

# **SZAKDOLGOZAT**

**NAGY TAMÁS**

**DEBRECEN**

**2010**



DEBRECENI EGYETEM  
INFORMATIKAI KAR

# MULTIMÉDIÁS TARTALMAK MEGJELENÍTÉSI ÉS ALKALMAZÁSI LEHETŐSÉGEI WEBOLDALAKON

TÉMAVEZETŐ:

DR. TORNAI RÓBERT

EGYETEMI ADJUNKTUS

KÉSZÍTETTE:

NAGY TAMÁS

PROGRAMTERVEZŐ INFORMATIKUS (BSc)

DEBRECEN

2010

# Tartalomjegyzék

<b>1.</b>	<b>BEVEZETŐ.....</b>	<b>4</b>
<b>2.</b>	<b>TECHNOLÓGIAI KÖRKÉP .....</b>	<b>5</b>
2.1.	TÖRTÉNELMI ÁTTEKINTÉS: AZ INTERNET MÚLTJA, JELENE ÉS JÖVŐJE .....	5
2.2.	AZ INTERNET, MINT GAZDASÁGI TÉNYEZŐ.....	10
<b>3.</b>	<b>A WEBES MULTIMÉDIA-TECHNOLÓGIÁK MEGHATÁROZÓ SZEREPLŐI, SZABVÁNYAI ÉS ALKALMAZÁSAI .....</b>	<b>11</b>
3.1.	WORLD WIDE WEB CONSORTIUM .....	11
3.1.1.	<i>A W3C főbb irányelvei .....</i>	<i>12</i>
3.1.2.	<i>HTML és XHTML.....</i>	<i>14</i>
3.1.3.	<i>DHTML .....</i>	<i>16</i>
3.1.4.	<i>JavaScript keretrendszerek.....</i>	<i>19</i>
3.1.4.1.	<i>jQuery.....</i>	<i>19</i>
3.1.4.2.	<i>MooTools .....</i>	<i>21</i>
3.1.4.3.	<i>Dojo.....</i>	<i>22</i>
3.1.5.	<i>OpenLaszlo.....</i>	<i>23</i>
3.1.6.	<i>HTML5 .....</i>	<i>24</i>
3.1.7.	<i>Két remek példa a HTML5 és a DHTML erejére .....</i>	<i>29</i>
3.2.	MPEG LA ÉS AZ AVC/H.264.....	31
3.3.	ADOBE ÉS A FLASH.....	33
3.3.1.	<i>Egy példa animáció készítése .....</i>	<i>35</i>
3.3.2.	<i>A példa animáció bővítése interakciós lehetőséggel .....</i>	<i>44</i>
3.3.3.	<i>A példa animáció HTML dokumentumban való megjelenítése .....</i>	<i>48</i>
3.4.	GOOGLE ÉS A YOUTUBE .....	51
3.4.1.	<i>Tartalom közzététele a YouTube-on .....</i>	<i>52</i>
3.4.2.	<i>YouTube videó HTML dokumentumba való beágyazása .....</i>	<i>55</i>
<b>4.</b>	<b>EGYÉB ALKALMAZÁSI TERÜLETEK .....</b>	<b>57</b>
4.1.	PREZENTÁCIÓK.....	57
4.1.1.	<i>Google Dokumentumok.....</i>	<i>58</i>
4.1.2.	<i>Prezi .....</i>	<i>59</i>
4.2.	MOBIL TECHNOLÓGIÁK.....	61
4.3.	AZ IPTV ÉS A SET-TOP BOXOK .....	62
<b>5.</b>	<b>ÖSSZEFOGLALÓ.....</b>	<b>66</b>
<b>6.</b>	<b>IRODALOMJEGYZÉK.....</b>	<b>67</b>

# 1. Bevezető

Az internet és főként a WWW, azaz a World Wide Web mindennapjainkat is meghatározó technológiájává nőtte ki magát. A kezdetben pusztán egyszerű, szöveges információcserére kifejlesztett eszközrendszer mára olyan fejlett és összetett lett, hogy lassan, de biztosan a multimédiás alkalmazások is teret hódítottak rajta. Ez már nem csak egyszerű webes felületeken elérhető tartalmakban nyilvánul meg, hanem arra alkalmas készülékekkel akár otthon a kanapén ülve a TV készülék előtt, vagy útközben egy – ma már egyszerűnek számító – mobiltelefon segítségével is élvezhetjük az internet forradalmával nyújtott előnyöket.

Dolgozatomban a legfontosabb technológiákat és alkalmazási lehetőségeket igyekszem bemutatni. Ezek egy része napi szintű felhasználásban is életünk részévé vált, míg egy része most kezd teret nyerni. Vannak technológiák, melyek még ugyan kidolgozás alatt állnak, de ha elkészülnek, a jövőben komoly hatással fognak bírni a multimédiás web felületek kialakítására.

Egyes cégek eszközei, megoldásai és alkalmazásai olyan innovatívnak bizonyultak a webes multimédia terén, hogy a korábbi üzleti modelleket meghazudtolva villámgyorsan piacvezető céggé és milliárdos üzletté tette őket. A legnagyobb ilyen cégeket és alkalmazásaikat is bemutatom, valamint az általuk felhasznált szabványos eszközökkel együtt.

A munkámhoz, mivel internetes eszközökről szól, magát az internetet fogom használni. Itt is elsősorban igyekszem a gyártók, szabványügyi cégek és a technológiákat birtokló cégek honlapját, dokumentációit referenciaként felhasználni, hogy első kézből kapott, hiteles adatokat közölhessek.

## 2. Technológiai körkép

Először is szeretném bemutatni az internetes technológiákat, ezen belül is a legfőbb multimédiás módszereket birtokló, vezető cégeiket és alkalmazásaikat. A jobb áttekintéshez elengedhetetlen, hogy tömören ugyan, de áttekintsük, honnan indult az internet maga, miért volt szükség ezen felül a World Wide Web létrehozására, és a gazdasági tényezőket is figyelembe véve, hogyan terjedhetett el ez olyan széles körben a polgári felhasználásban is.

### 2.1. Történelmi áttekintés: az internet múltja, jelene és jövője

A távíró, a telefon, a rádió és a televízió villámgyorsan az információközlés fő eszközeivé váltak az 1900-as évek első felében. Azonban hamar nyilvánvalóvá vált, hogy ezek a – ma már – hagyományosnak tekinthető eszközök nem elég hatékonyak, a visszacsatolás csekély mértékű, vagy nem is létezik. Ezért egyre nagyobb igény mutatkozott az interaktív információcserére, mely a felhasználó által szabadon- és személyre szabottan elérhető tartalmakat jelent. A számítástechnika az internet segítségével megteremtette ezt a lehetőséget.

Az első írásos feljegyzések a társadalmi interakciókról a Massachusettsi Műszaki Egyetem professzorától, J.C.R. Licklider-től származnak. Az ő szavaival élve ezen társadalmi folyamatok eszköze az ún. „Galaktikus Hálózat”. Leírása alapján ez nem más, mint egy globális, hálózatba kötött számítógépek sokasága, melyről bárki, bármely időben tetszőleges tartalmat érhet el, vagy oszthat meg. Ha belegondolunk, ez a mai internet koncepcióját takarja. Licklider lett 1962-től a DARPA<sup>1</sup> számítástechnikai fejlesztésekért felelős csoportjának vezetője. Itt a feletteseit sikerül meggyőznie a hálózati elméletének fontosságáról.

Szintén a Massachusettsi Egyetem egyik professzora, Leonard Kleinrock írt elsőként tudományos folyóiratban a csomagkapcsolt<sup>2</sup> technológiákról 1961-ben. Kleinrock meggyőzte Lawrence G. Robertset a csomagkapcsolt adatátvitel használatáról, aki Lickliderrel együtt dolgozott a DARPA-nál. Ez volt az első nagy lépés a számítógépes hálózatok megalkotása

---

<sup>1</sup> Defense Advanced Research Projects Agency (Fejlett Védelmi Kutatási Projektek Ügynöksége): az Egyesült Államok Védelmi Minisztériumának kutatásokért felelős részlege. Forrás: <http://hu.wikipedia.org/wiki/DARPA> (letöltve: 2010. április 7.)

<sup>2</sup> A hálózati adattovábbítás alapja. Az információ darabokra bontva továbbítódik a hálózaton. Minden darabhoz tartozik egy fejléc, mely az adattovábbítás módját határozza meg, valamint egy törzs, mely maga az adattörzsdék.

felé. A következő lépést az jelentette, hogy ezt a technológiát felhasználva 1965-ben az egyetem TX-2-es számítógépét összekötötte a Calfroniai Egyetem Q-32-es számítógépével egy egyszerű, alacsony adatátviteli sebességű telefonvonal segítségével. Habár ez mai szemmel egyszerűnek és „kicsinek” tűnik, akkoriban óriási eredménynek számított és gyakorlatilag innen eredeztethetjük a hatalmas távolságokat áthidaló számítógép hálózatok létrejöttét.

Robertsék erre az eredményre alapozva 1967-ig kidolgozták az ARPANET (Advanced Research Projects Agency Network) tervezetet, melyet még ebben az évben publikáltak is. Ez tekinthető a mai internet őséne. Gyakorlati alkalmazását szorgalmazta az a helyzet is, melyet az Egyesült Államok hidegháborús politikája okozott, azaz a számukra haszontalannak tűnő projektektől megvonták a támogatást. Így az egyetemek rászorultak arra, hogy a számítási kapacitás növelése érdekében méregdrága számítógépek vásárlása helyett megosszák a többi egyetemmel ezt a feladatot is. Az ARPANET alkalmas eszköz volt erre, és 1969-re már 4 egyetemi csomópont kapcsolódott a hálózathoz. Az információ megosztás növelte a kutatási hatékonyságot, és a számítási teljesítményt egyaránt, és ez olyan alapokat fektetett le, amelyeket ma is követnek.

1970-re kidolgozták az első alkalmazási protokollt, az NCP-t (Network Control Protocol), mely két végpont között tette lehetővé az egységes kommunikációt. Miután 1971-72 között a hálózatba kapcsolódó gépek implementálták ezt a protokollt, elkezdődhetett a valós, hálózati kommunikációra alapuló alkalmazásfejlesztés.

1972-ben Bob Kahn, aki az ARPANET architektúrális tervezésében nagy szerepet játszott, sikeres bemutatót tartott a hálózat működés közbeni állapotáról a Nemzetközi Számítástechnikai Kommunikációs Konferencián (ICCC). Ez volt az első nagy nyilvánosság előtt tartott bemutató. Még ebben az évben Ray Tomlinson, aki a BBN-nél<sup>3</sup> dolgozott, bemutatta az email alkalmazást, mely talán minden idők legsikeresebb internetes technológiájává nőtte ki magát. Ezen keresztül manapság milliárdok kommunikálnak naponta. Népszerűségét egyszerűségének is köszönheti, hiszen egyszerű szöveges üzeneteket, hagyományos leveleket gyorsan továbbíthatunk akár több személynek úgy, hogy azok szinte azonnal megkapják azt.

---

<sup>3</sup> Bolt, Beranek and Newman: kutatással és fejlesztéssel foglalkozó vállalat, melyet a három névadó Massachusettsi Egyetemi alkalmazott hozott létre, így ők is részt vehettek az ARPANET megalkotásában.

Az ARPANET kialakításánál – mivel a DARPA kutatási projektje volt – kulcsszerepet játszott, hogy egy esetleges atomtámadás esetén a megmaradó csomópontok továbbra is képesek legyenek kommunikálni egymással, vagy akár önmagukban is működjenek egységként. Ezt az elvet sikerült is implementálniuk, mint láthatjuk a kapcsolatban használt (kezdetben telefonos) közvetítívonal sokfelé ágazhat, az többféle útvonalon is felépítheti a kommunikációt. Ezt a mai internet is nagyban kihasználja. Egyes hálózati eszközök működési hibája esetén, közvetítői vonal sérülésekor ma már automatikusan képesek az útvonal választó eszközök a kommunikációt helyreállítani más vonalon.

Kahn, még 1972-ben felvetette egy nyílt architektúrájú hálózat létrehozását, mely mentes az operációs rendszerek felépítésétől, és a közvetítői hálózat kialakításától. Az NCP protokoll alkalmatlannak bizonyult erre a feladatra, hiszen nem rendelkezett hibakezeléssel, valamint kizárólag két végpont között volt képes kapcsolatot felépíteni, azaz nem tudott hálózatokat megcímezni. Mivel az ARPANET függött az NCP-től, és fordítva, így Kahn a felmerült igényeket teljesen kielégítő, új hálózati protokoll létrehozásába kezdett. Ez lett a mai is használatos TCP/IP (Transmission Control Protocol/Internet Protocol).

1973-ban Kahn egy korábbi ismerősével, Vinton Cerffel<sup>4</sup> együtt elkezdték kidolgozni a lehetőségét annak, hogy a meglévő számítógépes hálózatokat, melyek nem részei az ARPANET-nek, egyetlen nagy hálózatba lehessen kapcsolni úgy, hogy ezek a részegységek megőrizhessék individualitásukat. A TCP/IP protokoll családot felhasználva ez lett a ma is ismert internet. A TCP protokoll felelős a kapcsolat fenntartásáért és a minőségéért, míg az IP protokoll a címzésért és az adattovábbításért.

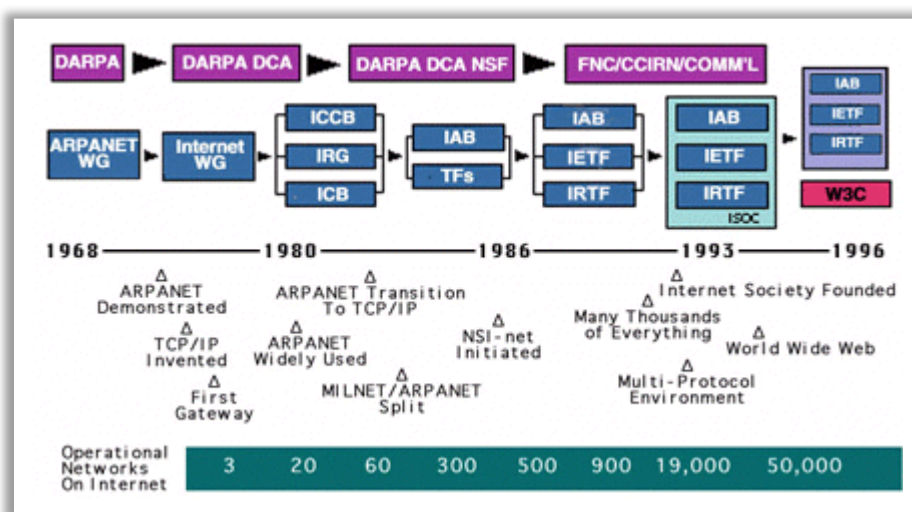
Az elkövetkező években az internet terjedésével olyan protokollokat és szabványokat kellett megvalósítani, amik képesek voltak az egyre szélesedő igényeket kiszolgálni az osztott szerver-rendszerektől az egyszerű asztali számítógépekig. Ilyenek pl. a hálózati osztályok, melyek a különböző méretű hálózatokat külön címtartományokra osztják fel, vagy a DNS (Domain Name System), mely egy hierarchikus adatbázis alapján neveket képes internet címekre fordítani. Maga a TCP protokoll is számos változáson esett át ez idő alatt.

1983-ban az ARPANET-ről leválasztották a hadászati részt MILNET néven, és ezután a Egyesült Államok Országos Tudományos Alapítványa (NSF – National Science Foundation)

---

<sup>4</sup> 1965-től az IBM munkatársa, de később folytatta tanulmányait számítógép tudományokban. Magasabb tudományos fokozatokat szerzett, és 1972-től a Stanfordi Egyetem professzora lett.

felismerte a hálózatokban rejlő, a tudományos kutatások szempontjából fontos lehetőségeket, így 1985-86 között kiépítették a 6 szuperszámítógépből álló hálózatukat, mely az NSFNET nevet kapta és közvetlenül az ARPANET-re kapcsolták. Az NSFNET ma is az USA domináns gerinchálózata, emellett a vezető telekommunikációs magáncégek kezében is jelentős rész van. Az ARPANET formálisan 1990-ben szűnt meg, hogy a fejlettebb, magasabb bonyolultságú hálózatoknak helyet adjon. 1994-ig 21 csomópontosra bővült a hálózat, szálanként 45Mbps sebességű adatátvitellel, világszerte több mint 50 ezer alhálózattal, amelyből csak az USA-ban kb. 29 ezer helyezkedett el (1. ábra).



1. ábra  
Az internet fejlődése<sup>5</sup>

1992-ben elérkezünk az internet történetének talán legfontosabb állomásához, a WWW létrejöttéhez. Ezen keresztül immár bárki kötetlenül „böngészhetett” az információk végtelennek tűnő tárházában, nem csak a fejlesztésben résztvevő tudományos csoportok. A WWW-t Tim Berners-Lee munkássága nyomán a CERN (Conseil Européenne pour la Recherche Nucléaire – Nukleáris Kutatások Európai Tanácsa, közismertebben az Európai Részecskekutató Intézet) publikálta. Három szabványra épül:

- URL – mely a dokumentumok egyedi címzéséért felelős
- HTTP – mely a dokumentumok átviteléért felelős protokoll, és a

<sup>5</sup> Forrás: <http://www.isoc.org/internet/history/timeline.gif>, Letöltve: 2010. április 07.

- HTML – mely a dokumentumok egyfajta leíró nyelve, meghatározza a dokumentumok egészének vagy részének megjelenési módját, így az minden eszközön egységesen jelenik meg.

1998-ra 300 millióra nőtt a web dokumentumok száma, és olyannyira elterjedt, hogy mára a köznyelv tévesen a WWW-t „összevonta” az internet fogalmával.

1993-ban megalakul az ISOC (Internet Society – Internet Társaság), mely az internet talán legfontosabb társasága. Nyílt, tagja lehet bármilyen cég vagy szervezet. Célja az internet fejlődését támogató személyek és cégek összefogása. Létrehozták a W3C alapítványt, azaz a World Wide Web Consortium, melynek egyik vezéralakja maga Tim Berners-Lee lett. Feladata a webes technológiák koordinálása.

A WWW a kezdeti formájához képest nem sokat változott. Míg az 1990-es években elsősorban a mások által közzé tett tartalmak között válogathattunk, addig az ezredforduló után eljött az ún. WEB2-es korszak, amikor is már maguk a böngésző emberek is részt vettek és részt vesznek a tartalmak létrehozásában. Ennek egyik leglátványosabb eszköze a Wikipédia, melyben szócikkek létrehozásával egy komplett, mindenre kiterjedő enciklopédiát hoztak létre, és bővítenek nap, mint nap önkéntes webezők. Fontos tulajdonsága, hogy nem csak új cikkeket vihetünk fel, amit csak mi, a tulajdonosai szerkeszthetjük, hanem gyakorlatilag mindenki, aki regisztrál a Wikipédiára.

*„A Wikipédia egy többnyelvű, nyílt tartalmú, a nyílt közösség által fejlesztett webes enciklopédia. A Wikipédiát a Wikimédia Alapítvány üzemelteti – egy floridai központú nonprofit alapítvány –, szerkesztését pedig önkéntes közösség végzi.”<sup>6</sup>*

Természetesen ez veszélyeket is hordoz magában, hisz a tartalmak szakmai ellenőrzés nélkül kerülhetnek fel, és csak a cikk szerkesztőjének jóindulatára van bízva, hogy mennyire hiteles információt oszt meg.

