapılmasını sağlamak üzere karmaşık algoritmaların ve diğer tasarımların uygulanmasını gerektirdiğinden kendi içinde ayrıntılı bir teknik tasarım haline gelmiştir. Bu yüzden bir yazılımda 3-B bir grafik motoru kullanılacaksa bu grafik motorunun dışarıdan hazır alınarak kullanılması yöntemi ortaya çıkmıştır. Günümüzdeki Birinci Tekil Şahıs Vurucu türündeki oyunların çoğunluğu başka bir oyunun grafik motorunu kullanmakta ve hatta orijinal oyunun başarısını gölgede bırakmaktadır. Quake motorunu kullanan Half-Life oyunu bu duruma örnek olarak gösterilebilir.

Bir grafik motorunun dışarıdan hazır alınarak kullanılması pizza yaparken hazır hamur kullanılmasına benzetilebilir. Hazır hamur kullanarak pizza yaparken yapmamız gereken tek şey pizzanın üzerine konulacak malzemeleri hazırlamaktır. Bununla birlikte nasıl hazır hamur kullanırken bazı kriterleri gözönüne almak zorundaysak – kullanılacak hamurun büyüklüğü gibi – hazır bir 3-B grafik motoru kullanırken de gözönüne alınması

gereken birtakım kriterler vardır: Hazır olarak alınan motorun saniyede gösterilebileceği poligon sayısı ve bu motorun üzerinde çalışacağı bilgisayar platformlarının çeşitliliği gibi.

QA projesinin başlarında yapılan ürün testleri sonucunda bu eğitsel oyunu kullanacak temsilci çocuk grubunun 2-Boyutlu bir sanal ortam yapısından (Şekil 1) memnun kalmaması tasarımcıları yeni tasarım arayışlarına itmiştir. Bunun sonucunda sanal oyun dünyasını 3-B, içerisinde gezinilebilir bir ortama aktarma fikri benimsenmiştir. Daha önce de değinildiği gibi böyle 3-B bir ortamı kullanmak için geliştirilmesi gereken grafik motoru zaman ve kaynak gerektireceğinden hazır bir grafik motorunu kullanmak üzere potansiyel motorlar incelemeye alınmıştır. Yapılan analizler sonucunda ActiveWorlds (AW, <http://www.activeworlds.com/>) firmasının ürettiği sanal gerçeklik ortamının projenin ihtiyaçlarına cevap verebileceği görülmüştür (Şekil 2).



Şekil 1: Başlangıçta QA'in sanal ortamı için 2-Boyutlu bir Web arayüzü düşünülmüştür. Şekilde sol tarafta 2-Boyutlu oyun dünyası, sağ tarafta ise bir kullanıcıya ait kişisel sayfa görülmektedir.



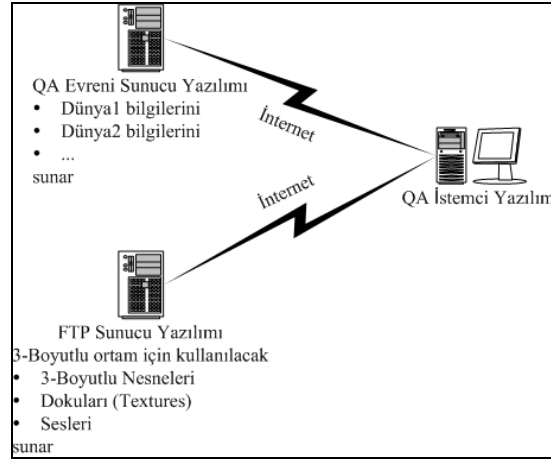
Şekil 2: Şekilde QA'in 3-B bir grafik motorunu kullanan versiyonu görülmektedir. Sol tarafta 3-B oyun dünyası, sağ tarafta ise bir kullanıcıya ait kişisel sayfa görülmektedir.

AW firmasının ürettiği ve İnternet'e bağlantısı olan herkese açık olan sanal gerçeklik ortamı Active Worlds (Aktif Dünyalar) olarak isimlendirilmektedir. Bu sanal ortam geniş bir sanal evrenden (universe) oluşmakta ve bu sanal evren de alt sanal dünyaları (virtual worlds) içermektedir. Kişiler bu sanal evrende bir sanal vatandaş hesabı (citizen) açarak diğerlerinin ortaya koydukları sanal dünyalar içerisinde gezebilecekleri gibi bu sanal evren içerisinde kendi sanal dünyalarını da açabilirler. Bu sanal dünyalar değişik amaçlar için kullanılabilir: Sanat yapıtlarının ortaya konması, gezintilerin düzenlenmesi, alışveriş, bilimsel verilerin incelenmesi, diğer insanlarla işbirliği yapılması, arkadaşlık etmek ve ticaret gibi. Eğitim alanındaki insanların da 3-B sanal dünyaları eğitsel amaçlarla kullanmak için girişimde bulunmalarından sonra Active Worlds kuruluşu eğitimcilerin artan taleplerine paralel olarak Active Worlds Eğitsel Evrenini (Active Worlds Educational Universe, AWEDU) kullanıma sunmuştur. Active Worlds Eğitsel Evreni bu tür teknolojiyi eğitim kurumlarının, öğretmenlerin ve öğrencilerin kullanımına sunan eğitsel amaçlı bir sanal dünyalar topluluğudur. Bu topluluk vasıtasıyla eğitimciler yeni kavramları, öğrenme teorilerini ve yaratıcı müfredat tasarımlarını araştırabilmekte ve sosyal öğrenme alanındaki yeni paradigmaları keşfedebilmektedirler.

