UNIVERSITATEA POLITEHNICA BUCUREŞTI

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

Proiect SMSCPFTM

Aplicaţie pe SMART TV

~ Music Player ~

Student: Văduva Marian-Cătălin

Grupa: 411-CMM

Bucureşti 2018

1. Introducere

1.1 Ce este un SMART TV?

SMART TV, denumit şi TV hibrid, este un televizor LED ce integrează funcţiile clasice ale unui televizor cu funcţie de conectare la Internet. În SMART TV sunt încorporate dispozitive precum media player-e digitale, Blu-ray, console pentru jocuri video şi alte dispozitive interactive. În plus, SMART TV-urile permit vizionarea unui conţinut multimedia de pe memory stick-uri USB sau HDD-uri externe şi conectarea prin Wi-Fi sau Bluetooth cu alte dispozitive precum smartphone, aparat foto sau video, tabletă, laptop sau PC. SMART TV-ul dispune de procesor intern şi memorie flash internă. Este capabil să ruleze un sistem de operare şi aplicaţii special create pentru acesta. SMART TV-ul nu trebuie confundat cu servicii sau echipamente precum IPTV.

1.2 Dezvoltarea aplicaţiilor pentru SMART TV Samsung

Aplicaţiile pentru SMART TV Samsung se pot realiza/dezvolta cu ajutorul mediului de dezvoltare Tizen Studio. Tizen Studio este un IDE (Integrated Development Environment) oficial pentru dezvoltarea aplicaţiilor web şi aplicaţiilor native Tizen. Este un set cuprinzător de instrumente. Pe lângă aceste instrumente, acesta cuprinde şi un emulator, multe exemple de proiecte de tip sample şi documentaţie. Tizen Studio poate rula pe Windows, Ubuntu şi macOS.

Figura 1.1: Tizen Studio

1.3 Ce este o aplicaţie de tip Music Player pentru SMART TV?

O aplicaţie de tip Music Player este o aplicaţie ce are rolul de a administra fişierele audio (melodii) prezente într-un mediu de stocare (stick USB, HDD, card SD etc.). De asemenea, acest tip de aplicaţie oferă utilizatorului o interfaţă grafică intuitivă. Funcţiile uzuale folosite în astfel de aplicaţii pentru manevrarea fişierelor audio sunt: pornire (play), oprire (stop), pauză (pause), melodia următoare (next), melodia precedentă (previous), fast-forward etc. Aplicaţiile de tip Music Player se găsesc preinstalate pe majoritatea dispozitivelor precum: smartphone, tabletă, laptop, PC, smartwatch, SMART TV etc. De asemenea, pe aceste dispozitive se pot instala alte aplicaţii de tip Music Player în funcţie de preferinţele utilizatorului.

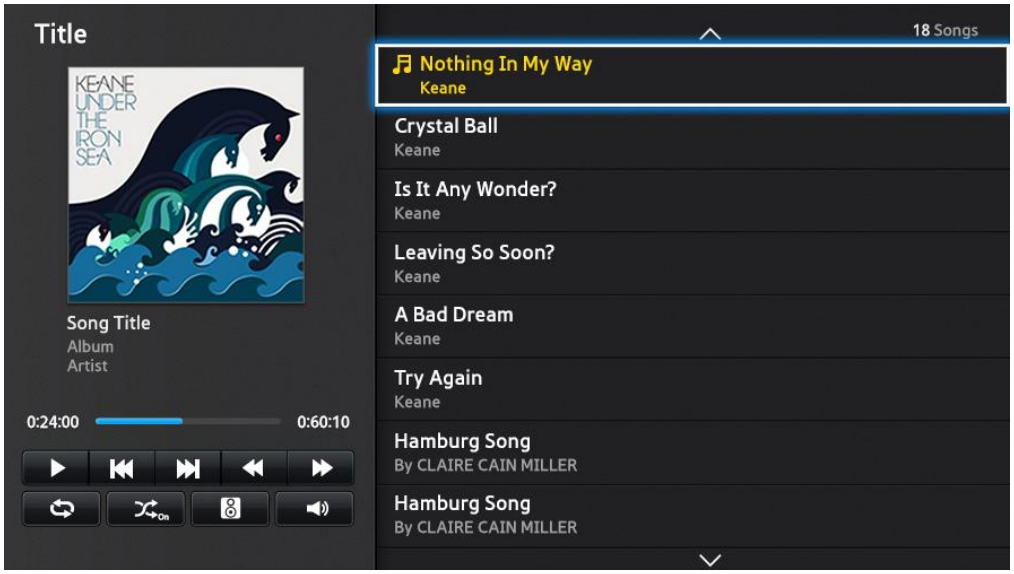

Figura 1.2: Exemplu de aplicaţie de tip Music Player pentru SMART TV

Scopul prezentului proiect este de a realiza o aplicaţie de tip Music Player pentru SMART TV Samsung folosind mediul de dezvoltare Tizen Studio.

2. Descrierea aplicaţiei realizate

Realizarea aplicaţiei de tip Music Player pentru SMART TV Samsung s-a început prin deschiderea unui exemplu de proiect de tip sample în Tizen Studio, acela fiind HelloTizen, pentru înţelegerea conceptului de aplicaţie pe SMART TV realizată în Tizen. Apoi s-a trecut la editarea treptată a proiectului-exemplu în funcţie de necesităţi. Observarea funcţionalităţii aplicaţiei s-a realizat cu ajutorul unui emulator de SMART TV instalat separat pe Tizen.