Elmondhatjuk, míg az internet legforradalmibb eszköze a WWW volt, addig a WWW-nél ugyanezt a WEB2-es alkalmazások megjelenése jelentette. Az olyan oldalak, mint a Facebook, a Twitter, a blog alkalmazások, fórumok magukkal hozták a hétköznapi emberek azon igényét is, hogy ne csak nyers dokumentumszerű tartalmak jelenjenek meg a

---

<sup>6</sup> Forrás: <http://hu.wikipedia.org/wiki/Wikipédia>, Letöltve: 2010. április 8.

weboldalakon, hanem látványos animációkkal körülölelve megoszthassák videóikat, zenéiket, fényképeiket egymással és egész közösségekkel. Ennek függvényében a WWW jövőjét az olyan ipari szabványok jelentik, mint a HTML5 is, mely ugyan jelenleg kidolgozás alatt áll, de pont az ilyen, növekvő multimédiás igényeket igyekszik kiszolgálni.

Mint láthatjuk, az internet viszonylag gyorsan, sőt egyre gyorsuló ütemben életünk részévé vált és válik. De vajon a növekvő multimédiás igényekben a gazdasági élet szereplői látnak-e fantáziát? Vajon megéri-e nekik egy olyan instabil környezetbe investálni, amelyik villámgyorsan változik, és szinte kontrollálhatatlan módon? Ezekre a kérdésekre keresek választ tömören a következő alfejezetben.

## **2.2. Az internet, mint gazdasági tényező**

Az internet gyors fejlődése előtt különálló rendszerek tárolták és közvetítették a képi és hanganyagokat, úgymint telefonok, televíziók és rádiók, videó lejátszók és önálló számítógépek. Ezek a szolgáltatások egyre inkább kezdenek áttevődni az internetre. Ez a konvergencia nem csak a fizikai eszközök fejlődésében mutatkozik meg, hanem olyan új, értéknövelt szolgáltatások jelentek meg, mint a VoD (Video on Demand) és a társa, az IPTV (Internet Protocol TV). Míg az előbbi a hagyományos videózást, a videotékák gyűrött kazettás élményét hivatott lecserélni, addig az utóbbi a televíziós műsorszórást modernizálja.

Ezekhez a szolgáltatásokhoz természetesen új, eddig nem használt technológiai háttér is szükségeltetik, így megjelennek a nagy gyártók (Samsung, Philips, Sony) olyan készülékekkel, többnyire kombináltakkal, melyek képesek mind a hagyományos formátumú médiumok kezelésére, mind az új, többletszolgáltatást adókéra.

A hagyományos, pusztán kommunikációs célokat szolgáló eszközöket leváltották a mobil eszközök. Ezek az eszközök ma már nem csak telefonálásra, egyszerű hangátvitelre képesek, hanem valós idejű képi átvitelre szintén az internetet használva közvetítő közegként. Az emberek multimédia igénye a mobil eszközök terjedésével kitágult, immár nem csak a négy fal korlátai között szeretnénk zenét hallgatni, filmet nézni, hanem út közben, buszon, vonaton ülve. A gyártók felismerték ezt, és mivel a technológia egyre inkább képes ezeket az igényeket kiszolgálni, így egy mobil telefon segítségével hallgathatunk zenét, nézhetünk

filmet és TV műsorokat, akár az interneten keresztül is. Napjaink egyik legnagyobb VoIP<sup>7</sup> szolgáltatásának, a Skype-nak 2008-ban közel 300 millió<sup>8</sup> felhasználója volt.

### **Néhány számadat az egyik vezető gyártótól, a Samsungtól<sup>9</sup>.**

A Samsung leányvállalata, a Samsung Electronics 2007-ben 150 ezer alkalmazottat foglalkoztatott, és 105 milliárd USA dolláros éves forgalmat bonyolított. Mobil kommunikációs részlegük kiemelkedő szerepet tölt be az internetre optimalizált eszközök gyártásában. 2007-ben 160 millió eszközt adtak el, valamint az ő mobil WiMAX<sup>10</sup> megoldásukat fogadták el a 3G-s mobil hálózatok alaptechnológiájaként. Ez a technológia teszi lehetővé a mobil videó telefonálást valamint a mobil TV-s műsorszórást.

## **3. A webes multimédia-technológiák meghatározó szereplői, szabványai és alkalmazásai**

Miután megismertük az internet múltját és jelenét, a felhasználók növekvő igényeit és tömören a lehetőségeket, most bemutatnám az internetes multimédia meghatározó területeit.

### **3.1. World Wide Web Consortium**

A World Wide Web Consortiumot (W3C) 1994-ben alapította Tim Berners-Lee, a WWW szülőatyja. A cél az volt, hogy a webes technológiák fejlesztésével foglalkozó cégek és kutatók az eredményeiket egymás közt osszák meg, a technológiákat egyszerűsítsék és egységesítsék, hisz a WWW alapelve az átjárhatóság, a platformfüggetlenség, az, hogy minden eszközön egyaránt ugyanúgy működjön. A W3C feladata, hogy elősegítse a webben rejlő lehetőségek minél teljesebb kihasználását olyan közös számítógépes protokollok kidolgozásával, amelyek előremozdítják a web fejlődését és garantálják széleskörű felhasználhatóságát. Ma már a W3C-nek több mint 450 tagja és közel 70 teljes munkaidős munkatársa van szerte a világon, akik hozzájárulnak a W3C specifikációk és szoftverek fejlesztéséhez.

---

<sup>7</sup> Voice over Internet Protocol – „Hang az interneten”. Jellemzően olyan szolgáltatások gyűjteménye, melyek a hagyományos telefonos szolgáltatásokat hivatottak leváltani internet segítségével. Mára ezek többsége videóhívást is lehetővé tesz.

<sup>8</sup> Forrás: OECD dokumentum (lsd.: Irodalomjegyzék)

<sup>9</sup> Adatforrás: <http://www.samsung.com/hu/aboutsamsung/corporateprofile/download/SamsungProfile2008.pdf>, Letöltve: 2010. április 08.

<sup>10</sup> IEEE 802.16e szabványnak megfelelő, szimmetrikus sávzélességű, fokozott kódolású és nagy távolságokat lefedő rádiós adatátviteli technológia.

### 3.1.1. A W3C főbb irányelvei

- A W3C a Webet a hálózaton keresztül elérhető információk rendszereként definiálja (elérhető számítógépen, telefonon, televízióon vagy akár hálózatra kötött hűtőszekrényen, stb. keresztül), tehát a cél annak megteremtése, hogy mindenki a hardverétől, szoftverétől, hálózati infrastruktúrájától, anyanyelvétől, kultúrájától és földrajzi helyzetétől függetlenül élvezhesse a web nyújtotta előnyöket.
- Ma még természetes nyelven osztjuk meg a tudásunkat a weben. A W3C egyik jelenkori feladata, hogy elterjessze a szemantikus webet, azaz egy jelentésen alapuló webet, melyen olyan kifejezéseket használhatunk, amit a számítógépek értelmezni és változtatni tudnak. Ez lehetővé teszi számunkra, hogy az információt gyorsan, platformtól függetlenül tehesük közzé. Főbb építőkövei a W3C-nyelvek: RDF, XML, XML séma és XML aláírás:
  - RDF – Resource Description Framework, azaz Erőforrás Leíró Nyelv. Olyan adatleíró nyelv, mellyel erőforrásokról szóló információkat ábrázolhatunk a weben. Ezt erőforrásokkal összefüggő metaadatok ábrázolási céljára fejlesztették ki, mint pl. cím, szerző, a weblap utolsó módosításának időpontja. Olyan esetekre tervezték, amelyekben az efféle információkat nem csak emberek számára kell megjeleníteni, hanem számítógép-programok segítségével is fel kell dolgozni.
  - XML – Extensible Markup Language, azaz Kiterjeszhető Leíró Nyelv. Általános célú leíró nyelv, speciális célú leíró nyelvek létrehozására. Különböző adattípusok leírására is képes. Az elsődleges célja a strukturált szöveg és információ megosztása az interneten keresztül. Az XML-en alapuló nyelvek formális módon vannak leírva, így lehetővé téve a programok számára a dokumentumok módosítását és validálását a formátum előzetes ismerete nélkül.
  - XML Séma, egy új keletű séma nyelv, amit a W3C a DTD<sup>11</sup> utódként definiál, de lényegesen többre képes annál. Sokoldalú adattípus

---

<sup>11</sup> DTD – Document Type Definiton (Dokumentum Típus Definíció): dokumentum típus azonosítására szolgáló leíró deklarációk egy halmaza. Olyan leíró nyelvekhez hozták létre, mint a web alapja, a HTML is.

rendszert használ, ami részletesebb megkötéseket tesz lehetővé az XML dokumentum logikai szintjén.

- XML aláírás: digitális aláírások létrehozására szolgáló szintaxist és feldolgozási szabályt definiál.
- A web alapja az együttműködés. Egy, a mainál is rugalmasabb környezet kidolgozásával szeretnék elérni, hogy a web a bizalomra épüljön, így lehetővé téve a weben való felelős publikálást. Ennek egyik eszköze az XML aláírás, de ilyen még többek között a kommentáló mechanizmus, a csoportazonosítás valamint a verziókezelés.
- A W3C, mint gyártófüggetlen szervezet, nyílt, azaz nem szabadalmaztatott nyelvek, protokollok tervezésével, támogatásával biztosítja az együttműködést, hogy egységesítse a piacot. Ennek megvalósításáért egy ipari megegyezés és egy nyílt vitafórum felel.
- Minden technológiai tervezésnél fontos az egyszerűség, a modularitás, a kompatibilitás és a kiterjeszhetőség, hogy a jövőben is könnyen módosítható, továbbfejleszthető legyen.
- A modern elosztott rendszereknek a decentralizálás az alapelve. Egy centralizált rendszerben minden üzenetnek, eseménynek keresztül kell haladnia a központon, ami nagy forgalom esetén torlódáshoz vezethet. A W3C tervezéskor limitálja a központok számát, hogy csökkentsék a web sebezhetőségét.
- A W3C hozzájárul multimédiás szabványok létrejöttéhez és kidolgozásához. Ilyenek pl. az SVG (Scalable Vector Graphics) és a SMIL (Synchronized Multimedia Integration Language). Ezeket egy alfejezetben részletesebben is tárgyalom.

A W3C által készített szabványok valójában többnyire nem igazi szabványok, hanem ahogy ők nevezik: ajánlások. Ezeket az ajánlásokat a szakma kezeli csak „de facto” szabványként.

### 3.1.2. HTML és XHTML

Ahogy az előzőekben is láthattuk, a web a platformfüggetlen tartalom közvetítés alapjait vallja. Ahhoz, hogy ez megvalósítható legyen minden területen, a W3C kidolgozott egy olyan leíró nyelvet, mely egységes, minden szoftveren az ajánlást követve egységesen jeleníthető meg. Ez a HTML – HyperText Markup Language, azaz „Hiperszöveg” Leíró Nyelv. A weboldalak készítőinek kezébe a HTML olyan eszközöket ad, mint a dokumentumok fejlécekkel, táblázatokkal, listákkal, képekkel, stb. való elláthatóságát, továbbá más dokumentumokra hivatkozhatnak linkekkel, űrlapokat hozhatnak létre a felhasználó által megadott adatok feldolgozására, valamint közvetlenül beágyazhatnak videoklipeket, hanganyagokat és más alkalmazásokat.

A HTML nyelvet még Tim Berners-Lee fejlesztette ki, amíg a CERN-nél dolgozott, majd a W3C-nél az 1990-es évek folyamán tökéletesítette azt. Legutolsó verziója az 1999. december 24.-én ajánlasként elfogadott 4.01. Minden egyes verzió kiadásánál figyelembe vették az iparági szereplők igényeit és javaslatait.

A HTML 4-es verziója a korábbi verziókkal szemben számos olyan újítást hozott, mint a stíluslapok, kliens oldali scriptek, keretek, beágyazott objektumok, a jobbról balra írt szövegek támogatását, fejlettebb táblázat formázó eszközöket. Ez a verzió szakértők bevonásával a nemzetköziesítés jegyében készült, azaz a leíró nyelv régiófüggetlen lett, a dokumentumok bármilyen nyelven íródhattak, a világ bármely pontján pontosan jelentek meg. Ezt az ISO/IEC:10646 szabvány implementálásával érték el, mely igen szigorúan rögzíti a világ különböző nyelveiből adódó számítástechnikai problémák megoldását.

2002-ben elfogadták ajánlasként az XHTML javított változatát, mely tulajdonképpen egy dokumentum típus és modul család, ami kiterjeszti a HTML 4-et. A dokumentum típus család tagjai XML alapúak, mellyel számos előny jár:

- Az XHTML dokumentumok megfelelnek az XML nyelvi követelményeinek, ezért bármely XML szerkesztővel könnyen megtekinthetők és módosíthatók.
- A szigorú nyelvi előírásoknak megfelelően hatékonyabb működésű dokumentumok készíthetők.
- Az XHTML dokumentumokra könnyű építeni olyan alkalmazásokat, melyek függenek a HTML-től és/vagy az XML-től.

- Ahogy az XML alapú web fejlődik, az XHTML alapú dokumentumok úgy válnak egyre átjárhatóbbá a különböző platformok között.
- Saját XML típusleírókkal új adattípusokat és elemeket adhatunk a meglévő dokumentumainkhoz a beépítettekén kívül.

Az XHTML dokumentumoknak három fő típusa létezik:

- Strict – szigorú szabványkövetési módot követel meg, kizárólag az XHTML nyelvi elemei használhatók.
- Transitional – átmeneti típust képez a HTML és az XHTML dokumentumok között, melyben a HTML nyelvi elemei is használhatók. A feldolgozó alkalmazásoknak (többnyire böngészőknek) mindkét nyelvi típusra kell tudniuk ellenőrizni a dokumentumot.
- Frameset – egymásba ágyazott dokumentumok (kereteket) definícióit hordozza.

Most pedig bemutatnám hogyan is néz ki egy nagyon alapvető XHTML dokumentum, melyet az 1. kód szemléltet.

```

1: <?xml version="1.0" encoding="UTF-8"?>
2: <!DOCTYPE html PUBLIC
3:     "-//W3C//DTD XHTML 1.0 Strict//EN"
4:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
5: <html xmlns="http://www.w3.org/1999/xhtml"
6:     xml:lang="en" lang="hu">
7:     <head>
8:         <title>Helló világ címe</title>
9:     </head>
10:    <body>
11:        <p>Helló világ!</p>
12:    </body>
13: </html>

```

**1. kód**  
**Egy alapvető XHTML dokumentum**

Az 1. sorban az XML deklaráció helyezkedik el, mely abban az esetben kötelező, ha más karakterkódolást használunk, mint az alapértelmezett, azaz nem UTF-8-at vagy UTF-16-ot. Ennek ellenére erősen ajánlott használni ezt a deklarációt minden esetben.

A 2-4. sorban helyezkedik el a dokumentumtípus deklaráció. Láthatjuk, hogy ez Strict típusú formázásnak felel meg.

Az 5-13. sor között helyezkedik el a tényleges dokumentum. A HTML tag attribútumaként elhelyeztem az XML dokumentumtípusainak nyelvét, azaz az angolt, míg a dokumentum nyelve a magyar lett. Egy XHTML dokumentum rendelkezhet fej résszel, melybe rendszerint a dokumentum meta adatai kerülnek. Ilyen a példában is elhelyezett `title` címke<sup>12</sup>, ami a dokumentum címét adja meg. Továbbá `body` címkék közé kell zárni a tényleges dokumentum részt. A példában nincs más, mint egy *'Helló világ!'* felírat, amit `p` címkék között helyeztem el. A `p` címke a HTML és XHTML dokumentumokban egyaránt bekezdéseket definiál.

### 3.1.3. DHTML

A DHTML egy, a szakmában elfogadott terminológia dinamikus, interaktív weboldalak létrehozására. A DHTML kombinálja a HTML, a JavaScript, a HTML DOM és a CSS előnyeit.

A HTML DOM, azaz a HTML Document Object Model egy W3C szabvány, mely leírja, hogy hogyan érhetjük el és manipulálhatjuk a már megjelenített HTML dokumentumokat. A DHTML olyan eszközökkel ruházza fel a dokumentumok készítőit, amelyekkel a felhasználók által generált eseményekre reagálhatunk. Ennek eszközei a HTML események, melyeket szintén a DOM definiál.

A JavaScript gyakorlatilag a web alap script nyelvéné vált és minden komolyabb böngésző támogatja. Pontosan arra lett tervezve, hogy interaktivitást teremthessünk meg vele a HTML dokumentumokban. Ma már hála a számítógépek megnövekedett teljesítményének

---

<sup>12</sup> A HTML dokumentumainak elemeit a W3C az angol tag névvel illeti, melynek jelentése a magyarban címke.

komoly webalkalmazások is készíthetők a segítségével. A JavaScript a nevével ellentétben nem egyenlő a Java programozási nyelvvel<sup>13</sup>.

A CSS, azaz a Cascading Style Sheets (lépcsőzetes stíluslapok) segítségével definiálhatjuk, hogy hogyan jelenjenek meg a HTML dokumentumok elemei. JavaScript és DOM segítségével manipulálhatjuk ezeket a megjelenítés után is, így látványosabbá téve a felhasználói események okozta állapot módosulásokat.

Egy tipikus JavaScript kód a 2. kódon is látható módon helyezhető el a dokumentumban.

```
1: <html>
2: <body>
3:     <script type="text/javascript">
4:         document.write('Hello');
5:     </script>
6: </body>
7: </html>
```

## 2. kód

### Egy tipikus JavaScript kód elhelyezése HTML dokumentumban

A `document` egy DOM objektum, mellyel magát az aktuális HTML dokumentumot érhetjük el. A `write()` metódus segítségével írhatunk a dokumentumba. Ez a kis HTML kód a kiegészített JavaScripttel nem tesz mást, mint kiírja a dokumentumban ahol elhelyezkedik, hogy „Hello”.

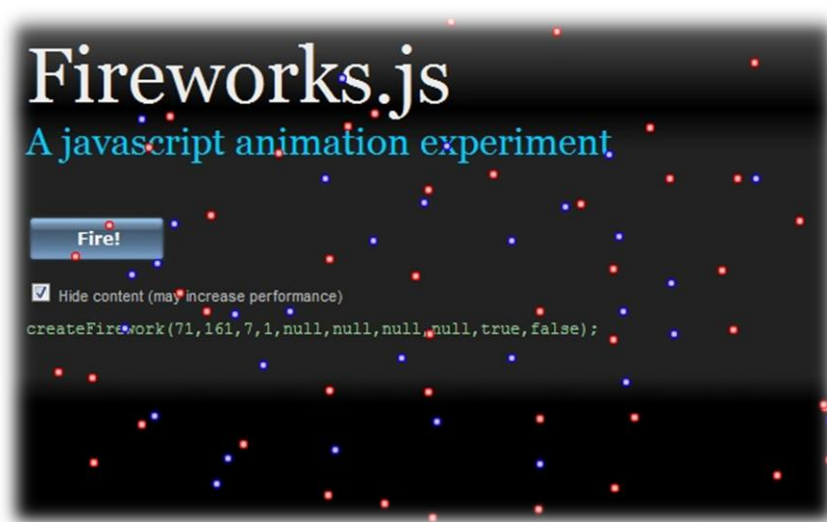
Ha a dokumentum egy adott elemét – jellemzően valamelyik címke tulajdonságát, vagy a közé zárt tartalmat – akarjuk megváltoztatni, akkor célszerű azt ellátni egy azonosítóval, melyet az `id` attribútummal tehetünk meg, és azon keresztül elérni a `document` objektum `getElementById()` metódusával.

---

<sup>13</sup> A JavaScript neve eredetileg ECMAScript, amit az ECMA szervezet fejleszt. Az ECMA-262 a hivatalos JavaScript szabvány. 1996-ban kezdték fejleszteni, és 1998 óta ISO (ISO/IEC 16262) szabvány, amit azóta is fejlesztenek. Forrás: [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp), letöltve 2010. április 11.

