

가상과학 박물관 구축에 관한 연구

강순덕*

요 약

본 논문의 목적은 가상 과학 박물관 구축에 있다. 소프트웨어의 신기술을 지닌 과학적이고도 기술적인 역량에 집중된 JAVA 3D 와 VR 3D, 멀티미디어와 교육적 자료의 공헌은 가상 과학 박물관을 현실화 시켰다. 즉, 이는 고수준의 정보서비스를 현실화 했다. 가상공간의 3차원 서비스와 유용한 과학 정보를 가진 교육적 자료를 제공한다. 결국에는 IT산업에 공헌한다.

I. 서론

현재 과학관련 Website는 많이 구축되어 있지만 서지 사항과 사진들로 구성되어 있는 단편적인 서비스로 제공되어지고 있다. 사용자의 입장에서 직접 박물관에서 물건을 보는 것과 같이 3차원적이고 동적이며 실제적인 기분을 느낄 수 있도록 하여 수요자 중심의 서비스를 제공하는 시스템이 필요하다.[12]

본 연구에서는 이 문제를 해결하기 위해서 수요자 위주의 서비스 시스템 설계 및 이미지 형상화로 이용자의 다양한 접근에도 응할 수 있는 시스템 구축을 구현하기 위해 박물관의 필요한 자료를 비디오 촬영 후 동영상으로 표현하는데 이때 쓰이는 기술로서 VRML, Plug-in에 의해 동영상의 움직임을 표현 구현한다.

이 가상 박물관의 사이트 구성은 각각의 자료의 Database화, 표현방법은 동영상 촬영을 통한

구현, 3D 환경을 지원하는 Virtual Reality, 서지 사항, 최적화된 고화질의 이미지, 등의 사용자 위주의 서비스 시스템 구축했다.

따라서 Internet에서 구현 및 서비스가 가능한 가상현실 기술의 도입과 서비스를 제공함으로써 고수준의 정보 서비스를 제공할 수 있게 한다. 그리고 다른 박물관과의 연계로 최신 자료의 검색 기능 탑재와 수요자 위주의 서비스 시스템 (User Friendly Interface) 설계 및 이미지 형상화 그리고 향후 CD-TITLE제작에 사용할 수 있는 자료의 준비와 이를 바탕으로 이 플랫폼에 연결 및 탑재하여 서비스를 가능케 하고 이용자의 접근이 다양한 구조를 구현을 했다.

II 가상박물관의 구현

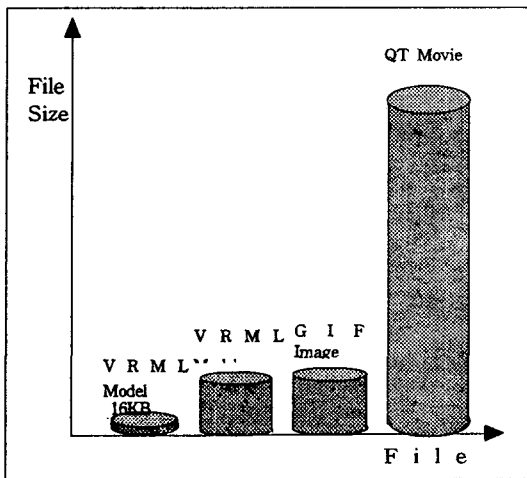
구현을 위한 도구로는 VRML, Java Script와 Java Applet, EAI(External Authoring Interface),

* 공주대학교 공과대학 정보통신공학부 교수

3차원 모델링 소프트웨어를 사용하였다.

2.1 VRML

3차원 세계와 사운드, 동영상 등의 멀티미디어를 월드 와이드 웹에서 모델링하기 위한 장면들과 웹의 하이퍼링크 개념을 접목시켜주며, 비교적 데이터 전송량이 적게 소요되어 즉, 시간적으로 유리해 14.4kbps의 모뎀을 사용한 접속에서도 실용적으로 사용되도록 했다. 이를 위해 다이얼업 유저 중심을 지향하는 방식이라 말할 수 있다. (그림 1)은 같은 내용의 3차원 형상을 표현하는데 필요한 파일 타입에 따른 파일 크기를 나타내주는 그림이다. 전송 속도의 최적화를 위해서 VRML에서는 GZIP 압축을 이용한다.



