

SZAKMAI BESZÁMOLÓ

E-HOSZTESZ INFORMÁCIÓS KIOSZK

Munkaszakasz: 2

Beszámolási időszak: 2004. május 1.-2005. január 31.

Szervezet: VerAnim Számítástechnikai és Prod. Iroda Bt.

Projektvezető: Takács Barnabás

Honlap: www.digitalElite.net/Pages/VerAnim/Ehosztesz.html

INFORMÁCIÓS ÉS KOMMUNIKÁCIÓS TECHNOLÓGIÁK PROGRAM

NYILVÁNTARTÁSI SZÁM: OMFB-00238/2004

TARTALOM

1. BEVEZETÉS	3
2. KORÁBBI MUNKASZAKASZOK EREDMÉNYEINEK ÖSSZEFOGLALÓJA	3
3. RÉSZFELADATOK LISTÁJA ÉS STÁTUSZA.....	4
4. A MUNKASZAKASZBAN ELKÉSZÜLT EREDMÉNYEK RÖVID BEMUTATÁSA.....	4
4.1 VALÓSIDEJŰ DIRECTX9 KÉPSZINTÉZIS MODUL	4
4.2 DIALÓGUSVEZÉRLŐ ALRENDSZER	5
4.3 MESTERSÉGES INTELLIGENCIA ÉS KOGNITÍV ARCHITEKTÚRA	6
4.4 MULTI-MODÁLIS FELHASZNÁLÓI FELÜLET	6
4.5 KÜLSŐ KÉPSZINTÉZIS MODULOK.....	7
5. CIKKEK ÉS PUBLIKÁCIÓK	7
6. MELLÉKLETEK.....	8
RÉSZLETES PROGRAM TERV	9
ELŐZETES FELHASZNÁLÓI DOKUMENTÁCIÓ.....	21
TESZTELÉSI TERV	35

1. Bevezetés

Az E-Hosztesz projekt alapvető célkitűzése olyan innovatív vizuális kommunikációs eszköz létrehozása, amely lehetővé teszi a felhasználókhöz eljuttatni kívánt információ érzelmi és személyre szóló modulációját, növelve ez által a befogadó képesség hatékonyságát. Az új eszköz egy vizuálisan gazdag, interaktív 3D média környezet, amely animált virtuális embereket, és mesterséges látás, illetve képfelismerés eszközeit használja a fenti célok elérésére. A virtuális E-Hosztesz - amely a számítógépes animáció és grafika segítségével lehet egy élő ember másolata, vagy stilizált, a valóságban nem létező, de életszerűnek tűnő személy - olyan egyénre szabott informatív környezet megteremtésére ad lehetőséget, amelyben az információ befogadás hatékonysága többszörösére növekedhet. A felhasználó és a virtuális E-Hosztesz kapcsolatát két-irányú információ-áramlás teszi lehetővé, amely egyrészt támaszkodik egy beépített arcfelismerő és képfeldolgozó rendszerre, másrészt a virtuális digitális arc azon kifinomult képességére, hogy az emberhez hasonló arc-mimikával, szemkontaktussal és más non-verbális eszközökkel teremtsen kapcsolatot a rendszer és használója között.

2. Korábbi munkaszakaszok eredményeinek összefoglalója

Az E-Hosztesz projekt három szakaszban valósul meg. A jelen szakasz a projekt második fázisát képezi. Az első munkaszakaszban munkánk megkezdésekor először egy olyan Web-alapú rendszert hoztunk létre, amely a projekt során elkészülő dokumentumokat tárolja, élettörténetüket követi. A Web oldal Internet elérhetőségét a címlap tartalmazza, ennek tartalmát folyamatosan frissítjük. Elkészült továbbá több, a projekt magas szintű minőségét és kivitelezését szabályozó „*Minősbiztosítási terv és dokumentum*” is.

A technikai megvalósulás tervezésekor az első munkaszakaszban megvizsgáltuk az E-Hosztesz kommunikációs és dialógus architektúra kialakításához szükséges interaktív környezetet, valamint lefektettük a fejlesztéshez szükséges animációs interfész (API) szabályait, a dialógus tervezés módszerének és a hasonló jellegű illesztendő külső programoknak az igényeit, valamint azok lehetséges szolgáltatásait. A képszintézist megvalósító módszerek terén különös figyelmet fordítottunk a virtuális E-Hosztesz élethűségét befolyásoló tényezők tanulmányozására, pl. a viselkedésre, vagy a megjelenés egyes részleteire, mint a haj, bőr és ruházat megjelenítésének kérdéseire. Kitérünk arra, hogy a modern grafikus kártyák pixel és vertex shader-ei milyen szinten támogathatják ezen anyagmodellek alkalmazását. A kapcsolódó módszereket egy a „*Valós-idejű képszintézis algoritmusok specifikációja*” és „*Részletes specifikáció*” című dokumentumokban foglaltuk össze.

3. Részfeladatok listája és státusza

1. Valós idejű DirectX9 E-Hosztesz képszintézis modul.	<i>elkészült</i>
2. Mesterséges intelligencia és kognitív architektúra.	<i>elkészült</i>
3. Dialógusvezérlő alrendszer.	<i>elkészült</i>
4. Multi-modális felhasználói felület.	<i>elkészült</i>
5. Külső képszintézis modulok illesztése.	<i>elkészült</i>

1. táblázat: Projekt státusz.

4. A munkaszakaszban elkészült eredmények rövid bemutatása

Az E-Hosztesz projekt jelen munkafázisában megvalósítottuk azt a prototípus animációs és fejlesztői környezetet, amelyben egy virtuális ember 3D modelljét hoztuk létre, majd azt E-hosztesként interaktív, intelligens formában használjuk. A fejlesztés öt párhuzamos fő irányvonala a valós időben létrehozható i) virtuális ember alaptechnológia továbbfejlesztése, a ii) valós idejű vizuális képi minőség gazdagítása, különös tekintettel a haj, bőr és ruházat modellezésére, a iii) dialógust megvalósító mesterséges intelligencia modulok kialakítása és a iv) multi-modális interfészek (pl. látás) funkcióinak kiterjesztése, és végül a v) külső képszintézis modulok illesztése volt. A fejlesztés során mind az öt különböző modul elkészült, ezek státuszát a 3. szakaszban foglaltuk össze.

4.1 Valós idejű DirectX9 képszintézis modul

Az E-Hosztesz virtuális ember animációs rendszer támogatja a legmodernebb grafikus kártyák által ajánlott képszintézis módszereket. Ahogy a hardvergyártók - elsősorban az *NVidia* és az *ATI* - egyre többet és többet valósítanak meg az ide vonatkozó grafikus szabványokból, úgy ezek a funkciók fokozatosan elérhetővé válnak az E-Hosztesz rendszer felhasználói számára is. A DirectX9 szabvány beépítésekor törekedtünk arra, hogy a virtuális ember minél élethűbb megjelenítésében fontos vizuális elemeket emeljük ki, így a kutatás elsősorban a bőr és haj megjelenítésre koncentrált. Mindkettő megvalósításában kulcs szerepet kapott, és szükségessé vált a grafikus kártyákra közvetlenül feltölthető, és ott a processzortól függetlenül, igen nagy sebességgel futó ún. *vertex-* és *pixel shader*ek használata.

A grafikus kártyákon közvetlenül programozható pixel-futószalag (pipeline) két legfontosabb eleme a 3D grafikai elemek *vertexeinek* és az egyes *pixelek színének* hardver szintű feldolgozása, dinamikus módosítása. A vertex feldolgozás további részekre bontható. Ezek az átalakítás (*transformation*), a megvilágítás (*lighting*), valamint korlátozott mértékben a textúra-koodináta transzformáció. Az átalakítás során a gráfstruktúrában (vertexstruktúrában) tárolt pozíció adatokat „képernyőtér” (*screen space*) adatokká alakítják át. A grafikus kártyán futó kis programnak (*fragment kód*) tehát arra van lehetősége, hogy minden egyes geometriai pontot kicsit „odébb toljon”. Ennek nagy szerep jut a haj kis mértékű mozgásában. Az architektúra másik fontos eleme a *pixel shader*, amely bizonyos értelemben forradalmi újítás. A *pixel shader* - a jelen fejlesztés szempontjából talán legfontosabb - képessége, hogy egy

textúrát egy másikkal együtt tud használni. Ezzel lehetőség válik arra, hogy pl. az emberi bőrre jellemző több rétegű fényvisszaverő struktúrát megvalósítsuk.

A vertex- és pixel shaderek a gyakorlati megvalósítása rövidke programok segítségével történik. Ezek a programok nem magas szintű nyelveken, hanem a gépi kódhoz erősen hasonlító, egyszerű műveletekkel történik. Az E-Hoztesz rendszer jelenleg támogatja mind az *NVidia* által kifejlesztett *CG*, és a *Microsoft* által támogatott *HLSL* nyelveket. A 2. táblázat egy CG nyelven íródott, az emberi bőr fényvisszaverő tulajdonságait szimuláló shaderre mutat példát. Ez a rövid program egyszerre több vezérlő textúrából és dinamikusan változtatható paramétereiből számítja ki, hogy egy adott irányból nézve mennyire csillogjon a az E-Hoztesz bőre.

```

struct vertout
{
    float4 Pos          : POSITION;
    float4 Col          : COLOR0;
    float4 uv           : TEXCOORD0;
    float4 normal       : TEXCOORD4;
    float4 view         : TEXCOORD5;
    float4 light        : TEXCOORD6;
    float4 lightColor   : TEXCOORD7;
};

void main(vertout IN,
uniform sampler2D texture,
uniform sampler2D bumpmap,
uniform float4 speccolor, // w = power of specular light
uniform float4 diffcolor, // w = bump map scale
uniform float4 ambient, // w = out float4 color : COLOR)
{
    float4 texCol      = tex2D(texture, IN.uv.xy);
    float4 bump        = tex2D(bumpmap, IN.uv.xy);

    float3 normal      = normalize(normalize( IN.normal.xyz) + (bump.xyz-0.5.xxx) * (diffcolor.w).xxx);
    float3 view        = normalize(IN.view.xyz);
    float3 light       = normalize(IN.light.xyz); //float3(1,0,0);

    float3 halfv       = normalize(light + view);
    float spec         = pow( dot(normal, halfv), speccolor.w);
    float diffuse      = max( dot(normal, light), 0);
    float whiteness    = ambient.w * ( 1 - dot(normal, view) + diffuse);
    float avgSkin      = (texCol.x + texCol.y + texCol.z) / 3;
    float3 skinColor   = texCol.xyz * (1 - whiteness) + whiteness * avgSkin.xxx;
    float3 diffuseColor = IN.lightColor.xyz * diffuse.xxx * diffcolor.xyz * skinColor.xyz;
    float3 specColor   = IN.lightColor.xyz * spec.xxx * speccolor.xyz;
    float3 ambientColor = ambient.xyz * skinColor.xyz;
    color.xyz          = diffuseColor.xyz + specColor.xyz + ambientColor.xyz;
    color.a            = IN.Col.a;
}

```

2. táblázat: CG pixel shader példa a bőr csillogásának megvalósítására.