A felhasználói események legtöbbször egér segítségével váltódnak ki HTML dokumentumokban kattintás útján. Ezekhez a dokumentumunkban a címkék `onclick` attribútumával rendelhetünk lefuttatandó kódot.

Most pedig szeretnék a DHTML erejére egy gyakorlati példát is mutatni. Ez egy egyszerű, ámde látványos tűzijáték effektus, a `Fireworks.js`, melynek paraméterei a felhasználó által dinamikusan változtathatók, és az animáció mellé hangok is társulnak. Az alkalmazás elérhető eredeti helyén, a <http://www.schillmania.com/projects/fireworks> címen<sup>14</sup>, a 2. ábrán pedig láthatjuk működés közben is.



2. ábra

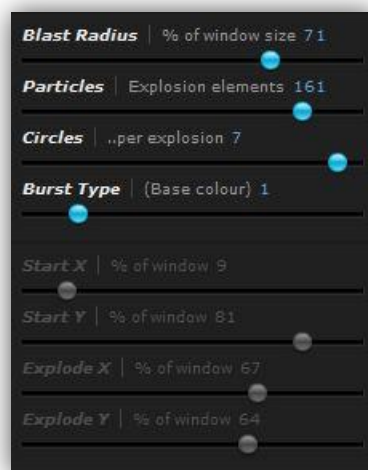
A `Fireworks.js` működés közben<sup>15</sup>

Komplex matematikai függvények, valós idejű, a felhasználó által is befolyásolt grafikai- és hang effektusok találhatóak a kisalkalmazásban, és mindez elsősorban a JavaScriptnek köszönhetően. A készítő elmondása szerint akár processzor teljesítmény mérésére is használhatjuk, hiszen ha a paramétereket – melyeket a 3. ábrán láthatunk – megfelelően módosítjuk, akkor olyan animációkat is létrehozhatunk, amelyek számítása komoly erőforrásokat igényel.

---

<sup>14</sup> 2010. április 12.

<sup>15</sup> Saját készítésű képernyőkép részlet. Készítve 2010. április 12.



3. ábra  
A Fireworks.js beállító panelja<sup>16</sup>

A beállításokat „csúszkákon” módosíthatjuk, melyek nem részei a HTML-nek, így a készítő itt is a DHTML-t hívta segítségül. CSS segítségével formázta meg a megjelenést, JavaScript segítségével a HTML DOM objektumok interaktív módosításával pedig lekezeli a felhasználói eseményeket, elérve a kívánt hatást.

#### 3.1.4. JavaScript keretrendszerek

A JavaScript keretrendszerek olyan alkalmazásfejlesztési gyűjtemények, melyek könnyebbé és gyorsabbá teszik a webalkalmazások készítését. Ezek általában előre deklarált és implementált osztályok és azok objektumai, valamint a hozzájuk kapcsolódó függvények. A legelterjedtebb ilyen keretrendszerek a DHTML eszközeinek használatát egyszerűsítik le, de többek között az AJAX<sup>17</sup> alapú adatfeldolgozást is megkönnyítik.

##### 3.1.4.1. jQuery

A jQuery 2006-ban jelent meg a Mozilla alapítvány munkatársa, John Resig által, melynek használata a MIT és a GNU GPL licencek által kötött. Több neves cég alkalmazza ezt a keretrendszert, mint pl. a Google, a Dell vagy a Microsoft. A jQuery alapja, hogy a

<sup>16</sup> Saját készítésű képernyőkép részlet. Készítve 2010. április 12.

<sup>17</sup> AJAX: Asynchronous JavaScript and XML – JavaScript utasítások segítségével aszinkron kérések küldését teszi lehetővé a kiszolgáló szerver felé, így létrehozhatók olyan weboldalak, melyekben a dinamikus adattartalmak frissítése megoldható a teljes dokumentum újratöltése nélkül.

lehető legjobban elválassza a JavaScript kódot a HTML-től és különböző eseményvezérlőkön, azonosítókön keresztül kommunikáljon a dokumentum elemeivel. A jQuery olyan eszközöket ad a fejlesztő kezébe, mint pl. az eseménykezelések, effektusok automatizálása, stíluslap módosítók és kijelölők, attribútum módosítók, AJAX utasítások és feldolgozók valamint a felhasználói felület megjelenését manipuláló eszközök.

A jQuery segítségével böngésző független JavaScript fejlesztéseket készíthetünk. Az egész keretrendszer viszonylag kisméretű, a jelenlegi verziója (1.4.2) kb. 24kB tömörítve<sup>18</sup>. A keretrendszer funkcionalitása bővíthető kiterjesztések használatával, melyek a <http://plugins.jquery.com> címről tölthetők le. Dokumentációja részletes, különböző könyvek is elérhetőek hozzá.

A jQuery keretrendszert a HTML dokumentum fejrészében kell belinkelnünk, melynek eszközeit ezután használhatjuk. Az alábbiakban egy mintakódot láthatunk (3. kód), mely az oldal betöltődése után felugró ablakban köszönti a felhasználót.

```
1: <html>
2: <head>
3:     <script type="text/javascript"
4:         src="jquery-1.4.2.min.js"></script>
5:     <script type="text/javascript">
6:         $(document).ready(function() {
7:             alert("Hello");
8:         });
9:     </script>
10: </head>
11: <body></body>
13: </html>
```

### 3. kód

#### Egy alapvető jQuery kód HTML-be ágyazva

A 3-4. sorban linkeljük be a dokumentumba a keretrendszert. A 6-8. sorban található a minta jQuery utasítás. Ez a `document` objektumon – mely a HTML dokumentumunk

---

<sup>18</sup> A tömörítés ebben az esetben azt jelenti, hogy a kódból a felesleges *white space* karaktereket, kommenteket törlik, valamint a változó- és függvényneveket egy-két karakterre rövidítik. Az ilyen fájl gyorsan letölthető a felhasználó által, így lassabb kapcsolattal rendelkezőknek is gyors az oldal betöltődése.

JavaScript referenciája – meghívja a `ready` metódust, melynek egy függvényt adunk át. Ez a függvény akkor hívódik meg, ha a böngésző befejezte a dokumentum feldolgozását és készen áll annak a manipulálására. A függvényünkben most nincs más, mint egy egyszerű JavaScript figyelmeztető ablak, ami kiírja a felhasználó számára a „Hello” üzenetet.

A `$` jel a jQuery utasítások hívó szimbóluma, ez viszont számos, más keretrendszerrel való együttes használat esetén ütközést okozhat. Erre a jQuery keretrendszer kínál egy megoldást, mellyel a hívó szimbólum feloldható, vagy akár tetszőlegesen helyettesíthető, az alábbi módon:

```
var JQ = jQuery.noConflict();
```

Ezen értékadó utasítás után bárhol a JavaScript kódunkban a JQ változónevet használva a már ismert módon adhatunk ki JQuery utasításokat. Ennek megfelelően a minta kódbeli utasítás így alakul:

```
JQ(document).ready(function() { ... });
```

#### 3.1.4.2. MooTools

A MooTools-t Valerio Pioretti készítette, melynek legelső publikus változatát 2006-ban tette közzé. A MooTools filozófiája szerint megpróbálja helyettesíteni a natív JavaScript objektumokat és függvényeket, természetesen kiegészítve őket számos hasznos tulajdonsággal és metódusokkal. Mindemellett a jQuery-hez hasonlóan külső JavaScript könyvtárakkal kiegészíthető, szintén tömörített formában elérhető, objektumorientált módon programozható és a HTML DOM objektumok kezelését is megkönnyíti.

A MooTools főbb komponensei:

- A *Core*, azaz a keretrendszer motorja. Ide tartoznak olyan függvények, melyeket a többi komponens igényel a működéséhez.
- A *Class*, azaz az osztály, melynek segítségével saját osztályokat definiálhatunk és implementálhatunk.
- A *Natives*, azaz a natív JavaScript objektumok kiterjesztései.
- Az *Element*, azaz JavaScriptben definiált, HTML DOM kezeléséhez szükséges Element objektum kiterjesztése.

- *FX*, mely különleges effektusok létrehozásához használható (elsősorban animációk).
- *Request*, mely a kliens-szerver közötti kommunikáció kezeléséért felelős.
- *Window*, mely segítségével böngésző független megjelenést adhatunk weboldalainknak.

Az előző fejezet jQuery-ben írt eseménykezelő MooTools-ban írt megfelelője a 4. kódon is láthatóak szerint alakul.

```
1: window.addEventListener('domready', function() {
2:     alert('Hello');
3: });
```

#### 4. kód

A 3. kód jQuery utasításának MooTools megfelelője

### 3.1.4.3. Dojo

A Dojot 2004-ben kezdte el fejleszteni Alex Russel, Dylan Schiemann, David Schontzler és számos más közreműködő. A Dojo terjesztésére alapítványt hoztak létre.

A Dojo tartalmaz egy ún. *widget* rendszert, mely JavaScript, HTML és stíluslapok segítségével böngészőfüggetlen interaktív felületi elemeket hoz létre. Ilyenek például a legördülő menük, fülek, rendezhető táblázatok, dinamikus grafikonok, kétdimenziós vektorgrafikák és számos animáció. Ezen felül ezeknek a widgeteknek a kinézete testreszabható, akár előre definiált témák segítségével is. A Dojo mindezek mellett hasonló eszközöket kínál, mint a MooTools és a jQuery.

A 3. kódban látható utasításunk Dojoban az alábbiak szerint alakul (5. kód).

```
1: dojo.addOnLoad(function() {
2:     alert('Hello');
3: });
```

#### 5. kód

A 3. kód jQuery utasításának Dojo megfelelője

### 3.1.5. OpenLaszlo

Az OpenLaszlo egy nyílt forrású rendszer ún. gazdag internetes alkalmazások fejlesztéséhez, melyet jelenleg olyan cégek is használnak, mint az IBM, Yahoo! és a Lockheed Martin.

Két fő eleme van, az LZX programnyelv és az OpenLaszlo szerver. Az LZX nyelv az XML-en és a JavaScripten alapul, melyben deklaratívan programozhatunk, mely lehetővé teszi a gyors alkalmazásfejlesztést. Maga a nyelv abban a szellemben készült, hogy a webalkalmazás fejlesztők meglévő tudásukat gyorsan átültethessék ebbe a környezetbe. Az OpenLaszlo szerver egy ún. Java servlet, ami lefordítja az LZX-ben írt alkalmazásokat és megjeleníti azokat a felhasználó böngészőjében. Ezen felül lehetséges DHTML alapú kódra vagy Adobe Flash 8-10 platformok valamelyikére is fordítani az alkalmazásokat.

```
1: <canvas>
2:     <window width="100" height="100">
3:         <text>Hello!</text>
4:     </window>
5: </canvas>
```

#### 6. kód

#### Egy alapvető LZX program

A 6. kódon láthatjuk, hogyan épül fel egy LZX program. A `canvas` címkével definiálunk egy rajzterületet, melyre írhatunk és rajzolhatunk. Ezen a területen helyeztem el kis 100x100 képpont méretű ablakocskát, melyet a `window` címkével adhatunk meg. Ezen belül pedig a `text` címke segítségével jelenítettem meg a „Hello!” szöveget.



4. ábra

A minta kód eredménye a böngészőben<sup>19</sup>

<sup>19</sup> Saját készítésű képernyőkép, készítve: 2010. április 28.

A 4. ábrán láthatjuk, hogy mindösszesen öt kódsor segítségével létrehoztunk egy olyan felületi elemet, mely a böngésző ablakban tetszőlegesen elmozdítható. A minta kód jól szemlélteti, hogy milyen egyszerű és gyors a fejlesztés az OpenLaszlo segítségével, így akár az alábbi ábrán látható komolyabb webalkalmazások készítése is jelentősen leegyszerűsödik.



5. ábra

Egy audio CD internetes áruház OpenLaszloban fejlesztve<sup>20</sup>

Dinamikus alkalmazásainkhoz az OpenLaszlo segítségével látványosan animált menüket, grafikákat és kimutatásokat is készíthetünk.

### 3.1.6. HTML5

A HTML5 a HTML nyelv teljes újragondolása lesz, melyen 2004 júniusa óta dolgoznak. A szakma már türelmetlenül várja, hogy W3C ajánlássá váljon, és elkezdhesse a webalkalmazások megújítását. Az új szabvány célja lesz többek között, hogy az olyan – egyébként erőforrás igényes – bővítmények szükségességét csökkentse, mint az Adobe Flash,

<sup>20</sup> Saját készítésű képernyőkép, készítve: 2010. április 28. Az alkalmazás címe:  
[http://www.openlaszlo.org/lps\\_demos/demos/amazon/amazon.lzx?lzt=html&lzr=dhtml](http://www.openlaszlo.org/lps_demos/demos/amazon/amazon.lzx?lzt=html&lzr=dhtml)

a Microsoft Silverlight és a Sun (Oracle) JavaFX, melyek jelenleg a piac meghatározó technológiái, valamint a DHTML eszközeiből is beépít, és továbbfejleszt elemeket.

A HTML5 számos új címkét és attribútumot definiál, melyeket számos fejlesztő már kerülő utakon így is használt, de ezáltal közvetlenül, magán a HTML nyelven keresztül lesz elérhető. Néhányuk le fog váltani ma már elavult, vagy nehezen kezelhető és feldolgozható elemeket, mint pl. a `div` (különböző tartalmi blokkok egymásba ágyazására való) és a `span` (blokkon vagy soron belüli elemek, szövegrészek formázásához használják), melyeket a `nav` (weboldal navigációs blokk) és a `footer` (jellemzően a weboldalak lábléce) váltja le. Az egyik legfontosabb változás, melyet a HTML5 hoz az az, ahogyan a scripteket fogja kezelni. Lehetőség lesz aszinkron futtatásra, mely azt jelenti, hogy mialatt a weboldal még feldolgozás alatt áll a böngésző által, a scriptet már ekkor is elindíthatjuk, így a dinamikus adatokat előkészíthetjük a betöltődő oldalnak, vagy akár az oldal tartalmát még megjelenés előtt megváltoztathatjuk. Ilyen lehet pl. az a felhasználási terület, hogy a különböző felbontású monitorokhoz még a betöltéskor hozzáigazíthassuk a tartalmat, ezzel is tehermentesítve a szerveret a feldolgozás alól, valamint a felhasználó számára is gördülékenyebb lesz a kezelhetőség.

A másik két nagy újítás, melyet a szakma nagyon vár, és a multimédiás alkalmazások szempontjából döntő szerepű lesz a jövőben, az a `video`, az `audio` és a `canvas` címkék megjelenése.

A `video` címke segítségével videókat ágyazhatunk a weboldalakba (7. kód), melynek megfelelő kezeléséről és lejátszásáról a böngészőnek kell rendelkezni, további operációs rendszerbe épülő eszközökre, *kodekekre*<sup>21</sup> nincs szükség.

- 1: `<video src="video.ogg" controls="controls">`
- 2: Az ön böngészője alkalmatlan a video címkék kezelésére.
- 3: `</video>`

#### 7. kód

##### A video címke alapvető alkalmazása

---

<sup>21</sup> Kodek: A magyar kodek szó az angol codec szóösszetételből származik. A compressor-decompressor szavak összevonása, és digitális jelek feldolgozására használatos eszközöket jelöl. Video-audio vonatkozásban olyan eszközöket, melyek segítségével képesek leszünk a velük kódolt adatfolyamok feldolgozására.

Ugyanez vonatkozik az `audio` címkére, mely kizárólag hanganyagok dokumentumba való ágyazásáért felelős (8. kód).

```
1: <audio src="zene.ogg" controls="controls">
2:     Az ön böngészője alkalmatlan az audio címék kezelésére.
3: </audio>
```

#### 8. kód

##### Az `audio` címke alapvető alkalmazása

A `canvas` címke segítségével közvetlenül a dokumentumba rajzolhatunk grafikus elemeket valamilyen script nyelv segítségével (9. kód).

```
1: <canvas id="myCanvas">
2:     Az ön böngészője nem támogatja a canvas címkét.
3: </canvas>
4: <script type="text/javascript">
5:     var canvas=document.getElementById('myCanvas');
6:     vas.getContext('2d');
7:     ctx.fillStyle='#FF0000';
8:     ctx.fillRect(0,0,80,100);
9: </script>
```

#### 9. kód

##### A `canvas` címke alapvető alkalmazása

A 6. ábrán pedig a `video` címke gyakorlati alkalmazását is láthatjuk, melyet a YouTube-ról mentettem. A YouTube HTML5 támogatása a dolgozatom elkészítésekor még csak béta állapotban, külön engedélyezve érhető el az arra alkalmas böngészőkben.



**6. ábra**  
**HTML5 video lejátszás közben a YouTube-on**

A HTML5, és benne a `canvas` címke fontos eszköze lehet az SVG, azaz Scalable Vector Graphics (Méretezhető Vektorgrafika). Az SVG tulajdonképpen egy XML alapú leírónyelv kétdimenziós grafikák és alkalmazások létrehozására. A legutóbbi W3C ajánlás az 1.1-es verziószámot viseli. Létezik belőle egy ún. *tiny* (apró, parányi) verzió kizárólag mobil alkalmazások számára. Az SVG háromféle objektum típust különböztet meg:

- vektorgrafikus alakzatok, úgymint útvonalak, melyek egyeneseket és görbéket tartalmaznak,
- képeket
- és szöveget.

A grafikus objektumok csoportosíthatók, formázhatók és korábban megjelenített alakzatokba transzformálhatók. A szövegek bármely XML névtérben<sup>22</sup> elhelyezhetők, mely megkönnyíti az SVG grafikák keresését és elérését. Az SVG eszköztárban megtalálhatók még a beágyazott transzformációk, vágógörbék, szűrő effektusok és a kiterjeszthetőség.

---

<sup>22</sup> Az XML névterek segítségével elkerülhetjük a címkék névtüközését a dokumentumban, ha azonos néven többféle tulajdonságú objektumtípus szeretnénk definiálni.

Az SVG dokumentum objektum modellje – mely magában foglalja a teljes XML objektum modellt – lehetővé teszi egyszerű és hatékony vektor alapú animációk létrehozását, melyekkel dinamizmust adhatunk rajzainknak. Akár csak a HTML DOM, az SVG DOM is rendelkezik eseménykezelő attribútumokkal, melyeket hozzárendelhetünk az alakzatainkhoz, így akár a felhasználó események kezelésével nem csak dinamikusak, hanem interaktívak is lehetnek az alakzatok, melyekkel pl. látványos legördülő menüket, vagy még tovább menve akár olyan multimédiás alkalmazásokat is létrehozhatunk, mint a böngészős játékok.

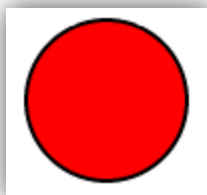
Példa kód SVG objektumra (10. kód):

```
1: <?xml version="1.0" standalone="no"?>
2: <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
3:     "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
4: <svg width="100%" height="100%" version="1.1"
5:     xmlns="http://www.w3.org/2000/svg">
6:     <circle cx="100" cy="50" r="40" stroke="black"
7:         stroke-width="2" fill="red"/>
8: </svg>
```

#### 10. kód

SVG objektumok alapvető kezelése HTML dokumentumokban

És a fenti kód eredménye (7. ábra):



7. ábra

A 10. kód eredménye webböngészőben<sup>23</sup>

<sup>23</sup> Saját készítésű kép a példa SVG objektum kód megjelenéséről. Készítve: 2010. április 14.

