

## **Szórakoztató elektronikai eszközök programozása (VIAUAV05)**

Tárgy hivatalos adatlapja:

<http://www.vik.bme.hu/kepzes/targyak/VIAUAV05>

A diasoron lévő kérdések kigyűjtve(1-8. előadás).

Készítette: Gróf Attila

Ha hibát találsz akkor kérlek jelezd itt: [grof.attila9@gmail.com](mailto:grof.attila9@gmail.com)

## 1. Előadás

- **Milyen szórakoztató elektronikai software-s technológiákat ismer?**
  - Web alapon
    - Flashalapon
    - HTML 5 alapon
  - Java alapon
    - BD-J, Java FX
  - Linux alapon
    - WebOS, Tizen, Android, ...
  - Egyedi
    - Roku, Xstreamer, ....
- **Mi a BD-J technológia két fő ága?**
  - Más néven: Java TV
  - Alkalmazást írhatunk a Blu-ray lemezre, amit a lejátszó futtat
  - MIDlet-hez hasonló Xlet-ek
    - Csak Blu-ray lemezre
    - GEM (Globally Executable Multimedia Home Platform) tartalomként
  - Több mint egy statikus DVD menü
    - „Bonus View” funkciócsoport
      - Kép a képben elérés
      - Hozzáférés a helyi tárolóhoz
    - „BD Live” funkciócsoport
      - Hálózati hozzáférés
- **Magyarázza el a SmartTV fogalmát!**
  - A Smart TV vagy Connected TV: „olyan televízió, amely a hagyományos TV-hez képest továbbfejlesztett képességekkel rendelkezik a kapcsolódás, tartalom és felhasználói élmény tekintetében”
  - Fizikai felépítés kétféle lehet
    - TV-be integrált internetes funkciók
    - Külön STB, a TV csak a megjelenítő eszköz
  - Egy adott platformhoz tartozó alkalmazásokat és plug-inokat/add-onokat telepíthetünk, futtathatunk
  - A hagyományos műsorszolgáltatás kicsit háttérbe szorul, a tv adás már csak egy funkció a sok közül
- **Mi a különbség a Chromecast és az AndroidTV (GoogleTV) között?**
  - GoogleTV
    - Andorid alapú, STB és TV integráció
    - GooglePlay, Chromeböngésző, Chromecasttámogatás, stb.
  - Chromecast
    - TV okosító HDMI stick
    - Videó streamelés Chrome-ból, Androidoskészülékekről
  - AndroidTV
    - Az elavult GoogleTV utódja
- **Milyen technológiákon akapul a HTML5?**
  - Webes technológiákon
- **Mi a WebGL, mire használható?**
  - Web-alapú grafikus könyvtár (Web-based Graphics Library)
  - A JavaScript programozási nyelvet kiegészíti 3D-s számítógépes grafikai lehetőségekkel
  - OpenGL ES 2.0-án alapuló, 3D-grafikai API-tnyújt
  - Implementációk:
    - MozillaFirefox
    - GoogleChrome
    - Safariés Opera fejlesztői változatai
  - 2011. március 3: 1.0-ás verzió
  - 2014-es Samsung SmartTV-k már támogatják