AW firması bu sanal gerçeklik ortamını özel olarak AW evreninden ayrı olarak kullanmak isteyen kişi ya da kurumlara bu imkanı sunmaktadır. QA projesi daha çok 9-12 yaş arası çocukların kullandığı bir proje olduğundan dolayı bu projede kullanılacak sanal ortamın herkese açık olmayan özel bir ortam olması gereksinimi ortaya çıkmış, ve bu gereksinim de QA için ayrı bir AW sanal evreninin kullanılmasıyla sonuçlanmıştır.

AW sanal evreninin ve sanal dünyalarının teknik olarak çalışması istemci ve sunucu (client/server) esasına dayanır. Buna göre kullanıcılar AW sanal evrenindeki sanal

dünyaları dolaşmak için bir istemci yazılımı kullanırlar. Bu istemci bir 3-B grafik motorunu içermektedir. Bu motor 3-B nesnelerin yorumlanarak gösterilmesi, bu nesnelerin doku ile kaplanması ve sanal ortam içerisindeki ışık kaynaklarının konumuna göre bu nesnelerin aydınlatılması görevlerini yerine getirir. İstemci yazılımı İnternet üzerinden sanal evren sunucusuna bağlanır. Bu sanal evren sunucusu istemciye nesnelerin 3-B dünyalarda nerelere ve nasıl konulması gerektiği konusunda bilgiler iletir. 3-B ortamda kullanılacak nesneler, dokular ve sesler ayrı bir FTP (File Transfer Protocol, Dosya İletim Protokolü) sunucusundan çekilir. Bu dosyalar İnternet üzerinden istemciye aktarılacağından dosyaların daha küçük olması daha az aktarım zamanı gerektirecektir. AW teknolojisi bu işi daha verimli hale getirmek için ZIP ile sıkıştırılmış dosyaları kullanmaktadır. İstemci ZIP formatında sıkıştırılmış haldeki dosyaları aldıktan sonra bunları 3-B ortam içinde kullanmadan önce kendi içinde açar. İstemci, evren sunucusu ve FTP sunucusu arasındaki iletişim Şekil 3’te görülebilir.



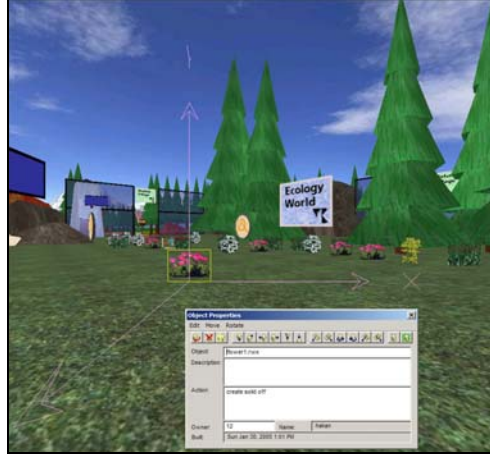
Şekil 3: İstemci, evren sunucusu ve FTP sunucusu arasındaki iletişim

QA Sanal Dünyaları ve Yapıları

QA 3-B sanal ortamı bir takım sanal dünyalardan oluşur. Bu sanal dünyalar çeşitli temalara göre gruplandırılmıştır: Ekoloji dünyası, kültür dünyası, birlik dünyası, sağlık

dünyası ve okyanus dünyası gibi. Her sanal dünya kendi içerisinde parçalara ayrılmış ve köy olarak isimlendirilmiştir. Örneğin ekoloji dünyası vahşi yaşam, su ve habitat köylerinden oluşur. Her bir köy içerisinde o köyün teması ile ilgili eğitsel etkinlikler yer alır. Bu dünyalarda kullanıcılar avatar olarak adlandırılan sanal karakterler tarafından temsil edilirler. Bu dünyalar çok-kullanıcılı sanal ortamlardır, yani farklı yerlerden QA 3-B sanal ortamına bağlanan kullanıcılar diğer kullanıcılarla gerçek zamanlı olarak etkileşimde bulunabilirler.