A példakódból jól látszik, hogy egy egyszerű XML dokumentumról van szó. Amiben több egy ilyen közönséges dokumentumtól, az az, hogy a 2-3. sorban található dokumentumtípus deklarációban `svg`-t adunk meg, illetve az objektumok típusára vonatkozó leírások a <http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd> címen érhetők el. A 4. sortól láthatjuk a lényeges SVG objektum leírást. Az 5. ábrán is látható képet a 6-7. sorban található `circle` címkéjű objektum hozza létre. A kör középponti koordinátáit a `cx` és `cy` attribútumok adják meg, a sugarát pedig az `r` attribútum. A `stroke` attribútummal adtam meg a körvonal fekete színét, melyet a `stroke-width` 2 képpont szélesre állít. A piros kitöltő színt a `fill` attribútum adja meg.

### 3.1.7. Két remek példa a HTML5 és a DHTML erejére

Az első példa elsősorban egy lehetséges felhasználási területet vetít előre, illetve a HTML5 képességeit boncolgatja. A <http://29a.ch/2010/3/24/normal-mapping-with-javascript-and-canvas-tag> címen található példa alkalmazás a `canvas` címke és egy kis JavaScript segítségével 3 dimenziós képeket állít elő, melyet a 8. ábrán láthatunk.



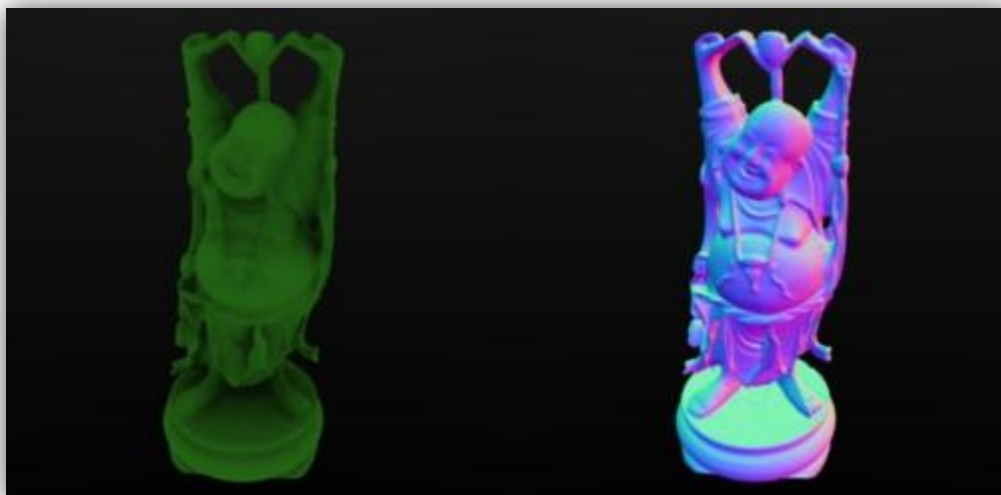
8. ábra

A HTML5 és a JavaScript segítségével előállított 3D képek<sup>24</sup>

Talán így állóképen nem látványos, de a weboldalra ellátogatva a fényforrás pozícióját manuálisan megválasztva mindjárt szembetűnő lesz a jelenség. A készítő leírása alapján az

<sup>24</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

effektust úgy hozza létre, hogy vesz egy képet, melyen a színeket tartalmazza, képpontról-képpontra pontosan, valamint egy másik képet, amelyen a felülete van árnyalásokkal együtt (9. ábra). A színeképet sima, közvetlen fényforrással képezi le, majd valós időben a második képpel együtt újra leképezi az ún. Phong árnyalási eljárással<sup>25</sup> együtt.



9. ábra

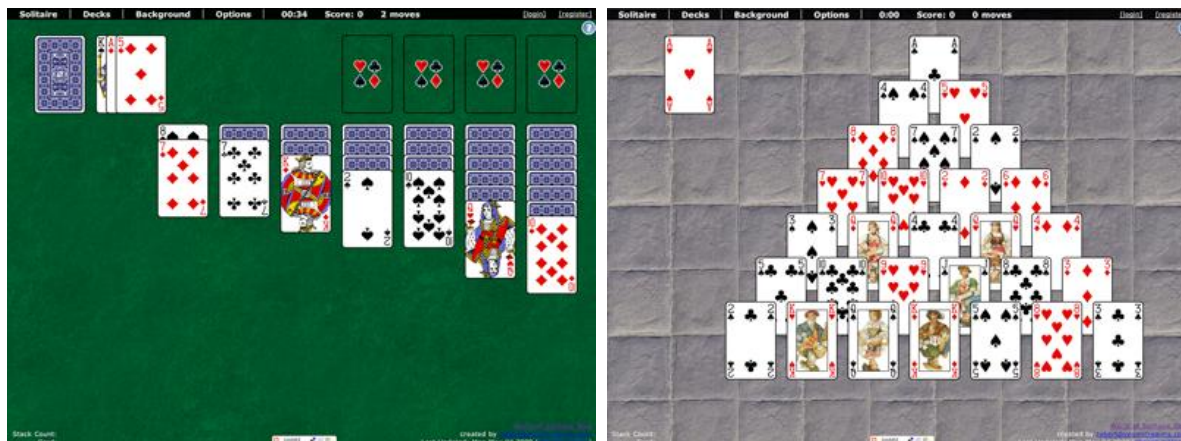
A Phong árnyalási eljárás során felhasznált képek<sup>26</sup>

A második példa leginkább DHTML demonstráció. A <http://worldofsolitaire.com> címen található kártyajáték a készítő szerint teljesen JavaScript alapú. Több mint 40 kártyajáték közül választhatunk, mindehhez kellemes grafikát is készített a programozó, hiszen választhatunk több háttér és paklifajta közül. Emellett lehetőség nyílik az eredmények közlésére is a játékban, így akár a többi játékosal is összemérhetjük tudásunkat. Mindemellett az animációk sebességét, megjelenését is tudjuk állítani.

<sup>25</sup> A Phong árnyalási eljárásról részletesebben az alábbi címen olvashatunk, melyre a készítő is utalást tesz: [http://en.wikipedia.org/wiki/Phong\\_shading](http://en.wikipedia.org/wiki/Phong_shading).

<sup>26</sup> Saját készítésű képernyő kép részlet az alábbi oldalról: <http://29a.ch/2010/3/24/normal-mapping-with-javascript-and-canvas-tag>, készítve: 2010. április 18.

A játékot a 10. ábrán láthatjuk.



10. ábra  
A World Of Solitaire játék<sup>27</sup>

### 3.2. MPEG LA és az AVC/H.264

Ez az alfejezet nem is igazán alkalmazásokról és alkalmazási lehetőségeikről szól, hanem egy olyan video tömörítési szabványról, mely jelenleg meghatározó, és talán egyeduralkodó szereplője a teljes multimédiás világnak, így nem kerülhetem el rövid ismertetését.

Az MPEG LA a világ egyik vezető szervezete a szabadalmak és szabványok kezelésében nem csak a szórakoztató elektronika, de a vegyipari, elektronikus kereskedelmi, oktatási, energiaügyi, egészségügyi és biotechnológiai, a gyártás és nyersanyagok felhasználása, a szállítás és vezeték nélküli technológiák területén egyaránt. A szabvány- és szabadalomkezelési folyamat során új piacot teremtettek, melynek célja, hogy a szabadalmak és szabványok fejlesztői, feltalálói szellemi termékükből profitálhassanak is, emellett az érintett piacok szereplői is kedvező feltételekkel juthassanak hozzájuk.

Az AVC/H.264 videó kódolási szabvány kezelője is az MPEG LA. Ez a szabvány ma a legelterjedtebb tömörítési eljárás a set top boxokban, média lejátszóknak, személyi számítógép szoftverekben, mobil eszközökön (mobil telefon és televízió), Blu-ray lemezekben,

<sup>27</sup> A képek forrása: <http://href.hu/x/c3d9> és <http://href.hu/x/c3da>, letöltve 2010. április 18.

lemezlejátszókon és rögzítő eszközein, játékgépeken, fényképező gépeken, videó kamerákon, és még sorolhatnám hosszasan a különféle eszközöket.

Az AVC/H.264 1999-ben nemzetközi ISO szabvánnyá<sup>28</sup> vált. A szabvány nem csak a videó képi tömörítését írja le, hanem a hozzá tartozó hangét is. A H.264 hatalmas előrelépést jelent a videó kódolásban: a korábbi uralkodó szabvánnyal, az MPEG-2-vel szemben (mely a DVD videók tömörítési eljárása is) ugyanazt a minőséget harmad akkora bitráta mellett teljesíti, de ugyanazon a bitrátán négyszer akkora felbontásra képes minőségromlás nélkül (11. ábra).



11. ábra

Az MPEG-2 és a H.264 összehasonlítása ugyanazon bitrátán<sup>29</sup>

Mindezek függvényében a H.264 a korábbi szabványokkal ellentétben jobb képminőséget nyújt, ha kihasználjuk a benne rejlő lehetőségeket. A H.264 ahelyett, hogy az egész képeket feldarabolná, és úgy kezelné az eljárás során – mely egyértelmű töredezettséget eredményez a képkockákban –, inkább magát az egész képet tömöríti, és ahogy növeljük a tömörítés mértékét, úgy egyre elmossa a részleteket. Fontos megemlíteni, hogy ez a tömörítési eljárás még igen fiatalnak számít, így az idővel, ahogy egyre finomítják az algoritmusát, jobb és szebb eredményeket várhatunk tőle pont úgy, ahogy az az MPEG-2 esetében is volt.

<sup>28</sup> ISO/IEC 14496

<sup>29</sup> Dokumentum részlet a forrásból: [http://images.apple.com/quicktime/pdf/H264\\_Technology\\_Brief.pdf](http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf)

Az 1. táblázatban láthatjuk, hogy adott felbontásokhoz mekkora bitráta elegendő, hogy minőségi képet kapjunk adott felhasználási területeken.

Felhasználási terület	Felbontás és képkockaszám/másodperc	Bitráta
Mobil alkalmazások (3G)	176x144, 10-24 fps	50-160 kbps
Webes videók (SD)	640x480, 24 fps	1-2 Mbps
Nagyfelbontású videók (HD)	1280x720, 24fps	5-6 Mbps
Full HD videók	1920x1080, 24fps	7-8 Mbps

1. táblázat

H.264 tipikus használata különféle területeken<sup>30</sup>

A felbontások közötti különbséget a 12. ábra szemlélteti.



12. ábra

Különböző felbontások és általuk hordozott képi információmennyiség<sup>31</sup>

### 3.3. Adobe és a Flash

Az Adobe vállalatot 1982-ben alapította Chuck Geschke és John Warnock Kaliforniában. Ők ketten egy olyan probléma megoldásával szálltak be a piacra, ami régóta égető volt: szöveg- és képi tartalom megjelenítése úgy a képernyőkön, hogy az a

<sup>30</sup> Dokumentum részlet a forrásból: [http://images.apple.com/quicktime/pdf/H264\\_Technology\\_Brief.pdf](http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf)

<sup>31</sup> Dokumentum részlet a forrásból: [http://images.apple.com/quicktime/pdf/H264\\_Technology\\_Brief.pdf](http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf)

nyomtatásban is azonosan jelenjen meg. Egy évvel később az Adobe PostScript technológiával forradalmasították a publikációs szoftvereket. Ez az innováció a cég részéről a későbbiekben sem hagyott alább. Olyan szoftverekkel jelentek meg, mint az Adobe Illustrator és az Adobe Photoshop, valamint a PDF<sup>32</sup> dokumentum formátum, melyek a maguk területén a mai napig piacvezető szerepet töltenek be.

A 2000-es években felismerték, hogy a WWW robbanásszerű terjedésével egy új, még kiaknázatlan piac nyílik meg a cégek előtt. Ennek jegyében egy gyorsan lebonyolított akvizíciós eljárás során 2005 áprilisában felvásárolták a Macromedia nevű céget, mintegy 3,4 milliárd USD összegért. Ez a cég innovációs eszközök sorát vonultatta fel a publikus internet rövid történetében. Az ő nevükhöz fűződnek olyan alkalmazások, mint a Flash, ShockWave és a DreamWeaver. Ezen fontos és nagysikerű alkalmazások fejlesztését az Adobe továbbfolytatta.

Dolgozatom szempontjából az Adobe Flash kulcsfontosságú szereplő, hiszen ez ma az uralkodó multimédiás platform az egész weben. Az internetes videók több mint 75%-a Flash alkalmazásokban jelenik meg. Több mint 1 milliárd olyan eszköz van ma is az embereknél valamint kereskedelmi forgalomban, mely képes kezelni a Flash alkalmazásokat. A Flash segítségével azonban nem csak videókat jeleníthetünk meg a felhasználónál, hanem készíthetünk vele a weboldalak funkcionalitását kiegészítő alkalmazásokat, játékokat vagy akár prezentációkat.

A Flash eredetileg egy animáció készítő programnak indult, mely megjelenésekor úttörőnek számított a dinamikus weboldalak létrehozásában. Kis mérete, egységes megjelenése, valamint megbízhatósága az egyik legstabilabb kliens oldali interaktív technológiává teszi. Létezik hozzá egy ún. Flash Player nevű program, mely a Flashben megírt tartalmakat hivatott lejátszani. Megjelenhet önálló programként, és böngészőbe beépülő alkalmazásként.

Tartva magamat a leírtakhoz, miszerint az Adobe Flash a legelterjedtebb multimédiás eszköz a teljes weben, szeretném bemutatni a használatát alapvető példákon keresztül.

---

<sup>32</sup> PDF – Portable Document Format: platform és alkalmazás független dokumentum formátum, melynek segítségével a dokumentumok a különböző eszközökön azonos megjelenéssel tekinthetők meg.

### 3.3.1. Egy példa animáció készítése

Az első példa egy, a mini-alkalmazásunkban keresztülpattanó labda lesz. Az animáció elkészítéséhez az Adobe saját Flash CS4 Professional próbaverziós alkalmazását használok, mely letölthető a gyártó honlapjáról a <http://www.adobe.com/support/flash/downloads.html> címen.

Először is kerestem egy, az animációhoz alkalmas képet az interneten, mely az 13. ábrán<sup>33</sup> látható. Választásom azért esett erre a képre, mert PNG<sup>34</sup> formátumú, így beállítható benne egy olyan szín, mely, ha az egészet egy másik képre helyezük, akkor áttetsző lesz. Így könnyen választhattam hozzá egy hátteret, mely a 14. ábrán<sup>35</sup> látható.



13. ábra



14. ábra

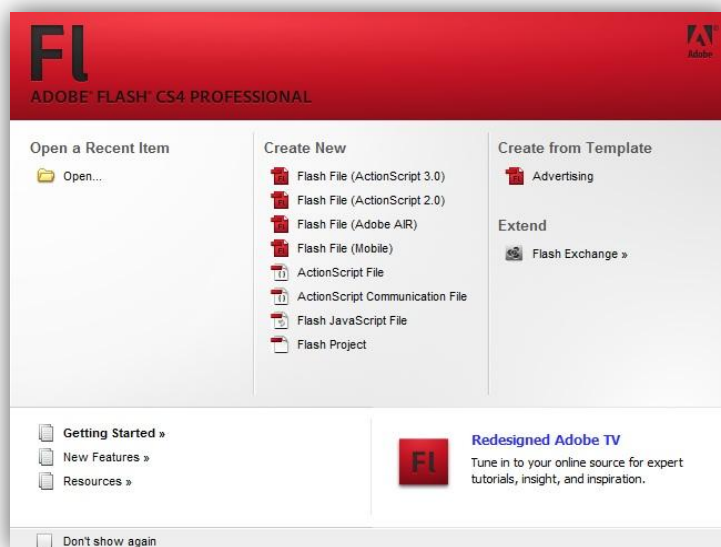
Az Adobe Flash CS4 Professional alkalmazást indítva a 15. ábrán látható képernyő (részlet) fogad minket.

---

<sup>33</sup> A kép az alábbi címen található: [www.iconarchive.com/icons/aha-soft/sport/256/football-icon.png](http://www.iconarchive.com/icons/aha-soft/sport/256/football-icon.png), letöltve: 2010. április 17.

<sup>34</sup> PNG – Portable Network Graphics: veszteségmentes tömörítésű képfórmátum, mely 1996 óta W3C ajánlás. Forrás: [www.w3.org/Graphics/PNG/](http://www.w3.org/Graphics/PNG/), letöltve: 2010. április 17.

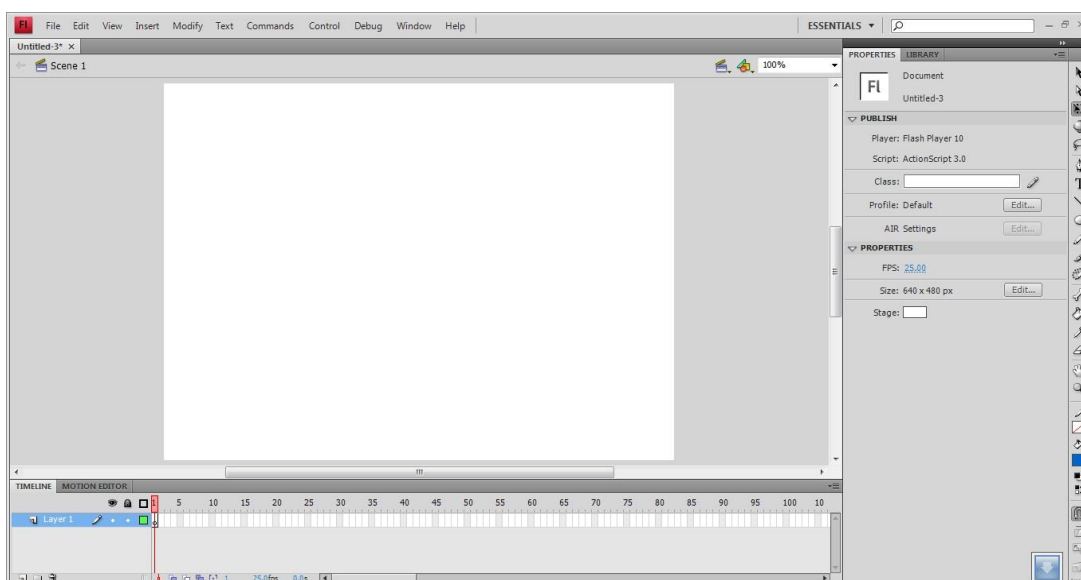
<sup>35</sup> A kép az alábbi címen található: [gallery.achituv.com/d/19382-2/soccer\\_field.jpg](http://gallery.achituv.com/d/19382-2/soccer_field.jpg), letöltve: 2010. április 17.



15. ábra

Az Adobe Flash CS4 Professional alkalmazás üdvözlő képernyője<sup>36</sup>

Ahogy látható, az alkalmazás már itt számos lehetőséget kínál számunkra, úgymint meglévő projektek megnyitása, újak létrehozása, sablonokból való új projekt létrehozása, ismertetőik megtekintése, stb. Számomra a 15. ábrán középen látható Create New Flash File (ActionScript 3.0) lehetőség szükséges.

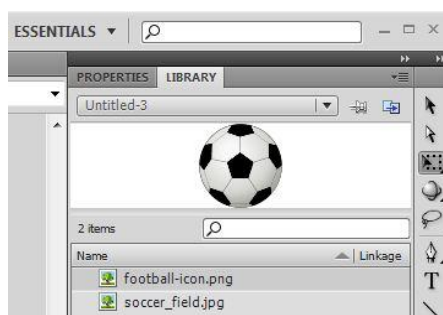


16. ábra

Az Adobe Flash CS4 Professional szerkesztő felülete

<sup>36</sup> Saját készítésű képernyőkép részlet. Készítve 2010. április 17.