## 2. Előadás

- **Mi a HTML 5, milyen technológiákat foglal magába?**
  - HTML = HyperTextMarkupLanguage
  - Régóta jelen van webes szabványként
  - XML alapú, több szabvány verzió
  - A HTML legújabb változata az 5. verzió
    - Felkészül: HTML 5.1, 5.2 ...
  - Nem platform, nem runtime, nemfejlesztő környezet, nem keretrendszer!
  - Szabvány(ok gyűjtőfogalma)
  - A HTML 4.01 már 15 éves!
    - Új igények jelentek meg azóta a weboldalak fejlesztése során
    - A HTML 5 szabványos megoldást ad
  - HTML5 =
    - Új HTML elemek, attribútumok
    - JavaScript újdonságok
    - CSS3
    - (WebGL, WebM, stb...)
- **Sorolj fel és ismertess röviden X új HTML vagy JavaScript újítást, amit a HTML 5-ben szabványosítottak. (X >= 10)**
  - JavaScript HTML5 related APIs
    - The canvas element for immediate mode 2D drawing. See Canvas 2D API Specification 1.0 specification
    - Timed media playback
    - Offline Web Applications
    - Document editing
    - Drag-and-drop
    - Cross-document messaging
    - Browser history management
    - MIME type and protocol handler registration
    - Microdata
    - Web Storage, a key-value pair storage framework that provides behaviour similar to cookies but with larger storage capacity and improved API.
  - HTML5
    - New Semantic/Structural Elements
      - HTML5 offers new elements for better document structure:
      - `<article>` Defines an article in the document
      - `<aside>` Defines content aside from the page content
      - `<bdi>` Defines a part of text that might be formatted in a different direction from other text
      - `<details>` Defines additional details that the user can view or hide
      - `<dialog>` Defines a dialog box or window és még sok.....
    - New Form Elements
      - `<datalist>` Defines pre-defined options for input controls
      - `<keygen>` Defines a key-pair generator field (for forms)
      - `<output>` Defines the result of a calculation
    - New Input Types
      - color
      - date
      - datetime
      - datetime-local
      - email és még sok.....
    - HTML5 - New Attribute Syntax
      - Empty `<input type="text" value="John" disabled>`
      - Unquoted `<input type="text" value=John>`

- Double-quoted `<input type="text" value="John Doe">`
- Single-quoted `<input type="text" value='John Doe'>`
- HTML5 Graphics
  - `<canvas>` Defines graphic drawing using JavaScript
  - `<svg>` Defines graphic drawing using SVG
- New Media Elements
  - `<audio>` Defines sound or music content
  - `<embed>` Defines containers for external applications (like plug-ins)
  - `<source>` Defines sources for `<video>` and `<audio>`
  - `<track>` Defines tracks for `<video>` and `<audio>`
  - `<video>` Defines video or movie content
- **Milyen új képességekkel rendelkezik a CSS 3?**
  - CSS2-vel ellentétben már modulokból áll
  - Az újonnan bevezetett szelektorok a gyakran felmerülő feladatokat könnyítik, pl.
    - Egy elem első/utolsó gyerekének közvetlen kiválasztása
    - Páros, páratlan sorok (táblázat színezés)
    - Rengeteg új, látványos attribútum (tükröződés, kerekített sarkok, árnyékolás, saját betűtípusok használata, stb...)
  - Megoldás CSS preprocessor-ok
    - Változók és függvények
    - Egyszerű matematikai műveletek
    - Átláthatóbb struktúra, és egyszerűbb szintaxis

### 3. Előadás

- Nincsenek kérdések

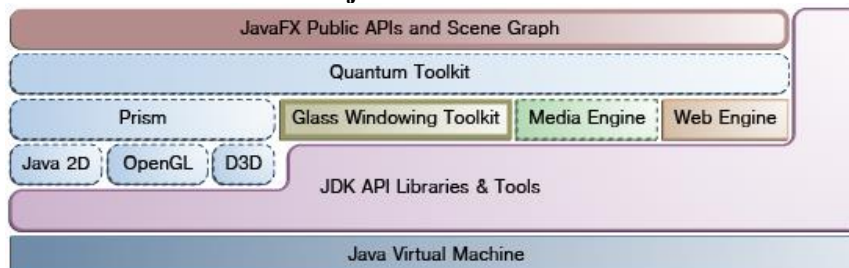
### 4. Előadás

- Nincsenek kérdések

### 5. Előadás

- **Milyen újdonságokat hozott a JavaFX?**
  - Java kliens platform új generációja
  - Gazdag felhasználói felület és RIA (Rich Internet Application) támogatás
  - Konzisztens működés több platformon
  - Gazdag Java alapú API grafikai és média támogatásra
  - Hardware támogatás
  - Széleskörű média támogatás