Bu dünyalar nesne-tabanlıdır ve yapımı programlama gerektirmez. Bir nesnenin ortaya konulması için ortamdaki başka bir nesnenin kopyalanması ve nesne isminin değiştirilmesi yeterlidir. Geriye kalan bu nesnenin ileri/geri, sağa/sola ve yukarı/aşağı hareket ettirilerek konumunun ve XYZ ekseninde döndürülerek pozisyonunun belirlenmesidir. Bu ayarlamalar nesneye ait diyalog penceresindeki basit kontroller yardımı ile yapılabilir (Şekil 4). Bu sanal ortam içerisinde yapıların oluşturulması aynı Lego parçaları ile yapılar oluşturulmasına benzer. Nesneler sanal ortamda biraraya getirilerek istenen yapılar ortaya konulur. AW teknolojisi içinde hazır olarak kullanılabilecek yüzlerce nesne gelmektedir. İhtiyaç duyulan bir nesne kütüphanede mevcut değilse 3-B grafik programları ile yapılabilir. Bu nesneler Renderware formatı olan RWX ya da TrueSpace formatı olan COB formatında olmalıdır, çünkü AW sanal ortamı sadece bu türde nesneleri kullanabilir. Değişik 3-B grafik programları ile üretilen diğer formatlardaki nesneler (örneğin 3D Studio ile üretilen 3DS formatındaki bir nesne) çeşitli dönüştürücü araçlar yardımı ile bu formatlara dönüştürülerek kullanılabilir.



Şekil 4: Bir çiçek nesnesinin Ekoloji dünyası içerisine konulması

Veritabanı Yönetim Sistemi ve Program Kodları

Her ne kadar QA'in sanal dünyaları ve bu dünyalar içerisinde yeralan yapılar bu oyunu oynayan çocukları eğitsel etkinlikleri tamamlama konusunda motive eden önemli bileşenler olsa da QA'in en az 3-B bileşeni kadar önemli olan diğer bileşenleri barındırdığı veritabanı yönetim sistemi ve program kodlarıdır.

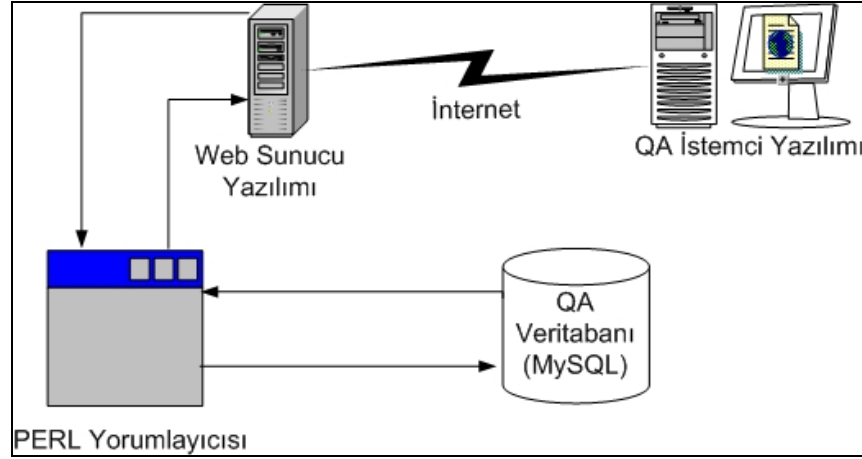
QA'i oyuncular için kalıcı ve sürekli bir deneyim haline getiren bileşen sahip olduğu veritabanı yönetim sistemi bileşenidir. QA veritabanında oyuncu verileri, sanal dünya verileri, Quest verileri, öğrencilerin bu eğitsel etkinliklere verdiği cevaplar ve buna benzer diğer birçok veri yeralır. QA için kullanılan veritabanı yönetim sistemi MySQL'dir. MySQL dünyanın en popüler açık kaynak kodlu veritabanı yönetim sistemidir. MySQL Google ve NASA'nın da dahil olduğu yaklaşık beş milyon web sitesinin ve diğer kritik iş sistemlerinin arkasındaki güçtür. MySQL bir ilişkisel veritabanı yönetim sistemidir, yani veriler tek bir büyük tablo yerine ayrı tablolarda tutulur. Bu özelliği ile MySQL QA verilerinin yaklaşık 100 ayrı tabloda tutulmasına ve tablolar arasında ilişkisel sorgulamalar yapılmasına olanak tanır. MySQL veritabanındaki veriler üzerinde işlem yapmak için Yapısal Sorgulama Dili'ni (Structured Query Language, SQL) kullanır. MySQL çoklu-

kullanıcılı (multi-user) bir veritabanı yönetim sistemidir, bu özellikleri ile birçok kullanıcı aynı anda QA veritabanı üzerinde işlemler yapabilir. MySQL çoklu iş-parçacıklı (multi-threaded) yapısıyla hız ve performans artışı sağlar.