2.1 Design-ul aplicaţiei

Secţiunea Butoanelor “Play”, “Pause”, “Stop”, “Next” şi “Previous”

Design-ul aplicaţiei s-a realizat prin intermediul limbajelor HTML şi CSS. S-a creat un element de tip “div” de clasă “container” în care s-au adăugat toate elementele aplicaţiei din punctul de vedere al stilului. Primele elemente introduse au fost butoanele cele mai uzuale pentru orice aplicaţie de tip Music Player şi anume: Play, Pause, Stop, Next şi Previous. Pentru fiecare buton s-a creat câte un element de tip “span” de clasă “btn” şi id-uri diferite în interioul “container-ului”. De asemenea, s-a creat încă un element de tip “span” pentru un selector de buton pentru ca utilizatorul să observe ce funcţie urmează să folosească. Imaginile pentru butoane au fost descărcate de pe Internet şi folosite în cadrul aplicaţiei, imaginile fiind introduse într-un director numit “image”.

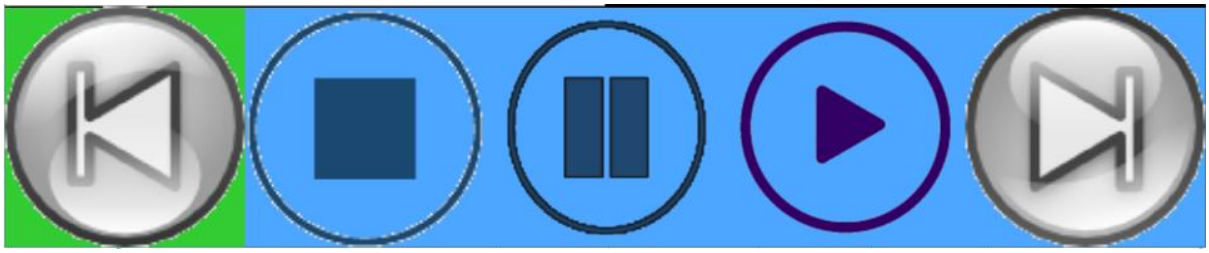

Figura 2.1: Secţiunea butoanelor “Play”, “Pause”, “Stop”, “Next” şi “Previous”

Cod HTML:

```
<body>
  <div class="container">

        <span class="sel_item" id="sel"></span>

        <span class="btn" id="prev"></span>
        <span class="btn" id="next"></span>
        <span class="btn" id="play"></span>
        <span class="btn" id="stop"></span>
        <span class="btn" id="pause"></span>
```

Cod CSS:

```css
body {
    margin: 0px auto;
    background-color:#222;
}

.container {

    position: absolute;
    top: 0px;
    height: 1080px;
    width: 1920px;
    background-color: #4da6ff;
}

#play{
    background-image: url("../image/play2.png");
    left: 1152px;

}

#stop{
    background-image: url("../image/stop.png");
    left: 384px;

}

#pause{
    background-image: url("../image/pause.png");
    left: 768px;
}
```

```css
.btn {

    position: absolute;
    bottom: 0px;
    height: 384px;
    width: 384px;


}

.sel_item {

    position: absolute;
    bottom: 0px;
    height: 384px;
    width: 384px;
    left: 0px;
    background-color: #33cc33;

}

#prev{
    background-image: url("../image/prev.png");
    left: 0px;

}

#next{
    background-image: url("../image/next.png");
    left: 1536px;

}
```

Secţiunea “Playlist”

În această secţiune s-au introdus mai multe elemente. Primul element creat este de tip “span” de clasă “playlist-class” ce reprezintă un patrulater de culoare albă (fundalul Playlist-ului). Următoarele elemente introduse au fost de tip “span” ce reprezintă textul afişat pe Playlist (titlul “Playlist” şi numele melodiilor prezente în aplicaţie scrise într-o ordine). Următoarele elemente introduse în această secţiune au fost trei imagini. Una dintre ele este o săgeată pe fundal gri ce are rolul de a arăta melodia selectată din Playlist de către utilizator. Celelalte două imagini au rolul de a arăta starea actuală a melodiilor ce sunt în curs de rulare.

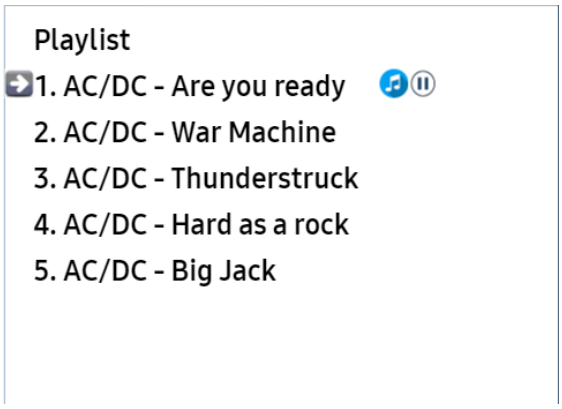

Figura 2.2: Secţiunea “Playlist”