- **Váolja fel a JavaFX architektúráját!**



- 
- **Hogy működik a SceneGraphmodell?**
  - Kezdőpont JavaFX alkalmazások fejlesztéséhez
  - Hierarchikus fa, minden csomópont a felhasználói felület egy eleme
  - Megjeleníthető
  - Beviteli eseményeket fogadhat
  - Minden csomópont tartalmazhat:
    - Egyedi ID
    - Stílus (styleclass)

- Minden csomópontnak egy szülője és 0 vagy több gyermek eleme lehet
- Csomópont további elemei
  - Effektek (elmosás, árnyék, stb), átlátszóság
  - Transzformáció
  - Eseménykezelő (egér, billentyűzet, stb.)
  - Alkalmazás specifikus állapot
- Létrehozható tartalmak:
  - Csomópontok: alakzatok (2D és 3D), képek, média, beágyazott böngésző, szöveg, UI vezérlők, diagrammok, csoportok, containerek
  - Állapotok: Valamilyen tartalom grafikus tulajdonságát írja le az egyes állapotokban (pozíció, orientáció, stb.)
  - Animációk: A SceneGraphlemeinek tulajdonságait változtatja az idő függvényében
  - Effektek: A SceneGraphcsomópontjainak megjelenítését befolyásolja (elmosás, árnyék, színátmenet, stb.), egyszerre akár többet is (gyerek elemek)
- **Milyen JavaFXszálakat ismer?**
  - A JavaFXegy időben kettő vagy több szálát folyamatosan futtat
  - JavaFXalkalmazás szál:
    - Fő szál, minden sceneebből a szálból érhető el, de lehet háttér szálakat is létrehozni
    - Más mint a Swingvagy az AWT szál, ezért figyelni kell amikor JavaFXkód kerül Swingalkalmazásba
  - Prismrenderszál:
    - A renderelésértfelelős szál, függetlenül az eseménykezelőtől
    - Lehetővé teszi, hogy az N-edik framerenderelődjön, amíg az N+1-edik feldolgozás alatt van
    - Konkurens feldolgozás
  - Média szál:
    - Háttérben fut
    - Szinkronizációsfeladatok
- **Hogyan telepíthetők a JavaFXalkalmazások?**
  - Desktopvagy böngésző alkalmazás
  - Háromféle telepítési mód:
    - Standalone: JAR telepítése/indítása
    - Browser: Weboldalba ágyazott objektum, JavaScript-en keresztül el tudja érni az oldalt
    - Web Start: Serverről töltődik le az alkalmazás, majd a JNLP állományra kattintva futtatható az alkalmazás
- **VázoljafegyJavaFXalkalmazáskódját, amely egy gombot jelenít meg és a gomblenyomás hatására egy tetszőleges szöveget jelenít meg “pop-up”-ban.**

```
publicvoidstart(finalStage primaryStage) {
```

```
    Buttonbtn= newButton();
    btn.setText("Show PopUp");
    btn.setOnAction(newEventHandler<ActionEvent>() {
        @Override
        publicvoidhandle(ActionEventevent) {
            Popuppopup= newPopup();
            HBoxbox= newHBox();
            box.getChildren().add(newLabel("Hello JavaFXcsoport!"));
            box.setPrefSize(150, 30);
            box.setStyle("-fx-background-color: gray;");
            popup.getContent().add(box);
            popup.setX(200);
            popup.setY(300);
        }
    });
}
```

```

    popup.show(primaryStage);
    }
});

StackPaneroot= newStackPane();
root.getChildren().add(btn);
Scene scene = newScene(root, 300, 250);
primaryStage.setTitle("Hello PopUp!");
primaryStage.setScene(scene);
primaryStage.show();
}

