

# Szórakoztató elektronikai eszközök programozása

---

Kelényi Imre

[imre.kelenyi@aut.bme.hu](mailto:imre.kelenyi@aut.bme.hu)

# Tartalom

---

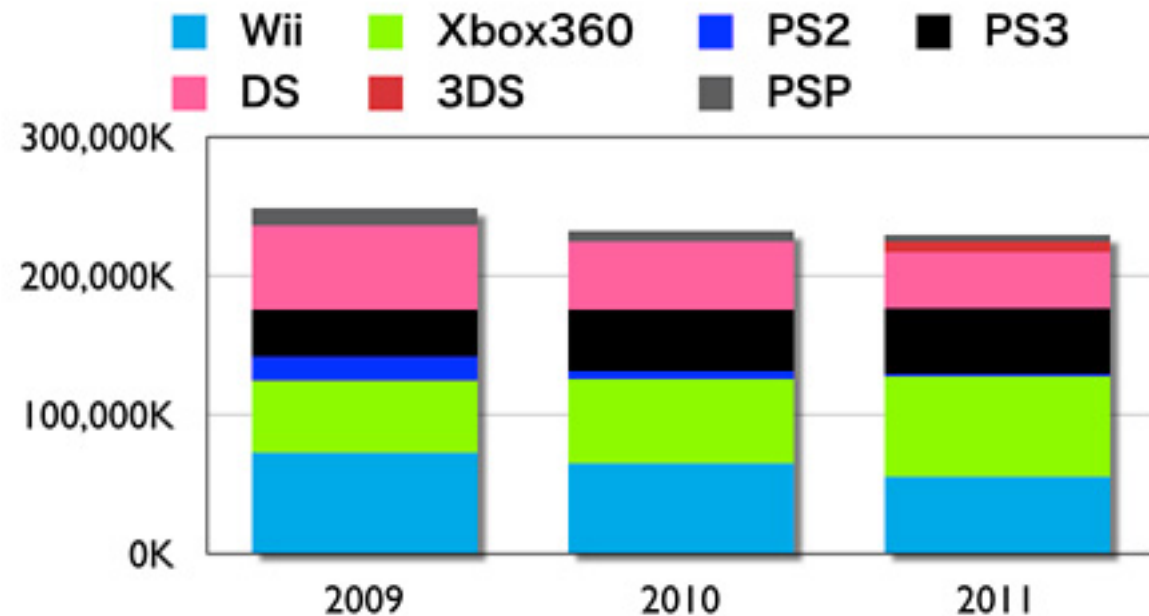
- Játékkonzolok
  - Playstation 3 és 4
  - Wii és Wii U
  - Xbox 360
- XNA
- Kinect

Szórakoztatóelektronikai eszközök programozása

# SZOFTVERFEJLESZTÉSS JÁTÉKKONZOLOKRA

## 7. generációs játékkonzol platformok

- Sony Playstation 3
- Microsoft Xbox 360
- Nintendo Wii



Eladott konzolok (USA)

Szórakoztatóelektronikai eszközök programozása

# PLAYSTATION 3

# Playstation 3

---

- 2006-ban jelent meg Japánban és Észak Amerikában
- Játékkonzol és média lejátszó
  - Eleinte sokan csak Blu-ray lejátszó miatt vették
- PlayStation Network (PSN):  
hálózat online játékhöz
- PlayStation Store:  
online játék/média bolt
- Részleges PlayStation 1-2  
kompatibilitás
- PlayTV: DVB-T Tuner és felvevő



# PS3 hardver

---

- Cell CPU
  - Sony, Toshiba, IBM közös fejlesztése
  - Több mag: 1 Power Processor Element (PPE), 8 Synergistic Processor Elements (SPEs)
  - Elméletben brutális teljesítmény
  - Nehezebb rá fejleszteni
- 20-320GB HDD
- Gigabit Ethernet, 802.11 b/g Wi-fi, USB 2.0, Bluetooth 2.0
- HD felbontás (720p és 1080p)
- Blu-ray lejátszó
  - Filmlejátszás

# Playstation Move + Eye

## ➤ PlayStation Move

- Sony válasza a Wii-re
- PlayStation Eye kamerát használja a játékos távolságának belövéséhez
  - A világító labda dinamikusan változtatja a színét, hogy kitűnjön a környezetéből
- Gyorsulásérzékelő, elfordulás mérő (rate sensor), magnetométer



## ➤ PlayStation Eye

- Kamera: 640x480 60 Hz vagy 320x240 120 Hz
- 4 tagú mikrofon tömb





# Játékfejlesztés PS3-ra

- Zárt rendszer, fejlesztés csak Developer Kit vásárlása után

- 2000\$

- PhyreEngine

- Sony által biztosított játékmotor
  - Támogatja PC-t (Direct3D) és PS3-at (OpenGL) valamint kézikonzolok közül PSP-t és PS VITA-t
  - Régóta ígéretik, hogy lesz open source, ingyenes változat, de még mindig nem elérhető

- Alternatív megoldás: jailbreak + homebrew



# PS3 és Linux

---

- A Sony eredetileg lehetővé tette a Linux futtatását PS3-on
  - OtherOS feature: GNU/Linux vagy FreeBSD futtatása
  - Sok disztribúciót kiadtak PS3-ra: Ubuntu, Fedora, OpenSUSE
- 3.21-es verzióban megszűnt a támogatás
  - Aki nem frissít, az tovább használhatja, de online játék és PlayStation Store nem lesznek elérhetőek

# PS3 Jailbreak

---

- Csak 2010 augusztusától
  - George Hotz (Geohot) törte fel (mint iPhone-t)
  - Voltak korábbi próbálkozások, de általánosan csak azóta
- Megvásárolható USB stick
  - Open-source változatok is kijöttek, ahol bármilyen saját USB eszköz használható
- Csak 3.55-ös vagy korábbi verzió, az újabb verziókkal nem!
  - Folyamatos küzdelem az új firmware-ekkel, részmegoldások vannak
- Homebrew programok, „játék backupok”
  - PSN elérés megszűnik



# PS3 Homebrew

---

## ➤ Két opció

### ■ Homebrew SDK : PSL1GHT

- <http://psl1ght.net>
- C/C++
- Fejlesztés Linux vagy Windows (Cygwin) alatt

### ■ Kiszivárgott hivatalos SDK

## ➤ Alkalmazások

- Fájlkezelők, FTP kliens
- Játékok
- Emulátorok (Gameboy, NES, SNES, Sega Saturn...)
- ...