Cod HTML:

```
<span class="playlist-class" id="playlist"></span>
<span class="playlist-text" id="text">Playlist</span>
<span class="playlist-1" id="1">1. AC/DC - Are you ready</span>
<span class="playlist-2" id="2">2. AC/DC - War Machine</span>
<span class="playlist-3" id="3">3. AC/DC - Thunderstruck</span>
<span class="playlist-4" id="4">4. AC/DC - Hard as a rock</span>
<span class="playlist-5" id="5">5. AC/DC - Big Jack</span>
<span class="sel_music" id="sel_m"></span>
<span class="playing" id="plying"></span>
<span class="paused" id="pd"></span>
```

Cod CSS:

```
.playlist-class {

    position: absolute;
    top: 0px;
    left: 0px;
    width: 960px;
    height: 696px;
    background-color: #ffffff;
    border-bottom: solid;
}
```

```
.playlist-text{
    position: absolute;
    top: 30px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;
}

.playlist-1 {
    position: absolute;
    top: 110px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;
}

.playlist-2 {
    position: absolute;
    top: 190px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;
}

.playlist-3 {
    position: absolute;
    top: 270px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;
}
```

```
.playlist-4 {
    position: absolute;
    top: 350px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;

}

.playlist-5 {
    position: absolute;
    top: 430px;
    left: 50px;
    font-weight:bold;
    font-size: 50px;
    color: #000000;
}
```

```
.sel_music {

    position: absolute;
    background-image: url("../image/sel_music.png");
    left: 0px;
    top: 110px;
    height: 50px;
    width: 50px;
}

.playing {

    position: absolute;
    background-image: url("../image/music.png");
    left: 650px;
    top: 110px;
    height: 50px;
    width: 50px;
}

.paused {

    position: absolute;
    background-image: url("../image/pause - Copy.png");
    left: 700px;
    top: 110px;
    height: 50px;
    width: 50px;
}
```

Secţiunea “Titlu”

Această secţiune este similară cu cea menţionată anterior. Aceasta este alcătuită din: un fundal de tip patrulater de culoare neagră şi mai multe elemente de tip text de culoare albă. Această secţiune are rolul de a informa utilizatorul în legătură cu numele aplicaţiei, numele dezvoltatorului şi anul în care aplicaţia a fost realizată. Aceasta s-a realizat prin intermediul unui element de tip “div” de clasă “proj”, în interiorul “container-ului”, şi mai multe elemente de tip “span” pentru fiecare text.

Figura 2.3: Secţiunea “Titlu”

Cod HTML:

```
<div class="proj">
    <span class="title-text" id="tlt-text">Proiect SMSCPFTM</span>
    <span class="apl-text" id="apl-text">Aplicaţie pe Smart TV</span>
    <span class="musicplayer-text" id="mp-text">~ Music Player ~</span>
    <span class="student-text" id="std-text">Student: Văduva Marian-Cătălin</span>
    <span class="group-text" id="grp-text">Grupa: 411-CMM</span>
    <span class="etti-text" id="etti">ETTI - 2018</span>
</div>
```

Cod CSS:

```css
.proj {

    position: absolute;
    left: 960px;
    top: 0px;
    height: 696px;
    width: 960px;
    border-bottom: solid;
    background-color: #000000;
}

.title-text{
    position: absolute;
    top: 30px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    left: 300px;
}

.apl-text{
    position: absolute;
    top: 110px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    left: 275px;
}

.musicplayer-text{
    position: absolute;
    top: 190px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    left: 325px;
}

.student-text{
    position: absolute;
    top: 430px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    right: 0px;
}

.group-text{
    position: absolute;
    top: 510px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    right: 0px;
}

.etti-text{
    position: absolute;
    bottom: 30px;
    font-weight:bold;
    font-size: 50px;
    color: #ffffff;
    left: 360px;
}
```

Fişierele audio au fost introduse în aplicaţie prin intermediul HTML. S-au creat elemente de tip “audio” de id-uri diferite pentru fiecare fişier. De asemenea, pentru simplitate, fişierele au fost redenumite şi introduse într-un director numit “audio”.

```html
<audio id='audio1'>
    <source src='audio/1.mp3'></source>
</audio>

<audio id='audio2'>
    <source src='audio/2.mp3'></source>
</audio>

<audio id='audio3'>
    <source src='audio/3.mp3'></source>
</audio>

<audio id='audio4'>
    <source src='audio/4.mp3'></source>
</audio>

<audio id='audio5'>
    <source src='audio/5.mp3'></source>
</audio>


 <script src="js/main.js"></script>
</body>
```

2.2 Funcţiile aplicaţiei

Funcţiile aplicaţiei sunt implementate în limbajul Javascript. Această aplicaţie poate fi controlată cu ajutorul telecomenzii. Butoanele folosite sunt: sânga, dreapta, sus, jos şi centru (secţiunea din chenarul roşu din figura de mai jos).

Figura 2.4: Butoanele folosite de la telecomandă pentru controlul funcţiilor aplicaţiei

Prin apăsarea butoanelor stânga/dreapta de la telecomandă se deplasează selectorul de culoare verde din Secţiunea Butoanelor “Play”, “Pause” etc. Astfel, utilizatorul va observa ce buton din acea secţiune va fi ulterior selectat (sau ce funcţie a aplicaţiei va fi apelată).