```

## 6. Előadás

- **Mi a JavaFX layout használat legnagyobb előnye?**
  - Alapértelmezetten megadhatjuk a komponensek X, Y koordinátáját
  - De! Sokkal jobb és egyszerűbb layout-okat használni
  - A JavaFX számos layout-ot támogat, melyeket egyszerű használni
  - Előny: Ablak átméretezéskor automatikusan jól rendezi el az elemeket
- **Írjon egy JavaFX programot, amely egy Label-t és egy Button-t jelenít meg. A Button lenyomására írja ki az aktuális időt a Label-re!**
- **Vázzon fel egy JavaFX alkalmazás kódját, amely lejátsza és megjeleníti a D:\sample.flv videót**

```

Media mMovie = newMedia( "http://www.mediacollege.com/video-gallery/testclips/20051210-w50s.flv");
finalMediaPlayer playerMovie= newMediaPlayer(mMovie);
MediaView mvMovie= newMediaView(playerMovie);
mvMovie.setLayoutX(5);
mvMovie.setLayoutY(30);
mvMovie.setRotate(30);
Media mMp3 = newMedia("file:///D:/voyager.mp3");
finalMediaPlayer playerMp3 = newMediaPlayer(mMp3);
Button btn= newButton();
btn.setLayoutX(5);
btn.setLayoutY(5);
btn.setText(„Play");
btn.setOnAction(newEventHandler<ActionEvent>() {
    public void handle(ActionEvent event) {
        System.out.println("Hello World");
        playerMovie.setVolume(0.0);
        playerMovie.play();
        playerMp3.play();
    }
});
root.getChildren().add(btn);
root.getChildren().add(mvMovie);

```

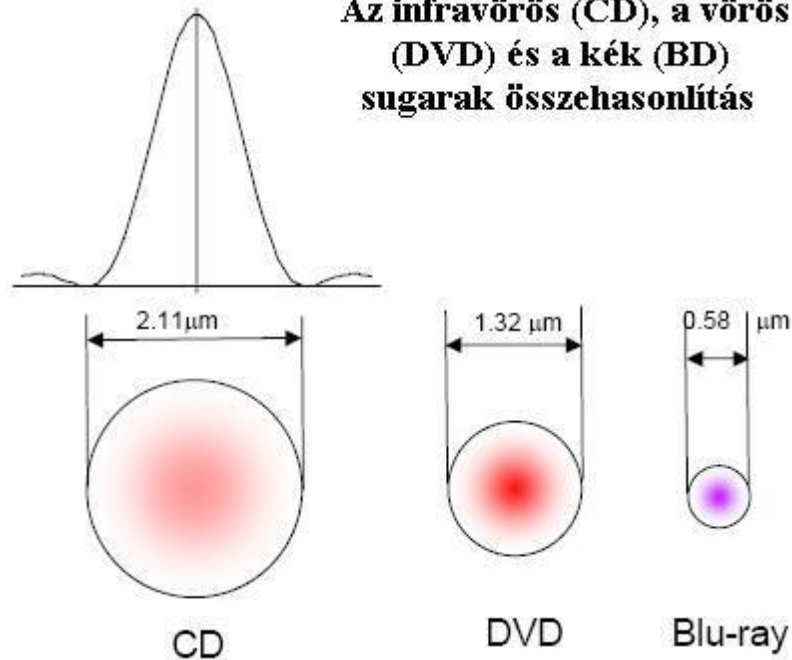
- **Mit jelent a PropertyBinding? Mutasson egy egyszerű példát!**
  - Tulajdonság kötés (PropertyBinding)
  - JavaBeans modellen alapszik, de továbbfejlesztették
  - Változók közötti direkt kapcsolat jelzése
  - Egyik objektum változása azonnal hatással van a vele összekötött objektumra
  - Például:
    - összes bevétel érték mindig az egyes bevétel elemek összege
    - GUI frissen tartása az adat objektumok összekötésével
  - Egy kötés (binding) egy vagy több forrástól függ (dependency), figyelni a függőségekben bekövetkezett változásokat és megfelelően frissíti magát