QA veritabanındaki bu verilerin sorgulanarak oyunculara yansıtılması ya da kullanıcıların veritabanına yeni veriler eklemesi QA program kodları aracılığı ile yerine getirilir. QA program kodları için kullanılan programlama dili PERL'dir (Practical Extraction and Reporting Language). PERL özellikle metin verilerini işlemek üzere tasarlanmış bir programlama dilidir, bu yüzden QA içerisinde veri yığınlarının işlenmesi için uygun bir dildir. PERL, C gibi yüksek seviyedeki programlama dillerine göre avantaj sağlar, çünkü PERL kodlarının çalıştırılması için derlenmelerine gerek yoktur, bu kodlar bir yorumcu tarafından yorumlanırlar. Böylece bir kod değiştirildiğinde o kodun tekrar tekrar derlenmesine gerek kalmaz. QA gibi içerisinde yüzlerce kod barındıran bir ortam için bu durum zamandan büyük tasarruf sağlar. PERL programları içine yerleştirilen SQL komutları gerek verilerin QA veritabanından okunması için gerekse verilerin QA veritabanına yazılması için etkilişimli bir erişim sağlar.

QA bilgileri PERL kullanılarak uygun parametrelerin kullanılması ile (örneğin bir kullanıcıyı, bir Quest'i ya da bir cevabı temsil eden bir numara) dinamik olarak oluşturulur. QA'e ait program kodları ve veritabanı yönetim sistemi bir web sunucusunda tutulmaktadır. QA istemcisinde bilgiler gösterilirken (örneğin bir kullanıcıya ait kişisel sayfalar gösterilirken) olan iş bir PERL kodunun gerekli parametrelerle (örneğin kişisel sayfası gösterilecek kullanıcıya ait bir numara) çalıştırılmasının istemci tarafından web sunucusundan istenmesidir. Web sunucu yazılımı çalıştırılması istenen PERL kodunu depolandığı yerde bulur. Bu kod PERL yorumlayıcısı tarafından işlenir ve bu sırada

veritabanından gerekli veriler okunur. Veriler okunduktan sonra PERL yorumlayıcısı bir HTML sayfası üretir ve tekrar web sunucu yazılımına gönderir, web sunucu yazılımı da bu HTML sayfasını QA istemci yazılımına gönderir. QA istemcisi tarafından alınan bu dosya standart bir web sayfası olarak görüntülenir (Şekil 5).



