

TÜRKİYE'DE E-ÖĞRENME Gelişmeler ve Uygulamalar



Editörler
Gonca TELLİ YAMAMOTO
Uğur DEMİRAY
Mehmet KESİM

TÜRKİYE'DE E-ÖĞRENME: **Gelişmeler ve Uygulamalar**

Editörler

Gonca Telli YAMAMOTO
Uğur DEMİRAY
Mehmet KESİM

Ankara-2010

Türkiye’de e-Öğrenme

1. İnternet (Bilgisayar ağı), Eğitimde 2. Etkileşimli
multimedya 3. Uzaktan öğretim 4. Eğitim teknolojisi
I. Telli Yamamoto, S. Gonca II. Demiray, Uğur III.
Kesim, Mehmet.
LB1044 .87 .T87 2010

ISBN: 978-605-88891-2-5

2010

ISBN 978-605-88891-2-5

Kapak Tasarımı: Gökhan KÜLE
Dizgi ve Sayfa Düzeni: Ahmet KİREZ
Baskı: Cem Web Ofset, Ostim, Ankara, 2010-12-01
Tel: 0312 385 37 27

© Uğur Demiray-2010

İÇİNDEKİLER

ÖNSÖZ.....	vi-ix
GİRİŞ.....	xi-xiii
EDİTÖRLERE İLİŞKİN.....	xv-xvi
YAZARLAR LİSTESİ.....	xvii
AYRINTILI İÇERİK.....	xix-xxxiii

E-ÖĞRENME GELEN DEĞERLER DİZİSİ (PARADİGMA) DEĞİŞİMLERİ VE EKONOMİK BOYUTU

BÖLÜM-1 Uzaktan Eğitimde Meydana Gelen Değerler Dizisi (Paradigma) Değişimlerinin E-Öğrenme Ekonomisi Alanına Yansımaları Eren KESİM, Anadolu Üniversitesi, Eskişehir.....	3-25
BÖLÜM-2 Uzaktan Eğitim Sistemine Geçişin Esas Öğeleri Olan, Öğretim Elemanları ve Öğrencilerinin Bakış Açıları Fazlı YILDIRIM, Okan Üniversitesi, İstanbul.	27-48

E-ÖĞRENME TASARIMLANMASI

BÖLÜM-3 E-Öğrenme ve Etkileşimli Ortam Tasarımı, Zafer GÜNEY, GATA, İstanbul.....	51-82
BÖLÜM-4 E-Öğrenme Programı Tasarım Süreçleri, Birim BALCI, Okan Üniversitesi, İstanbul.....	83-110

BÖLÜM-5	
E-Öğrenme ve Bilgi Teknolojileri Yönetimi	
Murat Paşa UYSAL, Kara Harp Okulu, Ankara.....	111-127

E-ÖĞRENME'DE ÖRNEKLER

BÖLÜM-6	
E-Öğrenme Sistemlerinde Verimlilik	
Artırımının Gözlenmesi: Afyon Kocatepe Üniversitesi Örneği	
Ömer DEPERLİOĞLU, Afyon Kocatepe Üniversitesi, Afyon	
Ertuğrul ERGÜN, Afyon Kocatepe Üniversitesi, Afyon	
Utku KÖSE, Afyon Kocatepe Üniversitesi, Afyon.....	131-161

BÖLÜM-7	
Kuzey Kıbrıs Türk Cumhuriyetinde E-öğrenme	
Murat HİSMANOĞLU, Lefke, KKTC.....	163-183

BÖLÜM-8	
PHP GD Kütüphanesi Kullanılarak	
Etkileşimli Uygulama Geliştirme	
Şevket YILMAZ, Pamukkale Üniversitesi, Denizli	
İsmail SARI, Pamukkale Üniversitesi, Denizli.....	185-202

BÖLÜM-9	
Eğitim İçerikleri Hazırlamada İnteraktif Uygulamalar	
Serdar ÖZKAŞ, Kliksoft, İstanbul	203-236

BÖLÜM-10	
Turizmde E-Öğrenme,	
Seha AKSÜ, Okan Üniversitesi, İstanbul.....	237-259

BÖLÜM-11	
Dünya Üzerine Yayılmış Çok-Kullanıcı Çevrim-İçi Eğitsel Bir	
Bilgisayar Oyununun Teknik Yapısı Ve Türkiye'de Yaklaşımlar	
Hakan TÜZÜN, Hacettepe Üniversitesi, Ankara.....	261-281

BÖLÜM-12

**Başarılı Bir E-Öğrenme Sisteminin Kurulmasına
ve Sürekliliğinin Sağlanmasına Yönelik Farkındalık
ve Tutum Ölçümü: Durum İncelemesi**