A 16. ábrán látható szerkesztő felületet kapjuk, melyet már beállítottam úgy, hogy a mini-alkalmazásunknak megfelelő legyen, azaz a dokumentum méretet 640x480 felbontásra állítottam, valamint az animáció másodpercenkénti képkocka számát 25-re. Innen nincs más dolgunk, mint importálni a két korábban már bemutatott képünket, megfelelően átméretezni és az animációt elkészíteni. Az importáláshoz a File menü Import, azon belül az Import to Library... opcióját kell választani, ami után egy fájlrendszer böngésző ablak jelenik meg, ahol a kívánt fájljainkat kell betallózni. Ezután a jobboldali eszközpanel Library fülén megjelennek az általunk megnyitott fájlok (17. ábra).

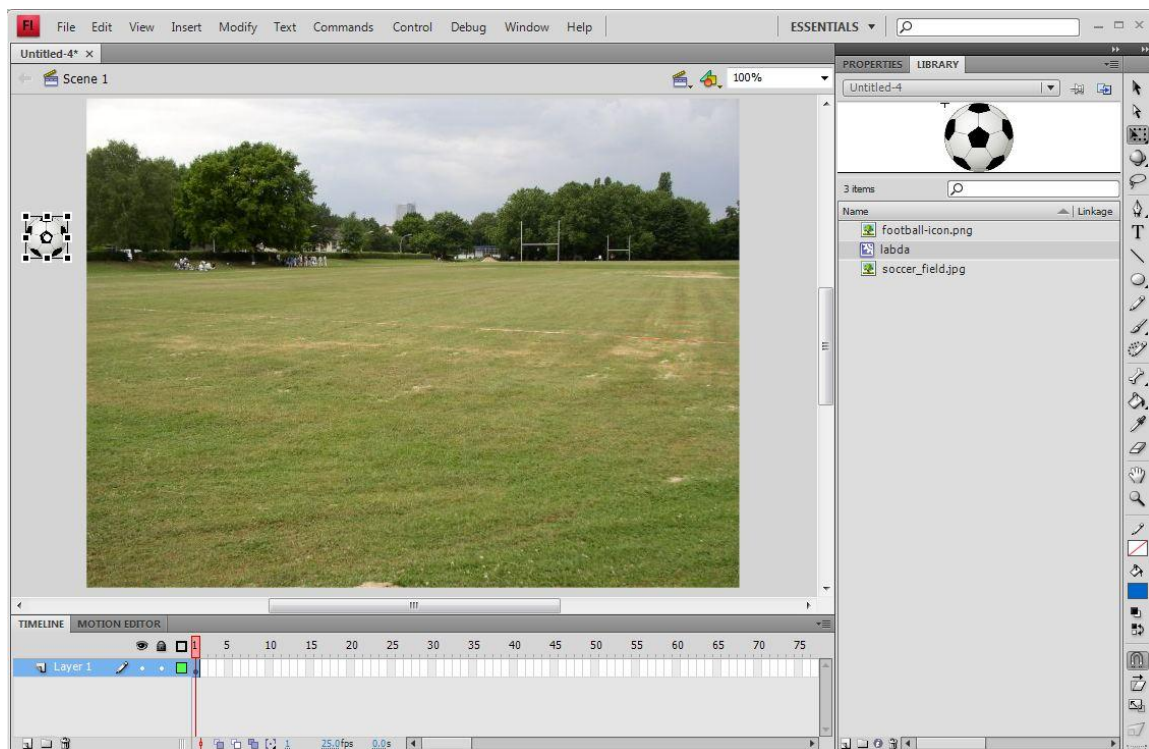


17. ábra

A megnyitott háttér és labda kép az eszköztár panel Library fülén<sup>37</sup>

Ahhoz, hogy a labdát animálhassuk, szimbólummá kell konvertálnunk. Miután kijelöltük, a Modify menü Convert to Symbol... opciójával megtehetjük ezt a felugró ablakban, melyben a Name mezőben meg kell neki adni egy nevet. Ha ezt megtettük, akkor a szimbólum szintén megjelenik a Library fülön. Ezek után elhelyezzük a szimbólumot és a háttérrel a kívánt pozíciókban úgy, hogy hasonló képet kapjunk, mint a 18. ábrán.

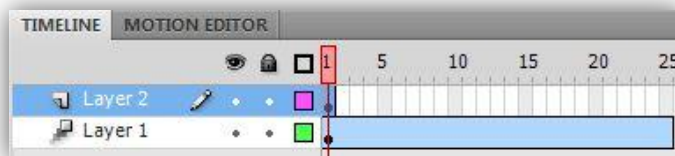
<sup>37</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 17.



18. ábra

A szerkesztő felület a szimbólum létrehozása és az objektumok elhelyezése után<sup>38</sup>

Ahogy látható, a szimbólumot konzekvensen labdának neveztem el. A labda szándékosan helyezkedik el a látható területen kívül, hogy az animáció kezdetén látványosabb legyen, ahogy kintről úszik be. A következő lépés a labdát animációs objektumként definiálni. Ehhez a labdára jobb egérgombbal kell kattintani, és a Create Motion Tween opciót választani a helyi menüben. Ekkor az ablak alján található időszávon az animáció a labda objektumra külön réteggént megjelenik.



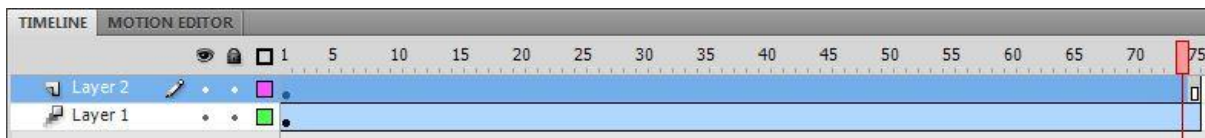
19. ábra

Az időszá, miután a labdát animációs objektumként definiáltuk<sup>39</sup>

<sup>38</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 17.

<sup>39</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 17.

A 19. ábrán látható, hogy a háttér objektum a Layer 2 nevű réteg maradt, és az animációban nem vesz részt teljesen. Ugyanakkor az is látható, hogy a Layer 1 nevű réteg, mely labda objektumot takarja, a 25-ös skáláig tart, mely annak megfelel, hogy a dokumentum tulajdonságait korábban 25 képkocka/másodperc formátumúra állítottam. Nyilvánvaló, hogy számunkra a Layer 2 nevű rétegnek is ugyanannyi ideig kell részt vennie az animációban, mint a labdát takaró Layer 1 rétegnek. Ugyanakkor kibővíttem az egész animációt 75 képkocka hosszúra, azaz 3 másodpercesre, hogy ne legyen túl gyors az egész. Az utóbbit könnyen megtehetjük, ha a labdát takaró réteg 25-ös végéhez mozgatjuk az egeret (tehát a világoskék sáv jobb széléhez), ahol megjelenik egy kettős nyíl egérkurzor. Így kattintva, folyamatosan nyomva tartva az egér gombot a kívánt hosszúságúra nyújtjuk azt. Ha ezt megtettük, akkor a rózsaszín jelölő, mely az aktuális képkockát jeleníti meg számunkra, szintén az animáció végére ugrik, és ezzel együtt a háttérünk fehérre válik. Ez is jelzi számunkra, hogy a háttérünk megjelenési idejét is ki kell terjesztenünk az animáció végéig. Ezt megtehetjük úgy, hogy a háttérrel jelképező réteg első képkockájára kattintunk, folyamatosan nyomva tartjuk az egér gombot, és az animáció vége előtti utolsó képkockáig húzzuk azt. Ekkor egy kijelölés jön létre, melyre jobb egér gombbal kattintva a helyi menüben az Insert frame opciót választva a kívánt eredményt érjük el, és a 20. ábrához hasonló eredményt kapunk.



20. ábra

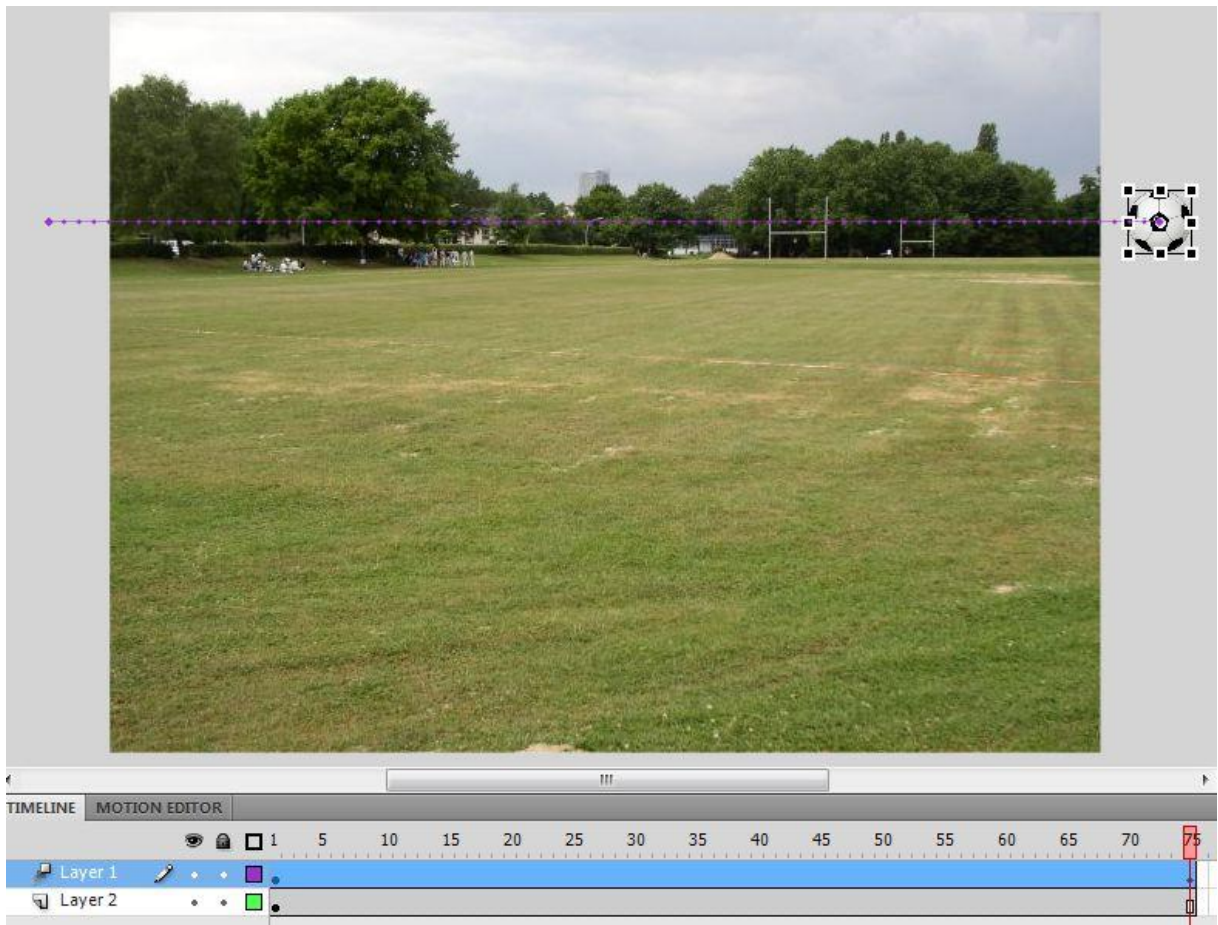
Az időszám, miután kiterjesztettük az animáció hosszát mindkét objektumra<sup>40</sup>

Egy fontos dolog maradt. Ugyebár a Layer 2 jelképezi a háttérrel, a Layer 1 pedig a labdát. Az időszám az elhelyezkedésük egymáshoz képest a megjelenítésük sorrendjét is jelenti, tehát a Layer 1-et előre kell mozgattunk. Ez egyszerű, rákattintunk, folyamatosan nyomva tartjuk az egér gombot, és felülre húzzuk.

Szinte már kész is van az animáció. Nem maradt más hátra, mint a labda konkrét elmozdulását hozzáadni. Ez egy egyszerű művelet, ugyanis magán az ábrán el tudjuk végezni.

<sup>40</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 17.

A rózsaszín jelölőt az időszávon az utolsó képkockára mozgatjuk, majd a labdát a 21. ábrán is látható pozícióra mozgatjuk.



**21. ábra**  
**A labda animálásának első lépése<sup>41</sup>**

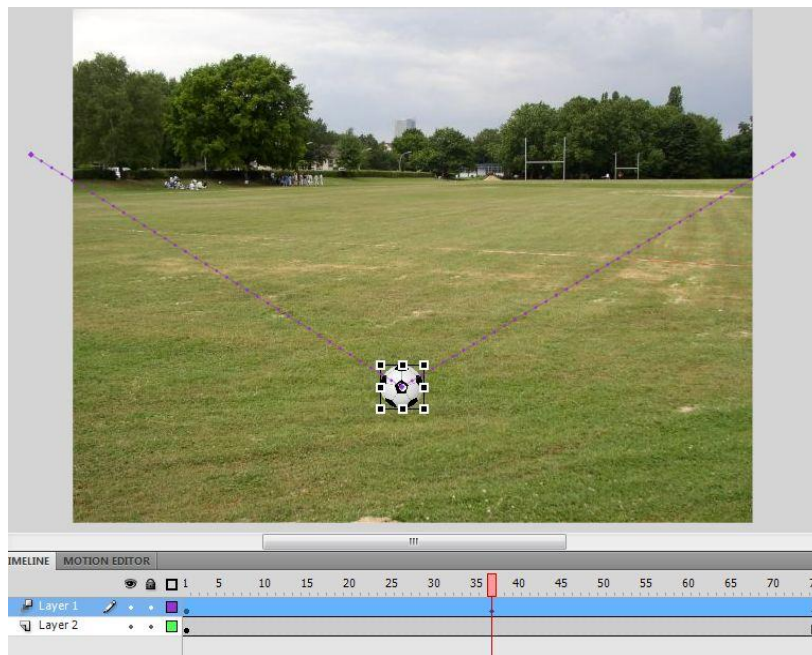
A Control menü Test movie menüpontjával megnézhetjük, hogy jelen állapotában hogy néz ki az animációnk. Nem nyújt túl nagy élményt és élethűséget, hiszen gyakorlatilag vízszintesen végigrepül a háttér előtt. Ezért egy kicsi módosítunk rajta úgy, hogy keresztül pattanjon az ablakon.

A rózsaszín jelölőt a 37-ik képkockához húzzuk, ez körülbelül az animáció fele. Itt a labdát a kívánt mélységig elmozdítjuk lefele. Az eredményt szintén megtekinthetjük az előbb ismertetett Control menü Test movie menüpontjával.

<sup>41</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 17.

Az eredmény közelít a kívánthoz, de még mindig kissé kiábrándító, hisz túl egyenes vonalakon mozog még mindig a labda, ami nem túl élethű. Hogy ezt kompenzáljuk, a beeső és kipattanó pályáit egy kicsit íveljük.

A jobb szélén található eszköztárról kiválasztjuk a legfelsőt, melynek a neve Selection Tool, azaz Kiválasztó Eszköz. Az egérkurzort a baloldali beeső egyenes fölé mozgatjuk – mely a 22. ábrán lila színnel van jelölve –, majd kattintva és folyamatosan nyomva tartva az egérgombot a kívánt ívűre formáljuk.



**22. ábra**  
**Kezdetleges pattanó effektus szerkesztése a labdán<sup>42</sup>**

Ugyanezt a műveletet ismételjük meg a jobboldali pálya egyenesre is mindaddig, míg mindkét oldalon a kívánt, 23. ábrán is látható eredményt el nem érjük.

<sup>42</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.



23. ábra

A beeső- és kipattanó pályák a módosítás után<sup>43</sup>

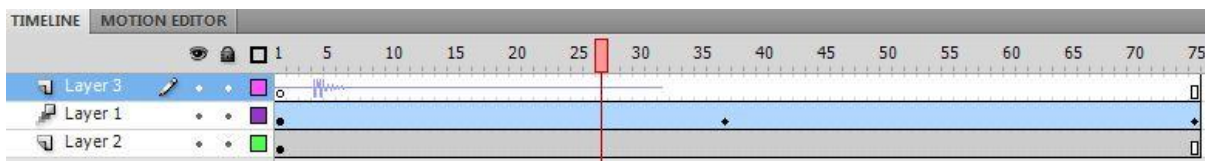
Ha ezt is leellenőrizzük a már ismert menüpontban, akkor láthatjuk, hogy már sokkal kellemesebb a labda mozgása. Ezek után a pattanáshoz egy hangeffektust<sup>44</sup> is hozzá adok.

Miután a megfelelő hangot kiválasztottuk, szintén a File menü Import és ezen belül az Import to Library... menüpontot választva megnyitjuk azt. Ekkor az eszköztár Library fülén megjelenik a hangunk. Ha nem megfelelő hosszú a hangunk, mint az én esetemben, akkor úgy kell az animációhoz hozzáadnunk azt, hogy a megfelelő helyre lehessen azt igazítani. Ezt úgy tudjuk elérni, hogy az Insert menü Timeline és azon belül a Layer menüpontot választjuk. Ekkor az időszávon megjelenik egy új réteg, mely alapértelmezetten az animációkkal megegyező hosszúságú lesz. Ezt a réteget kijelöljük, majd a szerkesztési területre húzzuk a hangunkat a Libraryból.

---

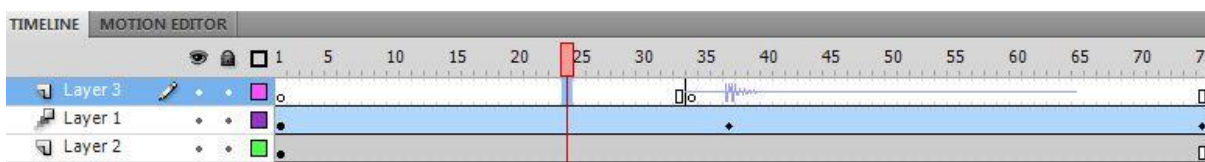
<sup>43</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

<sup>44</sup> A hangeffektust a [www.freesound.org/samplesViewSingle.php?id=37155](http://www.freesound.org/samplesViewSingle.php?id=37155) oldalról töltöttem le, mely regisztrációhoz van kötve. Letöltve: 2010. április 18.



24. ábra  
Az időszám a hang hozzáadása után<sup>45</sup>

A 24. ábrán láthatjuk az eredményt, ahogy azt is, hogy megjelenik a hangunk hullámmintája, amelyből egyértelműen látszik, hogy rosszul van pozicionálva. Ahhoz, hogy ezt korrigáljuk, a hanghullám mintát elejétől a végéig kijelöljük, majd egyszerűen az egérrel rákattintunk, folyamatosan nyomva tartjuk a gombot, és a kívánt pozícióba mozgatjuk (25. ábra).



25. ábra  
Az időszám a hang pozíciójának korrigálása után<sup>46</sup>

Ha túlnyúlna a hang az animáción, akkor a végét eltávolíthatjuk úgy, hogy a kívül eső részt kijelöljük, majd jobb egérgombbal a helyi menüt felhozzuk, és a Remove Frames opciót választjuk.

Ezután, ha megnézzük az ismert Test Movie menüponttal az animációkat, már a hangunkat is hallani fogjuk. Ha ezen a ponton szeretnénk olyan formátumban menteni az animációt, hogy a HTML dokumentumba ágyazható legyen és/vagy Flash Player segítségével le lehessen játszani, akkor a File menü Export, azon belül pedig az Export Movie... menüpontján keresztül tehetjük meg. Egy fájlböngésző ablak jelenik meg. Itt nincs más teendők, mint a kívánt mentési helyet kitallózni és a fájl nevet megadni. A kiterjesztést hagyjuk az alapértelmezett SWF-en. Ez az alapértelmezett formátuma a Flash fájloknak.