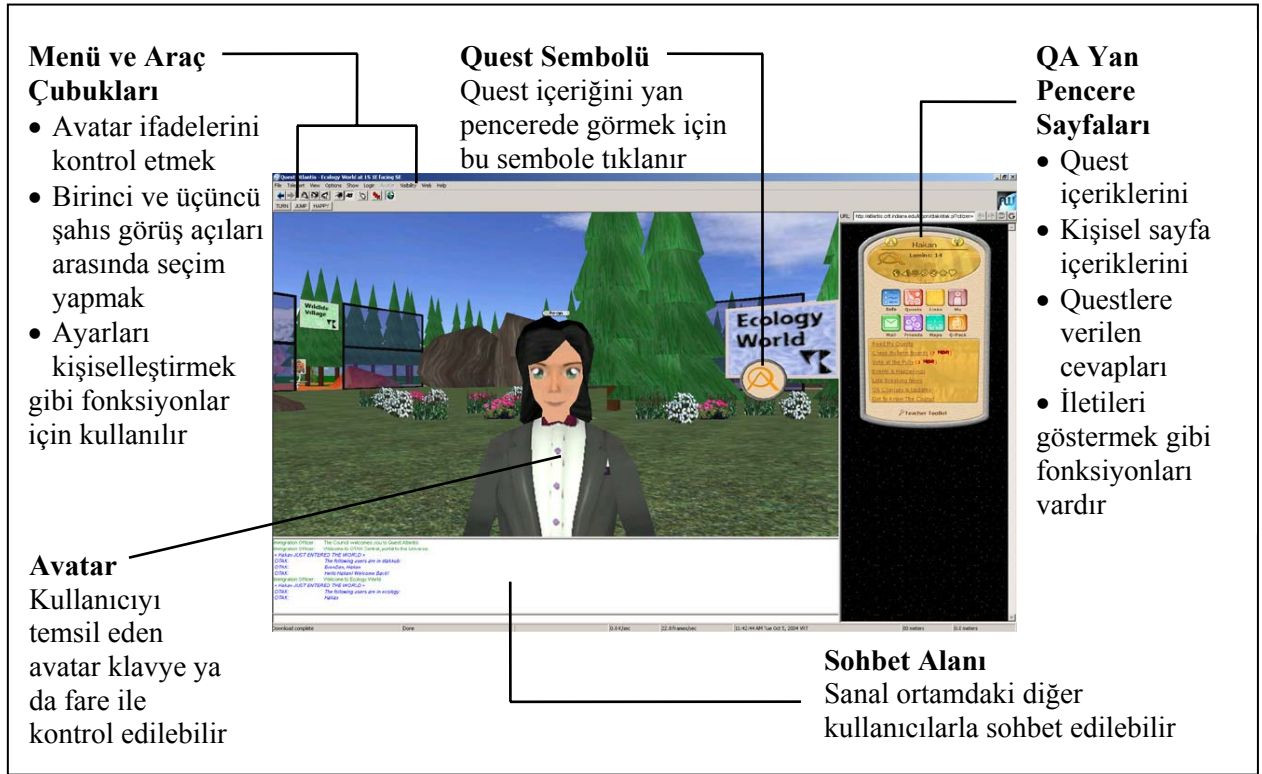
Şekil 5: Bir program kodunun çalışması

QA veritabanına veriler yazılırken de (örneğin bir Quest'e ait cevap) benzer işlemler yapılır. Yine bir PERL kodunun gerekli parametrelerle çalıştırılması istemci tarafından web sunucusundan istenir. Ek olarak bu istek yapılırken QA istemci yazılımında görüntülenen standart web sayfasında bulunan form alanlarındaki veriler de web sunucu yazılımına gönderilir. Perl yorumlayıcısı kodu çalıştırır ve form alanlarından toplanan verileri belirtilen tablolardaki alanlara yazmak için gerekli SQL komutlarını MySQL veritabanına iletir. Bu yazma ya da değiştirme işleminden sonra bir sonuç mesajı HTML formatında oluşturularak görüntülenmek üzere QA istemci yazılımına gönderilir.

QA Teknik Bileşenleri Entegrasyonu

AW tarafından üretilen sanal gerçeklik teknolojisini kullanmak için gerekli istemci standart olarak üç bileşeni bünyesinde barındırmaktadır: Bir 3-B grafik motoru ve grafik göstericisi, web sayfalarını göstermek için bir İnternet tarayıcısı ve bir sohbet aracı. QA

için bu teknoloji bir bütün olarak benimsenmiştir (Şekil 6). Öğrenenler QA sanal dünyaları ve yapıları içerisinde dolaşarak yapılacak etkinliklere ulaşırlar ve bu etkinlikleri tamamlamak için gerekli verileri sanal ortamda toplarlar. İsterlerse bu sanal ortama bağlı diğer kullanıcılarla yardımlaşma ve benzeri amaçlarla gerçek zamanlı olarak iletişimde bulunabilirler. Etkinliklere ait bilgiler ya da üretilen cevaplar istemciye bütünleşik İnternet tarayıcısında gösterilir.



Şekil 6: QA istemcisi arayüzü. Sol üstte sanal bir dünyanın bir parçası, sol altta sohbet penceresi ve sağda bir kullanıcıya ait kişisel sayfa görülmektedir.

Sorunlar ve Bu Sorunların Giderilmesi

Genel Sorunlar ve Bu Sorunların Giderilmesi

QA tasarımı için tasarıma-dayalı araştırma (Design-Based Research) (Brown, 1992; Collins, 1992) yöntemi benimsenmiştir. Bu yöntemin başlıca amacı doğal koşullardaki öğrenme ve öğretmeyi etkileyecek yeni kuramları ve uygulamaları ortaya koymaktır. Bu

anlamda bu yöntem değişik öğrenme formları tasarlamayı ve bu öğrenme formlarını çalışmayı gerektirir. Tasarlanan bağlam devamlı olarak tekrar eden testlere ve revizyonlara tabi tutulur. Bu yapısı ile tasarıma-dayalı araştırma yöntemi ADDIE gibi klasik tasarım yöntemlerinden çok farklıdır. Klasik tasarım yöntemlerinde bir ürüne ait tasarım ortaya konup geliştirildikten sonra test edilir, kullanıcılara sunulur ve tasarım tamamlanmış olur. Tasarıma-dayalı araştırma yöntemi izlenirken bir ürün devamlı olarak test ve revizyon aşamasındadır. Bundan dolayı QA tasarımı için organik bir süreçtir denilebilir. QA'e yeni katılan ve böyle bir organik süreçte çalışmaya alışmamış tasarımcılar ve programcılar bir bocalama evresi geçirmektedirler. Çünkü çoğu zaman yaptıkları tasarımın sonlandığını düşünmekte ve biten bir tasarımı değiştirmek bu yeni tasarımcılar ve programcılar için anlamsız gelmektedir. Zaman içerisinde bu tür takım üyeleri projenin kültürüne alışmakta ve tasarım sürecinde izlenen metoda ait test ve revizyon aşamalarını benimsemektedirler.

QA 3-B Grafik Motoru, Sanal Dünyaları ve Yapıları ile İlgili Sorunlar ve Bu Sorunların Giderilmesi

QA 3-B grafik motoru ve göstericisi istemci ve sunucu esasına göre çalışmaktadır. Bu istemci ve sunucu ağların ağı olarak adlandırılan İnternet üzerinden haberleşmektedirler. İstemci ve sunucu esasına göre çalışan sistemlerdeki başlıca sorunlardan birisi bilgi paketlerinin aktarılmasında yaşanan gecikme süresidir (latency). QA'de çalışan istemci ve sunucuların bir yerel alan ağından değil de İnternet üzerinden haberleştikleri gözönüne alınırsa bu gecikme süresi daha da artabilmektedir. Gecikme süresini azaltmak için AW teknolojisi içinde iki yöntem kullanılmaktadır.