# PlayStation 4

---

- A Sony 8. generációs konzolja
  - Megjelenés: 2013 karácsony
- Hardver
  - AMD „accelerated processing unit” (APU)
    - X86-64 CPU (8 magos, AMD)
    - AMD GPU
  - 8 GB RAM
- Mozcásérzékelős, érintésérzékeny kontroller
- Új PlayStation Eye, konzolhoz csomagolják, Kinect-szintű funkciók
- Szakítás a Cell architektúrával
  - Gyakorlatilag egy PC
  - Könnyebb fejlesztés, egyszerűbb portolás, nagyobb kompatibilitás meglévő eszközökkel/technológiákkal



Szórakoztatóelektronikai eszközök programozása

**WII (+ WII U)**

# Wii

---

- A Nintendo játékkonzola
- Mozcásvezérelt irányítás
- Opera böngésző („Internet Channel”)
- WiiConnect24:  
Wi-fi kapcsolat standby módban  
(pl. üzenetek, email-ek fogadása)
- Virtual Console: emulátor és  
játékbolt korábbi Nintendo platform  
játékokhoz
- GameCube kompatibilitás



# Wii controller

---

## ➤ Az alap csomagban két eszköz

- Wii Remote (WiiMote): infra + gyorsulásérzékelő
- Nunchuk: gyorsulásérzékelő

## ➤ Wii MotionPlus

- Kiegészítés Wii Remote-hoz
- Növeli a pontosságot
- Két tengelyes giroszkóp





# A controller árnyoldala



➤ A kézi szíj egyik szériáját visszahívták...

# Wii hardver

---

- PowerPC alapú, 90 nm, 730 MHz
- 88 MB RAM
- 512 MB flash
  - Bővíthető kártyaolvasón keresztül SD/SDHC-vel
- DVD olvasó
- Ethernet, 802.11b/g Wi-Fi, USB 2.0
- Csak SD felbontásokat támogat (480p, 576i)

# Játékfejlesztés Wii-re

---

- Hivatalosan zárt rendszer, fejlesztés csak developer kit révén
  - 2000\$-10000\$
  - Igazolni kell, hogy játékfejlesztéssel foglalkozó cégünk van
- Alternatív megoldás: homebrew



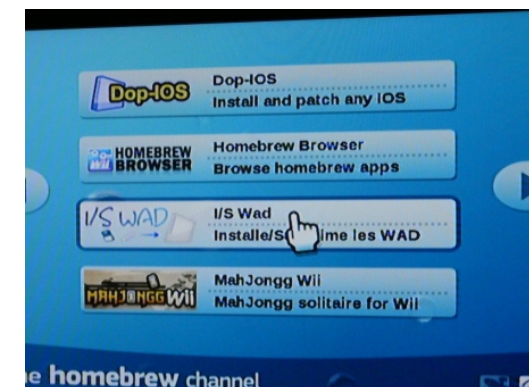
# Wii „jailbreak”

---

- Cél a HackMii installer lefuttatása a konzolon
- Jelenleg minden Wii verzió jailbreak-elhető
  - Több különféle megoldás
    - Savegame alapú
      - Egy adott játék módosított savegame-e futtatja a törést
      - Pl. Indiana Pwns, Return of the Jodi
    - System menu alapú
      - Játékszoftver nélkül működik
- Jailbreak után
  - Homebrew Channel
  - „Biztonsági mentések” futtatása

# Homebrew Channel

- Egy új csatorna (alkalmazás) a Wii rendszermenüjében
- Kilistázza a háttértáron/ memóriakártyán lévő homebrew alkalmazásokat
- Homebrew Browser: online alkalmazás-tár, netről közvetlenül Wii memóriakártyára tölthetjük le az alkalmazásokat



- Saját alkalmazásokat is felrakhatok, jelentkezés:  
[http://wiibrew.org/wiki/Talk:Homebrew\\_Browser#Wanted\\_Apps/](http://wiibrew.org/wiki/Talk:Homebrew_Browser#Wanted_Apps/)

# Wii homebrew SDK

---

## ➤ devkitPro

- Wii, Gamecube, DS, PSP homebrew SDK
- C/C++ API-k
- <http://devkitpro.org/>

## ➤ Debuggolás hálózaton keresztül vagy USB-n (USBGecko)

## ➤ Tutorialok:

<http://www.codemii.com/category/wii-programming-tutorials/>

# Wii homebrew példakód

---

```
#include <wiiuse/wpad.h>

int main() {
    Initialise(); // Wii inicializáló
    printf("Hello World!\n");

    while(1) {
        WPAD_ScanPads();

        u16 buttonsDown = WPAD_ButtonsDown(0);
        if (buttonsDown & WPAD_BUTTON_A ) {
            printf("Button A pressed.\n");
        }
        if (buttonsDown & WPAD_BUTTON_HOME) {
            exit(0);
        }

        return 0;
    }
}
```