**Damla YILDIRIM, Ilmenau Teknik Üniversitesi, Almanya
Paul KLIMSA, Ilmenau Teknik Üniversitesi, Almanya.....283-294**

BÖLÜM-13

**K-12 Seviyesinde E-Öğrenmenin
Türkiye'deki Yapılabilirliği ve Öneriler**

Ahmet BAYTAK, Erciyes Üniversitesi, Kayseri.....295-314

BÖLÜM-14

K12 Düzeyinde E-Öğrenme Örnekleri

Ülkü GÜRSOY, İstanbul.....315-328

BÖLÜM-15

**Üstün Yetenekli Öğrencilere Yönelik Matematik
Öğretim Programlarında E-Öğrenmenin Yeri**

Gülşah Batdal KARADUMAN, İstanbul Üniversitesi, İstanbul..329-344

E-ÖĞRENME BİLGİ HİZMETLERİ

BÖLÜM-16

**Uzaktan Eğitim Sisteminde E-Öğrenme Süreci Bilgi Güvenliği,
Akın MARŞAP, Aydın Üniversitesi, İstanbul**

Ebru YILDIRIM, Uludağ Üniversitesi, Bursa

Gizem AKALP, Uludağ Üniversitesi, Bursa.....347-373

BÖLÜM-17

E-Öğrenme Sistemlerinde

Ölçme Değerlendirme ve Sınav Uygulamaları

Hakan Güray ŞENEL, Anadolu Üniversitesi, Eskişehir375-412

E-ÖĞRENME AÇILIMLARI

BÖLÜM-18

IPTV: Uzaktan Görsel (Tele Visual)

Öğrenme ve Türkiye İçin Fırsatlar

Yasin ÖZARSLAN, Osmangazi Üniversitesi, Eskişehir.....415-435

BÖLÜM-19

Mobil Öğrenme Teknolojileri ve Eğitim Uygulamaları

Gonca TELLİ YAMAMOTO, Okan Üniversitesi, İstanbul

Özlem OZAN, Osmangazi Üniversitesi, Eskişehir

Uğur DEMIRAY, Anadolu Üniversitesi, Eskişehir.....437-464

BÖLÜM-20

E-Öğrenmede Başarı Faktörleri,

Birim BALCI, Okan Üniversitesi, İstanbul.....465-480

E-ÖĞRENMEDEN MOBİL ÖĞRENMEYE

BÖLÜM-21

E-Öğrenmeden Mobil Öğrenmeye Teknoloji Yol Haritası

Murat Paşa UYSAL Kara Harp Okulu, Ankara.

Yavuz GAZİBEY, Kara Harp Okulu, Ankara.....483-497

BÖLÜM-22

E-Öğrenmeye Bakış: TSK Örneği

M. Sabri ŞEKEREFELİ, Bilkent Ankara.....499-508

BÖLÜM-23

Türkiye’de E-Öğrenmenin Bugünü ve Geleceği

Ahmet HANÇER. Enocta, İstanbul.....509-526

SON SÖZ.....527-533

BÖLÜM-11

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

Yard. Doç. Dr. Hakan TÜZÜN
Hacettepe Üniversitesi, Ankara
htuzun@hacettepe.edu.tr

ÖZET

Günümüzde bilgisayar oyunları eğlenmeye ek olarak eğitim faaliyetleri için de kullanılan bir olgu haline gelmiştir. Gerek örgün gerekse yaygın eğitim için kullanılan böyle bir eğitsel bilgisayar oyununa örnek Quest Atlantis'tir. Quest Atlantis 6 kıtada yaklaşık 50.000 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 çalışmada, 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. İnternet üzerinden çalışan uygulamaların ve dağıtık sistemlerin geliştirilmesinin çok popüler hale geldiği günümüzde bu çalışmanın özellikle teknik açıdan benzer e-öğrenme uygulamalarının geliştirilmesine katkıda bulunacağı düşünülmektedir.

GİRİŞ

Quest Atlantis (QA, <http://www.QuestAtlantis.org>) kullanıcıların eğitsel faaliyetlerde bulunma amacı ile 3-Boyutlu (3-B) sanal bir ortama girdiği ve diğer kullanıcılarla gerçek zamanlı olarak etkileştiği bir oyun ortamıdır. Oyunda oyuncular efsanevi Atlantis şehrini yaklaşan bir felaketten kurtarmaya çalışmaktadır. Oyunun senaryosuna göre öğrenenler "Quest" olarak adlandırılan eğitsel faaliyetleri tamamlayarak Atlantis'in bu felaketten kurtarılmasına yardımcı olmaktadır.

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 50.000 kadar kullanıcının eğitsel faaliyetlerini başarılı bir şekilde desteklemiştir.