AW teknolojisi içinde gecikme süresini azaltmak için kullanılan ilk yöntem 3-B grafik motoru tarafından kullanılan nesnelerin sıkıştırılmış olarak ZIP formatında

bulunmasıdır. Bu nesneler istemciye ulaştıktan sonra gösterilmeden önce istemci içerisinde açılmaktadır. Sıkıştırılmış formattaki nesnelerin boyutu azalacağından bu nesnelerin İnternet üzerinden aktarılması için geçen gecikme süresi de azalacaktır. Bu yöntem QA için büyük bir performans artışı sağlamaktadır, çünkü her bir QA sanal dünyası birçok nesne ve doku kaplaması içermektedir. Örneğin ekoloji dünyası yaklaşık 5000 adet nesnenin biraraya gelmesiyle oluşmaktadır.

AW teknolojisi içinde gecikme süresini azaltmak için kullanılan ikinci yöntem sanal ortam içerisinde kullanılan avatar, nesne, doku kaplaması ve ses gibi öğelerin önbelleklenmesidir (cache). Bu öğelere ait olan bir dosya istemci tarafından elde edildikten sonra kullanıcının hard diskine kaydedilmekte, ileri bir zaman ya da tarihte bu dosyanın tekrar kullanılması gerektiğinde istemci bu dosyayı tekrar çekmek yerine hard diskteki önbellekten kullanmaktadır. QA sanal dünyalarında kullanılan öğelerin yaklaşık 150MB yer kapladığı gözönüne alınırsa bu yöntem özellikle bant genişliği düşük olan kullanıcılar için gecikme zamanını büyük ölçüde düşürerek büyük bir performans artışı sağlar.

Her ne kadar QA'nın sanal oyun dünyası için hazır bir 3-B grafik motorunun kullanılması bazı sorunları ortadan kaldırırsa da diğer bazı sorunları beraberinde getirmiştir. Bu sorunların en büyüğü AW teknolojisine olan bağımlılıktır. QA istemcisinin 3-B parçasında yapılabilecek işler AW teknolojisinin sağladığı olanaklar ile sınırlıdır. Her ne kadar bu olanaklar AW geliştiricileri tarafından sürekli artırılsa da (örneğin, 1995'den bu yana geliştirilen AW özellikleri bu topluluğun üyelerinden birisi olan Mauz tarafından <http://mauz.info/awhistory.html> adresinde belgelendirilmiştir) QA içerisinde geliştirilmesi düşünülen bazı eğitsel parçalar 3-B grafik motorunun buna olanak vermemesinden dolayı kavramsal taslak düzeyinde kalmaktadır. Kavramsal taslak düzeyinde kalan önemli böyle

fikirlerden bir tanesi sanal dünya içerisinde fizik kanunlarına dayalı simülasyonların gerçekleştirilebilmesidir. Normal şartlar altında AW teknolojisinde sanal dünyadaki nesnelere gerçek dünya koşulları hakim değildir. Örneğin bir nesne nerede bırakılırsa orada kalır, yere düşmez. Dahası sanal dünya içerisindeki nesneler birbirleri ile etkileşemezler ve birbirlerini etkileyemezler (örneğin bir kovaya suyun doldurulması ya da bir topun diğer bir topa çarpması gibi). AW, kendi teknolojisine olan bağımlılığı bir dereceye kadar azaltmak için bir yazılım geliştirme kiti (Software Development Kit, SDK) geliştirmiştir. AW yazılım geliştirme kiti C programlama dili için bir kütüphanedir. Bu kütüphanedeki fonksiyonlar kullanılarak AW sanal ortamında etkileşimler ortaya konulabilir. AW yazılım geliştirme kiti çoğunlukla kendiliğinden hizmet sunan avatarların geliştirilmesi için kullanılmaktadır (örneğin bir sanal dünyaya girenleri karşılayan avatar ya da bir sanal dünyayı gezdiren avatar gibi). Bu yazılım geliştirme kiti sanal ortamda bir takım fizik olaylarını geliştirmek için de kullanılabilir. Bununla birlikte böyle bir iş için geliştirilmesi gereken algoritmalar, programlar v.b. görevler için gerekli zaman ve kaynaklar devasa boyuttadır ki, bu durum QA için hazır bir 3-B grafik motoru kullanılması amacı ile ters düşmektedir. İleride QA içerisinde fizik olaylarını kullanan simülasyonların kullanılması ihtimali gözönünde bulundurularak diğer 3-B grafik motorları devamlı olarak değerlendirilmektedir.