# Wii U

- Az első 8. generációs konzol
- Speciális „táblagép” controller (Wii U Gamepad)
  - Érintésérzékeny, NFC, kamera, gyroszkóp, gyorsulásérzékelő
  - Extra kijelző
  - Játékok TV nélkül is játszhatók
  - Nem kötelező, hagyományos controller is használható
- 1080p felbontású játékok
- Nintendo egyik célja a „hardcore” játékosbázis visszaszerzése

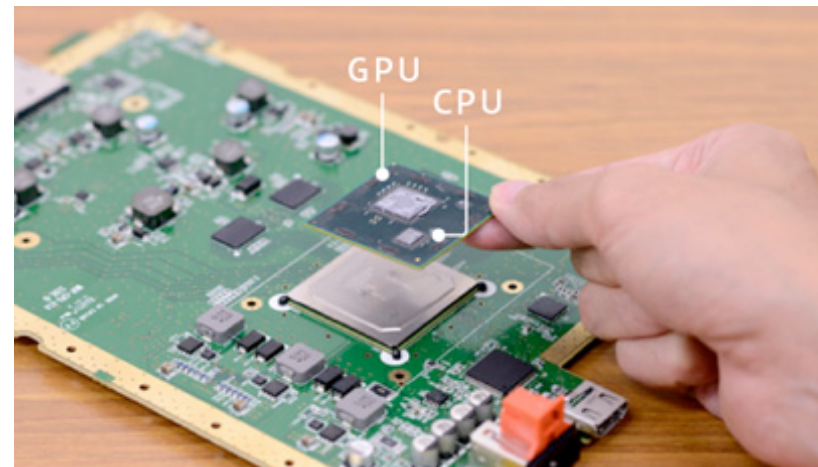




# Wii U hardver

---

- Inkább Xbox 360 és PS3 szintű...
- Három magos, 3 GHz-es PowerPC CPU a játékoknak
- Egy külön ARM CPU az OS-nek
- 2 GB RAM
  - 1 GB játékoknak
  - 1 GB OS-nek
- 8-32 GB háttértár
- Saját (Blu-ray-re épülő) lemezformátum (25 GB / lemez)



# Fejlesztés Wii U-ra

---

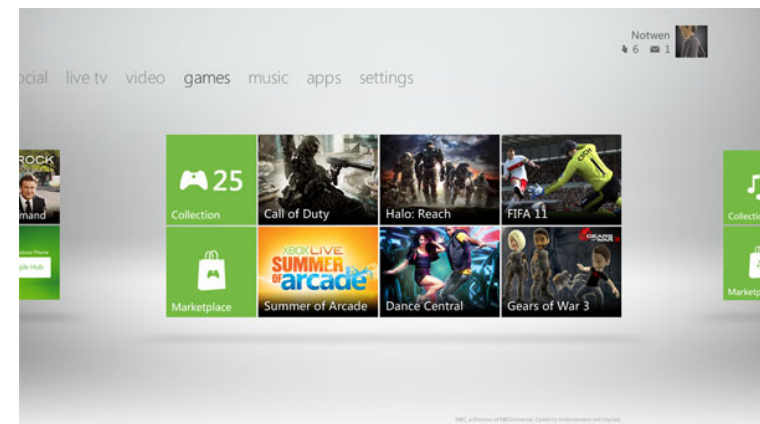
- Hivatalos út: fizetős devkit
- C++ API-k
- Megjelenés után nem sokkal feltörték: Homebre Channel telepítése
  - A Nintendo bezárta a biztonsági réseket

Szórakoztatóelektronikai eszközök programozása

# XBOX 360

# Xbox 360

- Microsoft 7. generációs játékkonzola
- 2005-ben jelent meg
- Médialejátszó funkciók
- Xbox Live: online játék
- Xbox Live Marketplace: online játékbolt
- Video store
- Kinect: teljes test mozgásérzékelés



# Xbox 360 hardver

---

- 3.2 GHz PowerPC Xenon
  - 3 mag, Cell PPE módosított változatai, IBM fejlesztette
- 512 MB RAM
- 802.11 b/g/n Wi-fi (Xbox 360 Slim)
- HD felbontás támogatása (720p, 1080p)
  - Legtöbb játék csak 720p-ben fut
- DVD olvasó
  - Xbox játékok dupla rétegű (DL) DVD-ken
- HDD (opcionális)

# Xbox Live más platformokon

## ➤ Windows Phone

- Elérhetjük a Live fiókunkat és avatarunkat
- Közös achievement rendszer
- Social network



## ➤ Android és iOS: SmartGlass app

## ➤ Windows 8



# SmartGlass

- Mobilalkalmazás (+ Windows 8/RT)

- Windows Phone, iOS, Android

- Távirányító Xbox-hoz

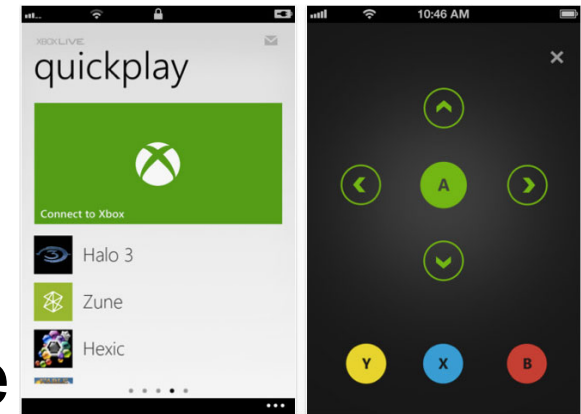
- Xbox Live áttekintés

- Játékok felületének kiterjesztése

- Pl. térképet a tableten mutatja
- Jelenleg kevés játék támogatja

- Információk a lejátszott médiatartalomról

- Pl. lejátszott film adatai



# Xbox 360 jailbreak

## ► DVD firmware hack

- Lehetővé teszi másolt játékok futtatását
- Homebrew nem futtatható
- Szoftveresen, a DVD olvasó firmware-jét kell átírni (pl. PC-re SATA kábellel rákötve)