Başka çalışmalarda Quest Atlantis oyununun pedagojik yönden yapılandırıldığı kuramsal çerçeve, oyunun eğitim, eğlence ve toplumsal sorumluluk boyutları, eğitsel oyunun eğitim ortamlarında uygulama süreci (Barab, Thomas, Dodge, Carteaux ve Tuzun, 2005; Tüzün, 2006;) ve Türkiye'de çeşitli uygulama ortamlarındaki uygulama sonuçları aktarılmıştır (Bakar, Tüzün ve Çağıltay, 2008; Bayırtepe ve Tüzün, 2007; Tüzün, 2007; Tüzün, Arkun, Bayırtepe-Yağız, Kurt ve Yermeydan-Uğur, 2008; Tüzün, Yılmaz-Soylu, Karakuş, İnal ve Kızılkaya, 2009).

Bu çalışmada ise, yukarıda 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. E-öğrenme tasarımı için farklı teknik yapılar ve yöntemler kullanılabilir. Bu bağlamda bölümün amacı çeşitli teknik yapıları ya da yöntemleri tanıtmak değil, e-öğrenme için kullanılan eğitsel bir bilgisayar oyununun teknik yapısını ve bu teknik yapıyı ortaya koymak için izlenen yöntemi tanıtmaktır.

YÖNTEM VE ARAÇLAR

Bu çalışmada biçimlendirici araştırma (formative research) tekniği kullanılmıştır. Biçimlendirici araştırma bir tasarım kuramıdır, yani amacı öğretimsel uygulamaları tasarlamak ya da iyileştirmek için gerekli modelleri ve ana noktaları ortaya koymaktır (Reigeluth, 1999). Biçimlendirici araştırma “Ne işe yaradı”, “Ne işe yaramadı” ve “Hangi iyileştirmeler yapılabilir” soruları rehberliğinde yürütülür. Belirli bir tasarımı ortaya koyarken izlenen uygulama sürecin güçlü ve zayıf taraflarını gösterir. Vakada izlenen sürecin ortaya konması ile uygulama süreci diğer uygulamalarda kullanılmak üzere geliştirilebilir. Reigeluth ve Frick (1999) tasarım kuramlarının üç niteliğe sahip olması gerektiğini belirtir:

Etkililik, verimlilik ve cazibe. Etkililik hedefe ulaşmayı ya da işi doğru yapmayı gerektirir. Verimlilik sonuçlara en uygun kaynaklarla hedefe ulaşmayı gerektirir. Cazibe ise sürecin tasarımcılar ve kullanıcılar için sevilme derecesidir.

Biçimlendirici araştırma tekniğinin alt yöntemleri vardır. Bu araştırma kapsamında tasarım vakası (designed case) alt yöntemi kullanılmıştır. Bu yöntemde zamana bağlı olarak aşağıdaki sıra izlenir:

- Tasarım için bir vakanın ortaya konması
- Vakaya ait biçimsel verilerin toplanıp analiz edilmesi
- Vakanın gözden geçirilerek geliştirilmesi
- Veri toplama ve düzeltme süreçlerinin tekrar edilmesi
- Tecrübelerin ortaya konması

Quest Atlantis eğitsel oyununun teknik yapısının ortaya konulmasında bu adımlar izlenmiştir. Ortaya konulan ya da üzerinde çalışılan tasarım devam etmekte olduğundan bulgulara bahsedilen teknik yapılar da sürekli gelişim halindedir.

Bu çalışmada yazarın kendisi Quest Atlantis araştırma ve geliştirme ekibinin bir üyesi olduğundan verilerin toplanması ve analizi için başlıca araç olmuştur. 2002 ile 2006 yılları arasındaki veri kaynakları bu çalışma dahilinde incelenmiştir. Quest Atlantis tasarımı için yapılan haftalık takım toplantılarında alınan notlar, tasarım ekibi üyeleri arasında yazılan e-posta mesajları ve yazarın gerek A.B.D.'deki gerekse Türkiye'deki çeşitli okullarda yaptığı Quest Atlantis uygulamalarındaki gözlem raporları bu araştırma için temel veri kaynağı olmuştur.

BULGULAR VE TARTIŞMA

Veri kaynaklarının analizi sonucunda Quest Atlantis eğitsel oyununun teknik yapısının 3 tematik alan altında gruplandığı ortaya çıkmıştır: QA 3-B grafik motoru, QA sanal dünyaları ve yapıları, ve veri tabanı yönetim sistemi ve program kodları. Aşağıda bu tematik alanlar ele alınmaktadır.

QA 3-B Grafik Motoru

Tasarımcılar Quest Atlantis oyununu tasarlarken projenin başlarında WebQuest (Dodge, 1995) 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 bulduklarını 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 (McGraw, 2003).