AW'in herkese açık olan sanal ortamına girebilmek için gerekli üyelik, AW evreni içerisinde bir sanal dünya açma ya da ayrı bir evrene sahip olma para gerektiren işlerdir. Ek olarak bunlardan herbirisi için yapılan lisanslandırma pek çok yazılımda olduğu gibi ömür boyu değil sınırlı bir süredir. Örneğin ayrı bir QA evren lisansı için ödenen ücret yıllıktır. Ek olarak bu evren içerisinde açılacak dünyalar için kullanılacak sanal alan miktarına göre

de ayrı bir ücretlendirme yapılmaktadır. QA içerisinde sık karşılaşılan bir sorun kullanılabilecek sanal alan miktarının bitmesidir. Böyle bir durumda ya bir miktar daha lisans ücreti ödenerek kullanılabilir sanal alan miktarı artırılmakta ya da o an kullanılmayan ya da geliştirme aşamasında olan sanal dünyaların çalıştırılması durdurularak üzerinde çalışılan dünyanın çalıştırılması sağlanmaktadır. Sanal evren sunucusu ile ilgili diğer bir sınırlama da bir dünyaya girebilecek toplam kullanıcı sayısıdır. QA için bu sınır başlangıçta 50 kullanıcıydı, fakat QA'in çok kullanıcı yapıları ile doğru orantılı olarak artan kullanıcı sayısı bu sınırın 300'e çıkarılmasını gerektirmiştir.

QA eğitsel etkinlikleri için bir sanal dünyayı ortaya koymak çok büyük kaynak ve çaba gerektiren bir süreçtir. Her ne kadar bu dünyalar nesne-tabanlı olsa ve yapımı programlama gerektirmese de süreçte yapılması gereken diğer görevler vardır: Sanal dünyanın kavramsal bir taslak olarak ortaya konması ve bu dünyada kullanılacak uygun nesnelerin bulunması ya da üretilmesi ve sanal dünya kütüphanesine yüklenmesi gibi. Bu görevlerin çeşitliliği ve karmaşıklığından dolayı QA geliştirme takımı içerisinde sadece sanal dünya geliştirme işi üzerinde çalışan 3 takım üyesi mevcuttur.

QA için bir sanal dünya geliştirilirken karşılaşılan en büyük sorunlardan birisi o sanal dünyada kullanılacak nesnelerin ya da avatarların kütüphanede olmayışıdır. AW teknolojisi 1995'den beri kullanılmaktadır ve bu teknoloji etrafında büyük bir kullanıcı topluluğu oluşmuştur. Bu kullanıcı topluluğu hem bu teknolojiyi kullanan kişileri hem de bu teknoloji için üretim yapan geliştiricileri içermektedir. Bu topluluktaki birçok geliştirici ürettiği birçok nesneyi ve avatarı ücretsiz olarak web sitelerinden sunmaktadır. QA sanal dünya geliştiricileri yeni bir dünya geliştirilmesi sözkonusu olduğunda ihtiyaç duyulan nesneleri bulmak için ilk olarak bu tür web sitelerinde nesne avına çıkmaktadır.

Bulunamayan nesne ya da avatarların geliştirilmesi için bu topluluktaki geliştiricilere başvurulabilmektedir. Örneğin QA içerisinde geliştirilen gergedan dünyasında avatar olarak kullanılmak üzere Afrika'da yaşayan hayvanlara ihtiyaç duyulmuştur, ve bu avatarların geliştirilmesi için bu topluluktaki bir geliştirici taşeron olarak kullanılmıştır.

QA dünyalarında sanal yapılar (örneğin binalar, fabrikalar, v.b.) ortaya konurken izlenebilecek bir yöntem küçük nesne parçalarının biraraya getirilmesidir. Örneğin bir evin ortaya konulması için kütüphanede mevcut olan duvarlar, pencereler, kapılar, çatılar, v.b. parçalar biraraya getirilerek sanal ortamda bir ev yapısı oluşturulabilir. Bununla birlikte sanal yapıların bu şekilde birçok parçadan oluşması çeşitli açılardan verimliliği düşürmektedir. Öncelikle bu sanal yapıyı oluşturan tasarımcı bu iş için çok zaman harcayacaktır, çünkü daha önce de belirtildiği gibi normal koşullar altında sanal dünyadaki nesneler birbirlerini etkilemediğinden yapının parçalarının konumunu ayarlamak zaman alan bir iştir (örneğin iki duvar nesnesinin birleşim noktalarının muntazam olarak yanyana getirilmesi gibi). Bu sorunu ortadan kaldırmak için kullanılan bir yöntem sanal yapıların tek parça olarak bir 3-B grafik programında tasarlanmasıdır. Bu şekilde tek parça olarak ortaya konan sanal yapılar kolayca bir sanal dünyaya yerleştirilebilir, hatta bu tür yapılar tek parça olduğundan sanal dünyadaki konumu çok parçadan oluşmuş yapılara göre çok daha kolay bir şekilde değiştirilebilir.