Hasonló módszereket dolgoztunk ki a haj megjelenítésére. Fontos különbség azonban, hogy míg a bőr felszínét közvetlenül modellezzük, a hajszálakat a megjelenítés előtt automatikusan generáljuk. A haj stílusát és megjelenését ily módon a programon kívülről módosítható ún. textúra térképekkel (*texture maps*) változtatjuk. Minden egyes textúra a hajszálak más és más tulajdonságát határozza meg. Konkrétan lehetőség van a i) haj szín, ii) sűrűség, iii) szálszámú hossz és iv) minden hajszál irányának beállítására is.

4.2 Dialógusvezérlő alrendszer

A dialógusvezérlő alrendszer feladata az interaktív folyamat fenntartása. Ez a modul hajtja végre a rendszer magasabb szintű reakcióit meghatározó moduloktól kapott utasításokat. Az itt megvalósított mechanizmusokat a Multi Track Manager (MMT) segítségével lejátszható apró mozgássorozatok formájában tároljuk. Ezek a mozgásminták három csoportra oszthatók. Egy részük kézzel animált és előre beállított test pozíciók és arckifejezések sorozatából áll,

másik részük az ún. *motion capture* technikájával felvett emberi mozgásokat tárol, végül az átkötő animációs elemeket, valamint az alacsony, reflex szintű ösztönös viselkedéseket automatikusan generálja a rendszer.

A fenti mechanizmusokat a *LUA* script interpreterben megvalósított viselkedés modul köti össze. Ezen a szinten könnyen és gyorsan lehet előre programozott, vagy dinamikusan változó viselkedéseket létrehozni, illetve egy adott szituációban lejátszani. Ezt segíti elő az E-Hosztesz megjelenését és belső állapotának minden aspektusát vezérlő *Disc Controller* alrendszer is. A *Disc Controller* mechanizmus közvetlen kapcsolatban van a szimbolikus *marker rendszerrel*, így segítségével olyan reakciók valósíthatók meg, mint pl. a felhasználó szemmel történő követése, vagy akár az is, hogy a virtuális emberünk szeme felcsillanjon egy érdekes válasz hallatán. A felhasználó és interaktív rendszer között ily módon kialakuló dialógust ezek a vizuális elemek nem-verbális, másodlagos jelekkel segítik, a szavakkal elmondott tartalom jelentését megerősítik és támogatják.

4.3 Mesterséges intelligencia és kognitív architektúra

Az E-Hosztesz architektúra intelligens reakcióit és viselkedését egy ún. kognitív vezérlő adja, amelyet a nemzetközileg több hasonló alkalmazásban már bevezetett *SOAR* motor adja. Ez a kognitív alrendszer egy a szimbolikus mesterséges intelligencia eszközeit használó és annak elemeire épülő megvalósítás, amelyet eredetileg az emberi gondolkodás modellezésére fejlesztettek ki. Az alrendszert a fő képszintézis modultól külön futó - jelenleg ugyanazon a számítógépen, de később akár külön szerveren elhelyezett – kliens-szerver módon valósult meg. Az E-Hosztesz egy adott szituációban lehetséges viselkedéseit és reakcióit egy szabály alapú rendszerben határozzuk meg, amely a belső szimbolikus változók, mint pl. a látó rendszer által észlelt események és emberi arcok, függvényében generál a szituációhoz illeszkedő válaszokat. Ezek a válaszok később az előző szakaszban röviden ismertetett dialógusvezérlő alrendszeren keresztül konkrét animált mozgások formájában jelennek meg. A projekt jelen fázisában sikerült megteremtünk a két program közötti adatforgalmat, a belső változók átadását és a generált reakciók megfelelő visszajuttatását. A viselkedést leíró szabályrendszer fejlesztése egy hosszabb folyamat, amelyet a harmadik fázisban is folytatunk.

4.4 Multi-modális felhasználói felület

A jelen második munkaszakaszban megterveztük és kifejlesztettük az E-Hosztesz rendszer multi-modális felhasználói felületeit. Ez a multi-modális alrendszer szemet, szájat és fület ad a rendszerben megvalósított virtuális embernek, ily módon megteremtve az emberhez közelebb álló közvetlen kommunikációt és dialógust. A legfontosabb bemenet a látó- és képfeldolgozó rendszer, amely kultúrától és nyelvtől függetlenül képes a felhasználó reakcióit érzékelni, valamint azokat olyan belső szimbolikus ábrázolásmóddá átalakítani, ami az E-Hosztesz viselkedést és reakcióit meghatározza.

A látórendszer megvalósításakor az *Intel* cég által kifejlesztett *OpenCV* rendszerre támaszkodtunk. Ennek segítségével valósítottunk meg többek között az arcdetektáló és követő algoritmusokat, illetve a látás más alapfunkcióit is. Konkrétabban, egy a számítógép monitor tetején elhelyezett webkamera segítségével folyamatosan figyeljük a képernyő előtt elhelyezkedő felhasználót. Ha az ily módon beolvasott képen nem található egyetlen arc sem,

a virtuális hosztesz úgy tesz, mintha egyedül lenne. Abban a pillanatban, hogy ha megjelenik valaki a kamera látóterében, ez az információ automatikusan elindítja az érdemi tartalom lejátszását és a dialógus folyamatát. Az arcfelismerő program a képmézőn belül akár egyszerre több arcot is megtalál, vagy követni tud. Ennek segítségével a virtuális hosztesz képes folyamatosan a felhasználó szemébe nézni, vagy mozgását követni, vagy érzékelni, ha elmegy. Hasonlóképpen, több arc megjelenésekor prezentációs stratégiát vált, azaz az „egy-az-egy”-ben történő beszélgetés helyett előadás módba kerül, és mindkét partnerre ránéz, odafigyel. Az így kialakított dialógusrendszer ily módon képes lesz a felhasználóban azt az érzést kelteni, hogy rá reagál a virtuálisan létrehozott E-Hosztesz.

4.5 Külső képszintézis modulok

Sok gyakorlati alkalmazásnál szükség lehet arra, hogy az E-Hosztesz animációs rendszer által valós időben, interaktívan generált képeket tároljuk, videó formájában rögzítsük. Ezért a rendszert felkészítettük arra, hogy AVI formátumban képes legyen kisfilmeket készíteni. Az ily módon létrehozott képi minőség azonban erősen limitált a valós időben alkalmazható képszintézis technikák miatt, azaz nem tartalmazhat olyan fény modelleket, mint pl. a globális illumináció, vagy finomabb anyag- és fény-visszaverődési tulajdonságok. Ezért fontosnak tartottuk egy olyan program- és adatfelület kialakítását is, ahol a rendszerben létrehozott és mozgatott vizuális elemeket külső képszintézis modulokkal is összekapcsolhatjuk. Ennek megvalósításához az animációs iparágban elterjedt *material* (mtl) és *OBJ* geometriai formátumokat használtuk. Az így kialakított export modul közvetlenül képes a Maya, illetve 3D Studio Max programokba beolvasható formátumok létrehozására. A következő munkaszakaszban ezt a modult olyan módon tervezzük tovább fejleszteni, hogy a fent említett professzionális modellező és animációs rendszerekkel minél magasabb integrálhatósági szintet érjünk el.

5. Cikkek és publikációk

A projekt második fázisában több hazai és nemzetközi cikk illetve publikáció készült. Ezek a folyamatosan fejlődő E-Hosztesz rendszer gyakorlati alkalmazásairól, valamint a kifejlesztés alatt álló technikai elemekről, azok szellemi háttéréről szólnak. Az E-Hosztesz rendszer egy korai verzióját jelenleg a SOTE Pszichiátriai Klinikáján, Budapesten teszteljük. A publikációk elsősorban a virtuális arcokon megjelenített érzelmek és arckifejezések analízisével foglalkoznak. Az ide vonatkozó publikációk az alábbiak:

- Csukly Gábor, Takács Barnabás, Simon Lajos (2005), „Érzelemkifejező arckifejezések felismerésének összehasonlító vizsgálata”, Magyar Pszichiátriai Társaság Jubileumi Vándorgyűlése, Jan. 25-27, Budapest.
- Simon Lajos, Csukly Gábor, Takács Barnabás (2005), „Facial Expression Recognition in Psychiatric Disorders Using Animated 3D Emotional Facial Expressions”, in CyberTherapy Conference, June 6 - 10, 2005 Basel, Switzerland (elbírálás alatt).

6. Mellékletek

RÉSZLETES PROGRAM TERV
E-HOSZTESZ INFORMÁCIÓS KIOSZK

Munkaszakasz: 2

Beszámolási időszak: 2004. május 1.-2005. január 31.

Szervezet: VerAnim Számítástechnikai és Prod. Iroda Bt.

Projektvezető: Takács Barnabás

Honlap: www.digitalElite.net/Pages/VerAnim/Ehosztész.html

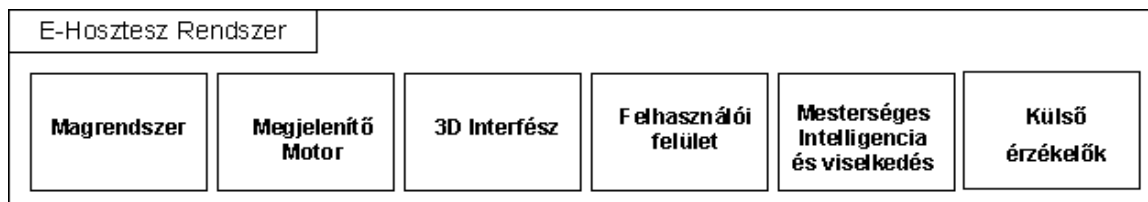
INFORMÁCIÓS ÉS KOMMUNIKÁCIÓS TECHNOLÓGIÁK PROGRAM

NYILVÁNTARTÁSI SZÁM: OMFB-00238/2004

Bevezetés:

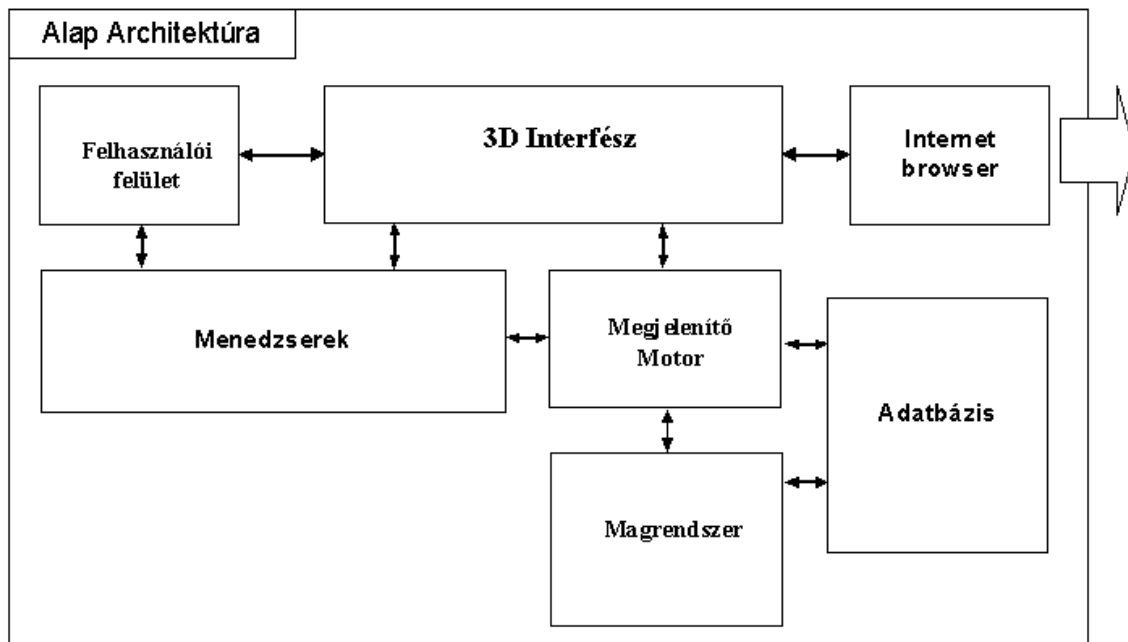
Ez a dokumentum összefoglalja az E-Hosztesz projekt megvalósításához szükséges részletes program modul terveket és programozási felületeket. Az E-Hosztesz architektúrában egy digitálisan létrehozott „virtuális” arc párbeszédben, illetve dialógusban vesz részt a felhasználóval. A rendszer intelligenciáját az E-Hosztesz „látó és halló rendszere” vezérli, amely egy arcfelismerő modul segítségével érzékeli, amikor a felhasználó megjelenik, majd az arc mozgását követve képes szemkontaktust tartani, és az adott kontextusnak megfelelően a beszédet különböző arkifejezésekkel követni. Ennek lefolyását a mesterséges intelligencia modul vezérli. A fenti funkciók megvalósításához több program modul tervét dolgoztuk ki. Ezek funkcióját az 1 ábrán foglaltuk össze.

- a. *Magrendszer*: A virtuális ember geometriáját (csontrendszerét, izomzatát, bőrét, haját, stb.) az interakció és a bemeneti paraméterek függvényében másodpercenként 30-szor újra számító program modul.
- b. *Megjelenítő Motor*: A deformációs és interaktív megjelenési paraméterek alapján az E-Hosztesz mozgását meghatározó egység.
- c. *3D Interfész*: A grafikus kártyát meghajtó driver, az alkalmazásban DirectX9 vagy Open GL.
- d. *Felhasználói felület*: Új tartalom tervezését, tesztelését és megvalósítását segítő grafikus felület.
- e. *Mesterséges Intelligencia és Viselkedés*: A hosztesz reakcióit, belső állapotát, valamint cselekedeteit és reakcióit vezérli.
- f. *Külső érzékelők*: A virtuális embert érzékszervekkel ruházzuk fel, hogy a mindennapi élethez hasonló módon tudjon a felhasználóval kommunikálni. Ebben a modulban valósul meg a mesterséges látás, elsősorban az arcfelismerés, valamint a beszéd felismerés is.



1. ábra: Az E-Hosztesz információs kiosk rendszer főbb elemei.

Az E-Hosztesz rendszer tovább részletezett alaparchitektúráját, a 2. ábra mutatja. Az itt szereplő három új program modul szerepe a külvilággal történő Internetes kommunikáció (Internet Browser), a felhasználói és belső folyamatoknak az elérhető erőforrások függvényében történő ütemezése és kezelése (Menedzserek), valamint egy belső adatbázis szolgáltatásainak megvalósítása, amely az egyes rendszerelemek működését összehangolja. Ebben az adatbázisban helyezkednek el az E-Hosztesz megjelenését meghatározó geometriai elemek, a viselkedéshez szükséges mozgás minták és egyéb rendszer paraméterek is.



2. ábra: A főbb modulok belső kapcsolódása.

A rendszer tovább lebontott részletes modul tervét az 1. munkaszakasz lezárásakor benyújtott „RÉSZLETES PROGRAM SPECIFIKÁCIÓ” című dokumentum tartalmazza. Az alábbiakban a fent bemutatott főbb program részek szubrutin szintű definícióját írjuk le.

Rendszer Elemek:

Az E-Hosztész program gyakorlati megvalósítása során az alábbi főbb program modulokat definiáltuk:

- *EH_CORE*: Az E-Hosztész alkalmazás alap magrendszere.
- *EH_ENGINE*: A 3D megjelenítéshez szükséges főbb rutinok.
- *EH_AI*: A dialógust vezérlő mesterséges intelligencia felület.
- *EH_Vision*: Mesterséges látás, arcfelismerés és képfeldolgozás.
- *EH_HumanIO*: Külső érzékelőket és hardver eszközöket illesztő felület.
- *EH_GUI*: A felhasználói felületet megvalósító program kód.
- *EH_API*: 3D grafikus kártyák támogatása (DirectX 7,8,9, Open GL).

A következő oldalakon a fenti modulok legfontosabb elemeinek specifikációja található. Minden rutinnál szerepel a rutin neve, a funkció és használat rövid leírása, valamint a rutin be- és kimenetei.

EH_CORE – program rutin specifikáció

Név: EH_Body()

Leírás: Az E-Hosztesz kinematikai láncát, a csontokat és az azokhoz kapcsolódó 3-dimenziós felszíni elemeket valósítja meg. A kinematika lánc hierarchikus felépítése lehetővé teszi, hogy az E-Hosztesz animálható, azaz csontonként mozgatható legyen.

Bemenet: Minden egyes csont hat szabadságfokú (3 rotációs és 3 translációs) mozgásának leírása, öröklési hierarchia, csukló forgatási sorrend.

Kimenet: Az E-Hosztesz csontvázának végleges állapota, a csontok kezdő és végpontjainak 3D koordinátái.

Név: EH_Graphics()

Leírás: Az E-Hosztesz geometriai megjelenését vezérlő modul. Itt valósul meg a csontokhoz erősített 3D geometria felszín pontonként történő mozgatása, deformációja és megjelenítése.

Bemenet: Az *EH_Body()* modul által kiszámított csont pozíciók.

Kimenet: A csontokhoz erősített felszíni elemek, pl. bőr, ruha, haj, ékszerek, kellékek, stb. felületek minden pontjának 3D megjelenítése.

Név: EH_Morphs()

Leírás: Az alap bőrfelszín lokális deformációját megvalósító transzformáció. Segítségével valósítjuk meg a finom arcmozgásokra jellemző apró mimikai jeleket, illetve a bőr alatt meghúzódó izomzat felszíni hatását.

Bemenet: Az *EH_Graphics()* rutin által kiszámított felszíni geometria minden pontjának pozíciója.

Kimenet: A lokális deformációnak megfelelő módosított pontonkénti 3D pozíciók.

Név: EH_Materials()

Leírás: Az E-Hosztesz megjelenését befolyásoló, a bőr, szemek, haj és ruházat anyagi tulajdonságait megvalósító képszintézis modul.

Bemenet: Az *EH_Body()*, *EH_Graphics()* és *EH_Morphs()* modulok által generált háromdimenziós felszínek és objektumok, valamint az anyagi tulajdonságokat meghatározó, illetve vezérlő textúrák és időben változtatható paraméterek.

Kimenet: A valósídejű képszintézis algoritmus által kiszámolt pixel intenzitás a képernyő egy adott pontján.

Név: EH_Channels()

Leírás: A csontok, a test és arc felszín, illetve az anyagi tulajdonságok időbeni változását ún. animációs csatornák (angolul *channels*) tárolják. A modul az ezek vezérléséhez szükséges adatformákat és algoritmusokat valósítja meg.

Bemenet: Egy adott időpont, illetve az abban az időpontban fixen megadott paraméter sorozat (csontok pozíciója és elforgatási szöge, arckifejezések és a bőrfelszín deformációja, vagy anyagi jellemzők, mint pl. a szem csillogásának mértéke).

Kimenet: Egy adott időponthoz rendelt adat *n-tuple*, vagy ún. *key frame*.

Név: EH_ParticleSystem()
Leírás: Az E-Hosztesz valóság-hű megjelenéséhez, elsősorban a haj és ruházat esetleges dinamikájának megvalósításához szükséges részecske rendszer szimulátor. A részecskék a szimuláció során egymással kölcsönhatásba lépnek, taszító és vonzó erőkkel hatnak egymásra, vagy esetleg környezetükkel is összeütköznek.
Bemenet: Az *EH_Graphics()* rutin által kiszámolt és deformált hajszál geometria, a részecske szimuláció perem feltételei, a szimulált anyag fizikai tulajdonságai, fellépő külső erők (pl. ütközés más objektumokkal, vagy interakció), szimulációs idő lépték és konstansok.
Kimenet: A digitális hajat, vagy ruhát leíró dinamikai rendszer egy állapota, illetve az ennek az állapotnak megfelelő módosított 3D geometria.

Név: EH_ClothSkin()
Leírás: A részecske rendszerre épülő dinamikus ruha és bőr szimuláció.
Bemenet: Az *EH_Graphics()*, *EH_ParticleSystem()* és *EH_Materials()* rutinok által meghatározott 3D geometria, a ruhára jellemző belső rugalmassági együtthatók.
Kimenet: A mozgó ruha, vagy bőr felszín változásait leíró dinamikai rendszer egy adott időpillanathoz tartozó belső állapota, illetve az ennek az állapotnak megfelelő módosított 3D geometria.

Név: EH_Hair()
Leírás: A részecske rendszerre épülő dinamikus haj szimuláció.
Bemenet: Az *EH_Graphics()*, *EH_ParticleSystem()* és *EH_Materials()* rutinok által meghatározott 3D haj geometria hajszálakkal modellezve.
Kimenet: A mozgó haj változásait leíró szimulációjához tartozó belső állapot, az ennek az állapotnak megfelelő módosított 3D geometriával.

Név: EH_FaceDesigner()
Leírás: Az E-Hosztesz arcának megjelenését a felhasználó ezzel az ún. arctervező modullal tudja módosítani.
Bemenet: Az *EH_Graphics()* és *EH_Morphs()* rutinok által kiszámolt arc felszín.
Kimenet: Új arc geometria.