<sup>45</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

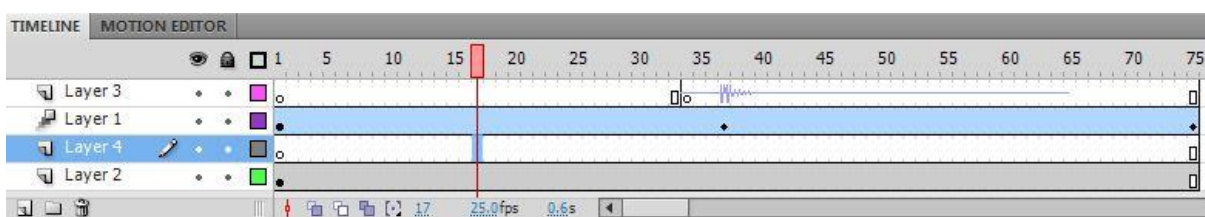
<sup>46</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

### 3.3.2. A példa animáció bővítése interakciós lehetőséggel

Az animációnk jelen állapotában igen egyszerű, a felhasználó számára kizárólag lejátsza egyszer a labda pattanását. Ahhoz, hogy demonstráljam az Adobe Flash képességeit, elengedhetetlen, hogy egy példán keresztül bemutassam, hogyan is lehet készíteni vele interaktív alkalmazásokat. Ebben az esetben ez egy egyszerű lejátszó gomb felvétele lesz az utolsó képkockán, majd egyedi programkód hozzáadása, melynek segítségével lekezeljük azt az eseményt, amikor a felhasználó a gombra kattint. A kiváltott esemény pedig nem lesz más, mint az animáció újrajátszása.

A Flash alkalmazások nem csupán képi és hanganyagok egyirányú közvetítésére képesek, hanem az ún. ActionScript segítségével komplett alkalmazások is készíthetők benne. Az ActionScript egy objektum orientált programnyelv, mely ECMAScriptre épül, ami alapja a JavaScriptnek is, ezt pedig a III.1.c. fejezetben korábban ismertettem. Tehát ha valaki járatos a JavaScript programozásban, igen gyorsan meg tudja tanulni az ActionScript programozását is. Az ActionScript kódok az ún. ActionScript Virtual Machine (AVM) virtuális gépben futnak, amit a Flash Player tartalmaz. Az ActionScript képes a kivételek és a felhasználói események kezelésére.

E rövid ismertető után visszatérve az előző animációnkhoz kezdetnek egy új réteget kell felvennünk. Ezt az előzőekben is ismertetett Insert menü Timeline és azon belül Layer menüpontja segítségével tehetjük meg.



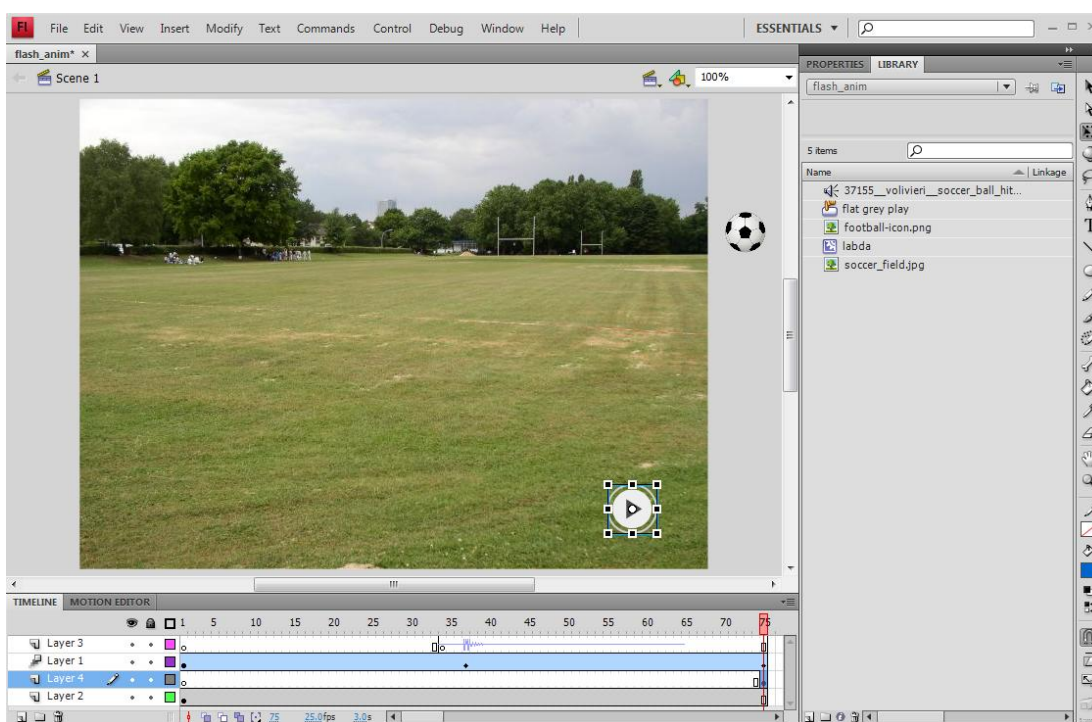
26. ábra  
Az időszám az új réteg felvétele után<sup>47</sup>

A 26. ábrán az új réteg Layer 4 néven szerepel. Látható, hogy a teljes animáció hosszával került ez is hozzáadásra. Számunkra ez nem jó, hisz azt szeretnénk, hogy az animáció végén jelenjen meg a lejátszó gomb. Ezért ki kell jelölni az első képkockától az

<sup>47</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

utolsó előtti képkockáig a Layer 4 réteg képkockáit, majd jobb egér gombbal felhozva a helyi menüt a Remove Frames opciót kell választani. Ekkor a megmaradó utolsó képkocka az első képkockára kerül, mellyel nincs más teendőnk, mint kijelölni, és az animáció végére mozgatni. Ha ez megtörtént, győződjünk meg róla, hogy az időszávon a kijelölés továbbra is az utolsó képkockán van a Layer 4 rétegen.

A Window menü Common Libraries és azon belül a Buttons menüpontot megnyitva megjelenik egy kis ablak, ahol kiválaszthatunk egy nekünk tetsző gombot, melyet elhelyezhetünk az animációban. Én most a listán a 'playback flat' menüből a 'flat grey play' gombot választom. Ha ez megvan, akkor egyszerűen az animációra kell húzni. Ezután ezt a Buttons ablakot akár be is zárhatjuk. Valószínűleg kicsinek fogjuk találni a gombot, ezért át kell méretezni. A jobboldali eszköztáron a Free Transform Toolt kell választani, minek hatására kis négyzetek jelennek meg a gombon. Ezekre kattintva az egérrel egyszerűen átméretezhetjük azt (27. ábra).



27. ábra

A szerkesztő felület a lejátszó gomb felvétele és átméretezése után<sup>48</sup>

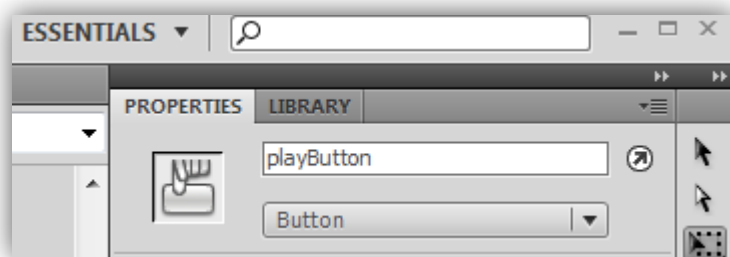
<sup>48</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

Egy kicsit csinosítsunk a gombon. A jobb oldali eszköztáron a Properties fülre kattintva, majd a Display opciót választva a Blending legördülőben adjunk meg neki áttetszési effektust. Legyen ez az Overlay. Az eredményt 28. ábra mutatja.



28. ábra<sup>49</sup>

A továbbiakban pedig felvesszük az eseménykezelést a gombhoz. A gomb, mivel egy ActionScript objektum, egy névvel kell ellátnunk. Ezt az előbbi Properties fülön rögtön legfelül tehetjük meg. A 29. ábrán látható, hogy én a playButton nevet adtam neki.



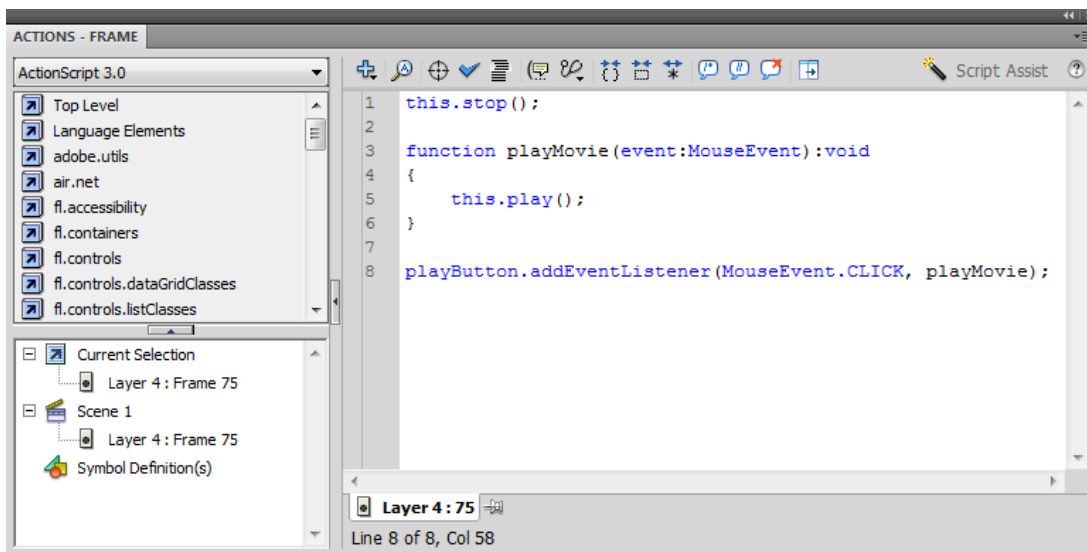
29. ábra<sup>50</sup>

Következő lépésben a jobb egérgombbal a Layer 4 réteg utolsó képkockájára kattintva a helyi menün az Actions menüpontot kell választanunk.

---

<sup>49</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

<sup>50</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.



30. ábra  
ActionScript szerkesztő ablak<sup>51</sup>

A megjelenő (30. ábra) ablakban szerkeszthetjük a hozzá tartozó ActionScript kódot. A 11. kódot illesztjük be.

```

1:  this.stop();
2:  function playMovie(event: MouseEvent): void {
3:      this.play();
4:  }
5:  playButton.addEventListener(MouseEvent.CLICK, playMovie);
  
```

11. kód  
Az eseménykezelő ActionScript kód

Az 1. sorban lévő `this.stop()` utasításra azért van szükség, mert a Flash Player alapértelmezésben ismétli folyamatosan előlről az animációt, ezzel pedig megakadályozzuk ezt. A `this` objektum az aktuális, teljes animáció referenciájára hivatkozik, a `stop` metódusa pedig a kívánt műveletet végzi el.

Ezután a 2-4. sorban definiálom a lejátszó gombra való kattintáskor kiváltódó esemény kezeléséért felelős műveletet. Ez nem lesz más ugyebár, mint a lejátszás újraindítása. Ezt egy eljárásba ágyazom, melynek az 5. sorban veszem hasznát. Először is röviden az eljárásról.

<sup>51</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

Ahogy látható, paraméterként átadunk neki egy `MouseEvent` osztályhoz tartozó objektumot, melynek példányai az egérrel kiváltott eseményekkor jönnek létre. Ezt az objektumot én most nem használom fel, de rajta keresztül ellenőrizhetnénk, hogy milyen objektumhoz tartozott az esemény, és pontosan milyen egérművelet volt az. Nálam erre azért nincs szükség, mivel konkrétan a lejátszó gombhoz rendeltem az eseménykezelést, valamint az 5. sorban kizárólag egérekattintásra definiálom az eseménykezelőt. A visszatérési típus `void`, azaz nem várunk vissza értéket az eljárástól. A 3. sorban pedig a tényleges művelet van, mely szerint az animációt – melyre a `this` objektum hivatkozik – újraindítom a `play` metódus segítségével.

Az 5. sorban a lejátszó gombhoz, melynek példányát korábban `playButton` névvel illettünk, az `addEventListener` metódus segítségével felvesszük az eseménykezelőt. Első paraméterként egy esemény típust adunk át, mely itt egy egérekattintás lesz, második paraméterként azt a függvény- vagy eljárásnevet, amelyet szeretnénk, hogy lefusson az esemény kiváltásakor. Ez utóbbi az előzőleg definiált `playMovie` lesz.

Így összességében, ha az animáció a végéhez ér, az egész kódsor feldolgozásra kerül, ám érvényes csak az 1. és az 5. sor lesz, azaz az animáció az 1. sornak megfelelően megáll, majd az 5. sor szerint a lejátszó gombhoz egy eseménykezelő rendelődik. Ha a felhasználó rákattint a lejátszó gombra, akkor az eseménykezelő szerinti `playMovie` metódus meghívódik, és az animáció újra lejátszásra kerül.

### **3.3.3. A példa animáció HTML dokumentumban való megjelenítése**

Bemutattam, hogy egy egyszerű animáció létrehozása nem igényel komoly ismereteket az Adobe Flash Professional szoftverében, valamint az ActionScript segítségével hogyan építhetünk fel akár webalkalmazásokat is, hiszen a bonyolultabbak létrehozása is ilyen alapokra épül. Ezek után nem marad más, amit ezen a példán keresztül be kell mutatnom, hogy hogyan is jeleníthetjük meg az animációnkat, és ebből következően bármilyen Flash animációt vagy alkalmazást HTML dokumentumokban.

Először is el kell mentenünk az animációt az arra alkalmas formátumba, melyet a III.3.a. alfejezet végén is ismertettem. Tehát a File menü Export és azon belül Export Movie... menüpontját kell választanunk, amire a felugró fájlböngésző ablakban ki kell

tallóznunk a mentési mappát és megadnunk a fájlnévet. Ahogy korábban is írtam, a fájl típus maradjon SWF, hisz ez az alapértelmezett formátuma az Adobe Flash fájloknak.

Ezután létrehozunk neki egy egyszerű XHTML dokumentumot, mely a beágyazással együtt így fog kinézni (12. kód):

```
1: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3: <html xmlns="http://www.w3.org/1999/xhtml"
4:     xml:lang="en" lang="en">
5: <head>
6:     <meta http-equiv="Content-Type"
7:         content="text/html; charset=iso-8859-2" />
8:     <title>Pattogó labda</title>
9: </head>
10: <body bgcolor="#ffffff">
11:     <h1>Pattogó labda</h1>
12:     <object width="640" height="480">
13:         <param name="labda" value="flash_anim.swf">
14:         <embed src="flash_anim.swf" width="640" height="480">
15:     </embed>
16:     </object>
17: </body>
18: </html>
```

#### 12. kód

#### A példa animáció HTML dokumentumba való beágyazása

Látható, hogy ez valóban egy egyszerű XHTML kód. Ami számunkra érdekes jelen esetben, az a 12-16. sorok között található. Ez a minimális kód mennyiség, amely szükséges ahhoz, hogy egy Flash objektum megjelenjen HTML dokumentumokban.

A 12. sorban található `object` címke az Internet Explorer kompatibilis böngészőinek szól, mert ezek csak ezen keresztül értelmezik a beágyazott tartalmakat. Attribútumként megadjuk az animáció méretét is, mely 640 pixel széles és 480 pixel magas lesz.

Továbbá az `object` címke közé kell zárnunk a beágyazott tartalom paramétereit konzekvensen `param` címkékkel megadva. A 13. sorban az animációnk labda nevet kap, és `value` attribútumon keresztül pedig a Flash fájl elérését adjuk meg. Mivel nálam a HTML dokumentum rögtön a Flash fájl mellett helyezkedik a könyvtárszerkezetben, így elég volt csak a fájl nevét megadnom.

A 14-15. sor `embed` címkéje a nem Internet Explorer alapú böngészőknek szól. Tulajdonképpen hasonló felépítésű, mint az `object` címke. Az `src` attribútumon keresztül megadjuk a Flash fájl elérési útvonalát, valamint szintén a `width` és `height` attribútumokkal az animáció szélességét és a magasságát..

Végül egy böngészőben megnyitva a HTML dokumentumot, a 31. ábrán látható módon jelenik meg az animációnk.



**31. ábra**  
Az animáció megjelenése webböngészőben<sup>52</sup>

<sup>52</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

### 3.4. Google és a YouTube

Korábban említettem a YouTube-ot, ami a webes multimédiára olyan jelentős hatással van, hogy kiemelten is kezelnem kell. A YouTube a Google tulajdonában lévő szolgáltatás. A Google alapítói Larry Page és Sergey Brin még a Stanford Egyetemi éveik alatt, 1996-ban kifejlesztett egy újfajta, máig nagyrészt titkos online keresési eljárást, mely forradalmasította a tartalomkeresést. Az eljárás tömör lényege, hogy megnézi hány link mutat az adott oldalra, és ez alapján állítja fel a találati listát. A Googlet, mint céget 1998-ban jegyezték be. A vállalat egészen 2004-ig adományokból tartotta fenn önmagát, amikor is a cég részvényeit a tőzsdére vitték. Azóta a Google az internetes reklámpiac átalakításával milliárdos üzletté nőtte ki magát: 2009-ben a cég értéke több mint 23 milliárd dollár volt, ami mellé 6 milliárd dollárt meghaladó éves profit társult a 20 ezernél kicsivel több alkalmazotthoz összesen.

A YouTube-ot 2005 februárjában alapították és ma a domináns online videó közvetítő- és megosztó szolgáltatás. 2006 novemberében a Google felvásárolta, és az egyik legvitatottabb akvizíciónak tartják, hisz a Google számára a mai napig veszteséges szolgáltatás. Ennek ellenére fejleszti és életben tartja, mivel a reklám felületeit ide is igyekszik behozni, hogy profitot termelhesen. A YouTube számos tartalomszolgáltatóval áll kapcsolatban, mint pl. a CBS, BBC, Universal Music Group, Sony Music Group, hogy ezzel még több, jó minőségű tartalmat közvetíthessen. Ennek volt eredménye például 2009. október 25-én, hogy 16 országban élőben közvetítették a U2 zenekar koncertjét<sup>53</sup>. Mára a YouTube számos optimalizált felületen képes megjeleníteni a videó tartalmakat, olyanokon mint pl. Nintendo Wii, Sony Playstation, Apple TV, TiVo és számos IPTV készüléken. Ezekről egy későbbi fejezetben részletesebben is tárgyalok.

A YouTube jelenleg az Adobe Flash-t használja a videók közvetítésére. Ezen túlmenően lehetőséget ad a felhasználók számára, hogy videóikat megoszthassák bárkivel, akár HTML oldalakba is beágyazhassák. Lehetőség van a videókat közvetlenül a megjelenési oldalukon értékelni, kommentálni. Ma már képes a nagyfelbontású, full HD-s felbontás kezelésére, azaz az 1920 pixel széles és 1080 pixel magas videók kezelésére, mindezt H.264 kodek segítségével. A fejlesztések között van pl. a HTML5-ben rejlő képességek kihasználása, úgymint a `video` címkére való átállás, a mostani Adobe Flash platformról. Ez

---

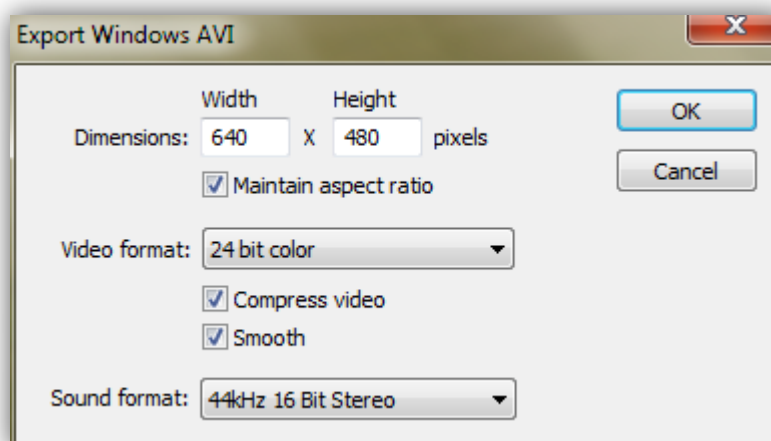
<sup>53</sup> Forrás: <http://quart.hu/cikk.php?id=4297>, letöltve: 2010. április 18.

jelenleg beta teszt állapotban elérhető a <http://www.youtube.com/html5> címen az arra alkalmas böngészőkön keresztül.