## ► JTAG hack

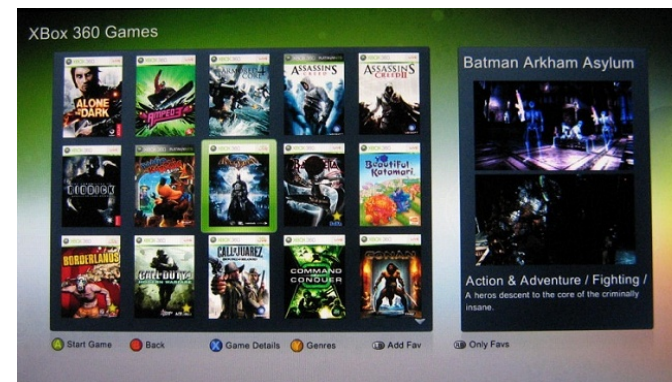
- Lehetővé teszi a teljes hozzáférést a rendszerhez
- Hardveres beavatkozást vagy egy külső USB-s eszközt (xKey) igényel
- Csak 2009 augusztusi vagy régebbi Dashboard verzió esetén ☹



# Xbox 360 homebrew

---

- A JTAG hekkelések megjelenéséig nem volt, mostanában kezdenek elterjedni
  - További infók, alkalmazások:  
<http://dev360.wikia.com>
- Főbb projektek
  - Alternatív rendszermenük (dashboard)
  - Emulátorok



# Játékfejlesztés Xbox 360-ra

---

- Xbox 360 Registered Developers Program
  - Fizetős, külön kell egyeztetni Microsofttal
  - Development kit
- Xbox Live Arcade
  - Az online store-ban árult, kisebb játékokhoz
  - Fizetős, külön kell egyeztetni Microsofttal
- Xbox Live Indie Games
  - Windows Phone és „indie” Xbox játékok fejlesztéséhez, XNA-val
  - 99\$ / év
  - Magyar Xbox Live-on még (mindig) nem elérhető ☹
  - <http://xbox.create.msdn.com>

# XNA fejlesztés megkezdése

---

## ➤ Kell

- Xbox 360
- XBOX Live Indie Games regisztráció (99\$/év)
- SDK: XNA Game Studio (IDE: Visual Studio)
- Xbox Live-ről le kell tölteni az „XNA Game Studio Connect nevű alkalmazást”

## ➤ Game Studio összekapcsolódik Xbox-al

## ➤ Mehet a deploy



Szórakoztatóelektronikai eszközök programozása

**XNA**

# XNA

---

- XNA is Not an Acronym
- Gazdag programozói keretrendszer és eszközcsoomag elsősorban játékfejlesztéshez
- Render loop modell, 2D és 3D támogatás (shaderek is)
- Támogatott platformok:
  - Xbox 360
  - Windows (XP, Vista, 7, 8 desktop)
  - Zune
  - Windows Phone 7.x



# XNA Windows 8, Windows Phone 8

---

## ➤ Windows 8

- Alapeset: XNA csak desktop alkalmazásban (nem Metro), VS 2010 (VS 2012-ben nincs XNA)
- Megoldás: MonoGame, platformfüggetlen XNA kompatibilis library, VS 2012 kompatibilis, Modern UI alkalmazásokban is használható
  - <http://www.wpcentral.com/xna-dead-long-live-xna>

## ➤ Windows Phone 8

- WP 7.x-et megcélzó XNA játékok működnek WP 8-on is
  - WP 8 API-k nem használhatók
- Ha WP 8 a target, akkor nincs XNA, megoldás a MonoGame!