(그림 1) file type에 따른 file size

따라서 사용자 입력에 의한 상호작용과 네비게이션이 가능. 웹을 기반으로 하며 인터넷을 통한 다중 참여자를 지원하는 가상환경 시스템 구축 가능하고 웹의 사용을 전제로 설계됨으로서 플러그인 만 설치하게 되면 웹 사용이 가능

한 어떤 시스템에서도 볼 수 있다. 빠른 전송을 위해서 아스키 파일 형태로 되어 있으며 다른 VRML 문서, HTML 문서와 연결되어 사용한다. 텍스트 화일 형식으로 되어 있으며 제공하는 정보의 양에 비하여 화일의 크기가 작다.

즉, VRML 파일은 평균 이미지 파일의 크기와 비슷하고 압축할 경우에는 5분의 1로 줄일 수 있다.

그리고 다음과 같은 다섯 가지 경우를 고려해서 VRML를 사용하였다.[7]

1. 상호작용성(Interactive)은 VRML 월드 내에서 사용자는 주위의 사물과 상호작용이 가능. [11] 집안을 돌아다니다가 닫힌 문을 만나면 열고 들어갈 수 있고 불을 켜거나 끌 수도 있음.

2. 네비게이션(Navigation)은 사용자는 VRML 월드 내를 마음대로 돌아다닐 수 있어서 흥미 있는 물체를 마음대로 돌려보거나 이동시켜 볼 수도 있게끔 된다. HTML에서처럼 Anchor를 통하여 다른 VRML문서나 HTML문서로 이동할 수 있다.

3. 장면의 구성은 그래픽이나 비디오 같은 매체는 사용자가 실제로 보고자 하는 장면에 대한 데이터를 미리 준비해서 파일 속에 저장.

VRML은 국제 표준인 ASCII나 유니코드(Unicode) 형식으로 저장하는데 유리하고, 장면에 대한 데이터를 저장하는 대신에 장면을 이루는 요소들에 대한 정보를 저장된다.

QuickTime VR은 미리 찍어둔 몇 개의 사진을 조합.

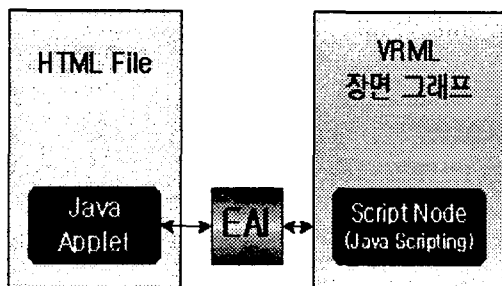
4. 3D 애니메이션은 VRML이 아니고 "3D Studio MAX" 같은 3차원 그래픽 툴을 이용하여 만든 모델링 데이터를 애니메이션으로 만든 것이다. VRML과 같이 마음대로 네비게이션 할 수 없다.

5. 스크립트는 VRML만 사용 가능하며 단순한 상호작용뿐 만 아니라 여러 가지 조건을 종합하여 복잡한 상호작용을 할 수 있다. 노드를 마음대로 제어하며 현재의 상태를 저장(변수를 사용할 수 있다)하여 나중에 사용 가능. [9]

2.2 Java Script와 Java Applet

프로그램은 다양한 일을 처리할 수 있다. 이 문제를 해결할 수 있는 기술이 Java Script와 Java Applet인데, 이벤트를 다른 노드로 보내 장면을 변화시키거나, 계산을 수행하거나, 인터넷상의 다른 호스트와 통신하는 등 여러 가지 작업을 할 때, 필요한 것으로 첫 번째 단계는 스크립트 노드에서 자바와의 연결을 기술로 스크립트 노드는 이벤트를 받았을 때 구동되는데, 브라우저는 노드안의 url 필드에서 기술한 프로그램을 수행시킨다. 이때 필요하다면 외부 인터프리터에 프로그램을 넘겨준다. 애플릿은 클라이언트 측에 다운로드 되어 사용된다. [6]