Cod Javascript:

```
//left and right buttons function

function move_selector(keyCode)
{
    var moveLeftLimit = 0,
    moveRightLimit = 1536,
    margin;

    var selectorStyle = window.getComputedStyle(selector, null),
    selectorMarginLeft = parseInt(selectorStyle.marginLeft.replace("px", ""));


    if (keyCode === LEFT_ARROW_BUTTON)
    {
        margin = selectorMarginLeft - 384;
        margin = margin < moveLeftLimit ? moveLeftLimit : margin;
        selector.style.marginLeft = margin + "px";
    }
     else if (keyCode === RIGHT_ARROW_BUTTON)
    {
        margin = selectorMarginLeft + 384;
        margin = margin > moveRightLimit ? moveRightLimit : margin;
        selector.style.marginLeft = margin + "px";
    }
}
```

Prin apăsarea butoanelor sus/jos se controlează selectorul de melodie/fişier audio din Secţiunea “Playlist” (săgeata pe fond gri). Astfel, utilizatorul va putea alege melodia preferată din listă ce se doreşte a fi ascultată.

Cod Javascript:

```
//up and down buttons function

function move_music_selector(keyCode)
{
    var moveUpLimit = 0,
    moveDownLimit = 320,
    margin;

    var selmusicStyle = window.getComputedStyle(music_sel, null),
    music_selMarginTop = parseInt(selmusicStyle.marginTop.replace("px", ""));

    if (keyCode === UP_ARROW_BUTTON)
    {
        margin = music_selMarginTop - 80;
        margin = margin < moveUpLimit ? moveUpLimit : margin;
        music_sel.style.marginTop = margin + "px";
    }
     else if (keyCode === DOWN_ARROW_BUTTON)
    {
        margin = music_selMarginTop + 80;
        margin = margin > moveDownLimit ? moveDownLimit : margin;
        music_sel.style.marginTop = margin + "px";
    }
}
```

Butonul din centru al telecomenzii are mai multe roluri. Acesta apelează funcţiile butoanelor de “Play”, “Pause” etc. Funcţia apelată la apăsarea butonului este “center(keyCode)”, prezentată în anexă în codul sursă Javascript.

În cazul în care se selectează butonul “Play”, va începe să ruleze melodia aleasă de utilizator cu ajutorul săgeţii de pe Playlist.