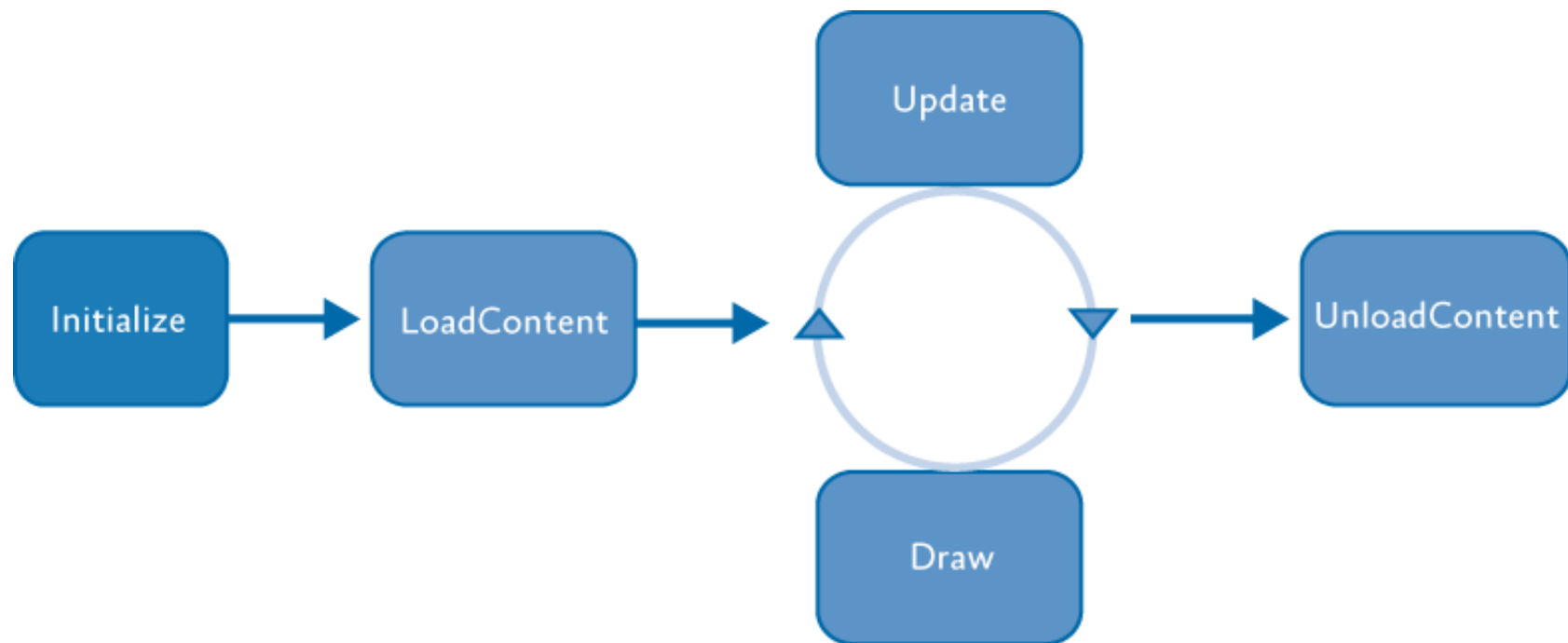
# XNA játék felépítése

---

- Microsoft.Xna.Framework.**Game** osztályból kell leszármaztatni
- Felüldefiniálendő metódusok
  - void Initialize()
  - void LoadContent()
  - void UnloadContent()
  - void Update(GameTime gameTime)
    - Periodikusan hívódik, modellt frissíthetjük, feldolgozhatjuk az „idő múlását”, itt valósítjuk meg a játéklogikát
  - void Draw(GameTime gameTime)
    - Periodikusan hívódik, mikor rajzolni kell valamit

# Game Loop

---





# Update és Draw

---

- Alapesetben fixed-step loop: a rendszer arra törekszik, hogy Update() lehetőleg minden másodpercben ugyanannyiszor hívódjon meg (pl. 60-szor)
  - Update() és Draw() hívása felváltva történik
  - Ha Update() nem fut le elég sokszor, akkor kihagy Draw() hívásokat
- A játék “pause” módba kerül, akkor nem hívódik Update()

# 2D rajzolás XNA-ban

---

## ➤ Sprite: bitkép

- A megjelenítendő képet Texture2D osztály egy példánya tárolja

```
Texture2D spriteTexture;
```

- Kirajzolás: SpriteBatch osztályon keresztül

- Begin(): SpriteBatch inicializálás
- Draw(): kirajzolás (itt lehet átméretezni és forgatni a sprite-ot)
- End(): SpriteBatch lezárása

```
spriteBatch.Begin();
```

```
spriteBatch.DrawString(font, "Hello World",  
    new Vector2(30, 30), Color.Black);
```

```
spriteBatch.Draw(spriteTexture,  
    new Vector2(30, 100), Color.White);
```

```
spriteBatch.End();
```

# XNA játék solution

---

- Egy tipikus XNA játék solution-ja 2 project-ből áll
  - **Game Project:** maga a játék forráskódja
  - **Content Project:** a játékhoz tartozó erőforrások (modellek, textúrák, videók, font-ok, stb.)
- A Game Project egy referenciával hivatkozik a Content Projectra, így könnyű kicserélni, pl. különböző célplatformok esetén
  - Egy Game Project több Content Project-re is hivatkozhat

# XNA HelloWorld 1/4

## ➤ A játékosztály váza

```
public class Game1 : Microsoft.Xna.Framework.Game
```

```
{
```

```
    GraphicsDeviceManager graphics;
```

```
    SpriteBatch spriteBatch;
```

```
    Texture2D spriteTexture;
```

```
    SpriteFont font;
```

```
    public Game1()
```

```
    protected override void Initialize()
```

```
    protected override void LoadContent()
```

```
    protected override void UnloadContent()
```

```
    protected override void Update(GameTime gameTime)
```

```
    protected override void Draw(GameTime gameTime)
```

```
}
```

Minden XNA játéknak kell egy GraphicsDeviceManager

SpriteBatch a sprite-ok kirajzolásához

Betűtípus

Sprite bitmap

# XNA HelloWorld 2/4

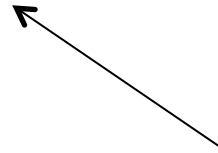
## ➤ Konstruktor

```
public Game1()  
{  
    graphics = new GraphicsDeviceManager(this);  
    Content.RootDirectory = "Content";  
}
```

A játék által használt  
GraphicsDeviceManager létrehozása

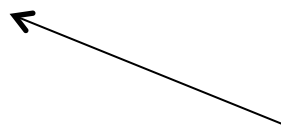


Megadjuk honnan fogjuk  
betölteni a  
„tartalmat” (erőforrásokat).  
Ha több Content  
projektünk lenne, itt lehet  
meghatározni melyikkel  
dolgozunk



Teljes képernyős megjelenítéshez:

```
graphics.PreferredBackBufferWidth = 480;  
graphics.PreferredBackBufferHeight = 800;  
graphics.IsFullScreen = true;
```



# XNA HelloWorld 3/4

---

- Tartalom betöltése (esetünkben sprite textúra és betűkészlet beállítása)

```
protected override void LoadContent()
```

```
{
```

```
    spriteBatch = new SpriteBatch(GraphicsDevice);
```

```
    spriteTexture =
```

```
        Content.Load<Texture2D>("target_small");
```

```
    font =
```

```
        Content.Load<SpriteFont>("SpriteFont1");
```

```
}
```