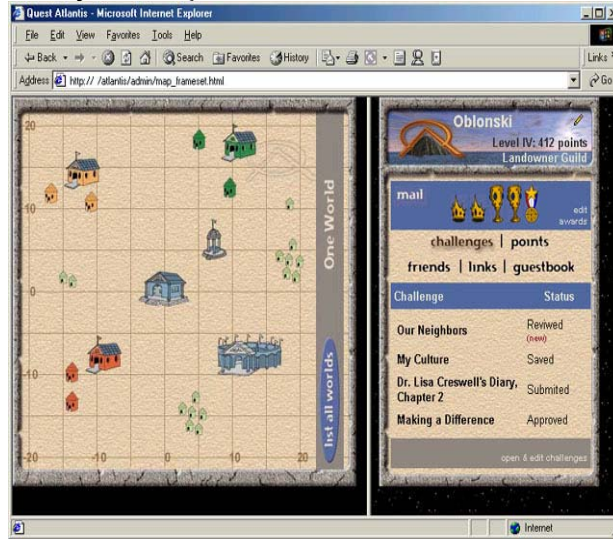
Günümüzde birçok bilgisayar oyunu bir 3-B grafik motorunu bünyesinde barındırmaktadır. Bu yazılımların başını ise bilgisayar oyunları çekmektedir. Her ne kadar bilgisayar oyunlarında 3-B ortamları kullanmak için değişik teknikler daha önceden denendiyse de (Atari platformundaki "Battlezone" oyunu gibi) 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 aktarılarak yorumlanması, nesnelerin doku ile kaplanması ve oyun içerisindeki ışık kaynaklarının konumuna göre bu nesnelerin aydınlatılması gibi görevleri yerine getirir (Hearn ve Baker, 1986).

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 ölçütleri 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.



Şekil 1.

Başlangıçta QA'nın 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.

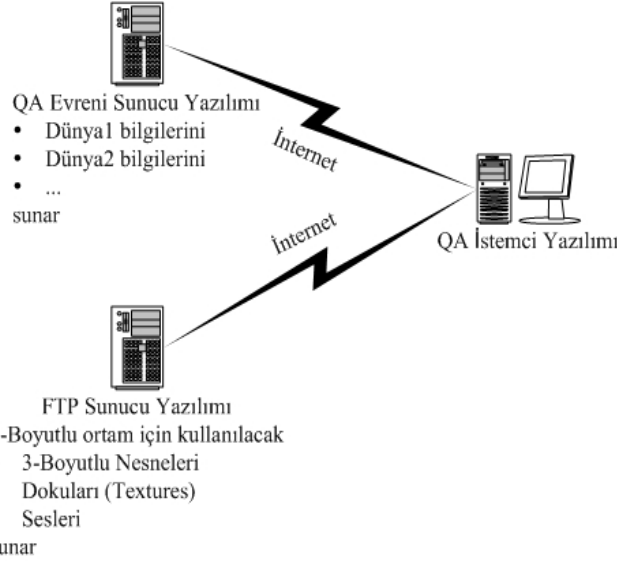
QA'nın 3-B bir grafik motorunu kullanan versiyonu görülmektedir. Sol tarafta 3-B oyun dünyası, sağ tarafta ise bir kullanıcıya ait kişisel sayfa görülmektedir.

AW firmasının ürettiği ve İnternet'e bağlantısı olan herkese açık olan sanal gerçeklik ortamı Aktif Dünyalar (Active Worlds) olarak isimlendirilmektedir. Bu sanal ortam geniş bir sanal evrenden (virtual 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 gezintilerin düzenlenmesi, alışveriş, bilimsel verilerin incelenmesi, diğer insanlarla işbirliği yapılması, arkadaşlık etmek ve ticaret gibi değişik amaçlar için kullanılabilir. Eğitim alanındaki akademisyenlerin 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 keşfedebilmektedir.

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

AW sanal evreninin ve dünyalarının teknik olarak çalışması istemci/sunucu (client/server) esasına dayanır. Buna göre kullanıcılar AW evrenindeki dünyaları dolaşmak için bir istemci yazılımı kullanırlar. Bu istemci bir 3-B grafik motorunu içermektedir. 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 nesnelere, dokular ve sesler ayrı bir FTP (File Transfer Protocol, Dosya İletim Protokolü) sunucusundan çekilir. İstemci, evren sunucusu ve FTP sunucusu arasındaki iletişim Şekil 3'te görülebilir.