- A bindingAPI két fő részből áll:
  - MagasszintűAPI: Egyszerű kötések létrehozása (pl. összeg), IDE támogatás
  - Alacsony szintű API: Tetszőleges kötések létrehozhatók
- **Mire használható az FXML?**
  - JavaFX2.0 újdonsága
  - XML alapú deklaratívleírnyelv felhasználói felületek tervezésére
  - Ideális statikus felületek, mint például űrlapok, táblázatok megjelenítésére
  - Támogatja dinamikusfelületek létrehozását, script-ekfelhasználásával
  - Java kóddal együtt tud működni
- **Sorolja fel az FXML legfontosabb előnyeit?**
  - XML alapú, ezért minden fejlesztő számára ismert nyelv
  - Főként a web fejlesztők kedvelik, de sok helyen előfordul (pl. Android)
  - Nem kell lefordítani a kódot, ha változik az FXML
  - Átláthatóbbá teszi a SceneGraphstruktúráját
  - Könnyebben tud több fejlesztő együtt dolgozni a UI-on

## 7. Előadás

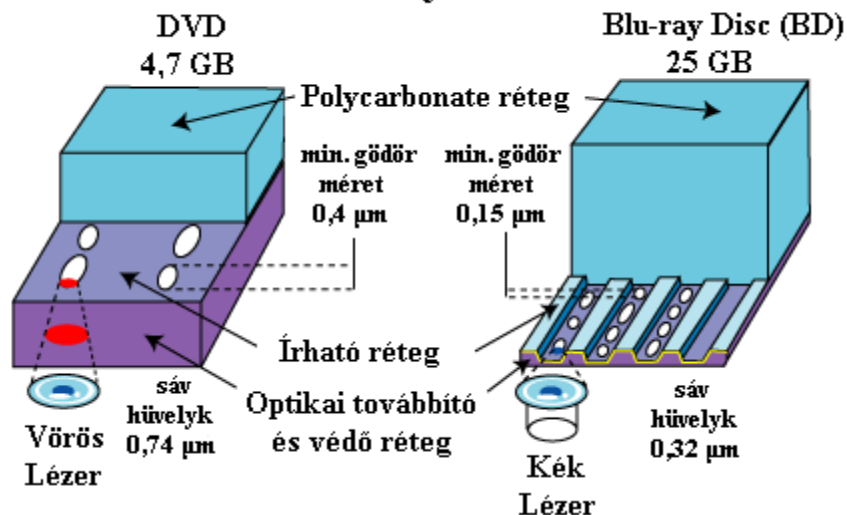
- **Ismertesse a Blu-ray Disc technológiát! Térjen ki a DVD és a Blu-raylézerre, a tárhelykapacitásra (annak okaira), valamint a sebességre.**
  - Egy egyoldalas, hagyományos DVD 4,7 GB adatot képes tárolni, ami elegendő egy átlagos 2 órás, normál felbontású filmnek és néhány extra adatnak
  - Egy nagy felbontású film (HDTV) körülbelül 5-ször több helyet igényel
  - Egy 2 órás felvételhez 22 GB szükséges
  - Nélkülözhetetlen olyan lemezek gyártása, amin sokkal több adat fér el, mint egy DVD-n, ahogy a stúdiók is egyre jobb minőségben gyártják a filmeket
  - Ahhoz, hogy a HDTV-rőlfelvegyünk egy több, mint 2 órás műsort, Blu-raylemezre van szükség
  - A Blu-ray-technológiaára egyenlőre még relativemagas
  - DVD vörös lézert használ, míg a BD kéket
  - A kék lézer rövidebb hullámhosszon működik (405 nanométer), mint a vörös lézer (650 nanométer)
  - A kisebb fénynyalábbal pontosabban lehet fókuszálni, lehetővé téve, hogy olyan kis gödröcskéből („pits”) is kiolvassunk adatokat, amik csak 0,15 mikrométer hosszúak –ez kevesebb, mint fele a DVD-n található kis gödrök méretének
  - Továbbá a Blu-raylecsökkentette a sávok szélességét 0,74 mikrométerről 0,32 mikrométerre
  - A kisebb gödröcskék, a kisebb fénysugár, és a rövidebb sáv-hüvelyk együttese révén az egyrétegű Blu-raylemezen több mint 25 GB információt tudunk tárolni
  - Dupla rétegű Blu-raylemez50 GB-t képes tárolni
  - Az 50 GB-nyi tárhely tökéletes FullHD-sfilmek tárolására 1920×1080-as felbontásban (a DVD 720×576)