### 3.4.1. Tartalom közzététele a YouTube-on

Mindenekelőtt szükségünk van egy felhasználói fiókra a YouTube rendszerében. Ezt, ha rendelkezünk Google fiókkal, egyszerűen megtehetjük, hisz a Google szolgáltatások átjárhatók, így a regisztrációs lépést ki is hagyhatjuk. Az előző fejezet minta Flash animációját fogom feltölteni.

Az Adobe Flash Professionalban, ha meg van nyitva a projektünk, akkor válasszuk a File menü Export és azon belül az Export Movie... menüpontot. Ezúttal a fájl kiterjesztését avi-nak válasszuk, azaz Windows AVI formátumot. Ezután a mentésre kattintva megjelenik egy beállító ablak, itt a projektünknek megfelelő értékeket válasszunk, ahogy az alábbi képen is látszik.



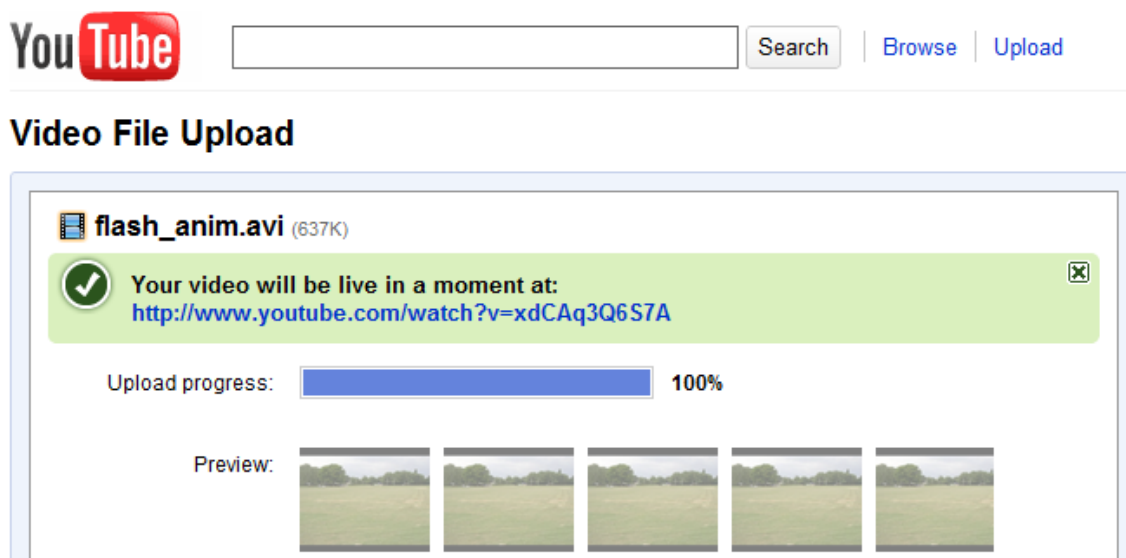
32. ábra<sup>54</sup>

Ezután még felugrik egy ablak, amiben a videó tömörítési kodeket tudjuk kiválasztani. Én a H.264-gyel kompatibilis tömörítőt választottam, hiszen a YouTube ezzel a formátummal jól együttműködik. Végezetül egy videó fájlt hoztunk létre, melyet egy közönséges lejátszó programmal is meg tudunk tekinteni. Természetesen a lejátszó gombunk az utolsó képkockán

<sup>54</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

értelmét veszti, hiszen egy közönséges videóról van szó, mely nem képes hordozni az ActionScript kódokat.

Következő lépésben a YouTube főoldalán rögtön a legfelső sorban közvetlenül a keresőmező jobb oldalán találjuk az Upload feliratot, melyre kattintva a feltöltő oldal nyílik meg, ahol kiválaszthatjuk, hogy feltöltünk egy fájlból videót, vagy a webkameránkról közvetlenül rögzítünk egyet, ugyanis a YouTube erre is ad lehetőséget. Nekünk most a videó felöltés opció kell, tehát az Upload video gombra kell kattintanunk. Egy fájlböngésző ablak nyílik meg, amelyben kiválaszthatjuk az előzőleg mentett videónkat. A YouTube felülete ekkor tudatja velünk, hogy megkezdte a feltöltést, majd feldolgozza azt (33. ábra).




33. ábra

A YouTube videó feltöltés után<sup>55</sup>

Ezután közvetlenül ezen a képernyőn megadhatunk alapvető beállításokat, úgymint sorban a videó nevét, egy rövid leírást róla, mely a látogatóknak szól, címkéket, melyek a keresést könnyítik meg, kategóriát, mely szintén ezt a funkciót tölti be, valamint a megosztási beállításokat, mely lehet nyilvános vagy meghívásos alapú (34. ábra).

<sup>55</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

**Video information and privacy settings** 

Title:

Description:

Tags:

Category:

Privacy:  Share your video with the world (Recommended)  
 Private (Viewable by you and up to 25 people)

**34. ábra**  
**A feltöltött videó alapbeállításai<sup>56</sup>**

Ezután ha a 33. ábrán is látható zöld keretben lévő linkre kattintunk, meg is tekinthetjük a videónkat (35. ábra).



**35. ábra**  
**A feltöltött videó lejátszás közben a YouTube-on<sup>57</sup>**

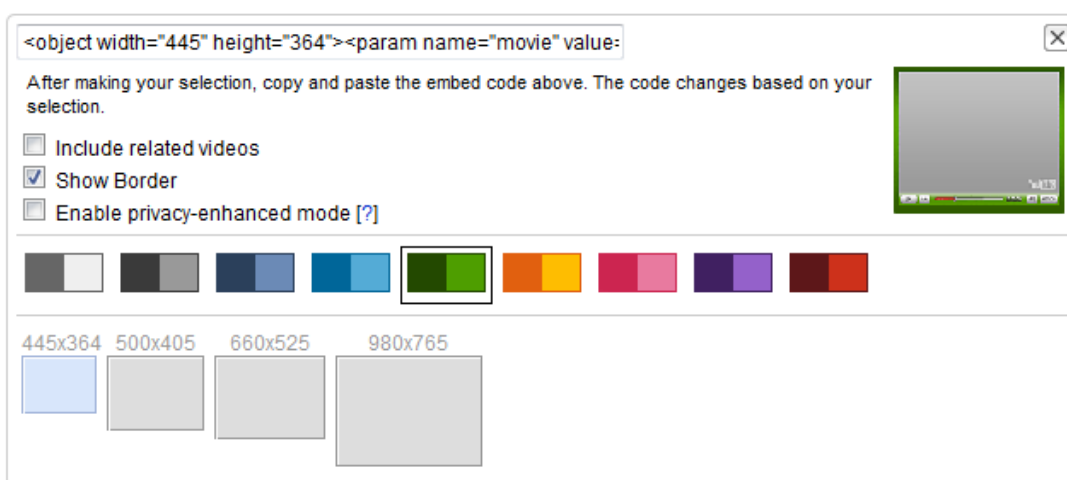
<sup>56</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

<sup>57</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

Természetesen ezek után részletesebb beállításokat is elérhetünk a videóhoz, ahol többek között megadhatjuk azt is, hogy be lehessen-e ágyazni weboldalakra a videót, a látogatók fűzhetnek-e hozzá megjegyzéseket, térképen geolokációs kódok segítségével megadhatjuk, hogy hol rögzítettük a videót, stb.

### 3.4.2. YouTube videó HTML dokumentumba való beágyazása

Ha a videónkat szeretnénk egy HTML dokumentumba beágyazni, akkor a YouTube ebben is a segítségünkre lesz. Ahogy a 35. ábrán is látszik, a videó lejátszó felület alatt jobbra található egy <Embed> feliratú gomb. Erre kattintva a 36. ábrán is látható keret jelenik meg az oldalon.



36. ábra

A beágyazást segítő beállító panel a YouTube lejátszási felületén<sup>58</sup>

Látható, hogy a beágyazó kódot a YouTube felülete előre generálja nekünk, így ezt kimásolva csak be kell illesztenünk a HTML dokumentumunkba. Megadhatunk néhány beállítást, úgymint a kapcsolódó videók megjelenhetnek-e a beágyazott videóban lejátszás után, adjon-e hozzá keretet, valamint ennek a színét, illetve a megjelenített videó lejátszó méretét is beállíthatjuk. Ezek mellett megadható egy biztonsági beállítás, az ún. privacy-enhanced mód, ami a beágyazott videók esetében a beágyazó oldalt megtekintve a YouTube lejátszója nem ment le a felhasználók azonosítására felhasználható sütiket<sup>59</sup>.

<sup>58</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

<sup>59</sup> Forrás: [www.google.com/support/youtube/bin/answer.py?answer=141046](http://www.google.com/support/youtube/bin/answer.py?answer=141046), letöltve: 2010. április 18.

A beágyazó kód a beállítások testreszabása után (13. kód):

```
1: <object width="445" height="364">
2:   <param name="movie"
3:     value="http://www.youtube.com/v/xdCAq3Q6S7A&hl=en_US&fs=1&
4:     rel=0&color1=0x234900&color2=0x4e9e00&border=1"></param>
5:   <param name="allowFullScreen" value="true"></param>
6:   <param name="allowscriptaccess" value="always"></param>
7:   <embed src="http://www.youtube.com/v/xdCAq3Q6S7A&
8:     hl=en_US&fs=1&rel=0&color1=0x234900&
9:     color2=0x4e9e00&border=1"
10:    type="application/x-shockwave-flash"
11:    allowscriptaccess="always"
12:    allowfullscreen="true" width="445" height="364">
13: </embed>
14: </object>
```

### 13. kód

#### A YouTube video beágyazó kódja

Beágyazva ezt a kódot a korábbi 3.3.3. fejezetben lévő HTML dokumentumba, az ott lévő videó helyére, egy böngészőben megnyitva azt, a 37. ábrán is látható eredményt kapjuk.



37. ábra<sup>60</sup>

## 4. Egyéb alkalmazási területek

Most, hogy bemutattam a főbb technológiákat és felhasználási lehetőségeiket, tömören megmutatnám, hogy milyen egyéb alkalmazási módok léteznek a internet segítségével a multimédiás tartalmak prezentálásában.

### 4.1. Prezentációk

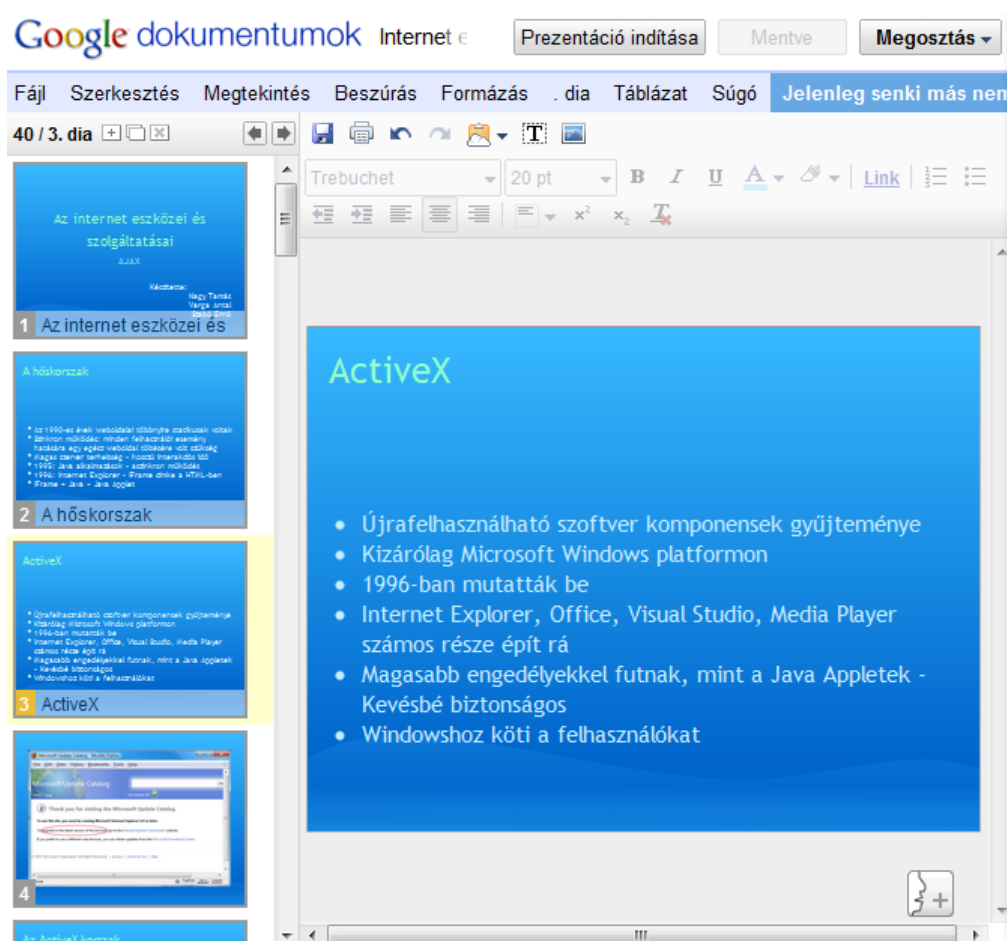
Az első kiemelendő csoport a prezentációkat érinti. A prezentáció fontos tartalomközvetítő eszköz, melyet első sorban személyes előadással kísérvé tehetünk igazán hatékonyá. A prezentáció készítő eszközök is egyre inkább kihasználják a internet és a rajta elterjedt technológiák nyújtotta lehetőségeket.

---

<sup>60</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 18.

#### 4.1.1. Google Dokumentumok

Szinte mindenki ismeri a Google Dokumentumok nevű szolgáltatását, mely a <https://docs.google.com/> címen érhető el. Online felületen szerkeszthetünk prezentációkat az offline (pl. Microsoft PowerPoint) megszokotthoz hasonló módon. Eszköztára viszonylag kezdetlegesnek tűnik (38. ábra), hisz pl. nem tudunk a különböző diák és elemek között áttűnéseket definiálni, de ha figyelembe vesszük, hogy pár éve még elképzelhetetlennek tartottuk a prezentációk hasonló módon történő szerkesztését, akkor láthatjuk a benne rejlő, előre mutató lehetőségeket.



38. ábra

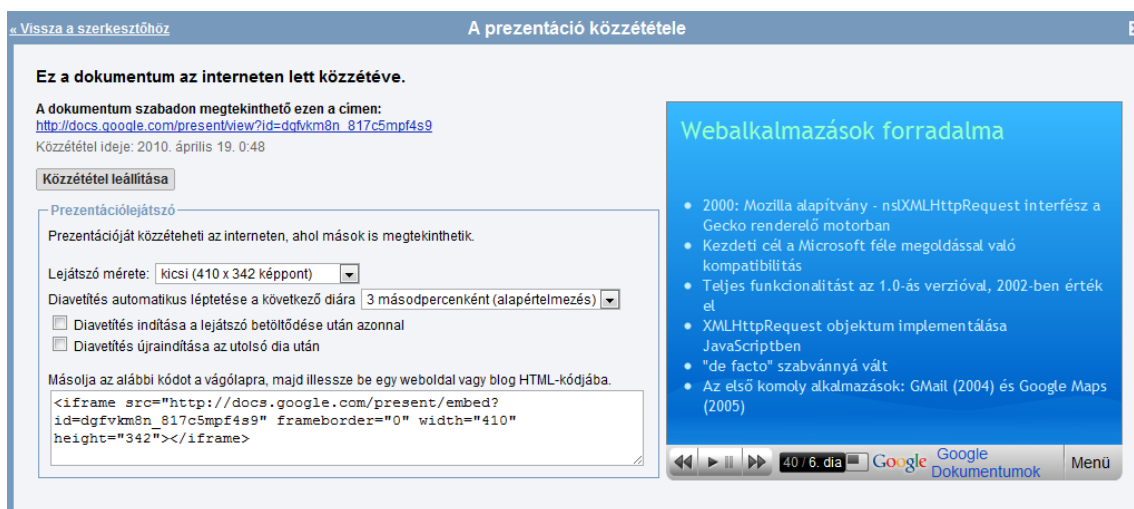
Egy prezentáció a Google Dokumentumokban<sup>61</sup>

A Google folyamatosan fejleszti, és ma már akár videókat, egyedi rajzokat, táblázatokat is beilleszthetünk. Emellett fő ereje a DHTML-en keresztüli kollaborációs

<sup>61</sup> Sajat készítésű képernyőkép részlet, készítve: 2010. április 19.

módban van, aminek köszönhetően egy időben több személy is szerkesztheti a dokumentumokat. A szerkesztés alatt álló elemek a többiek számára is azonnal láthatóak kiemelt módon. A dokumentumokat tetszőleges számú személlyel megoszthatjuk, valamint publikálhatjuk is egyúttal a nagyközönség felé.

Mindezekon felül a Google Dokumentumok szolgáltatás lehetőséget biztosít arra, hogy egy egyszerű HTML kód segítségével bármilyen weboldalba beágyazzuk azt. A 39. ábrán láthatjuk, hogy milyen módon is jelenik meg egy ilyen beágyazott prezentáció.



39. ábra

Egy prezentáció közzététele és  
HTML beágyazó kód készítése a Google Dokumentumokban<sup>62</sup>

#### 4.1.2. Prezi

A Prezi egy új, magyar fejlesztésű prezentációs eszköz. Elmondásuk szerint ők nem a divatot, a hagyományos eszközöket követik, hanem a kultúrák különbözőségéből merítenek ötleteket és ezt technológiai újításokban foganatosítják meg.

A <http://prezi.com> címen elérhető alkalmazást 2007-ben kezdte el fejleszteni Somlai-Fischer Ádám és Halácsy Péter annak kapcsán, hogy a hagyományos prezentációkészítő alkalmazásokat túl korlátolt eszköztárúnak találták, és ez gátat szabott kreativitásuknak. A Prezi hivatalosan 2009 áprilisa óta érhető el, de még ez év novemberében nyitottak egy irodát San Franciscóban is.

<sup>62</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 19.

A Prezi különlegessége abban rejlik, hogy pár egérgattintással látványosan animált, mégis reprezentatív előadásokat készíthetünk. Az animációk fiatalosak, lendületesek, mégsem zavaróak. A <http://prezi.com/learn/getting-started/> címen 6 egyszerű lépésben vázolják a szerkesztés alapvető lépéseit angol nyelven, melyet én is bemutatok röviden, valamint az oldalon található képeket is felhasználom szemléltetés végett.