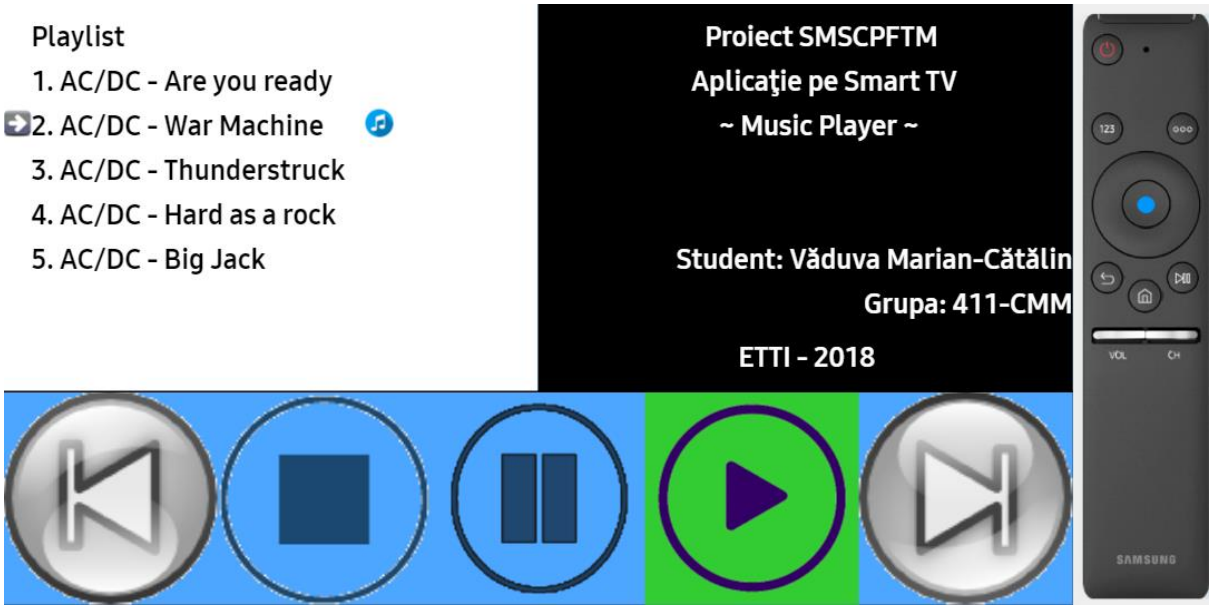

Figura 2.5: Butonul “Play” - Exemplu de rulare a unei melodii

Cod Javascript (exemplu pentru primele două melodii, restul codului fiind în anexă):

```
//play button

if (selectorMarginLeft === 1152 && music_selMarginTop === 0)
    {
    playing_icon.classList.add("playing");
    playing_icon.style.marginTop = 0;
    audioElement1.play();

    audioElement2.addEventListener('playing', audioElement2.load());
    audioElement3.addEventListener('playing', audioElement3.load());
    audioElement4.addEventListener('playing', audioElement4.load());
    audioElement5.addEventListener('playing', audioElement5.load());

    audioElement2.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement3.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement4.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement5.addEventListener('paused',  paused_icon.classList.remove("paused"));
    }

else if (selectorMarginLeft === 1152 && music_selMarginTop === 80)
    {
    playing_icon.classList.add("playing");
    playing_icon.style.marginTop = 80 + "px";
    audioElement2.play();

    audioElement1.addEventListener('playing', audioElement1.load());
    audioElement3.addEventListener('playing', audioElement3.load());
    audioElement4.addEventListener('playing', audioElement4.load());
    audioElement5.addEventListener('playing', audioElement5.load());

    audioElement1.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement3.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement4.addEventListener('paused',  paused_icon.classList.remove("paused"));
    audioElement5.addEventListener('paused',  paused_icon.classList.remove("paused"));
    }
```

În cazul în care se selectează butonul “Pause”, melodia în curs de rulare va intra în repaus. Dacă se doreşte rularea în continuare a melodiei în repaus, se apasă butonul “Play”, săgeata din Playlist fiind în dreptul melodiei în repaus.

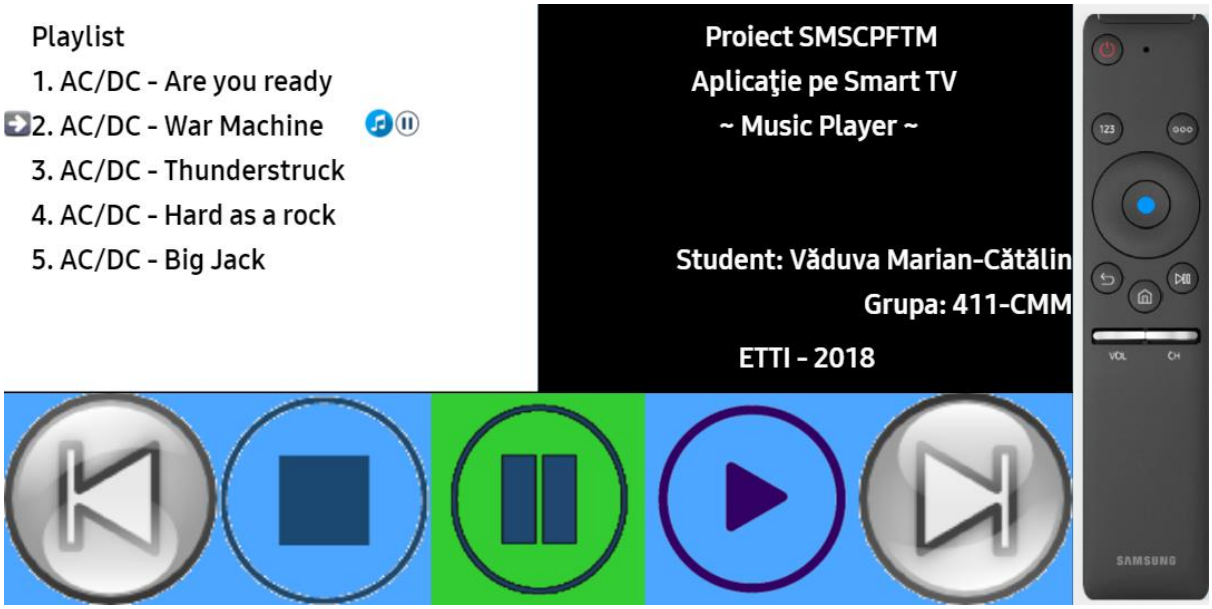

Figura 2.6: Butonul “Pause” - Exemplu de punere în repaus a unei melodii

Cod Javascript (exemplu pentru primele două melodii, restul codului fiind în anexă):

```
//pause button

else if (selectorMarginLeft === 768 && playing_icon.style.marginTop === 0 + "px")
    {
    audioElement1.addEventListener('playing', audioElement1.pause());
    audioElement1.addEventListener('playing', paused_icon.classList.add("paused"));
    audioElement1.addEventListener('playing', paused_icon.style.marginTop = 0 + "px");
    }

else if (selectorMarginLeft === 768 && playing_icon.style.marginTop === 80 + "px")
    {
    audioElement2.addEventListener('playing', audioElement2.pause());
    audioElement2.addEventListener('playing', paused_icon.classList.add("paused"));
    audioElement2.addEventListener('playing', paused_icon.style.marginTop = 80 + "px");
    }
```

În cazul în care se selectează butonul “Stop”, melodia în curs de rulare se va opri şi reseta. Dacă există o melodie deja în modul de repaus şi se apasă pe “Stop”, acea melodie, de asemenea, se va opri şi reseta.