## Az infravörös (CD), a vörös (DVD) és a kék (BD) sugarak összehasonlítás



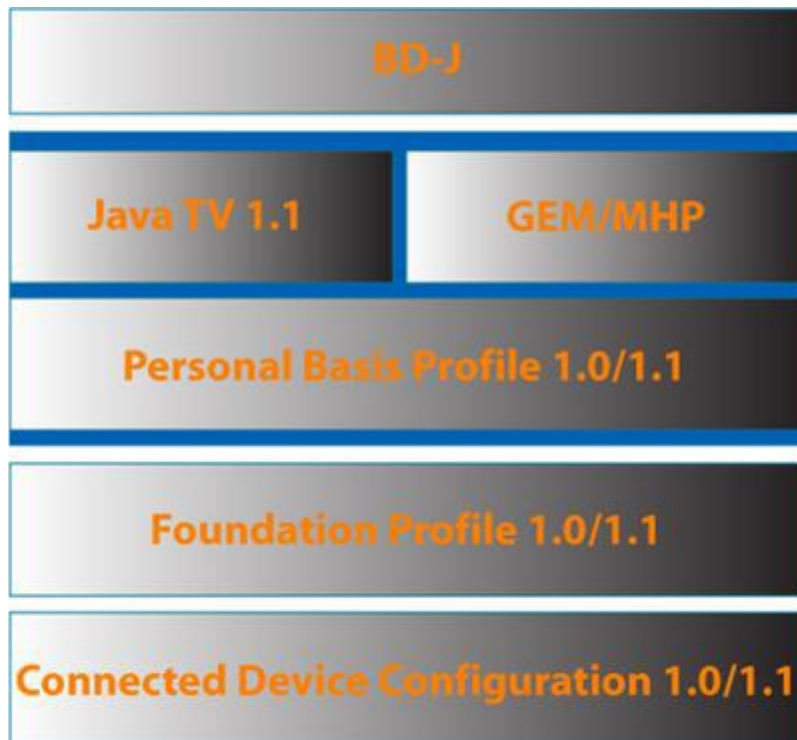
- 
- Előállítási költsége alacsonyabb, mint a DVD-nek
- Az adatátviteli sebességben is a BD a gyorsabb
- DVD: 10 Mbps
- Blu-raylemez: 36 Mbps
- 25 GB adat másolásához elegendő csupán másfél óra
- Védelem szempontjából is jobb a Blu-raylemez a mai DVD-kinél
- Egyedi biztonsági titkosító rendszerrel rendelkeznek, ami annyit jelent, hogy van egy egyedi ID-jük(azonosító), ami védelmet nyújt a film-és szoftverkalózkodok ellen, illetve a szerzői jogok megsértése ellen
- A JVC kifejlesztett egy Blu-ray/DVD Combolemezt megközelítőleg 33,5 GB kapacitással, ami lehetőséget teremt arra, hogy mindkét formátumú filmet tároljunk egyetlen lemezen

## A DVD és a Blu-ray összehasonlítása



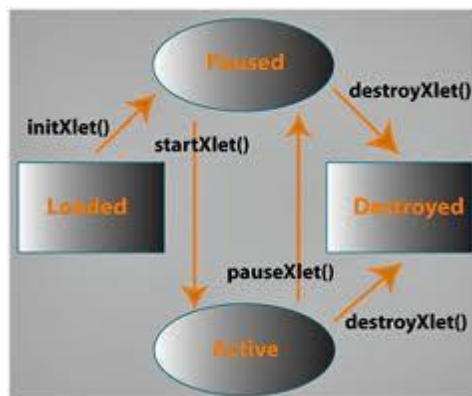
- 
- Rajzolja fel a BD-J API Stack-et!