SpriteBatch létrehozása  
rajzoláshoz

Sprite textúra  
betöltése

Betűkészlet  
betöltése

# XNA HelloWorld 4/4

## ➤ A kirajzoló metódus megvalósítása

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    spriteBatch.DrawString(font, "Hello World",
        new Vector2(30, 30), Color.Black);
    spriteBatch.Draw(spriteTexture, new Vector2(30, 100),
        Color.White);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

Sprite  
„átszínezhető” (tint).  
Fehér esetén nincs



Szórakoztatóelektronikai eszközök programozása

**KINECT**



# Kinect

---

- „Natural User Interface”
  - Teljes test mozgásérzékelés
  - Gesztus alapú vezérlés
  - Hangfelismerés
- Irányítás bármiféle kontroller nélkül
- A világ leggyorsabban fogyó szórakoztató elektronikai cikke: megjelenés utáni 60 napban 8 millió adtak el
- Optimális használathoz 2-3m távolság kell a szenzortól

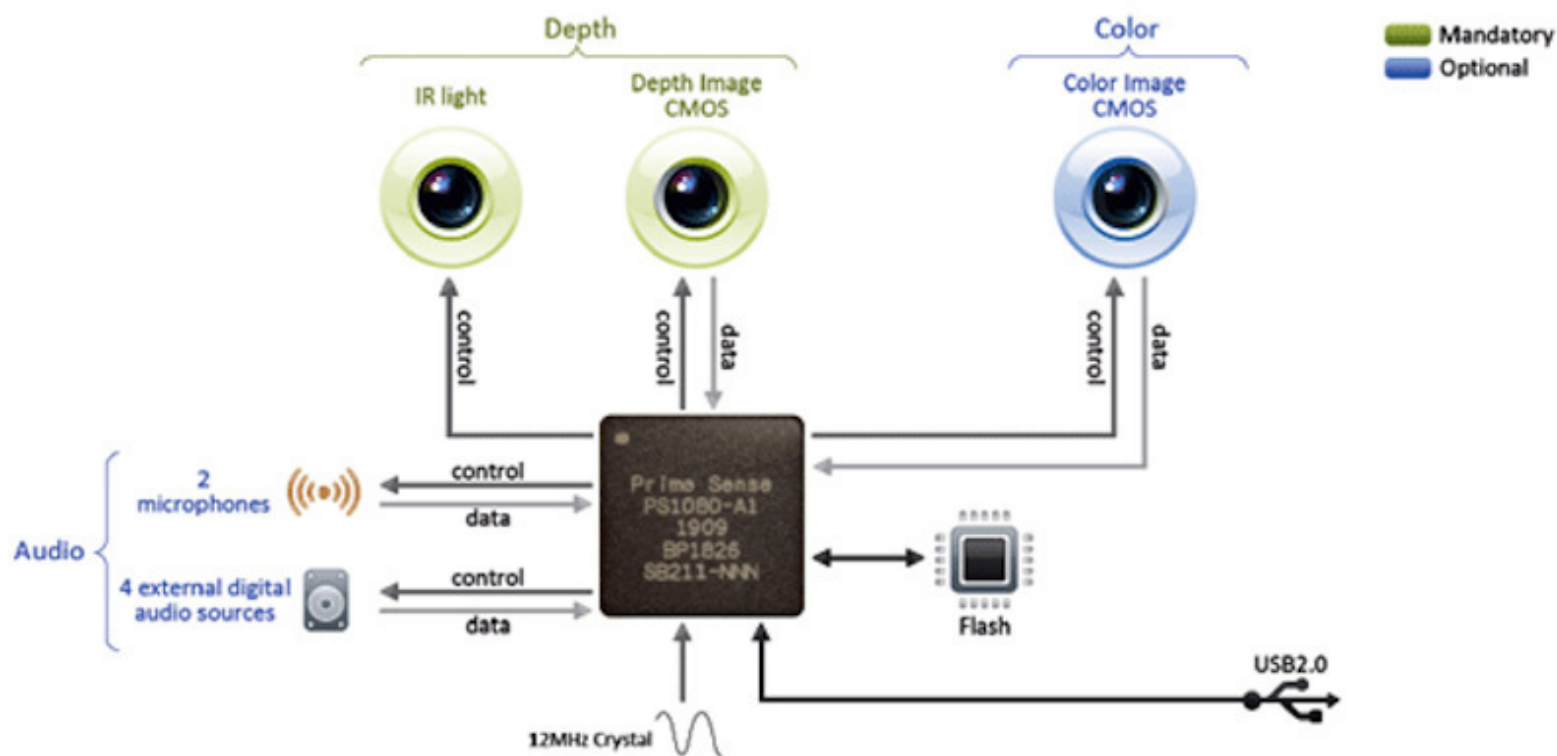
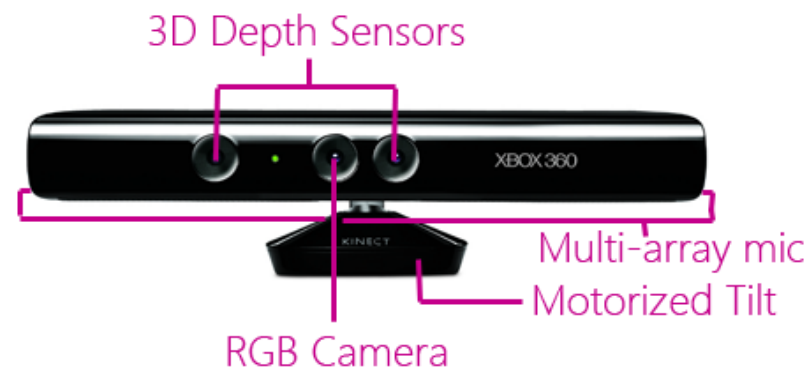
# Kinect hardver