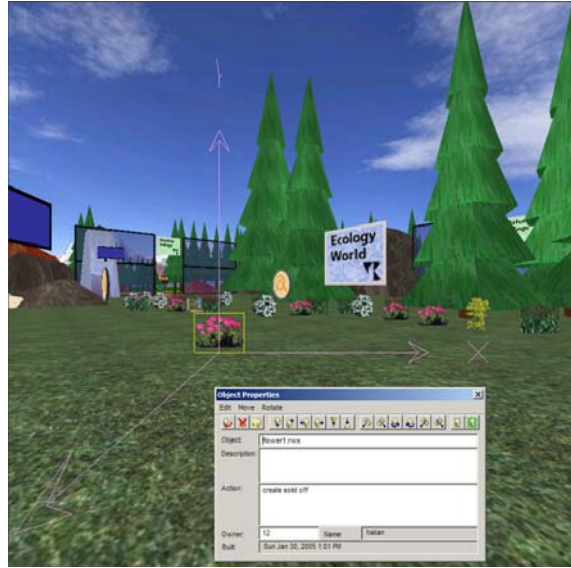


Şekil 3.
İstemci, evren sunucusu ve FTP sunucusu arasındaki iletişim

QA Sanal Dnyaları Ve Yapıları

QA 3-B sanal ortamı bir takım sanal dnyalardan oluşur. Bu sanal dnyalar çeşitli temalara göre gruplandırılmıştır: Ekoloji dnyası, kültür dnyası, birlik dnyası, sağlık dnyası ve okyanus dnyası gibi. Her sanal dünya kendi içerisinde parçalara ayrılmıştır, örneğin ekoloji dnyası vahşi yaşam, su ve habitat köylerinden oluşur. Her bir köy içerisinde o köyün teması ile ilgili eğitsel etkinlikler yer alır. Bu dnyalarda kullanıcılar avatar olarak adlandırılan sanal karakterler tarafından temsil edilirler. Bu dnyalar çok-kullanıcılı sanal ortamlardır, yani farklı yerlerden QA 3-B sanal ortama bağlanan kullanıcılar diğer kullanıcılarla gerçek zamanlı olarak etkileşimde bulunabilirler.

Bu dnyalar 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.



Şekil 4.

Bir çiçek nesnesinin Ekoloji dnyası içerisinde konulması

Nesneler sanal ortamda biraraya getirilerek istenen yapılar ortaya konulur. AW teknolojisi içinde hazır olarak kullanılacak 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 nesnelere 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.

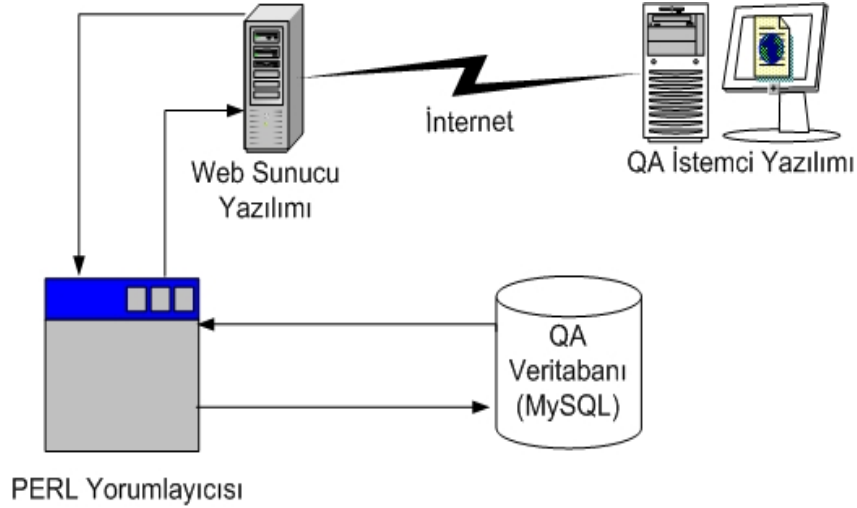
Veritabanı Yönetim Sistemi Ve Program Kodları

Her ne kadar QA'in sanal dünyaları ve bu dünyalar içerisinde yer alan 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, oyuncular için kalıcı ve sürekli bir deneyimdir, yani oyunculara ait tüm kişisel veriler ve etkinlik verileri bir oturumdan diğerine taşınmaktadır. Bu sürekliliği sağlayan QA içerisindeki 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 yer alır. QA için kullanılan veritabanı yönetim sistemi MySQL'dir. MySQL dünyanın en popüler açık kaynak kodlu veritabanı yönetim sistemidir (Sheldon ve Moes, 2005). 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 (Hansen ve Hansen, 1996). 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-kullanıcı (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 (Brown, 2001), 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 derlenmesine gerek kalmaz. QA gibi içerisinde yüzlerce kod barındıran bir ortam için bu durum programcılar için zamandan büyük tasarruf sağlar. Perl programları içine yerleştirilen SQL komutları verilerin QA veritabanından okunması ve QA veritabanına yazılması için etkileşimli bir erişim sağlar. QA bilgileri Perl kullanılarak uygun parametrelerin kullanılması ile 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 sayfa 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.