Név: EH_Tracks()
Leírás: Az animációs rendszer minden időben változó paramétereit kezelő adat struktúra és az ezekhez tartozó operátorok. Az E-Hosztesz mozgásának, viselkedésének és megjelenésének vezérlése mellett, ez a modul felel a 3D szintetikus környezetben megjelenő összes vizuális és audiális elem időzítésére.
Bemenet: Az *EH_Channels()* modul által meghatározott mozgás, megjelenítés és képszintézis paraméterek.
Kimenet: Az interakciót és képszintézis algoritmust vezérlő, időben folyamatosan változó animációs paraméterek.

Név: EH_Networking()
Leírás: Az E-Hosztesz rendszer külső világgal történő összeköttetését megvalósító protokoll interfész.
Bemenet: Külső eszközöktől és belső processzektől kapott kommunikációs parancsok, adatok, és adat struktúrák, a kommunikációs csatorna paraméterei.
Kimenet: Két program, vagy külső eszközök és az E-Hosztesz rendszer közötti adatcsere.

Név: EH_DataBase()
Leírás: Az E-Hosztesz rendszer vezérlését meghatározó központi adatbázis vezérlő felülete. Az adatbázis tartalmazza a virtuális ember geometriáját, viselkedés mintáit, a szintetikus környezet elemeit, illetve az interaktív dialógus lefolyásának paramétereit. Az interakció során keletkező eredményeket és megfigyeléseket szintén itt tároljuk.
Bemenet: Minden, a program működésével kapcsolatos adat
Kimenet: A program számára SQL felületen lekérdezhető paraméterek.

Név: EH_AnimPackage()
Leírás: Az E-Hosztesz rendszer, a valós idejű képszintézis megvalósítása mellett, több professzionális animációs program csomagot (Maya, 3D Studio Max) is támogat. A virtuális ember megjelenítéséhez és mozgatásához szükséges rendszer specifikus rutinok valósulnak meg itt. Ezeket a program fordításakor compiler direktívák segítségével lehet beállítani.
Bemenet: Minden megjelenítő és interakciót kezelő rutin kimenete.
Kimenet: Az adott külső programban értelmezett üzenetek és parancsok.

EH_ENGINE – program rutin specifikáció

Név: EH_VirtualHumanInterface()
Leírás: E-Hosztesz rendszer alsóbb szintjein megvalósított rutinok egységesített belépő pontja és adatcsere felülete.
Bemenet: Minden az *EH_Core*, *EH_Vision*, *EH_HumanIO*, *EH_AI* alrendszerekben található rutin.
Kimenet: Az *EH_Engine()*, *EH_MotionServer()* és *EH_SpeechAnim()* moduloknak megfelelő adat struktúrák.

Név: EH_EnginePriorityQueue()
Leírás: Az E-Hosztesz virtuális ember viselkedését és reakcióit harmonizáló, azokat prioritási sorrendbe helyező modul.
Bemenet: Az *EH_VirtualHumanInterface()*, *EH_AI()* és *EH_Vision()* és *EH_HumanIO()* modulokon keresztül elérhető animációs, viselkedési és szenzor paraméterek.
Kimenet: A virtuális ember egy adott konfigurációja (pl. test pozíció, arckifejezés, mozdulat vagy reakció).

Név: EH_MotionServer()
Leírás: Az E-Hosztesz az interakció és dialógus folyamata során a reakciókat megvalósító mozgások egy részét a belső adatbázisból veszi. E modul feladata egy adott pillanatban a virtuális ember „kérésére” az elérhető, illetve megvalósítható (plauzibilis) mozgások halmazát visszaadni. A hosztesz minden pillanatban ezekből választ reakcióinak végrehajtása során.
Bemenet: A virtuális ember belső állapota és környezeti változói.
Kimenet: Az adott pillanatban megvalósítható mozgás sorozat kódja az *EH_Tracks()* modul számára érthető formában.

Név: EH_SpeechAnim()
Leírás: A dialógus folyamata során speciális szerephez jut a beszéd automatikus animációja. A modul feladata egy adott hangsorhoz megfelelő szájmozgás létrehozása.
Bemenet: Beszédet leíró szöveg, fonéma sor, vagy hang fájl.
Kimenet: Az E-Hosztesz szájmozgását vezérlő animációs paraméterek az *EH_Tracks()* modul számára érthető formában.

EH_AI – program rutin specifikáció

Név: EH_LowLevelBehaviour()
Leírás: Az E-Hosztesz alacsony szintű, ösztönszerű reakcióit megvalósító program modul.
Bemenet: Minden az *EH_Vision*, *EH_HumanIO*, és az *EH_AI* alrendszeréken keresztül rendelkezésre álló külső benyomás.
Kimenet: Az E-Hosztesz testmozgását és arckifejezéseit vezérlő animációs paraméterek az *EH_Tracks()* modulon keresztül megjelenő formában.

Név: EH_GoalSeeker()
Leírás: Az E-Hosztesz közép szintű céljait megvalósító modul. A virtuális ember alkalmazásban feladata a dialógus folyamatosságának fenntartása és a megkívánt produkciós, információs, vagy oktatási tartalomnak a felhasználóhoz történő optimális eljuttatása.
Bemenet: Minden az *EH_Vision*, *EH_HumanIO*, és az *EH_AI* alrendszeréken keresztül rendelkezésre álló külső benyomás, illetve az *EH_DataBase()*-ben tárolt tartalom és interakciós gráf.
Kimenet: Az virtuális ember mozgását *EH_LowLevelBehaviour()*, *EH_Tracks()* vezérlő animációs paraméterek.

Név: EH_AICommunication()
Leírás: Az E-Hosztesz viselkedése és reakciói mögött rejlő klasszikus, szabály alapú mesterséges intelligencia architektúra, adat struktúrák és kommunikáció megvalósítása.
Bemenet: Minden program és környezeti változó.
Kimenet: Az virtuális ember mozgását *EH_LowLevelBehaviour()*, *EH_Tracks()* vezérlő animációs paraméterek.

Név: EH_ArtificialEmotion()
Leírás: A racionalitást megkerülő mesterséges érzelmeket megvalósító program modul. Az E-Hosztesz kedvét és belső „lelki állapotát” egy több állapotú *Markov modellel* ábrázoljuk, ahol az egyes állapotok közötti átmenetet a felhasználó reakciói és az interakció folyamata modulálják.
Bemenet: Minden program és környezeti változó.
Kimenet: A virtuális ember belső érzelmi állapota illetve azok keveréke (semleges, boldog, szomorú, undorodott, féltő, mérges, kétségbeesett).

Név: EH_CognitiveModule()
Leírás: Az E-Hosztesz magasabb szintű globális viselkedését, a hosszú távú belső célok és a rövid távú reakciók összehangolását egy belső kognitív modul szabályozza. Ezen modul megvalósításakor a *SOAR* architektúrát használjuk.
Bemenet: Minden környezeti és érzékelés változó.
Kimenet: A E-Hosztesz belső kognitív állapota és ennek megjelenése az *EH_LowLevelBehaviour()*, *EH_Tracks()* animációs paraméterek formájában.

EH_Vision – program rutin specifikáció

- Név:** EH FaceDetector()
Leírás: A mesterséges látó rendszer speciális funkciót betöltő modulja, amely a web-kamerából érkező képen arcokat és szempárokat keres. Ennek célja, hogy az E-Hosztesz a terminál előtt jelenlevő személyek számának függvényében tudjon reagálni, illetve szükség esetén képes legyen szemkontaktust teremteni. Megfelelő arc adatbázis használatával a rendszer a felhasználó azonosítására is képes.
- Bemenet:** Videó input.
Kimenet: A képen található arcok száma, azok koordinátái és azonosítója, a megtalált, illetve felismert arcok az *EH_MarkerServer()* alrendszeren keresztül válnak a magasabb szintű kognitív modulok számára láthatóvá.
-
- Név:** EH ImageProcessor()
Leírás: Általános célú mesterséges látó- és képfeldolgozó rendszer, amely a látás alap funkcióit (pl. mozgásérzékelés, újdonság detektálás, követés, stb.) látja el.
- Bemenet:** Videó input.
Kimenet: Az egyes algoritmusok kimenetét képező feldolgozott, módosított képek és *EH_MarkerServer()* adatok.
-
- Név:** EH MarkerServer()
Leírás: Az E-Hosztesz animációs rendszer ún. *markerek* segítségével ábrázolja a külvilág eseményeit, a virtuális ember számára érthető szimbolikus formában. Marker lehet bármely objektumhoz erősített dimenzió nélküli adatsomag. Az E-Hosztesz látó- és képfeldolgozó rendszerben a külvilág egy virtuális monitoron jelenik meg, ahol a képfeldolgozás eredményei (pl. egy felismert arc) 3D markerek formájában jelennek meg.
- Bemenet:** Videó image, 3D objektum.
Kimenet: E-Hosztesz animációs rendszer marker adat reprezentáció.
-
- Név:** EH FeatureTracker()
Leírás: A felhasználó arcán megjelenő események (nem-verbális kommunikáció) feldolgozását elősegítő program modul, amely egyes pontok (pl. száj sarok, orr, pupilla, stb.) mozgását képes követni.
- Bemenet:** Videó input.
Kimenet: A követett arcpontok pozíciója *EH_MarkerServer()* adatok formájában.
-
- Név:** EH FacialExpressionAnalysis()
Leírás: A felhasználó arcán megjelenő eseményeket elemző modul, amely az érzelmek, figyelmi szint és fáradtság szintjét próbálja érzékelni.
- Bemenet:** *EH_FeatureTracker()* adatok.
Kimenet: Szimbolikus marker paraméterek *EH_MarkerServer()* adatok formájában.

EH_HumanIO – program rutin specifikáció

Név: EH_DataBaseInterface()
Leírás: Az E-Hosztesz rendszer működését vezérlő adatbázis felület. A megvalósításban az *SQL adat modellre* és a *Microsoft Access* adatbázis kezelőre támaszkodunk.
Bemenet: SQL query.
Kimenet: SQL kimeneti változók.

Név: EH_CameraController()
Leírás: A mesterséges látó modulhoz kapcsolódó külső aktív robot kamerát vezérlő program felület. Az E-Hosztesz rendszer képes a hozzá kapcsolt kamera *pan*, *tilt* és *zoom (PTZ)* paramétereit a „látottak” függvényében módosítani.
Bemenet: Videó.
Kimenet: Kameravezérlő paraméterek (PTZ) a soros porton keresztül (COM).

Név: EH_ScriptInterpreter ()
Leírás: Általános célú belső script interpreter a virtuális világ és az E-Hosztesz parancsokkal történő vezérléséhez. A megvalósításban a *Nebula* architektúrát használjuk.
Bemenet: Script parancs sorozat.
Kimenet: Belső rendszerváltozók és függvényhívások.