2.3 EAI(External Authoring Interface)



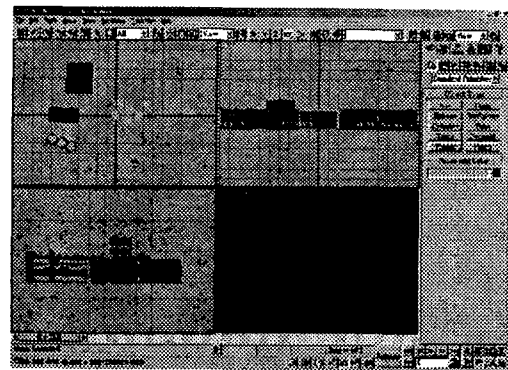
(그림 2) VRML, EAI, Java Applet의 관계

(그림 2)와 같이 VRML 세계와 외부 환경과

의 인터페이스를 제공하는 것으로 Java applet이 같은 페이지에 있는 VRML세계를 제어하는 것이 EAI이다. [14]

2.4 3차원 모델링 소프트웨어

VR 시스템의 구성요소들을 통합하고 관리하는 작업을 수행. 가상공간 저작 기능(자체 저작 기능 또는 외부 데이터 import), 사용자 입출력 처리로서 (그림 3)과 같다.[13]



(그림 3) 3D Stdio MAX

VR응용 개발 소프트웨어

- 가상세계에 3차원 모델링 객체의 배치, 관리기능
- 객체의 행위 기술 및 3차원 애니메이션 기능

구현에 따른 기술에는 VRML을 좀더 구체적으로 제어하기 위해서는 EAI를 통해서만 가능하다. 여기서는 동영상의 소스코드를 가지고 기술적인 부분만을 설명하겠다. 기본적으로 3가지 소스코드가 필요한데, VRML코드인 확장명이 .wrl인 파일과 Java Applet 소스코드인 .java를 컴파일하여 얻은 .class 그리고 두 파일을 포함하는 html문서가 필요하다.

```
<HTML><HEAD></HEAD>
<BODY>
<APPLET CODE="Applet1.class"
WIDTH=570 HEIGHT=80></APPLET>
<EMBED SRC="theater.wrl" WIDTH=570
HEIGHT=300>
</BODY>
```

<html>

```
#VRML V2.0 utf8
## 생략
DEF PLAY Switch (
  whichChoice -1
  choice [
#####1번 동영상#####
DEF playingscreen1 Transform ( ##생략
  texture DEF M1 MovieTexture (
    url "01.avi"
  stopTime -1
  loop TRUE )
#####2번 동영상#####
DEF playingscreen2 Transform ( ##생략
##### 기타 동영상#####
|
```

<java 코드⇒ 컴파일시 확장명 .class>

```
import java.awt.*;
import java.applet.*;
import vrml.external.Browser;
import vrml.external.Node;
import vrml.external.field.EventInSFInt32;
public class Applet1 extends Applet {
  EventInSFInt32 c_movie = null;
  int c;
  // 생략
  public void init() {
  // 생략
  Browser browser = Browser.getBrowser(this);
  Node n_movie = browser.getNode("PLAY");
```

<vrml 코드>

2.5 웹 VR기술

- VRML 기반, Quicktime VR 기반, 기타 3차원 모델 플러그인
- 3차원 구현 회사들의 독립적인 기술과 브라

우저 제공

- VRML은 사용자 입력에 의한 네비게이션이 가능하고, 애니메이션, 사운드 및 스크립팅 기능을 지원하여 동적인 3차원 환경 연출.
- QuickTime VR은 연속적인 사진을 파노라마 형식으로 배열하여 네비게이션 효과를 연출하는 방식으로 웹 VR기술을 사용했다.[8]