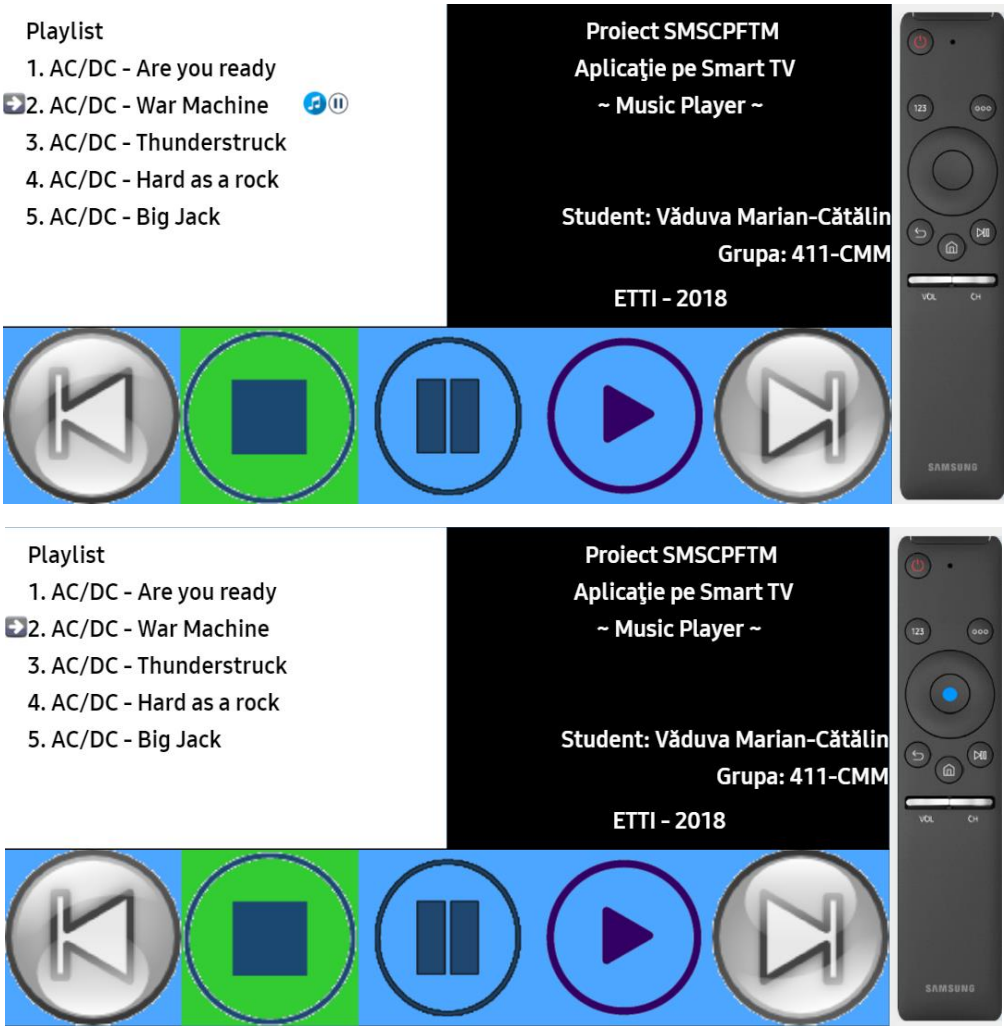

Figura 2.7: Butonul “Stop” - Exemplu de oprire şi resetare a unei melodii

Cod Javascript:

```
//stop button

else if (selectorMarginLeft === 384)
    {
    audioElement1.load();
    audioElement2.load();
    audioElement3.load();
    audioElement4.load();
    audioElement5.load();
    playing_icon.classList.remove("playing");
    paused_icon.classList.remove("paused");
    }
```

La selectarea butonului “Next” melodia în curs de rulare se va opri şi va rula melodia următoare din listă. În cazul în care melodia ce rulează este ultima din Playlist şi se apasă butonul “Next”, următoarea melodie ce va rula va fi prima din listă.