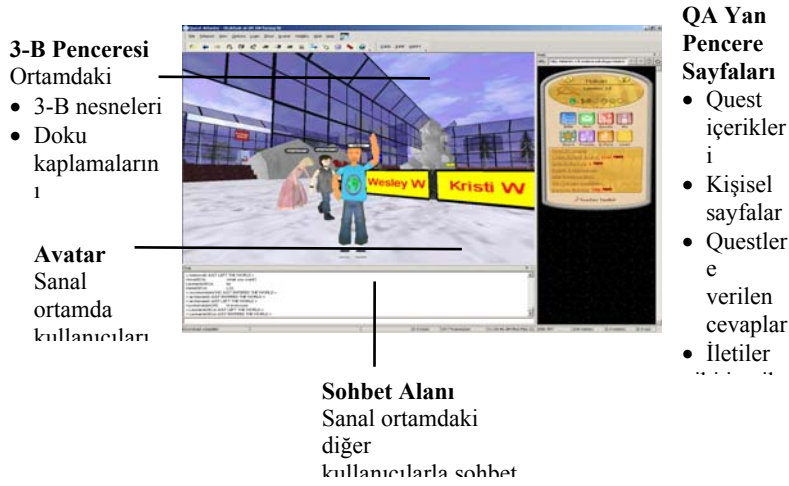


Şekil 5.
Bir program kodunun çalışması

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). QA veritabanına veriler yazılırken de (örneğin bir Quest'e verilen 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).



Şekil 6.

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

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 bütünlük İnternet tarayıcısında gösterilir.

SORUNLAR VE BU SORUNLARIN GİDERİLMESİ

Genel Sorunlar ve Bu Sorunların Giderilmesi

QA tasarımı için tasarıma-dayalı araştırma yöntemi (Design-Based Research) (Brown, 1992; Collins, 1992) benimsenmiştir. Bu yöntemin başlıca amacı doğal koşullardaki öğrenme ve öğretmeyi etkileyecek yeni kuramları ve uygulamaları ortaya koymaktır. Bu anlamda bu yöntem değişik öğrenme biçimleri tasarlamayı ve bu öğrenme biçimlerini çalışmayı gerektirir. Tasarlanan bağlam devamlı olarak tekrar eden testlere ve yenilenmelere tabi tutulur. Bu yapısı ile tasarıma-dayalı araştırma yöntemi ADDIE (Analiz, Tasarım, Geliştirme, Uygulama ve Değerlendirme) (Molenda, Pershing, & Reigeluth, 1996) 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 sürekli olarak test ve yenilenme 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 geçirmektedir.

Çünkü çoğu zaman yaptıkları tasarımın sonlandığını düşünmekte ve biten bir tasarımı değiştirmek bu yeni tasarımcılar ve programcılar için anlamsız ve zor gelmektedir. Zaman içerisinde bu tür takım üyeleri projenin kültürüne alışmakta ve tasarım sürecinde izlenen yöntemle ait test ve yenilenme aşamalarını benimsemektedir.

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

QA 3-B grafik motoru ve göstericisi istemci/sunucu esasına göre çalışmaktadır. Bu istemci ve sunucu ağların ağı olarak adlandırılan İnternet üzerinden haberleşmektedir. İstemci ve sunucu esasına göre çalışan sistemlerdeki başlıca sorunlardan birisi bilgi paketlerinin aktarılmasında yaşanan gecikme süresidir (latency) (Jenkins ve Schatt, 1995). QA'de

çalışan istemcilerin 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 nesnelere 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 (caching). 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 içerisindeki bir sanal dünyada kullanılan öğelerin toplam boyutunun 150MB'a kadar çıkabildiği 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'ın 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 artırılrsa 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 modüller 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 hâkim değildir. Örneğin bir nesne havada bırakılırsa orada kalır, yere düşmez.

Dahası sanal dünya içerisindeki nesnelere 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 bunların 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 kullanılacak 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 evrene aynı anda girebilecek toplam kullanıcı sayısıdır.

QA için bu sınır başlangıçta 50 olmasına rağmen 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 söz konusu olduğunda ihtiyaç duyulan nesnelere 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 duyulmuş 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 nesnelere 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 öngörmeyebilir. 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 olanaksızdır.