Név: EH_Script_TCLTK()
Leírás: *TCL/TK* script interpreter a külső rendszerekkel digitális hálózaton (pl. Internet) keresztül történő kommunikáció támogatására. A megvalósításban a *Nebula* architektúrát használjuk.
Bemenet: *TCL/TK* script parancs sorozat.
Kimenet: Belső rendszerváltozók és függvényhívások.

Név: EH_Script_LUA()
Leírás: *LUA* script felület egy hatékony valósídejű script rendszer megvalósításához.
Bemenet: *LUA* script parancs sorozat.
Kimenet: Belső rendszerváltozók és függvényhívások.

Név: EH_USBDeviceDriver()
Leírás: Külső perifériális eszközök illesztését lehetővé tevő modul.
Bemenet: USB-alapú Human Interface Device.
Kimenet: Belső rendszerváltozók és függvényhívások.

EH_GUI – program rutin specifikáció

Név: EH_BoneAnimDlg()
Leírás: A virtuális ember (E-Hosztesz) csontjainak kézi beállítását és animációját lehetővé tevő felhasználói felület. Minden csont hat transzformációs és három a geometriát módosító szabadsági fokkal rendelkezik.
Bemenet: Az E-Hosztesz csontrendszer pillanatnyi állapota.
Kimenet: Az E-Hosztesz csontrendszer módosított, új állapota az *EH_Tracks()* modul számára érthető formában.

Név: EH_FaceDesignerDlg()
Leírás: Az E-Hosztesz arctervező modul (*EH_FaceDesigner()*) felhasználói felülete.
Bemenet: Az *EH_Graphics()* és *EH_Morphs()* rutinok által kiszámolt jelenlegi arc felszín, interaktív felhasználói parancsok és paraméterek.
Kimenet: Új virtuális arc geometria.

Név: EH_SceneManagerDlg()
Leírás: A digitális hoszteszt körülvevő 3D virtuális világ (angolul *scene*) elemeinek és azok tulajdonságainak megváltoztatására szolgáló felhasználói felület.
Bemenet: Minden virtuális elem és objektum, illetve azok képi megjelenést befolyásoló paraméter.
Kimenet: Az egyes 3D elemek új állapota.

Név: EH_MultiTrackManagerDlg()
Leírás: Az *EH_Tracks()* modul által kezelt virtuális elemek kezelésére szolgáló felhasználói felület.
Bemenet: Minden az E-Hosztesz animációs rendszer időben változtatható objektum és paraméter.
Kimenet: Az egyes objektumok, vizuális elemek, illetve az E-Hosztesz egy következő állapota.

Név: EH_TimeSliderDlg()
Leírás: Egy magnetofon, vagy CD lejátszó gombjait utánzó felhasználói felület, amely segítségével a rendszeridő változtatható.
Bemenet: A felhasználó által meghatározott időpont.
Kimenet: Az animációs rendszer globális állapota az adott időpontban.

Név: EH_RenderManagerDlg()
Leírás: A nem valósídejű külső képszintézis eljárásokat támogató felhasználói felület, ahol az elkészült interaktív dialógus AVI videó fájlba menthető.
Bemenet: Az *EH_Tracks()* modul és a felhasználó által meghatározott idő paraméterek.
Kimenet: AVI formátumú videofájl, vagy képsorozat.

Név: EH_ImageProcessorDlg()
Leírás: A mesterséges látást és a valós idejű képfeldolgozást megvalósító modul (*EH_Vision*) felhasználói felülete.
Bemenet: Videó bemenet.
Kimenet: Az egyes algoritmusok kimenetét képező feldolgozott, módosított képek és *EH_MarkerServer()* adatok.

EH_API – program rutin specifikáció

Név: EH_ActiveX()
Leírás: Az E-Hosztesz programrendszer Interneten keresztül történő használatát támogató *ActiveX* beágyazás. Segítségével az alkalmazás egy HTML weboldalba beágyazható, és az oldalon megjelenített gombok, valamint egyéb vizuális elemeken keresztül *Java script* segítségével közvetlenül vezérelhető.
Bemenet: E-Hosztesz ablak, HTML interfész leírás.
Kimenet: Weboldalba beágyazott alkalmazás.

Név: EH_Direct3D7()
Leírás: Direct3D-7 grafikus programozói felület (API) régebbi számítógépek és grafikus kártyák támogatására.
Bemenet: E-Hosztesz programrendszer valós idejű képszintézis modul.
Kimenet: Képi megjelenítés.

Név: EH_Direct3D8()
Leírás: Direct3D-8 grafikus programozói felület (API) átlagos teljesítményű számítógépek és grafikus kártyák támogatására.
Bemenet: E-Hosztesz programrendszer valós idejű képszintézis modul.
Kimenet: Képi megjelenítés.

Név: EH_Direct3D9()
Leírás: Direct3D-9 grafikus programozói felület (API) a legmodernebb nagy teljesítményű számítógépek és grafikus kártyák támogatására.
Bemenet: E-Hosztesz programrendszer valós idejű képszintézis modul.
Kimenet: Képi megjelenítés.

Név: EH_OpenGL()
Leírás: OpenGL grafikus programozói felület (API) grafikus kártyák támogatása.
Bemenet: E-Hosztesz programrendszer valós idejű képszintézis modul.
Kimenet: Képi megjelenítés.

**ELŐZETES FELHASZNÁLÓI DOKUMENTÁCIÓ
E-HOSZTESZ INFORMÁCIÓS KIOSZK**

Munkaszakasz: 2

Beszámolási időszak: 2004. május 1.-2005. január 31.

Szervezet: VerAnim Számítástechnikai és Prod. Iroda Bt.

Projektvezető: Takács Barnabás

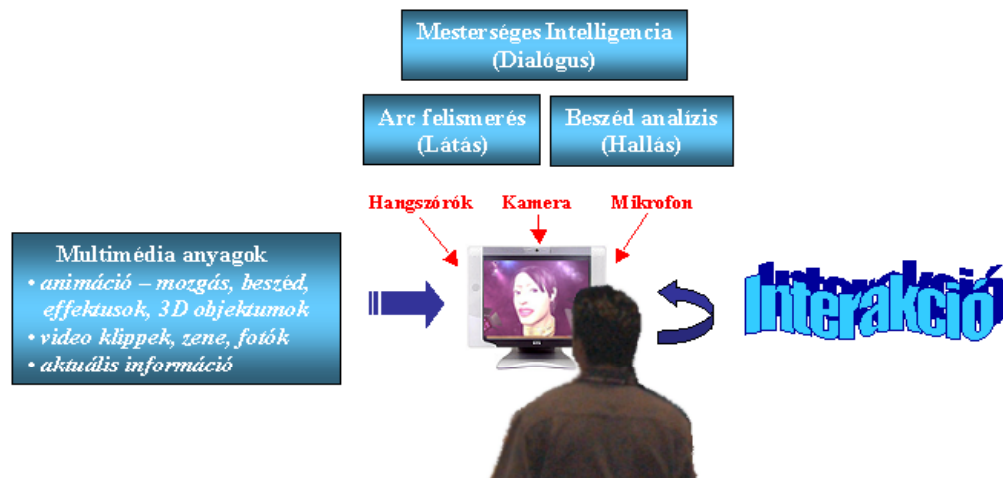
Honlap: www.digitalElite.net/Pages/VerAnim/Ehosztesz.html

INFORMÁCIÓS ÉS KOMMUNIKÁCIÓS TECHNOLÓGIÁK PROGRAM

NYILVÁNTARTÁSI SZÁM: OMFB-00238/2004

1. Bevezetés

Az *E-Hosztesz* rendszer egy nagy teljesítményű képszintézis és képfeldolgozási eszköz, amely a számítógépes animáció segítségével megjelenített digitális hosztesz és a felhasználó között interaktív kapcsolatot és interfészt hoz létre. Az *E-Hosztesz* a világon elsőként teremti meg a felhasználó és digitális ember társa közti interaktív dialógus élményét. A valósidejű rendszer legfontosabb elemeit és felépítését mutatja az 1. ábra. A számítógéppel animálható *E-Hosztesz* ugyanakkora méretben jelenik meg a monitoron, vagy egy fali kivetítőn, mint egy valódi emberi fej. Az így megjelenített digitális arc képes az érzelmek és a nem verbális jelek kifejezésének széles skálájára. Ezen belül több mint hatvan kifinomult arckifejezésre, beszédre, valamint az ezeket kísérő fej- és szemmozgásokra is. A beépített webkamera szállítja a képeket a mesterséges látó alrendszerbe, amely észleli és kielemez az arcokat, ahogy elhaladnak, vagy megjelennek az *E-Hosztesz* kiosk előtt. Az arcokon túlmenően a látó rendszer a felhasználó más cselekedetei is észleli, feldolgozza, majd az ily módon érzékelt információt közvetlenül a mesterséges intelligencia (*Artificial Intelligence – AI*) egységbe küldi, ahol a virtuális *E-Hosztesz* reakciói és viselkedése ennek megfelelően adaptálható. Ez a kölcsönhatás és zárt kommunikációs lánc a mindennapok beszélgetéseiben előforduló természetes párbeszédet utánozza. Célja, hogy interaktív módon olyan hasznos tartalmat és információt juttasson el a felhasználóhoz, amely szórakoztató, izgalmas, nagyszerű és érdemes rá emlékezni. Az *E-Hosztesz* rendszer ezáltal tehát képes arra, hogy a felhasználó a lehető legkönnyebben meg tudja jegyezni a hallottakat. Ezt az érzelmi kötődés, illetve pontosabban kifejezve *érzelmi modulációt*, az interaktív élményt kiegészítő multimédiás tartalom (animáció, széles arckifejezések repertoárja, beszéd, különböző vizuális effektek, valóság-hű 3-dimenziós videó klippek, zene) valósítja meg. Az ehhez szükséges elemeket a rendszer egy dinamikus frissíthető és új tartalommal feltölthető külső adatbázisban tárolja.



1. ábra: Az *E-Hosztesz* információs kiosk rendszer főbb elemei.

Az E-Hosztesz interaktív információs kiosk rendszer több magas szintű és összetett program modulból áll, amelyek önmagukban is technikailag kimagasló és páratlan teljesítményt tesznek lehetővé. Ezek röviden a következők:

1. Egy nagy teljesítményű vizualizációs motor, amely képes - másodpercenként 30 képkocka megjelenítése mellett is - több mint 300,000 poligon méretű 3 dimenziós objektumokat megjeleníteni.
2. Egy valósidejű képfeldolgozó egység, amely folyamatosan dolgozza fel a webkamerából érkező képeket, felismeri az arcokat, egyes tárgyakat, valamint nyomon követi azok mozgását.
3. Általános bemeneti eszköz felület, amely integrálja a beérkező multi-modális adatáramot, és egységesen kezeli az egyéb külső perifériákat, mint mozgásérzékelők, virtuális kesztyűk, fejre erősíthető virtuális valóság sisak.
4. Egy hálózati protokoll interfész (TCP/IP) sokoldalú programozói felülettel, amely lehetővé teszi a külső rendszerek számára a probléma mentes kapcsolatteremtést, vezérlést és adat struktúrák átadását.