III. 가상박물관 3D 프로그램 소스 및 단면도

3.1 프로그램 소스

```
#include <pr.h>
#include <prchar.h>
#include <TerrainEngine.h>
```

```
#include <winutil.h>
#include "prdev.h"
```

```
#include <direct.h>
```

```
void resize (int width, int height);
```

```
PR_LIGHTLIST userlights; /* Holds a list of lights */
```

```
PR_DWORD Quit = 0;
```

```
/* Frame Rate Control */
```

```
PR_DWORD ticks; /* Total number of ticks passed */
```

```

    PR_DWORD updates; /* Total number of
updates */
    PR_DWORD framerate; /* Resulting frame
rate (updates/ticks/tickrate) */

    EntityNode *MyPlayer;
    PR_REAL Gravity = -0.1f;
    PR_REAL JumpStrength = 50;
    PR_DWORD FreeFall;

    /* ----- */
    /* Makes sure the camera is above the
landscape for the first frame */
    /* ----- */
    void SetInitialPosition (void)
    {
        PR_REAL hgt;

        hgt = PR_GetTerrainHeight (Terrain_Globals.
terrain, camera->source.x, camera->source.z,
Terrain_Globals.terrain->level1_heightpic,
Terrain_Globals.terrain->level1_colorpic) + 300;
        PR_PositionCameraSource (camera, camera->
source.x, hgt + 300, camera->source.z);
    }
    /* ----- */
    /* Checks for ground collision */
    /* ----- */
    void CheckForTerrain (EntityNode *node)
    {
        PR_REAL hgt1;
        hgt1 = PR_GetTerrainHeight (Terrain_Globals.
terrain, node->Info.Location.x,
node->Info.Location.z,
Terrain_Globals.terrain->level1_heightpic,
Terrain_Globals.terrain->level1_colorpic)
+ 300;
        if (PR_InputModel == MODEL_WALKING)
        {
            if (node->Info.Location.y < hgt1)
            {
                OM_MoveEntity (node, 0, hgt1-node->Info.
Location.y, 0, 0);
                MyPlayer->Info.Velocity.y = 0;
            }
        }
        else /* MODEL_FLYING */
        {
            if (camera->source.y < hgt1)
            {
                camera->dest.y += hgt1 - camera->source.y;
                camera->source.y = hgt1;
            }
        }
    }
    /* ----- */
    /* Check the mouse and keyboard input */
    /* ----- */
    void UserInput (PR_DWORD ticks)
    {
        PR_ReadInput ();
        /* Move around depending on what
movement model we are using */
        PR_CameraDirection (camera);

        if (kbdon[KEY_SPACE])
        {
            MyPlayer->Info.Velocity.y = JumpStrength;
        }
    }

```

```

MyPlayer->Info.Velocity.x           = width * Terrain_Globals.terrain->world_scale;
-camera->direction.x * currentspeed;
MyPlayer->Info.Velocity.z           = height * Terrain_Globals.terrain->world_scale)
-camera->direction.z * currentspeed;
                                     camera->source.z -= Terrain_Globals.terrain->
height * Terrain_Globals.terrain->world_scale;
PR_CameraDirection (camera);

if (PR_InputModel == MODEL_WALKING)
MyPlayer->Info.Velocity.y += ticks * Gravity;

FreeFall = OM_MoveEntity (MyPlayer,
MyPlayer->Info.Velocity.x, MyPlayer->Info.Velocity.
y, MyPlayer->Info.Velocity.z, 1);
if (!FreeFall)
MyPlayer->Info.Velocity.y = 0;

if (Terrain_Globals.OutsideVisible)
CheckForTerrain (MyPlayer);

PR_PositionCameraSource (camera, MyPlayer->
Info.Entity->orientation.location.x,
MyPlayer->Info.Entity->orientation.location.y,
MyPlayer->Info.Entity->orientation.location.z);
PR_CameraDirection (camera); /* Automati-
cally generated destination for camera */

if (camera->source.x < 0)
camera->source.x += Terrain_Globals.terrain->
width * Terrain_Globals.terrain->world_scale;
if (camera->source.z < 0)
camera->source.z += Terrain_Globals.terrain->
height * Terrain_Globals.terrain->world_scale;

if (camera->source.x >= Terrain_Globals.
terrain->width * Terrain_Globals.terrain->world_
scale)
camera->source.x -= Terrain_Globals.terrain->
width * Terrain_Globals.terrain->world_scale;
if (camera->source.z >= Terrain_Globals.terrain->
height * Terrain_Globals.terrain->world_scale)
camera->source.z -= Terrain_Globals.terrain->
height * Terrain_Globals.terrain->world_scale;

PR_SetD3DPipeline (1);
PR_SetActiveCamera (camera);
/* Updates the transformation matrix */

if (kbdon[KEY_F1])
{
PR_SetInputModel (MODEL_WALKING);

TRANSLATE_SPEED = 20.0f;
MAX_TRANSLATE_SPEED = 100.0f;
TRANSLATE_DRAG = 10.0f;

PITCH_SPEED = 0.3f;
MAX_PITCH_SPEED = 2.0f;
PITCH_DRAG = 0.1f;
}
else if (kbdon[KEY_F2])
{
PR_SetInputModel (MODEL_FLYING);

TRANSLATE_SPEED = 20.0f;
MAX_TRANSLATE_SPEED = 100.0f;
TRANSLATE_DRAG = 10.0f;

PITCH_SPEED = 0.6f;
MAX_PITCH_SPEED = 4.0f;
PITCH_DRAG = 0.2f;
}