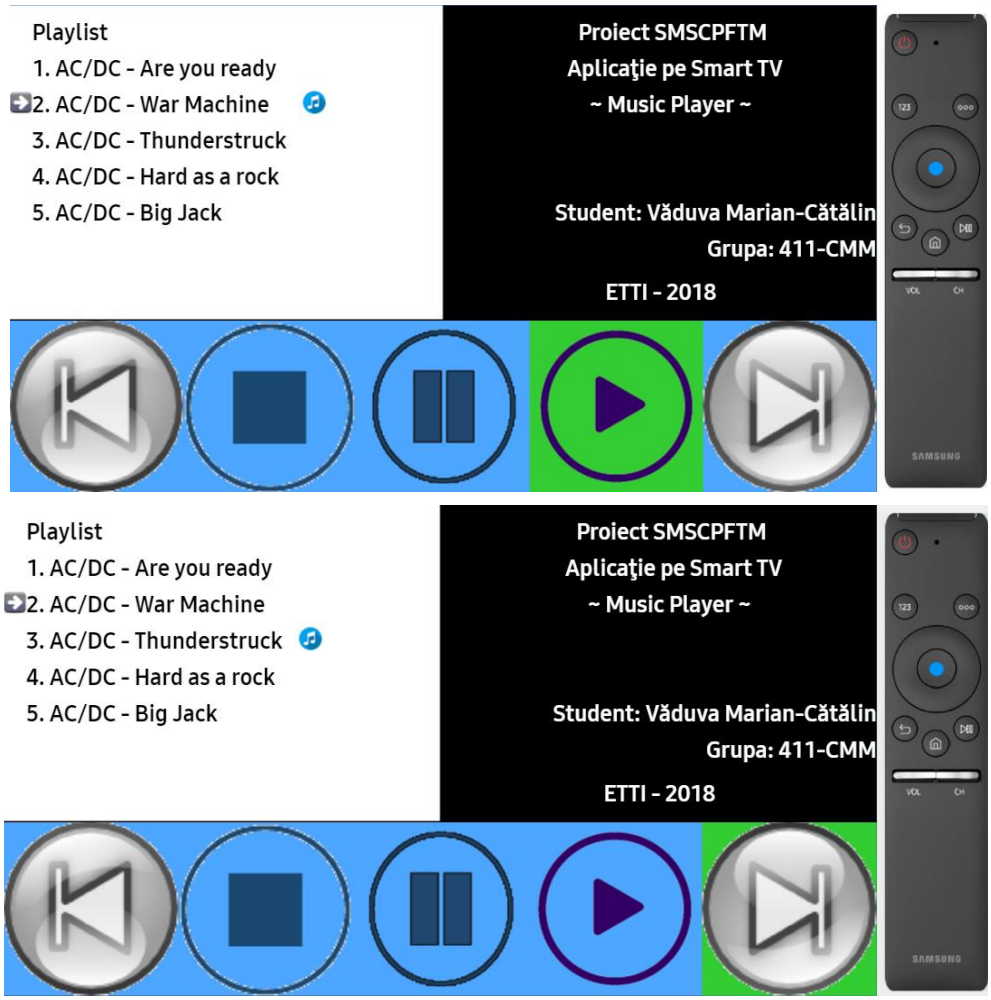

Figura 2.8: Butonul “Next” - Exemplu de rulare a următoarei melodii din listă

Cod Javascript (exemplu pentru primele două melodii, restul codului fiind în anexă):

```
//next button

else if (selectorMarginLeft === 1536 && playing_icon.style.marginTop === 0 + "px")
    {
    audioElement1.addEventListener('playing', audioElement1.load());
    audioElement1.addEventListener('playing', paused_icon.classList.remove("paused"));
    audioElement1.addEventListener('playing', playing_icon.classList.add("playing"));
    audioElement1.addEventListener('playing', playing_icon.style.marginTop = 80 + "px");
    audioElement1.addEventListener('playing', audioElement2.play());
    }

else if (selectorMarginLeft === 1536 && playing_icon.style.marginTop === 80 + "px" )
    {
    audioElement2.addEventListener('playing', audioElement2.load());
    audioElement2.addEventListener('playing', paused_icon.classList.remove("paused"));
    audioElement2.addEventListener('playing', playing_icon.classList.add("playing"));
    audioElement2.addEventListener('playing', playing_icon.style.marginTop = 160 + "px");
    audioElement2.addEventListener('playing', audioElement3.play());
    }
```

Butonul “Previous” are o funcţie similară cu cea a butonului “Next” doar că sensul este inversat. La apăsarea acestuia, melodia în curs de rulare se va opri şi va rula melodia precedentă din listă. De asemenea, în cazul în care melodia ce rulează este prima din listă şi se apasă butonul “Previous”, melodia ce va rula va fi ultima din listă.

Figura 2.9: Butonul “Previous” - Exemplu de rulare a melodiei precedente din listă

Cod Javascript (exemplu pentru primele două melodii, restul codului fiind în anexă):

```
//previous button

  else if (selectorMarginLeft === 0 && playing_icon.style.marginTop === 0 + "px")
      {
      audioElement1.addEventListener('playing', audioElement1.load());
      audioElement1.addEventListener('playing', paused_icon.classList.remove("paused"));
      audioElement1.addEventListener('playing', playing_icon.classList.add("playing"));
      audioElement1.addEventListener('playing', playing_icon.style.marginTop = 320 + "px");
      audioElement1.addEventListener('playing', audioElement5.play());
      }

  else if (selectorMarginLeft === 0 && playing_icon.style.marginTop === 80 + "px")
      {
      audioElement2.addEventListener('playing', audioElement2.load());
      audioElement2.addEventListener('playing', paused_icon.classList.remove("paused"));
      audioElement2.addEventListener('playing', playing_icon.classList.add("playing"));
      audioElement2.addEventListener('playing', playing_icon.style.marginTop = 0 + "px");
      audioElement2.addEventListener('playing', audioElement1.play());
      }
```

Această aplicaţie mai dispune de încă o funcţie interesantă, prezentă în marea majoritate a aplicaţiilor de tip Music Player. Această funcţie face ca la terminarea unei melodii să se redea următoarea melodie din Playlist. De asemenea, dacă melodia care se finalizează este ultima din listă, următoarea melodie care va rula va fi prima din listă.