2. A program installálása

Ha installálni akarja az E-Hosztesz interaktív információs kiosk programot, először helyezze be a telepítő CD-t és kattintson kétszer a „Setup.exe” varázsló ikonra. Kövesse a képernyőn megjelenő utasításokat. A folyamat alatt felmerülő esetleges problémákat (pl. kompatibilitási kérdések az Ön rendszerével kapcsolatban) a telepítő program automatikusan, és szükség esetén speciális hibaüzenettel kezeli.

Ha az automatikus telepítés valami oknál fogva nem működik kielégítően, az E-Hosztesz rendszert kézzel kell telepíteni. Ebben az esetben további információért kérjük forduljon ahhoz a technikai csoporthoz, amely az Ön rendszerének konfigurációjával foglalkozik.

2.1. Virtuális rendszer lemez (S: meghajtó) és fájl hierarchia

Az E-Hosztesz rendszer egy speciális virtuális meghajtót használ (S:) azért, hogy elkerülje a konfliktusokat más rendszerforrásokkal. Ez a telepítő program automatikusan létrehoz egy *.BAT* parancsfájlt a Windows StartUp direktoriájában és a *subst* rendszer parancsot használva létrehozza a szükséges virtuális rendszer lemezt (meghajtót). Az így módon létrehozott meghajtó ugyanúgy működik, mint bármely másik lemez a rendszerben. Mielőtt futtatja a programot, kérjük, ellenőrizze, hogy az S: meghajtó valóban létezik-e. Ha valami oknál fogva nem, akkor létrehozhatja a fentebb említett *.BAT* parancs fájl futtatásával, vagy azzal, ha a *DOS* promtnál közvetlenül beírja a következő parancssort:

Subst s: „C:\Program Files\EHostess”

Ennek eredményeként az S: meghajtó létrejön a helyi mappában, ezen a néven:

„C:\Program Files\EHostess”

Megjegyzés: Az E-Hosztesz fájl hierarchiája rögzített. A rendszer helyes működéséhez minden könyvtárnak az S: meghajtón kell lennie. Ha módosítja ezeknek a nevét vagy elérési útját, a rendszerben hiba keletkezhet, egyes program funkciók hibásan vagy csak részlegesen működnek.

2.2. Központi adatbázis (ODBC felület)

Az E-Hosztesz rendszer egy *Microsoft Access* formátumú (.mdb) központi adatbázist használ, amely tartalmazza és irányítja az interakcióhoz, illetve multimédia tartalomhoz kapcsolódó összes elemet (képek, 3D objektumok, animációs adatok, haj és bőr anyagi jellemzőket), valamint a kiosk használatához szükséges sok más fontos elemet. Az alapértelmezés szerinti adatbázis az „S:/SA_Database.mdb” helyen tárolódik. Ha szükséges, Ön ennek az adatbázisnak tartalmát saját igényei alapján módosíthatja, vagy kicserélheti egy másikkal.

2.3. Külső program modulok

Az E-Hosztesz alkalmazás több szabad-felhasználású, ún. *open source software* elemet használ. Ezek közé tartozik a Nebula Device 1, Intel OpenCV képfeldolgozó könyvtár, LUA script interpreter, TCL/TK felület, ActiveX, ODE és SOAR. Ezek a jogcímeik a megfelelő tulajdonosok birtokában vannak és a kereskedelmi felhasználást nyílt licencekkel (*open source*) korlátozzák. Az E-Hosztesz telepítő varázsló ezeket a rendszer elemeket és a szükséges könyvtárakat automatikusan installálja. További információért tekintse meg a felhasználói dokumentáció végén található licence mellékletet.

3. Rendszerkövetelmények

Az E-Hosztesz rendszert a következő operációs rendszerekkel teszteltük: Windows NT 4.x, Windows ME, és Windows 2000 Professional, különböző konfigurációjú és teljesítményű processzorok (CPU) és grafikus kártyák használata mellett. A legjobb teljesítményhez a következő minimális konfigurációt javasoljuk:

Minimális rendszerkövetelmény:

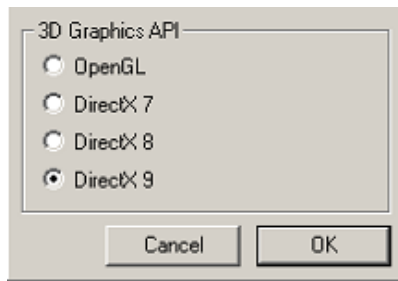
Processzor:	Intel 2.0 GHz
Memória:	512 Mbyte
Merevlemez:	30 Gbyte
Grafikus kártya:	Radeon 9700 vagy GE-Force 4
Hálózati kártya:	10 Mb/s

Megjegyzés: Az E-Hosztesz jelenlegi verzióját Unix vagy Apple OS gépeken még nem teszteltük.

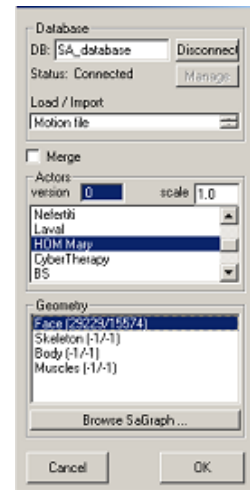
4. A program futtatása

Az E-Hosztesz rendszert sokrétűen lehet használni. A különböző alkalmazásokat a program egységes felhasználói felületén keresztül indíthatjuk. Ennek legegyszerűbb módja, ha duplán rákattintunk az *SAFrame.exe* fájl ikonjára az *S:\SW_DEV\C++\bin* könyvtárban. Először az grafikus felületet kiválasztó ablak jelenik meg (lásd 2. ábra), ahol kiválaszthatja a négy elérhető *API (Application Programmin Interface)* réteg egyikét. Annak függvényében, hogy milyen grafikus kártya van az Ön rendszerében telepítve, a megfelelő felületet kiválasztásával innen szabályozhatja, hogy az E-Hosztesz animációs rendszer grafikus motorja milyen különböző megjelenítési módokat használjon.

Megjegyzés: *Néhány különleges számítógép konfiguráció esetében a rendszer esetenként nem indul helyesen. Ezekben az esetekben kérjük, próbálja újraindítani az alkalmazást és válasszon egy másik grafikai felületet a 2. ábrán látható menüből.*



2. ábra: Az E-Hosztesz információs kioszk grafikus felület kiválasztó dialógus.



3. ábra: Adatbázis kezelő dialógus felület.

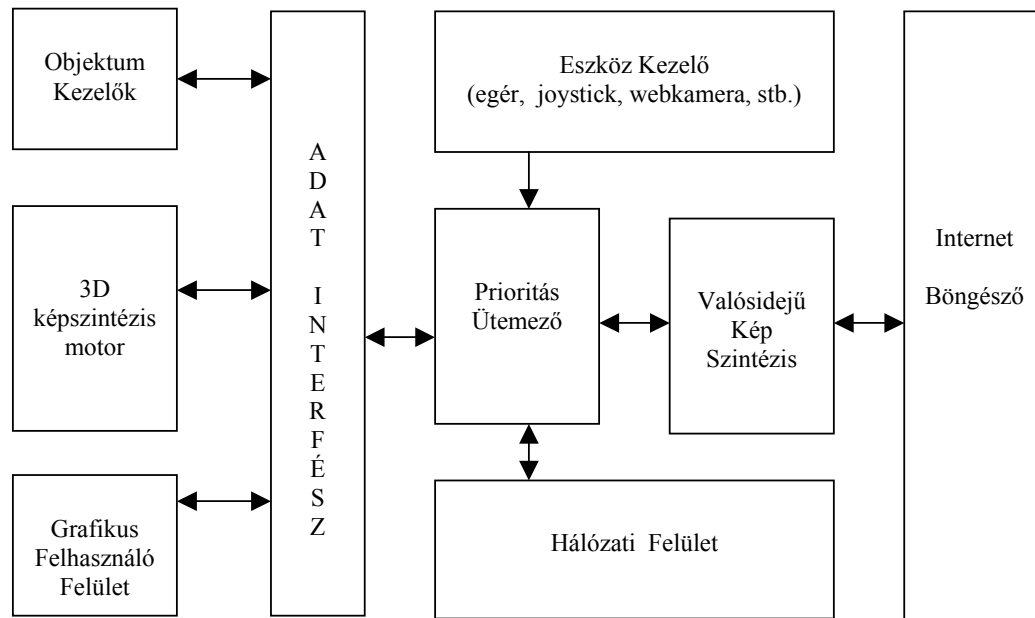
Miután kiválasztotta a kívánt grafikus felületet, a következő lépésben az *Adatbázis Manager* ablak jelenik meg (lásd 3. ábra). Az *Adatbázis Manager* teszi lehetővé, hogy

- hozzákapcsolódjon a különböző alkalmazásokat szolgáló adatbázisokhoz,
- új E-hoszteszeket töltsön be,
- szabályozza azok megjelenését, viselkedését, és mozgás repertoárját.

A program elindításához válasszon egy hoszteszt, vagy területet úgy, hogy kijelöli a megfelelő nevet a középső listából, majd válasszon ki egy verziót az alatta található másik listából. Ha mindkettőt kiválasztotta, nyomja meg az *OK gombot* a jobb alsó sarokban. Ennek hatására a rendszer folytatja az animációs fájlok betöltését, beleértve a megfelelő szintereket, animációs adatokat és más vizuális elemeket.