```

```

    }
    void CreateCharacter (void)
    /* This will search for the first spawn
    location and create our character there. */
    {
        EntityNode *SpawnNode;
        PR_POINT Location;
        /* Find the first spawn location and start
        there */
        SpawnNode = OM_FindEntityName ("Spawn1",
        Terrain_Globals.RootEntity);
        if (SpawnNode != NULL)
            Location = SpawnNode->Info.Location;
        else
            Location.x = Location.y = Location.z = 0;
        /* No spawn, so start at origin */

        MyPlayer = LS_CreateGamePlayer ("Player",
        &Location);
        if (MyPlayer == NULL) // missing player
        type */
            PR_FatalError ("Cannot create game player!",
            " ");

        MyPlayer->Info.Cylinder.Radius = 150;
        MyPlayer->Info.Cylinder.TopHeight = 150;
        MyPlayer->Info.Cylinder.BottomHeight = -150;
        MyPlayer->Info.Flags |= OFLAG_COLLISION_
        SLIDE;
    }

    void main (void)
    {
        PR_UDWORD BLACK; /* Color index/value
        for black */
        PR_DWORD move_count;

        //PR_Settings.EnableLogging = 1;
        // PR_Settings.UseWindowed = 1;

        PR_Settings.MissingTextureFatal = 0;

        Terrain_Globals.InitializationParameters.ScreenWidth = 800;
        Terrain_Globals.InitializationParameters.ScreenHeight = 600;
        Terrain_Globals.InitializationParameters.MaxPolys = 10000;
        Terrain_Globals.InitializationParameters.MaxTextures = 1024;
        Terrain_Globals.InitializationParameters.MaxMaterials = 1024;
        Terrain_Globals.InitializationParameters.MaxCharacters = 32;
        Terrain_Globals.InitializationParameters.MaxArticles = 5000;
        Terrain_Globals.InitializationParameters.MaxEmitters = 500;
        Terrain_Globals.InitializationParameters.MaxVoices = 32;
        Terrain_Globals.InitializationParameters.MaxWAVs = 32;
        strcpy (Terrain_Globals.InitializationParameters.PackagePath, "packages");

        /* PR Initialization */
        ChooseDevice ();
        PR_Settings.UseAGP = 1;
        PR_Initialize (Terrain_Globals.InitializationParameters.MaxPolys);
    }