---

- VGA kamera (640x480), 30 fps
- Depth (mélység/távolság) kamera
- Infra lézer fényforrás
- 4 mikrofon

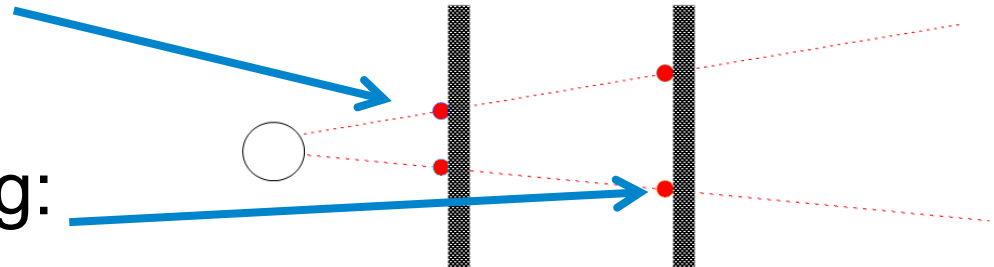
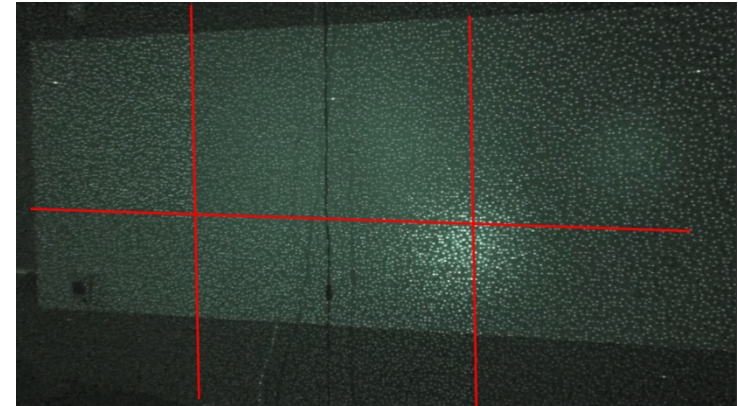


# Szenzorok



# Kinect mélységérzékelés

- Az infra lézer egy pontokból álló rácsot vetít ki
- A rács pontjait egy infra kamera érzékeli
- A rácspontok közötti távolságból következteti ki a mélységet
  - Kisebb távolság: közelebb
  - Nagyobb távolság: távolabb



# Kinect for Windows / Xbox 360

---

➤ 2 Kinect hardver, mindkettő csatlakoztatható PC-hez



- Kinect for Xbox 360

- Licenc: csak amatőr/tesztelési célokra
- \$150

- Kinect for Windows

- Fő különbség a firmware
- Near Mode: 40 cm-ről is képes a csontváz felismerésre
- Licenc: üzleti célokra is
- \$250



# Szoftverfejlesztés Kinect-re

---

➤ Xbox 360: lépj kapcsolatba a Microsofttal

➤ PC:

- Kinect for Windows SDK 1.7

- ingyenes
- C# és C++ API-k .NET-es fejlesztéshez
- nagy hardverigény (béta óta sokat gyorsult)
- egy PC-hez több Kinectet is köthetünk
- Windows 7-8, Visual Studio 10 vagy 2012 kell

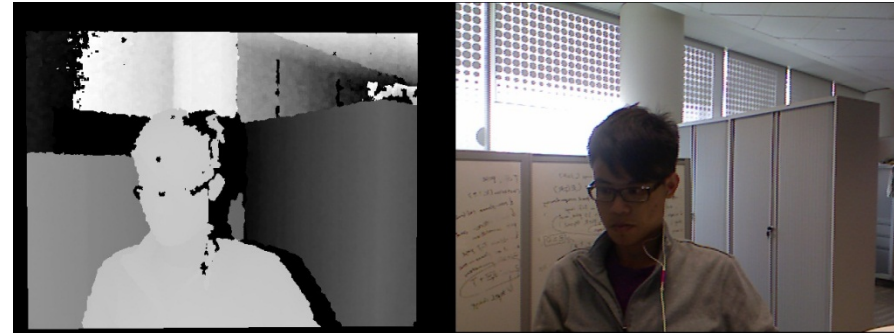
- Kinect for Windows Developer Toolkit

- Példaprogramok, dokumentáció, stb.

# Kinect SDK funkciók

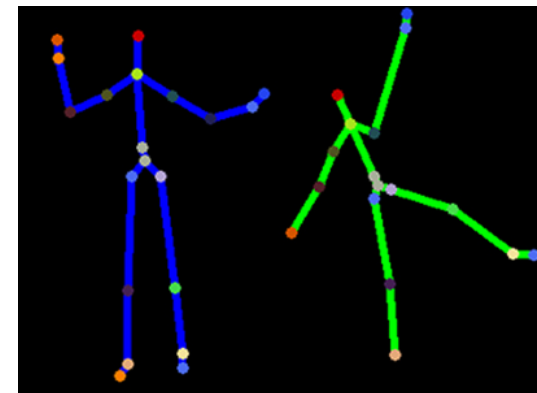
## ➤ Nyers szenzor adatfolyam

- Mélység szenzor
- VGA kamera
- Audio folyam



## ➤ Csontváz követés (skeletal tracking)

- A Kinect 20 pontra bontja az emberi testet és ezen pontok koordinátáit folyamatosan jelenti
- Egy folyamban kapjuk meg folyamatosan a csontváz adatokat