- 
- **Mutassa be és magyarázza el az Xlet életciklus modellt!**

- A konstruktor meghívása után az Xlet a Loaded állapotba kerül
- Ezt követően az `initXlet()` automatikusan meghívódik és Paused állapotba kerül az Xlet
- Indításkor a `startXlet()` hívódik meg, melynek hatására Active állapotba kerül az Xlet
- Látható, hogy befejezés előtt a Paused és az Active között többször is válthat az Xlet
- `startXlet()` függvényben érdemes ellenőrizni az erőforrásokat és megnyitni őket, ha még nincsenek
- Destroyed állapotba kerülés előtt a `destroyXlet()` függvényben szabadítsuk fel a lefoglalt erőforrásokat



- 
- **Vázolja fel az Xletvázát kód szinten!**

```

public class MyFirstExampleXlet implements javax.tv.xlet.Xlet {
    public MyFirstExampleXlet() {
    }
    public void initXlet(javax.tv.xlet.XletContext context)
        throws javax.tv.xlet.XletStateChangeException {
    }
    public void startXlet()
        throws javax.tv.xlet.XletStateChangeException {
    }
    public void pauseXlet() {
    }
}
  
```

```

        public void destroyXlet(boolean unconditional)
            throws javax.tv.xlet.XletStateChangeException {
        }
    }
}

```

- **Adja meg egy *Xlet* kódját, mely megjeleníti a „Hello CE csoport!” szöveget!**

```

import java.awt.Color;
import java.awt.Container;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Rectangle;

import javax.tv.xlet.Xlet;
import javax.tv.xlet.XletContext;

import org.havi.ui.HScene;
import org.havi.ui.HSceneFactory;
import org.havi.ui.HSceneTemplate;

public class XletDemoBasic implements Xlet {

    private class MyContainer extends Container
    {
        private Font font = new Font(null, Font.PLAIN, 24);

        public void paint(Graphics g) {
            g.setFont(font);
            g.setColor(new Color(10, 10, 10));
            g.fillRect(10, 10, getWidth()-20, getHeight()-20);
            g.setColor(new Color(245, 245, 245));
            int message_width = g.getFontMetrics().stringWidth("Hello Xlet!");
            g.drawString("Hello Xlet!", (getWidth() - message_width) / 2, 100);
        }
    }

    // --- Xlet main code

    private HScene scene;
    private MyContainer container;

    public XletDemoBasic() {
    }

    public void initXlet(XletContext context) {
        initScene();

        container = new MyContainer();
        container.setSize(scene.getBounds().width, scene.getBounds().height);

        scene.add(container, BorderLayout.CENTER);
        scene.validate();
    }

    public void startXlet() {
        container.setVisible(true);
        scene.setVisible(true);
    }
}

```

```

public void pauseXlet() {
    container.setVisible(false);
}

public void destroyXlet(boolean unconditional) {
    scene.remove(container);
    scene = null;
}

// ---

public void initScene() {
    // HSceneFactory lekérése
    HSceneFactory factory = HSceneFactory.getInstance();
    // HSceneTemplate, hogy a teljes képernyőt lefedő HScene-t elkérjük
    HSceneTemplate hst = new HSceneTemplate();
    hst.setPreference(
        HSceneTemplate.SCENE_SCREEN_DIMENSION,
        new org.havi.ui.HScreenDimension(1,1),
        HSceneTemplate.REQUIRED);
    hst.setPreference(
        HSceneTemplate.SCENE_SCREEN_LOCATION,
        new org.havi.ui.HScreenPoint(0,0),
        HSceneTemplate.REQUIRED);
    // HScene beállítása
    scene = factory.getBestScene(hst);
}
}

```

- **Honnan származnak a billentyű események BD-J alkalmazásokban?**
  - KeyListener

## 8. Előadás

- Nincsenek kérdések