Bunun için QA programcılarını bir program kodu çalışmadığında, sunucuda duran ve kaydı Perl yorumlayıcısı tarafından tutulan hata kayıt dosyasını incelemektedir. 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 görebilir. 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. Ayrıca bir işi yaptıracak program kodu birçok şekilde yazılabilir.

Bunlardan dolayı QA programcılarını 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 çalışmaktadır. 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 bozmuş 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 şikâyet ederek programcılardan yardım istemekte, programcılar da yeni kod gelişimini durdurarak mevcut yapıdaki program kodlarını ve veritabanı yapısını optimize etmektedir. QA programcıları geliştirme süreci boyunca bu 'DejaVu' durumundan geçmektedir.

SONUÇ VE ÖNERİLER

Quest Atlantis'in dünya çapında birbirinden farklı ortamlarda başarı ile uygulanmasını sağlayan 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 çalışmada bu bileşenlerin çalışması, bu bileşenlerin birbirleri ile bütünleştirilmesi ve ortaya çıkan sorunlar ve bu sorunların giderilmesi ele alınmıştır.

Her ne kadar bazı bileşenler QA'e özgü gibi görünse de İnternet üzerinden çalışan bir bilgisayar oyunu, benzer bir dağıtık sistem ya da e-öğrenme

sistemi geliştirilirken benzer bileşenler kullanılacak, bu bileşenler biraraya getirilecek ve bunlarla ilgili birtakım sorunlar yaşanacaktır. İnternet üzerinden çalışan uygulamaların ve benzer dağıtık sistemlerin geliştirilmesinin çok popüler hale geldiği günümüzde bu çalışmanın özellikle teknik açıdan benzer e-öğrenme uygulamalarının geliştirilmesine katkıda bulunacağı düşünülmektedir.

Çalışmada ortaya konulan bulgulara göre aşağıdaki sonuç ve öneriler çıkarılabilir:

- Çok-kullanıcı, dağıtık, 3-B perspektifi bünyesinde barındıran ve çoklu ortam öğeleri yönünden zengin bir eğitsel bilgisayar oyunu tasarımı zaman, kaynak ve nitelikli insan gücü gerektiren bir süreçtir.
- Bilgisayar oyunları çeşitli ve karmaşık teknik yapıları içerdiğinden öğrencilerin beklentilerine hitap edecek eğitsel bilgisayar oyunlarını ortaya koymak teknik açıdan bir takım çalışmasını gerektirmektedir.
- Böyle bir takımın üyeleri arasında güçlü bir iletişim olmalıdır. Bunu sağlamak üzere teknik ekip fiziksel olarak birbirine yakın bulunmalıdır.
- Eğitim ortamlarında kullanılan bilgisayar oyunları gibi karmaşık teknoloji uygulamalarının tasarımında ADDIE gibi klasik tasarım yöntemleri yetersiz kalmaktadır. Bu tür ortamların geliştirilmesinde tasarıma-dayalı araştırma yöntemi gibi bu tür teknolojilerin hem tasarımına hem de araştırılmasına katkıda bulunacak güncel yöntemler kullanılabilir.
- Günümüz çocukları için tasarlanan eğitsel bilgisayar oyunu ortamları 3-B perspektifi bünyesinde barındırmalıdır.
- 3-B perspektif bir oyun için her ne kadar gerekli olsa da tek başına yeterli değildir. Bu ortamı kullanışlı hale getirmek onu veri ile desteklemekten geçer. İlişkisel bir veritabanı yönetim sistemi ve bu veritabanından verileri okuyup yazabilecek uygun bir programlama dili bu amaç için kullanılabilir.
- Perl ve MySQL gibi açık kaynak kodlu yazılımlar büyük ölçekli, çok-kullanıcı ve dağıtık bir oyun sistemini profesyonel düzeyde ortaya koymak üzere kullanılabilir.
- Oyun ortamlarındaki verileri tutan veritabanı ve oyun ortamındaki fonksiyonelliği sağlayan kodlar kullanıcıların bu

- ürünü kullanmadaki verimliliğini artırmak üzere geliştirme süreci boyunca optimize edilmelidir.
- Gerek geliştiriciler gerekse tasarımcılar tasarım süreci boyunca eğitsel oyun ortamını sık sık test etmelidir.
 - Belirli bir teknoloji etrafında (Örneğin AW teknolojisi gibi) zaman içerisinde o teknolojinin kullanılması ve geliştirilmesi ile ilgili olarak bu teknolojiyi üreten kişi ya da kurumların dışında bir geliştirici ve kullanıcı topluluğu oluşur. Bu topluluktan gerek bu teknolojileri kullanma konusunda, gerek bu teknolojileri kullanırken yaşanan sorunları çözme konusunda gerekse bu teknolojiler için üretim konusunda faydalanılabilir.