# Kinect SDK funkciók 2/2

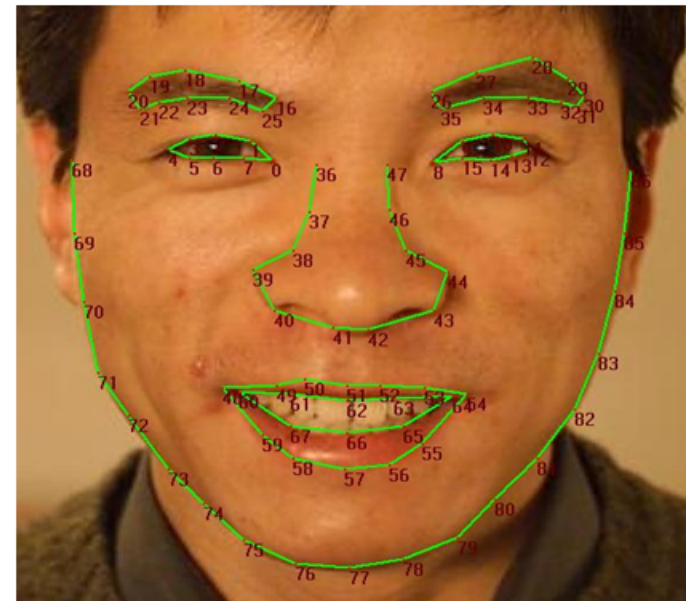
---

- Hangforrás irányának érzékelése,
- Hangfolyam zaj és visszhang szűréssel
- Beszédfelismerés (Microsoft Speech SDK kell)
- Arcfelismerés (Face Tracking SDK)
- Kinect Interactions: bizonyos mozdulatok/ gesztusok könnyű felismerése
  - Pl. drag, push
- Kinect Fusion: 3D szkennelés a Kinect szenzorral



# Face Tracking SDK

- Kinect SDK része
- Arcfelismerés, az arc jellegzetes pontjainak kiválasztása
- Az arc felépítésének leírása és 3D modelljének elkészítése
- Az arcfelismerés elindítása után folyamatosan kapjuk az adatokat FTResult objektumokban
  - Hasonlóan mint csontváz követésnél



# Face Tracking adatok (FTResult)

---

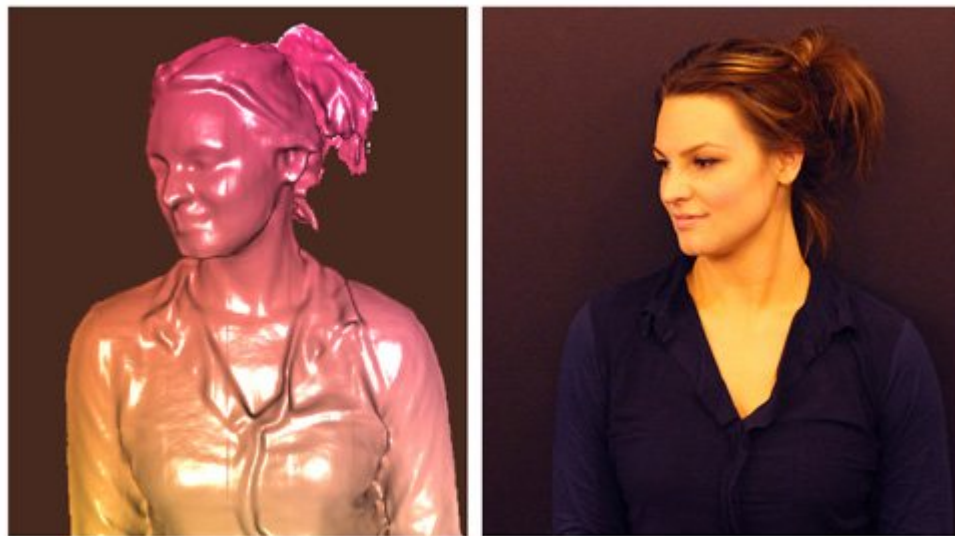
- 87 2D pont az arcról
- 6 Animation Unit (AU) és 11 Shape Unit (SU)
  - SU: az arc felépítésének egy paramétere
    - Pl. szem szélesség és magasság
  - AU: az arc mimikájának egy paramétere
    - pl. AU1 – Jaw Lowerer: -1 és 1 közötti érték, mennyire van kinyitva a száj
- Az arc 3D-s modellje



# Kinect Fusion

---

## ➤ 3D szkennelés



# KinectSensor class

---

1. KinectSensor osztály példányosítása
2. Start() metódus hívása egy Kinect inicializálásához
3. Stream-ek lekérése és feldolgozása
  - Color (nyers kameraképek)
  - Depth (nyers mélység adatok)
  - Skeleton (csontvázak és ízületi pontok)
4. Stop() metódus hívásával leállítjuk a Kinect-et

# Color Stream lekérése

- Események jelzik ha rendelkezésre áll egy új kamerakép

Color stream inicializálása

```
kinectSensor.ColorStream.Enable(  
    ColorImageFormat.RgbResolution640x480Fps30);
```

```
kinectSensor.ColorFrameReady +=  
    new EventHandler<ColorImageFrameReadyEventArgs>(  
        kinectSensor_ColorFrameReady);
```

Feliratkozás az eseményre

```
void kinectSensor_ColorFrameReady(object sender,  
    ColorImageFrameReadyEventArgs e)
```

```
{
```

```
    ...
```

```
}
```

Esemény lekezelése (pl.  
kamerakép eltárolása  
textúrában)

# Skeletal tracking

- A Skeleton stream-ből Skeleton típusú objektumokat tudunk kiolvasni
- Minden játékoshoz tartozik egy külön Skeleton
- A Skeleton 20 Joint-ból áll, ezeket a Joints property-ből tudjuk kiolvasni

```
Joint rightHand =  
    playerSkeleton.Joints[  
        JointType.HandRight];
```

- A Joint.Position egy 3 dimenziós vektor, mely tartalmazza a Joint pozícióját (-1 és 1 közötti érték)