```

```

InitializeDevices ();

/* Landscape Studio Initialization */
LS_InitializeTerrainEngine ();
PRBSP_InitEngine (20000, 180000); /*
Initialize the BSP engine */

resize (PR_Globals.PR_DisplayDevices[PR_Globals.
PR_CurrentMonitor].D3DViewport.Width,
PR_Globals.PR_DisplayDevices[PR_Globals.P
R_CurrentMonitor].D3DViewport.Height);

/* Initialize the camera */
camera = PR_AllocCamera ();
PR_InitializeCamera (camera);
PR_PositionCameraTarget (camera, 0, 0, 0);
PR_SetCameraMode (camera, CAMFLAG_
ANGLE_BASED);
camera->nearclip = 10;
camera->aspect_ratio = 1.0f; //3333f;
PR_SetActiveCamera (camera);

/* Initialize the light */
PR_AllocLights (&PR_Globals.scenelights, 8);
PR_AllocLights (&userlights, 1);
PR_SetLightPosition (&userlights, 0, 4000,
4000, 4000);
PR_SetLightOn (&userlights, 0);
PR_SetLightType (&userlights, 0, DIREC-
TIONAL_LIGHT);
PR_SetLightColor (&userlights, 0, 1.0, 1.0,
1.0);
PR_SetLightStrength (&userlights, 0, 1.0);
userlights.NumLights = 1; /* This sets
how many lights are actually used */

PR_OpenScreen (PR_BACKBUFFER);
BLACK = PRGFX_MakeColor (0, 0, 0);
PR_InitTimer (60);

LS_InitializeCharacters ();
PR_SetD3DPipeline (1);

chdir ("level1");
LS_LoadLevel ();

CreateCharacter ();

/* Initialize the input devices */
installkbd ();
PR_SetInputDevice (INPUT_MOUSE);
PR_SetInputModel (MODEL_WALKING);
PR_SetInputEntity (NO_ENTITY, NO_SEG-
MENT);
PR_SetInputCamera (camera);

TRANSLATE_SPEED = 20.0f;
MAX_TRANSLATE_SPEED = 100.0f;
TRANSLATE_DRAG = 10.0f;

PITCH_SPEED = 0.2f;
MAX_PITCH_SPEED = 2.0f;
PITCH_DRAG = 0.2f;

/* main program loop */
while (!kbdon[KEY_ESC])
{
UpdateMessages ();

move_count = PR_GetTicks ();

```



```

if (Terrain_Globals.terrain != NULL)
    UserInput (ticks);

if (Terrain_Globals.terrain != NULL)
    LS_DrawDynamicShadows ();

PR_SetActiveCamera (camera);
PR_OpenScreen (PR_BACKBUFFER);

    CHR_UpdateEmitters (move_count);

if (Terrain_Globals.terrain != NULL)
{
    OM_UpdateEntities (Terrain_Globals.Root
Entity, move_count);
    PR_UpdateTerrain (Terrain_Globals.terrain);
        LS_DrawWorld (move_count);
}
else
{
    PRGFX_SetColor(0);
    PRGFX_ClearScreen ();
}

PR_Globals.scenelights.NumLights = 0; /*
Reset scene lights, so previous lights only
affect precalculated lighting methods */

SND_Update3DSounds ();

ticks += move_count;
if (ticks > 60)
{
    framerate = updates;
    ticks = 0;

    updates = 0;
}
else
    updates++;

    PRGFX_SetTextForeground
(PRGFX_MakeColor (63,63,63));
    PRGFX_SetTextTransparent
(TEXTFG);
    PR_printf (0, 0, "freefall: %i", FreeFall);
    PR_printf (0, 10, "inside: %i", Terrain_
Globals.RootEntity->Info.BSPEntity->CameraIn
side);
    PR_printf (0, 20, "sky: %i", Terrain_
Globals.RootEntity->Info.BSPEntity->DrawSky);

    PR_Flip (0); /* Show the background
page */
}

PRBSP_CloseEngine ();
LS_CloseCharacters ();
LS_FreeLevel ();

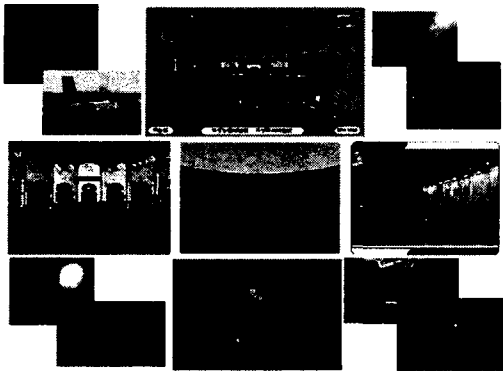
/* Free all allocated memory */
PR_FreeLights (&userlights);
PR_FreeLights (&PR_Globals.scenelights);
PR_FreeCamera (camera);
PR_DeleteAllTextures ();
PR_DeleteAllMaterials ();
uninstallkbd ();

PR_ShutDown ();
}

```

3.2 가상박물관의 단면도

가상박물관의 단면도는 (그림 4)와 같다.