KAYNAKLAR

- Bakar, A., Tüzün, H., & Çağıltay, K. (2008). Öğrencilerin eğitsel bilgisayar oyunu kullanımına ilişkin algıları: Sosyal bilgiler dersi örneği. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 35, 27–37.
- Bayırtepe, E., & Tüzün, H. (2007). Oyun-tabanlı öğrenme ortamlarının öğrencilerin bilgisayar dersindeki başarıları ve öz-yeterlik algıları üzerine etkileri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 33, 41-54.
- Barab, S.A., Thomas, M.K., Dodge, T., Carteaux, B., & Tuzun, H. (2005). Making learning fun: Quest atlantis, a game without guns. *Educational Technology Research and Development*, 53(1), 86–107.
- Brown, A.L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The Journal of the Learning Sciences*, 2(2), 141–178.
- Brown, M.C. (2001). *Perl: The complete reference* (2nd ed.). New York, NY: McGraw-Hill/Osborne.
- Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O’Shea (Eds.), *New directions in educational technology* (pp. 15-22). New York, NY: Springer-Verlag.
- Dodge, B.J. (1995). WebQuests: A technique for internet-based learning. *Distance Educator*, 1(2), 10-13.

- Hansen, G.W., & Hansen, J.V. (1996). *Database management and design*. Upper Saddle River, NJ: Prentice Hall.
- Hearn, D., & Baker, M.P. (1986). *Computer graphics*. Englewood Cliffs, NJ: Prentice-Hall.
- Jenkins, N., & Schatt, S. (1995). *Understanding local area networks*. Indianapolis, IN: Sams Publishing.
- McGraw, W. (2003). *Classic video games may be a hard cell*. <http://www.cnn.com/2003/TECH/12/10/hln.game.cell.games/index.html>
Retrieved March 08, 2010.
- Molenda, M., Pershing, J.A., & Reigeluth, C.M. (1996). Designing instructional systems. In R.L. Craig (Ed.), *The ASTD training and development handbook* (4th ed., pp. 266–293). New York, NY: McGraw-Hill.
- Reigeluth, C.M. (1999). What is instructional-design theory and how is it changing?. In C.M. Reigeluth (Ed.), *Instructional-design theories and models, Volume 2: A new paradigm of instructional theory* (pp. 5-28). Mahwah, NJ: Erlbaum.
- Reigeluth, C.M., & Frick, T. (1999). Formative research: A methodology for improving design theories. In C.M. Reigeluth (Ed.), *Instructional-design theories and models, Volume 2: A new paradigm of instructional theory* (pp. 633–651). Mahwah, NJ: Erlbaum.
- Sheldon, R., & Moes, G. (2005). *Beginning MySQL*, Indianapolis, IN: Wiley Publishing.
- Tüzün, H. (2006). Eğitsel bilgisayar oyunları ve bir örnek: Quest atlantis. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 30, 220-229.
- Tüzün, H. (2007). Blending video games with learning: Issues and challenges with classroom implementations in the Turkish context. *British Journal of Educational Technology*, 38(3), 465-477.
- Tüzün, H., Arkun, S., Bayırtepe-Yağız, E., Kurt, F., & Yermeydan-Uğur, B. (2008). Evaluation of computer games for learning about mathematical functions. *i-manager's Journal of Educational Technology*, 5(2), 64–72.

Tüzün, H., Yılmaz-Soylu, M., Karakuş, T., İnal, Y., & Kızılkaya, G. (2009). The effects of computer games on primary school students' achievement and motivation in geography learning. *Computers & Education*, 52(1), 68–77.

YAZARA İLİŞKİN

Yrd. Doç. Dr. Hakan TÜZÜN Hacettepe Üniversitesi, Ankara



Yrd. Doç. Dr. Hakan Tüzün Hacettepe Üniv., Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümünde öğretim üyesi olarak görev yapmaktadır. Doktora eğitimini 2004 yılında Indiana Üniversitesinde Öğretim Teknolojileri bölümünde tamamlamıştır. Araştırma alanları arasında zengin öğrenme ortamları tasarımı yer alırken bu kapsamda öğrenenlerin kültürünü ve topluluğunu göz önüne

almaktadır.

Yrd. Doç. Dr. Hakan TÜZÜN
Hacettepe Üniversitesi, Eğitim Fakültesi,
Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü,
Beytepe/ANKARA
Tel: 90–312–297 7176
Eposta: htuzun@hacettepe.edu.tr
Web adresi: <http://yunus.hacettepe.edu.tr/~htuzun>
Facebook: <http://www.facebook.com/HakMan>
Academia: <http://hacettepe.academia.edu/HakMan>
MSN: hakantzn@yahoo.com
Skype: htuzun