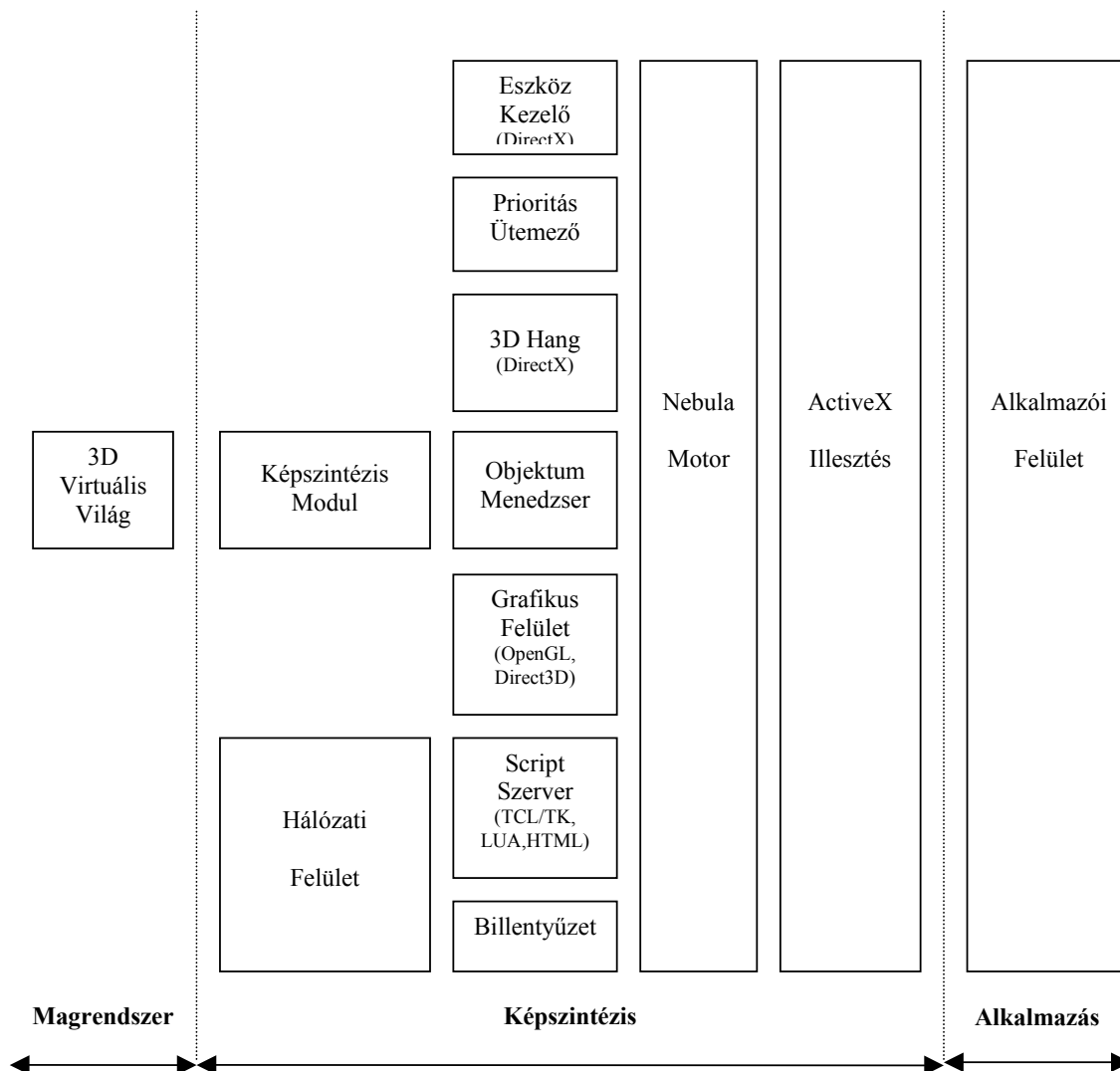
5. Rendszer blokk diagram

Az E-Hosztesz interaktív valós idejű képszintézis rendszer belső felépítése a 4. és 5. ábrákon látható. A 4. ábrán vannak a rendszer magjának legfontosabb részei. Ezek az *Objektum Kezelők*, a *3D Képszintézis Motor* és a *Grafikus Felhasználói Felület* (GUI). A valós idejű 3D képszintézis modul egy belső *Adat Interfészen* keresztül kommunikál a *Prioritás Ütemező* csomóponttal, ami a *Eszköz Kezelőtől* kapott utasítások függvényében közvetlenül vezérli a *Valós idejű Kép Szintézist* megvalósító algoritmusokat. Végül két másik kapcsolódási felület biztosítja, hogy a generált képek és események a külső illeszkedő programok számára is megjelenjenek a *Hálózati Felületen*, illetve az *Internet Böngészőn*.



4. ábra: Rendszer blokk diagram.

Az 5. ábrán a rendszer részletesebb belső felépítése látható. A *Magrendszer* tartalmazza a külvilág ábrázolási módjait, beleértve a 3D-s tárgyakat és más objektumokat úgy, ahogy azt az E-Hosztész „látja” a virtuális környezetén belül. A képszintézis modul több kisebb részhez csatlakozik a hálózati felületen, illetve a belső kommunikációs csatornákon keresztül. A további kiszolgálók felelősek azért, hogy véghezvigyék a képszintézis bonyolult folyamatának minden aspektusát, így vezérlik a grafikus kártyát, fogadják és feldolgozzák a külső eszközökből beérkező jeleket, kommunikálnak a különböző scripteket értelmező rétegekkel (LUA, HTML, XML, TCL), és végül generálják a képernyőn megjelenő képet. A következő réteg felelős az információ Nebula kernel-be juttatásáért, ami az ActiveX rétegen keresztül válik a web böngésző számára is hozzáférhetővé. Végül az utolsó réteg tartalmazza a különböző alkalmazásokhoz tartozó speciális modulokat.



5. ábra: Rendszer blokk diagram (folytatás).

6. Grafikus kártya támogatás (OpenGL/DirectX)

Az E-Hosztesz rendszer támogatja mindkét széles körben elterjedt grafikai szabványt, azaz az *OpenGL* és a *DirectX API (Application Programming Interface)* több változatát. Attól függően, hogy az Ön számítógépe konkrétan milyen hardver szintű megvalósítást támogat, futtathatja az *API* más és más verzióját, ahogy azt az előző fejezetekben már leírtuk. A különböző jelenleg elterjedt szabványok közül E-Hosztesz támogatja mind az OpenGL 1.0-t és 2.x-et, mind a DirectX 7, 8 és 9-es verzióit.

7. Grafikus felhasználói felület (GUI)

A *Grafikus Felhasználói Felület*, vagy más néven *GUI* valósítja meg a felhasználó és az interaktív képszintézis rendszer közötti magas színvonalú kapcsolatot. A *GUI* több rétegekből tevődik össze. Ezeket ún. „menedzsereknek” hívjuk. A menedzserek kezelik az interakció, a képszintézis és a dialógus folyamat minden összetett aspektusát. A jelenleg megvalósított E-Hosztesz program verzióban öt jól elkülönülő menedzser van. Ezek a következők:

1. *Idő Menedzser*: A felhasználó ezen a modulon keresztül állíthatja, vagy változtathatja a képszintézis rendszer belső óráját és az ehhez tartozó paramétereket.
2. *Szintér Menedzser*: Minden 3 dimenziós tárgy és képi elem megjelenítéséért felelős felület, amelyeken keresztül e tárgyak jellemzői időben módosíthatók.
3. *Animációs Menedzser*: Az időtől függő információk és adatok tárolásáért, illetve irányításáért felelős felület.
4. *Képszintézis Menedzser*: Innen lehet AVI videofájlokat létrehozni.

Az alábbi szakaszok röviden bemutatják a fenti modulok legfontosabb jellemzőit, funkcióit és a hozzájuk kapcsolódó interfészek használatát.

7.1. Idő Menedzser (IM)

Az *Idő Menedzser (IM)* interfész (lásd 6. ábra) egy egyszerű, a magnetofon, vagy CD lejátszó vezérléséhez hasonló grafikus felület funkció, amely lehetővé teszi, hogy a felhasználó vagy tervező az animált képsorozatot az idő függvényeként, azaz képkockáról irányítsa. A konkrét megvalósítás olyan billentyűket tartalmaz, mint a Lejátszás, Következő kép, Előző kép, Ugrás az elejére, vagy ugrás a képsor végére. Ezen túlmenően a felhasználó rendelkezésére áll még egy csúszka (slider) felület is, amelynek segítségével lehetőség nyílik egy konkrét rész kiválasztására. Minden alkalmazás más és más hosszúságú lehet. A maximális időt a képi elemek, pl. videofájlok együttes hossza adja meg. Ez az érték azonban szükség esetén kézzel is felülírható, módosítható a jobb oldalon található „*Length*” pipa bejelölésével és a kívánt hossz másodpercben történő értékének beírásával.

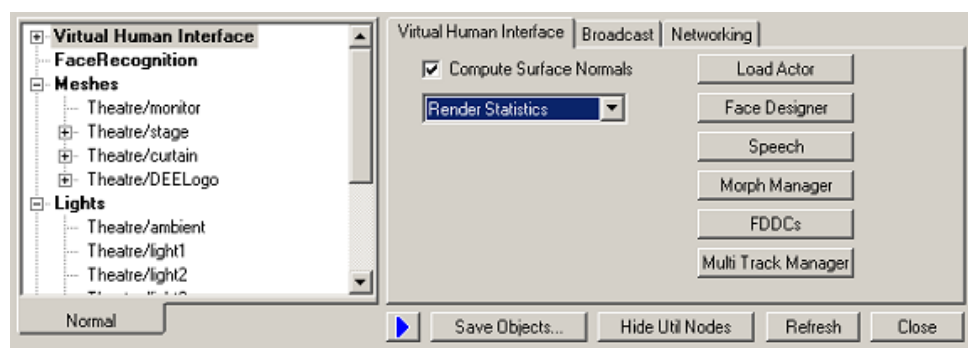


6. ábra: Idő Menedzser felhasználói felület (lásd szöveg).

7.2. Szintér Menedzser (SzM)

A 7. ábra mutatja a **Szintér Menedzser** (SzM) ablak felületet. Az interfész egy, az összes elemet tartalmazó fa szerkezetű navigációs részből (baloldal), és egy munkaterületből áll (jobb oldal).

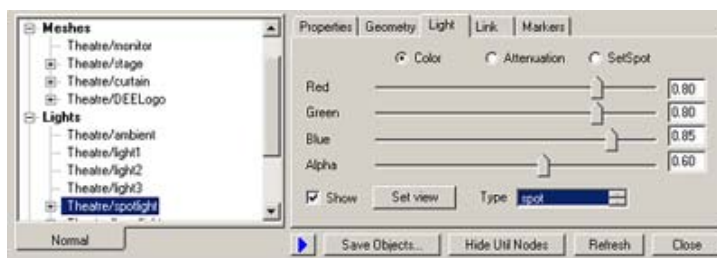
A felhasználó a fában navigálva választhatja ki az egyes tárgyakat és vizuális elemeket, valamint a jobb oldalon megjelenő lapokon módosíthatja a hozzájuk tartozó tulajdonságokat. A rendszer a vizuális elemeket öt különböző kategória szerint csoportosítja. Ezek névszerint az *Objektumok*, *Fények*, *Kamerák*, *Speciális effektusok* és *Hangszórók*. Ezen elemek mindegyike különböző tulajdonságokkal és ennek megfelelően más és más grafikus interfésszel rendelkezik.



7. ábra: Szintér Menedzser felhasználói felület (lásd szöveg).