Veritabanı Yönetim Sistemi ve Program Kodları ile İlgili Sorunlar ve Bu Sorunların Giderilmesi

QA veritabanındaki veriler yaklaşık yüz kadar tabloda bulunan yüzlerce alana yayılmıştır, veritabanındaki bu tablolara erişim yapan yüzlerce QA program kodu bulunmaktadır. Bir QA program kodu diğer QA program kodları ile etkileşimde

bulunabilir. Veritabanı ve program kodları arasında gerçekleşen devasa boyutlardaki bu tür iletişimlerden dolayı programcılar kod yazarken bazı hataları öngörmeyebilirler. Dolayısı ile programcılar bir kod üzerinde çalışırken bir hata mesajı ile karşılaşılması olağan bir durumdur. Bir program kodu çalışmadığında QA İnternet tarayıcısında genel bir hata mesajı görüntülediğinden bu hata mesajına dayanarak sorunu çözmek neredeyse imkansızdır. Bunun için QA programcıları bir program kodu çalışmadığında, sunucuda duran ve kaydı PERL yorumlayıcısı tarafından tutulan hata kayıt dosyasını incelemektedirler. Bir kod çalışmadığında bu dosyaya bunun hangi kod olduğu ve kod içerisindeki hangi satırlarda hata olduğu kaydedilir. Böylece programcılar bir program kodundaki hatayı bulmak için onu satır satır incelemek yerine kayıt dosyasını inceleyerek hatanın tam olarak nereden kaynaklandığını görebilirler. Ek olarak program kodları çalışır görüldüğü halde ortaya çıkabilecek mantıksal hataların giderilmesi için QA yapıları ortaya konduktan sonra bu yapılar gerek programcılar, gerek QA tasarım ekibi, gerekse son kullanıcılar tarafından test edilmektedir.

PERL yorumlanan bir dil olduğu için bu dilde yazılan bir program kodunun çalıştırılması daha yüksek seviyede, derlenerek çalıştırılan program kodlarına göre daha yavaştır. Bir işi yaptıracak program kodu birçok şekilde yazılabilir. Bundan dolayı QA programcıları PERL program kodlarının yorumlanmasından dolayı oluşan zaman kaybını telafi etmek için gerek program kodlarını gerekse bu program kodları içerisinde bulunan ve veritabanında sorgulama yapan SQL komutlarını zaman içerisinde optimize ederek bu yordamların mümkün olan en kısa zamanda belirlenen işleri yapmalarını sağlamaya çalışmaktadırlar. Program yapılarının optimizasyonu programcılar ile tasarım ekibi arasında gerginliğe de neden olmaktadır. Çünkü QA tasarımı organik bir süreç olduğu için her

zaman eklenmesi ya da değiştirilmesi gereken programlama işleri vardır. QA tasarımcıları perspektifinden bakıldığında önemli olan bu programlama işlerinin mümkün olan en kısa zamanda yapılarak çalışır hale getirilmesidir. QA programcıları perspektifinden bakıldığında önemli olan ise bu programlama işlerinin çalışır hale getirilmesi, fakat aynı zamanda program yapılarının optimize halde olması ve mevcut optimize yapıyı bozmamış olmasıdır. QA tasarım süreci boyunca bir noktadan sonra QA tasarım ekibi baskın çıkmakta, ve kısa zamanda yapılan fakat optimize olmadığından dolayı yavaş çalışan program yapıları ortaya çıkmaktadır. Fakat bir süre sonra tasarım ekibi bu kodların verimsiz çalıştığından şikayet ederek programcılardan yardım istemekte, programcılar da yeni kod gelişimini ya da değişimini durdurarak mevcut yapıdaki program kodlarını ve veritabanı yapısını optimize etmektedirler. QA programcıları geliştirme süreci boyunca bu ‘DejaVu’ durumundan geçmektedirler.

Sonuç

Quest Atlantis, dünya çapındaki popülerliğini birçok faktöre borçludur. Bu faktörlerden birisi onu şu ana dek başarı ile ayakta tutan teknik iskelettir. Bu teknik iskelet bir 3-B grafik motorundan, 3-Boyut içerisindeki sanal dünyalar ve yapılardan, bir veritabanı yönetim sisteminden ve program kodlarından meydana gelmektedir. Bu makalede bu bileşenlerin çalışması, bu bileşenlerin birbirleri ile bütünleştirilmesi ve ortaya çıkan sorunlar ve bu sorunların giderilmesi ele alınmıştır. Her ne kadar bazı bileşenler QA’ye özgü gibi görünse de herhangi bir eğitsel bilgisayar oyunu hatta bir bilgisayar oyunu geliştirilirken benzer bileşenler kullanılacak, bu bileşenler biraraya getirilecek, ve bunlarla ilgili birtakım sorunlar yaşanacaktır. Bilgisayar oyunu ve eğitsel bilgisayar oyunu geliştirilmesinin çok popüler hale geldiği günümüzde bu makalenin özellikle teknik açıdan

gerek bilgisayar oyunlarının gerekse eğitsel bilgisayar oyunlarının geliştirilmesine katkıda bulunacağı umulmaktadır.