(그림 4) 가상과학관의 단면도

IV 결론

가상박물관의 연구 결과에 대한 기여도, 기대 효과 및 활용 방안은 다음과 같다.[5]

1. WEB을 통해 3차원 영상, 동영상을 사용해서 즉, 가상공간의 3차원 서비스 및 과학정보를 제작 및 보급함으로써 기관 홍보 효과를 높이고 자료활용도를 극대화시킬 수 있으며 향후에 CD-ROM TITLE제작에 사용할 수 있어서 교육 자료로도 활용 가능케 할 수 있다. 이는 멀티미디어 교육의 총화라 할 수 있다.

2. 초고속 정보 통신망의 활용영역의 확대와 투자효과의 최대화 기여

3. 가상박물관시스템의 구축으로 Venture기업의 활성화

4. 문화적인 역량이 국가경쟁력을 결정하는 문화 우위의 시대에 세계문화의 화합과 창조성

을 주도하는 확고한 입지 확보 및 정보화 시대에 맞추어 인터넷을 활용한 한국 과학의 우수성과 독창성을 세계에 홍보하며 지역의 정보화 인프라 구축 및 도시 기반 연계를 통하여 세계적인 과학교류 및 인류과학화 부흥의 중심지로 부상

5. 전통 문화 상품의 제작과 시연 및 참여를 통한 한국 문화의 이미지 제고

6. 박물관 구조조정의 계기가 됨

향후에 가상박물관에서 오브젝트에 대한 다양한 시각화와 사용자와의 인터페이스, 최소 데이터량 전송으로 서버부하의 최소화 및 자유로운 오브젝트 배치를 통한 시뮬레이션, 브라우저와의 impoer/export기능에 대해 연구해야한다.

참고문헌

- [1] MicroSoft Ware 2000
<http://maso.zdnet.co.kr>
- [2] MicroSoft Agent Document 2000
<http://msdn.microsoft.com/workshop/imedia/agent/documentation.asp>
- [3] PcBee 2000
<http://www.pcbec.co.kr>
- [4] Kbench 2000
<http://www.kbench.com>
- [5] DAVID JONES , COLIN McCORMACK
BUILDING A Web-Based Education System 1997
- [6] David E. Culler Jaswinder Pal Singh with Anoop Gupta Paraller Computer Architecture A Hardware / Software Approach 1998

- [7] Koster, Martijn. "Guidelines for Robot Writers" Nexor Corp, 1994a
<http://info.webcrawler.com/mak/projects/robots/guidelines.html>
- [8] Koster, Martijn. "A Standard for Robot Exclusion". Nexor Corp..1994b
<http://info.webcrawler.com/mak/projects/robots/norobots.html>
- [9] Koster, Martijn. "WWW Robot Frequently Asked Questions". 1996
<http://info.webcrawler.com/mak/projects/robots/faq.html>
- [10] Steve G. Steinberg. "Seek and Ye Shall Find (maybe)". 1996
<http://www.hotwired.com/wired/4.05/features/in dexweb.html>
- [11] Franklin. Stan.. Graesser, Art "Is it Agent, or just a Program?: A Taxonomy for Autonomous Agents". Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag. 1996
- [12] Gray. Matthew "Web Growth Summary". 1996
<http://dutifp.twi.tudelft.nl:8000/checkbot/>
- [13] Victor Parada. "Web Copy". 1996
<http://www.inf.utfsm.cl/vparada/webcopy.html>
- [14] "Introducing to HTML", WWW Consortium 1996
<http://www.w3.org/pub/WWW/MarkUp/Wilbur>

A Study on the Cyber Science Museum Construction

Soon-Duk, Kang*

Abstract

This paper aims at making cyber science museum construction. JAVA 3D and VR 3D which concentrate the scientific capacity and the technical capacity with the new technology of software, contribution of multimedia and educational data realized the Cyber Science Museum. So, It realized high level information service. It offered a user the educational data which has three dimension service of cyber space and useful science information. After all, It contributed to IT industry.

* Division of Information & Communication Engineering, Kongju National University