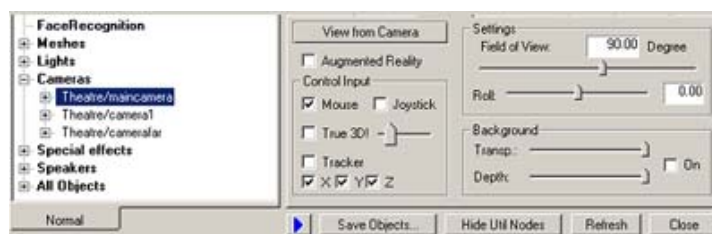
Objektumok: Az E-Hosztész rendszerben használt virtuális tárgyakat 3 dimenziós koordinátákkal rendelkező térbeli pontok listájaként, és az ezeket a pontokat összekapcsoló élek segítségével definiált poligonokként ábrázoljuk. Ezeknek a poligonoknak a segítségével írjuk le minden objektum felszínét. Ezek a felszínek anyagi tulajdonságokkal, színekkel és textúrákkal ruházhatók fel, és jeleníthetők meg.

Fények: A valósídejű képsztézis során digitális fényforrásokat használunk. Ezek digitális ábrázolásakor az E-Hosztész rendszer figyelembe veszi a fények típusát, színét, azok számát, és a relatív távolságát és még sok más időfüggő jellemzőt. A 8. ábrán a *Szintér Menedzser* egy digitális fényforrás (reflektor) tulajdonságait szemlélteti, amit a baloldalon látható fa menüből választottunk ki.



8. ábra: Szintér Menedzser – egy digitális fényforrás felhasználói felülete.

Kamerák: Az E-Hosztesz vizualizációs rendszer lehetővé teszi, hogy a virtuális világot, a képet, több kamerával, vagy nézőpontból szemlélhessük. Ezek a digitális kamerák szabadon mozgathatók, forgathatók, és akár külső eszközökhöz kapcsolva közvetlenül is vezérelhetők. A 9. ábrán látható példa egy kamera tulajdonságainak megváltoztatását mutatja. Természetesen egyszerre több kamera megvalósításával elérhető, hogy a felhasználó ugyanazt a 3D-s képet nézheti több szögből, miközben szabadon válthat közöttük.



9. ábra: Szintér Menedzser – egy virtuális kamera paramétereit.

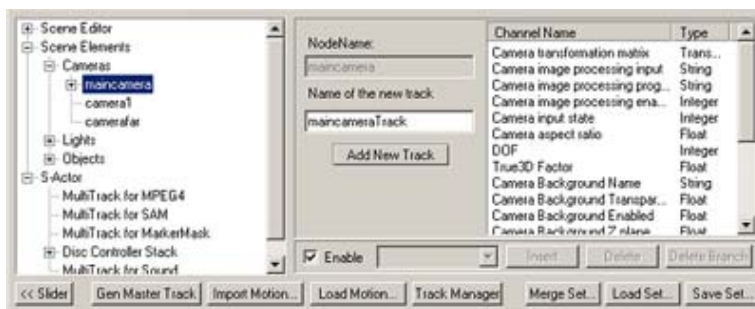
Speciális effektusok: Olyan elemek csoportja, amelyekkel légköri és részecske effektusok hozhatók létre, illetve azok paramétereit dinamikusan változtathatók. Az E-Hosztesz rendszer ezek segítségével valósít meg sok, a valóságérzetet fokozó vizuális elemet, mint pl. a tüzet, vizet, ködöt, vagy hasonló természeti jelenségek szimulációját.

Hangszórók: Az E-Hosztesz rendszerben programozható 3D hangforrásokat is megvalósítottunk, amelyek élethű fizikai tulajdonságokkal rendelkeznek (*Doppler effektus*, térbeli hanghullámok, stb.).

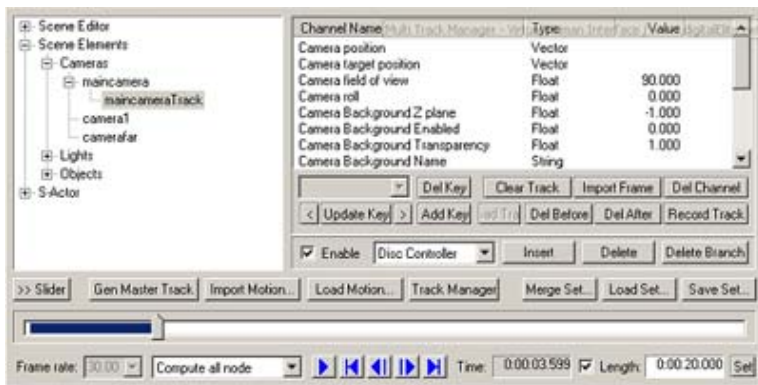
7.3. Animációs Menedzser (AM)

Az E-Hosztesz *Animációs Menedzser* (AM) modul célja, hogy tárolja és irányítsa az időben változó adatokat és információkat, mint pl. a hosztesz arckifejezéseit, testének mozgását, valamint az ezekkel összehangolt multimédia elemeket. Ezeket az időben változó információkat az *Animációs Menedzser* ún. adat csatornákon tárolja. Ezek a változók lehetnek diszkrét, vagy folyamatosak. Diszkrét változókat használhatunk pl. különböző események előidézésére, míg a folyamatos változókat a beépített interpolációs mechanizmus fix pontjait (kulcsok) képezik.

Az *Animációs Menedzser* felhasználói felülete a 10. ábrán látható. A baloldalon található fa navigátor segítségével mozoghatunk, választhatunk a rendszer elemei között. Az ábrán látható példában egy kamerát választottunk ki, amelyhez tartozó animálható paraméterek/változók a jobb oldalon láthatók. Az „Add Track” gomb megnyomásával egy új kameravezérlő csatornát (*track*) hozhatunk létre. A következő 11. ábra szintén az *Animációs Menedzser* használatát szemlélteti.



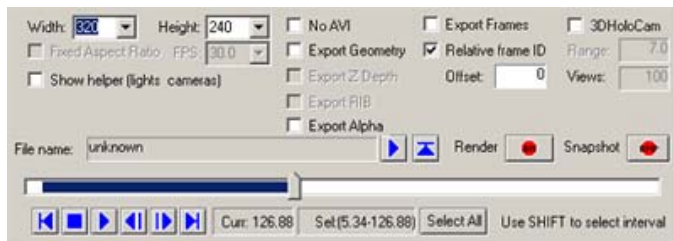
10. ábra: Animációs Menedzser felhasználói felület (lásd szöveg).



11. ábra: Animációs Menedzser felhasználói felület (folytatás).

7.4. Képszintézis Menedzser (KM)

Az E-Hosztész rendszer **Képszintézis Menedzserének (KM)** feladata, hogy más programok és prezentációk számára érthető *AVI* kódolású média fájlt hozzon létre. A 12. ábra ennek megvalósított grafikus felületét mutatja. Ez a dialógus felület legegyszerűbben a jobb oldalt alul található „Snapshot” gomb megnyomásával használható, aminek eredményeképpen a képernyőn éppen megjelenített képet egy *bitmap* (*BMP*) formátumú fájlban tároljuk el. *AVI* formátumú videofájlok létrehozásához először az alul található idő csúszka (*time slider*) használatával kell kiválasztani a kívánt időtartományt, majd megnyomni a „Render” gombot.



12. ábra: Képszintézis Menedzser felhasználói felület.

8. Licencek

8.1. VerAnim/Digital Elite

Szerzői jog © 1999-2005. VerAnim Bt, Budapest, Magyarország és Digital Elite Inc., Los Angeles, CA, USA – minden jog fenntartva.

8.2. Intel License Agreement For Open Source Computer Vision Library

Copyright (C) 2000, 2001, Intel Corporation, all rights reserved. Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistribution's of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistribution's in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The name of Intel Corporation may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

8.3. TCL/TK License Agreement

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

8.4. LUA License Agreement

Lua is licensed under the terms of the MIT license reproduced below. This mean that Lua is free software and can be used for both academic and commercial purposes at absolutely no cost.

Copyright (C) 2002 Tecgraf, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.5. Nebula Device

This software is copyrighted by the Radon Labs GbR, Germany <This are: Andre Weissflog, Andreas Flemming, Bernd Beyreuther (the Authors)> and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The Authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

TESZTELÉSI TERV
E-HOSZTESZ INFORMÁCIÓS KIOSZK

Munkaszakasz: 2

Beszámolási időszak: 2004. május 1.-2005. január 31.

Szervezet: VerAnim Számítástechnikai és Prod. Iroda Bt.

Projektvezető: Takács Barnabás

Honlap: www.digitalElite.net/Pages/VerAnim/Ehosztesz.html

INFORMÁCIÓS ÉS KOMMUNIKÁCIÓS TECHNOLÓGIÁK PROGRAM

NYILVÁNTARTÁSI SZÁM: OMFB-00238/2004

Bevezetés:

Az E-Hosztesz projekt következő munkaszakaszában egy teljes értékű prototípus környezetet állítunk össze, amelyet részletes tesztelésnek vetünk alá. A tesztelés folyamata két jól megkülönböztethető szakaszra bomlik. Ezek a i) technikai és technológia paraméterek ellenőrzése és mérése, valamint a ii) rendszer hatékonyságára vonatkozó pszichológiai kiértékelés, a korai eredmények megalapozása.

Technikai és technológiai paraméterek ellenőrzése:

A tesztelési folyamat során ellenőrizzük, hogy a képszintézis és a hálózat átviteli paraméterek megfelelnek-e a projekt korábbi szakaszaiban meghatározott célkitűzéseknek. A tesztelés folyamata vizsgálni fogja a kép generálás sebességének változását a 3D E-Hosztesz modell komplexitásának, valóságosságának függvényében (pl. alap geometria, pixel shaderek és bőr, haj, különböző grafikus kártyák, stb.). A fentiek paraméterek ismeretében a technikai tesztelés második fázisában megvizsgáljuk az elosztott hálózati módban történő adatátviteli karakterisztikákat, valamint az interakció sebességét.

Pszichológiai és alkalmazhatósági kiértékelés:

A tesztelés második fázisában folytatjuk a SOTE Pszichiátriai Klinikán - már részben a jelen munkaszakaszban is - megkezdett vizsgálatokat. Konkrétan a klinikai vizsgálat során a projektben kialakított programrendszer segítségével kívánjuk kiértékelni a módszer gyakorlati alkalmazhatóságát. Ennek keretén belül, statisztikailag értékelhető számú kísérlet végrehajtása után, az interakció újszerű metodikával leírható jellemzőinek, objektív paramétereinek rögzítése válik majd lehetővé. A klinikai vizsgálat további célja olyan vizsgálati protokoll kialakítása, ami alkalmas lesz az információfeldolgozás kognitív folyamatainak és az érzelmi állapot változásainak objektív paramétereken keresztül történő mérésére. Az 5-10 fős csoportra tervezett vizsgálat során a vizsgált személyek kognitív és érzelmi állapotában bekövetkező változásait egy jól definiált paramétermezőben jelenítjük meg, ami lehetőséget nyújt majd az eljárás hatékonyságának vizsgálatára.

Budapest, 2005. február 01.

.....
projektvezető