1. Új Prezi létrehozásakor egy rajzterületet kapunk, melyen bárhova duplán kattintva szöveget vihetünk fel. A rajzterület mérete nem fix, ahogy nő a prezentációnk mérete, úgy növekszik az is.
2. Ahogy felvittünk a rajzterületre kellő mennyiségű szöveget, azokra ráközelítve fejleszthetjük azokat, hogy még jobban kifejezzék gondolatainkat.
3. Az ún. zebra segítségével változtathatunk az objektumok méretén és elhelyezkedésén, valamint elforgathatjuk azokat. A zebra egy több körből álló eszköz. A középső kör segítségével mozgathatunk, a középső körgyűrűvel méretezhetünk, a külső körgyűrűvel pedig elforgathatunk.
4. Prezentáció készítésekor bal felül elérhető egy menü, melyben a szerkesztéssel kapcsolatos műveleteket végezhetjük. Itt illeszthetünk be képeket is. Ha egy beillesztett képre kattintunk, akkor szintén megjelenik a zebra eszköz, amit a 3-as pontban ismertetett módon használhatunk.
5. Miután elhelyeztük az objektumainkat, a szerkesztő menün keresztül elérhető útvonal eszköz segítségével megadhatjuk a prezentáció lefolyásának sorrendjét. Ez egy dinamikus animációs átvezetést jelent egy egyik objektumról a másikra, ha szükséges nagyításokkal és kicsinyítésekkel együtt. És ez a



Prezi egyik nagy ereje. Nem szükséges új diákat felvenni, hivatkozni azok közt, ismételni önmagunkat, a Prezi a szükségtelen elemek elrejtését elvégzi helyettünk, ugyanakkor az adott objektumok között részleteket is elrejthetünk, melyek dinamikus közelítve a prezentáció során láthatóvá válnak ott, ahol szükség van rájuk.

6. Végezetül nem marad más dolgunk, mint megtekinteni a prezentációt, hogy kedvünkre való-e. A prezentációkat publikálhatjuk a Prezi oldalán keresztül is, kérhetünk hozzá HTML beágyazó kódot, és ekkor az előző alfejezetben ismertetett Google Dokumentumokhoz hasonló módon, a saját weboldalunkon is megjeleníthetjük.



A fejezet célja természetesen nem dokumentáció, vagy részletes bemutatás készítése volt a Preziről, hanem az olvasó figyelmének a felkeltése egy új, a hagyományosokhoz képest merőben más prezentációkészítő eszköz ismertetése.

## 4.2. Mobil technológiák

Említést érdemelnek a mobil eszközök, melyek rohamos fejlődésen és térhódításon mennek keresztül évről-évre. Ma már nem csak mobiltelefonok, de nagyobb méretű táblaszámítógépek is kaphatók, melyek az internetre csatlakozva pl. mobiltelefon hálózaton keresztül képesek nekünk bárhol, bármikor az internet nyújtotta multimédiás előnyöket közvetíteni.

A leglátványosabb fejlődésen talán a mobiltelefonok mennek át, melyek 10 év alatt az egyszerű SMS képes, kétszínű kijelzővel és egyszólamú csengőhangokkal felvértezett készülékekből igazi mobil multimédiás szórakoztató eszközzé nőttek ki magukat. Ahogy régen tényleg csak kapcsolattartásra használtuk őket, a szociális igények fejlődésével együtt szükségessé vált olyan eszközök megjelenése is, melyen útközben akár zenét hallgathatunk, filmet nézhetünk, internetezhetünk, fényképezhetünk vagy videó felvételeket készíthetünk. Mára mindezt a mobiltelefonok nyújtják a fejlett társadalmak számára.

A korábban részletesen tárgyalt Adobe Flash platformja itt is meghatározó tényező az internetes tartalmak közvetítésében. Pontosabban annak egy egyszerűsített, a korlátozottabb eszközháttérre optimalizált, ún. Flash Lite verziója. Egy 2008-as felmérés szerint több mint 800 millió Flash Lite képes eszköz volt forgalomban, és becslések szerint 2010 végére ez a szám eléri a 2,5 milliárdot is.

Ez az egyszerűsített platform lehetővé teszi az eszközgyártók számára, hogy egyénileg testreszabott felületen olyan internetes tartalmakat jeleníthessenek meg, mint a YouTube videók, játékok, űrlapok, internet alapú TV adások, stb. A Flash Lite támogatja a H.264 kodeket is, így internet alapú TV-ken akár nagyfelbontású műsorokat is nézhetünk. A fejlesztők számára integrált környezetet és automatizált tesztrendszert is biztosítanak, mellyel akár 85%-kal is felgyorsíthatják a fejlesztési időt. A Flash Lite segítségével létrehozhatunk olyan dinamikus alkalmazásokat is, melyet a III.3.b fejezetben mutattam. Ezenfelül képes alkalmazás-specifikus beállítások mentésére is, ezáltal a felhasználó számára állapotmentési lehetőséget is nyújthatnak a fejlesztők.



40. ábra

Kép forrása:

[www.knowyourmobile.com/siteimage/scale/800/600/16390.png](http://www.knowyourmobile.com/siteimage/scale/800/600/16390.png)

letöltve 2010. április 19.

### 4.3. Az IPTV és a Set-top Boxok

Az IPTV az Internet Protocol Television rövidítése, és olyan rendszereket jelöl, melyek képesek fogadni és megjeleníteni valamilyen kódolási eljárással tömörített videó adatfolyamokat interneten keresztül. Az IPTV fő eszközei a hagyományos kialakítású TV készülékek mellett a Set-top Boxok. Ezeket csatlakoztathatjuk az internetes hálózatra (jellemzően kábeles megoldással), és ezen keresztül akár a jó öreg katódsugárcsöves televíziókon is élvezhetjük az IPTV nyújtotta előnyöket.

A Set-top Boxok elvégzik az adatfolyamok kódolását és megjelenítését. Mint azt nyilván sejti az olvasó, ezt a funkciót egy egyszerű felépítésű számítógéppel is el lehet látni, de ezek mellett a készülékek mellett az alacsonyabb fogyasztás, zaj, fenntartási költség és

természetesen ár is szól. Természetesen ma már kaphatók ilyen felhasználási területre szánt személyi számítógépek, ún. media centerek, melyek kizárólag arra vannak tervezve, hogy TV-re és megfelelő hangrendszerre kötve az általános házi szórakozási igényeket kiszolgálja. Mindemellett megjelentek fejlett IPTV képes TV készülékek is, így ezekhez Set-top Boxra sincs szükség.

Az IPTV képes készülékek többnyire hardveresen tartalmazzák a videó adatfolyamok, tehát leggyakrabban MPEG-2 és H.264 kódolásához szükséges eszközöket. Ezek a készülékek képesek a hálózat QoS (Quality of Service) szolgáltatását kihasználni, ami a hálózati adatforgalom prioritizálását végzi, így a videónk megfelelő adatátviteli sebesség mellett a szemlélő számára folyamatosan fog érkezni a hálózaton keresztül, nem lesznek késések és kimaradások. Ennek ellenére a technológia magában hordozza a magasabb hibalehetőségeket is, hiszen bármilyen fennakadás a hálózati kapcsolatban akár a teljes közvetítés kimaradását, vagy akár a megszűnését vonhatja maga után.



**41. ábra**

Kép forrása:

[http://www.sg.hu/kep/2007\\_05/0513ip1.jpg](http://www.sg.hu/kep/2007_05/0513ip1.jpg),

letöltve 2010. április 19.

Természetesen, ha nagyfelbontású, tehát 1920x1080 képpont felbontású full HD videót szeretnénk nézni, a nagy adatméret miatt legalább 20-25 Mbps sebességű internetkapcsolat szükséges.

Mivel az IPTV képes eszközök tulajdonképpen egyszerű internet képes készülékek, így a hagyományos értelemben vett csatornakorlátok elméletben eltűnnek. Tehát bármikor, bármennyi tartalomközvetítő adását nézhetjük.

Gyakorlatilag az internetszolgáltatónk, mint közvetítő is korlátozó jelleggel léphet fel, illetve a készülék gyártójától is sok függ, hiszen ő teszi alkalmassá az adatfeldolgozásra azt. Például dönthet úgy, hogy kihagyja a YouTube támogatást.

Az IPTV képes készülékek alkalmasak a PIP képmegjelenítésre, azaz a Picture-In-Picture – Kép a képben megjelenítésre. Természetesen ez kettő vagy több egyidejű videó adatfolyam töltését igényli, ami különösen terhelő lehet a hálózatnak. Egy full HD felbontású

videó H.264-gyel kódolva általában 7-8 Mbps-os bitrátával van kódolva, amihez még jön egy vagy több hangsáv. Látható, hogy két ilyen adatfolyammal akár a teljes sávszélességünket lefedhetjük. Emellett ha lassulás található a hálózaton, tehát a kapcsolat nem stabil, akkor ezt a nagyon hasznos funkciót már ki sem tudjuk használni ekkora felbontás esetén.

Felhasználói részről előnyök közé tartozik többek között még az is, hogy sokkal nagyobb interaktivitást biztosítanak, mint a hagyományos műsorszórási formák. Megfelelően programozott eszközökkel akár az emailjeinket is elolvashatjuk, válaszolhatunk is rá, riasztást állíthatunk be kedvenc műsorunkhoz, sőt, igény szerint nézhetjük meg őket, azaz nem vagyunk kötve fix műsoridőhöz.

Mindezekon felül olyan szolgáltatók is megjelentek a piacon, mint a világ egyik legnagyobb azonnali üzenetküldő szolgáltatása, a Skype. A Skype ez évtől, vagyis 2010-től H.264 kódolással akár HD minőségben is lehetővé teszi videó hívások, sőt konferenciák lebonyolítását televízió készülékeken keresztül<sup>63</sup>. Nincs másra szükségünk, mint egy IPTV képes, Skype szolgáltatással ellátott TV-re és egy webkamerára.

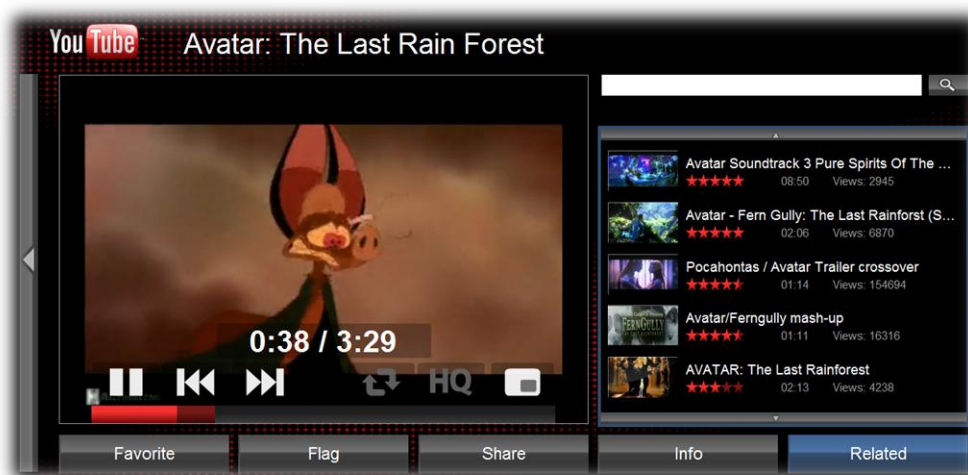


**42. ábra**  
**Skype konferencia IPTV-n<sup>64</sup>**

<sup>63</sup> Forrás: <http://www.skype.com/allfeatures/tv>, letöltve: 2010. április 19.

<sup>64</sup> Kép forrása: [stuff.tv/csfiles/blogs/usa/WindowsLiveWriter/StuffmakesworldsfirstSkypecallonTV\\_A32B/LG\\_SkypeTV3\\_2.jpg](http://stuff.tv/csfiles/blogs/usa/WindowsLiveWriter/StuffmakesworldsfirstSkypecallonTV_A32B/LG_SkypeTV3_2.jpg), letöltve: 2010. április 19.

Kaphatók olyan TV készülékek és Set-top Boxok is, melyek egy korlátozott eszköztáru webböngészőt is tartalmaznak. Ezen keresztül akár a YouTube ún. XL<sup>65</sup> felületét is használhatjuk videó megtekintésre.



43. ábra  
YouTube XL felület lejátszás közben<sup>66</sup>

Ezen a felületen TV készülékre optimalizált felületen, nagyméretű nyomógombok segítségével navigálhatunk, melyek között a távirányítóval is könnyen lépkedhetünk.

<sup>65</sup> Cím: <http://www.youtube.com/xl>

<sup>66</sup> Saját készítésű képernyőkép részlet, készítve: 2010. április 19.

## 5. Összefoglaló

Célkitűzésem alapján szerettem volna minél részletesebb képet adni az internet és a WWW nyújtotta multimédiás lehetőségekről. Ez részben sikerült, hisz a domináns technológiákat bemutattam, képességeiket felvázoltam. Ugyanakkor a téma jellegéből és méretéből adódóan nehéz mindent írásos formában prezentálni.

Igyekeztem mindenhol az adott témát legjobban prezentáló weboldalak elérhetőségét feltüntetni. Mindemellett a dolgozat írása közben áthelyeztem a hangsúlyt a részletességről a prezentálásra, azaz egy átfogó képet igyekeztem bemutatni az alkalmazásokról, mellyel felkelthetem az olvasó érdeklődését, miután az maga is könnyedén, akár a hivatkozott címek segítségével részletesebben utánajárhat a dolgoknak.

Ezt figyelembe véve a dolgozatom nem a szűk szakmai köröknek szól, hisz nem is vázoltam fel mélyreható esettanulmányokat, valamint alkalmazástervezési koncepciókat. Ahol viszont szükséges volt egy kissé részletesebb, szakmai leírást adnom, természetesen megtettem. Ilyen részek pl. a HTML nyelvek ismertetése illetve az Adobe Flash bemutatása, ezek a részek a dolgozat terjedelméhez mérten is a teljesség igénye nélkül készültek, hisz ezen technológiák referencia dokumentációja önmagában is tekintélyes méretű.

Végezetül pedig őszintén remélem, hogy sikerült felkeltenem az olvasó érdeklődését, és egy kis elszántsággal maga is utánajár az őt érdeklő területeknek.

## 6. Irodalomjegyzék

### Nyomtatott források

1. Adobe Flash CS4 Professional – Tanfolyam a könyvben, kiadó: Perfact-Pro Kft., kiadás éve: 2009, szerzők: Adobe Creative Team, ISBN: 9789639929036.
2. ActionScript 3.0 a gyakorlatban, kiadó: Pult Kft., kiadás éve: 2008, szerző: Colin Moock, ISBN: 9789639637474.

### Internetes források

3. Anderson, N. (2006. március 12). *Ars Technica*. Letöltés dátuma: 2010. április 19, forrás: An introduction to IPTV:  
<http://arstechnica.com/business/news/2006/03/iptv.ars>
4. Clark, D. D., Cerf, V. G., Barry, L. M., Kahn, R. E., Kleinrock, L., Lynch, D. C., és mtsai. (2010). *Internet Society (ISOC)*. Letöltés dátuma: 2010. április 7, forrás: All About The Internet: History of the Internet:  
<http://www.isoc.org/internet/history/brief.shtml>
5. Grossman, G. (2006. június 27). *Adobe Developer Connection*. Letöltés dátuma: 2010. április 18, forrás: ActionScript 3.0 overview | Adobe Developer Connection:  
[http://www.adobe.com/devnet/actionscript/articles/actionscript3\\_overview.html](http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html)
6. Group, W. H. (2002. augusztus 1). *W3C Recommendation*. Letöltés dátuma: 2010. április 10, forrás: XHTML 1.0: The Extensible HyperText Markup Language (Second Edition): <http://www.w3.org/TR/xhtml1/>
7. Hickson, I., & Hyatt, D. (2010. március 4). *W3C Working Draft*. Letöltés dátuma: 2010. április 12, forrás: HTML5 - A vocabulary and associated APIs for HTML and XHTML: <http://www.w3.org/TR/html5/>

8. Inc., A. (2010). *Adobe Flash Platform*. Letöltés dátuma: 2010. április 18, forrás: Adobe Flash Platform - Working with objects:  
[http://help.adobe.com/en\\_US/as3/learn/WS5b3ccc516d4fbf351e63e3d118a9b90204-7fd9.html](http://help.adobe.com/en_US/as3/learn/WS5b3ccc516d4fbf351e63e3d118a9b90204-7fd9.html)
9. Inc., A. (2010). *Adobe Systems Incorporated*. Letöltés dátuma: 2010. április 19, forrás: Adobe - Flash Lite: <http://www.adobe.com/products/flashlite/>
10. Inc., A. C. (2006. október 17). *Apple Inc.* Letöltés dátuma: 2010. április 15, forrás: H.264 Technology Brief:  
[http://images.apple.com/quicktime/pdf/H264\\_Technology\\_Brief.pdf](http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf)
11. Inc., G. (2010). *Mindent a Google-ról*. Letöltés dátuma: 2010. április 18, forrás: Mindent a Google-ról: <http://www.google.hu/intl/hu/about.html>
12. Incorporated, A. S. (2010. február). *About Adobe*. Letöltés dátuma: 2010. április 16, forrás: Adobe Systems Incorporated:  
<http://www.adobe.com/aboutadobe/pressroom/pdfs/profile.pdf>
13. Iroda, W. M. (4. február 2010). *W3C Magyar Iroda*. Letöltés dátuma: 2010. április 9, forrás: W3C 7 pontban - W3C Magyar Iroda:  
<http://www.w3c.hu/forditasok/7pont.html>
14. LA, M. (2009-2010). *MPEG LA - The Standard for Standards*. Letöltés dátuma: 2010. április 15, forrás: MPEG LA - The Standard for Standards - AVC Introduction: <http://www.mpegla.com/main/programs/AVC/Pages/Intro.aspx>
15. Lilley, C., & Schepers, D. (2009. december 11). *W3C*. Letöltés dátuma: 2010. április 13, forrás: Scalable Vector Graphics (SVG):  
<http://www.w3.org/Graphics/SVG/>
16. OECD. (2008. június 5). *OECD*. Letöltés dátuma: 2010. április 8, forrás: Organisation for Economic Co-operation and Development:  
<http://www.oecd.org/dataoecd/20/41/40789235.pdf>

17. Opendir.hu. (2007-2010). *Opendir.hu*. Letöltés dátuma: 2010. április 27, forrás: JQuery alapok 1 / JQuery alapozó:  
[http://opendir.hu/?freedom=/tartalom/jquery/jquery\\_alapozo](http://opendir.hu/?freedom=/tartalom/jquery/jquery_alapozo)
18. Raggett, D., Hors, A. L., & Jacobs, I. (24. december 1999). *W3C Recommendation*.  
Letöltés dátuma: 2010. április 9, forrás: HTML 4.01 Specification:  
<http://www.w3.org/TR/html401/>
19. W3Schools. (1999-2010). *W3Schools*. Letöltés dátuma: 2010. április 18, forrás:  
Flash in HTML: [http://www.w3schools.com/flash/flash\\_inhtml.asp](http://www.w3schools.com/flash/flash_inhtml.asp)
20. W3Schools. (1999-2010). *W3Schools Online Web Tutorials*. Letöltés dátuma: 2010. április 10, forrás: DHTML Tutorial: <http://www.w3schools.com/dhtml/default.asp>
21. Wikipedia. (2010. április 24). *Wikipedia, the free encyclopedia*. Letöltés dátuma: 2010. április 27, forrás: MooTools: <http://en.wikipedia.org/wiki/MooTools>
22. Wikipedia. (2010. április 21). *Wikipedia, the free encyclopedia*. Letöltés dátuma: 2010. április 27, forrás: Dojo Toolkit: [http://en.wikipedia.org/wiki/Dojo\\_Toolkit](http://en.wikipedia.org/wiki/Dojo_Toolkit)
23. Wikipedia. (2010. február 28). *Wikipedia, the free encyclopedia*. Letöltés dátuma: 2010. április 28, forrás: OpenLaszlo: <http://en.wikipedia.org/wiki/OpenLaszlo>
24. YouTube. (2010). *About YouTube*. Letöltés dátuma: 2010. április 18, forrás: YouTube - Broadcast Yourself.: <http://www.youtube.com/t/about>