Cod Javascript (exemplu pentru primele două melodii, restul codului fiind în anexă):

```
//end of each track

audioElement1.onended = function()
{
    audioElement1.load();
    playing_icon.classList.add("playing");
    playing_icon.style.marginTop = 80 + "px";
    audioElement2.play();
};

audioElement2.onended = function()
{
    audioElement2.load();
    playing_icon.classList.add("playing");
    playing_icon.style.marginTop = 160 + "px";
    audioElement3.play();
};
```

Aici este un link către o demonstraţie practică a aplicaţiei:

https://www.youtube.com/watch?v=TRjGxrBPAX8

3. Concluzii

Această aplicaţie de tip Music Player pentru SMART TV Samsung conţine funcţiile de bază ce le au majoritatea aplicaţiilor de acest tip. De asemenea, aceasta poate fi ulterior dezvoltată din orice punct de vedere (design, funcţii etc.).

Pentru design-ul aplicaţiei se pot realiza îmbunătăţiri ce pot ieşi uşor în evidenţă. De exemplu, se pot adăuga animaţii ce pot arăta faptul că un fişier audio este în curs de rulare sau se poate adăuga o bară de "seek" ce arată progresul fişierului audio în timpul rulării. De asemenea, se pot adăuga imagini cu artiştii prezenţi în Playlist şi acestea să apară pe ecran în momentul în care se apasă pe "Play".

Există multe funcţii ce se pot implementa pe această aplicaţie pentru a o îmbunătăţi. De exemplu, se poate implementa funcţia de "Repeat" ce are rolul de a reda un fişier audio, seletat de utilizator, în mod continuu (imediat ce se termină, se redă acelaşi fişier audio). De asemenea, se poate adăuga un fader care să aibă rolul de a varia volumul melodiei. Se poate implementa o secţiune numită "Equalizer", alcătuită din trei sau mai multe fader-e, ce are rolul de a introduce efecte audio melodiilor în funcţie de preferinţele utilizatorului (de exemplu: bass profund). Desigur, se pot implementa şi butoane de "Shuffle" sau de "Fast-forward", funcţii ce sunt prezente în majoritatea aplicaţiilor de tip Music Player.

Bibliografie

1) https://ro.wikipedia.org/wiki/SmartTV
2) https://developer.samsung.com/tv/develop/tools/tizen-studio/
3) https://www.w3schools.com/
4) https://developer.samsung.com/tv/develop/guides/multimedia/media-playback/using-audio-elements
5) https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement#Properties
6) https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Media_events
7) https://developer.samsung.com/tv/develop/legacy-platform-library/2014/06_media_player

Cod sursă HTML:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=1.0">
    <meta name="description" content="Tizen TV basic template generated by Samsung
TV Web IDE"/>

    <title>Proiect SMSCPFTM - Music Player</title>

    <link rel="stylesheet" type="text/css" href="css/style.css"/>

</head>

<body>
  <div class="container">



            <span class="sel_item" id="sel"></span>
            <span class="btn" id="prev"></span>
            <span class="btn" id="next"></span>
            <span class="btn" id="play"></span>
            <span class="btn" id="stop"></span>
            <span class="btn" id="pause"></span>


            <span class="playlist-class" id="playlist"></span>
            <span class="playlist-text" id="text">Playlist</span>
            <span class="playlist-1" id="1">1. AC/DC - Are you ready</span>
            <span class="playlist-2" id="2">2. AC/DC - War Machine</span>
            <span class="playlist-3" id="3">3. AC/DC - Thunderstruck</span>
            <span class="playlist-4" id="4">4. AC/DC - Hard as a rock</span>
            <span class="playlist-5" id="5">5. AC/DC - Big Jack</span>
            <span class="sel_music" id="sel_m"></span>
            <span class="playing" id="plying"></span>
            <span class="paused" id="pd"></span>



      <div class="proj">
            <span class="title-text" id="tlt-text">Proiect SMSCPFTM</span>
            <span class="apl-text" id="apl-text">Aplicaţie pe Smart TV</span>
            <span class="musicplayer-text" id="mp-text">~ Music Player ~</span>
            <span class="student-text" id="std-text">Student: Văduva Marian-
Cătălin</span>
            <span class="group-text" id="grp-text">Grupa: 411-CMM</span>
            <span class="etti-text" id="etti">ETTI - 2018</span>
      </div>



  </div>


<audio id='audio1'>
      <source src='audio/1.mp3'></source>
</audio>
```

```
<audio id='audio2'>
       <source src='audio/2.mp3'></source>
</audio>

<audio id='audio3'>
       <source src='audio/3.mp3'></source>
</audio>

<audio id='audio4'>
       <source src='audio/4.mp3'></source>
</audio>

<audio id='audio5'>
       <source src='audio/5.mp3'></source>
</audio>


 <script src="js/main.js"></script>
</body>
</html>
```

Cod sursă CSS:

```css
body {
    margin: 0px auto;
    background-color:#222;
}

.container {

       position: absolute;
       top: 0px;
       height: 1080px;
    width: 1920px;
    background-color: #4da6ff;
}



.playlist-class {

       position: absolute;
       top: 0px;
       left: 0px;
       width: 960px;
       height: 696px;
       background-color: #ffffff;
       border-bottom: solid;
}


.btn {

       position: absolute;
       bottom: 0px;
       height: 384px;
       width: 384px;
}

.sel_item {

       position: absolute;
       bottom: 0px;
       height: 384px;
       width: 384px;
       left: 0px;
       background-color: #33cc33;

}


#prev{
       background-image: url("../image/prev.png");
       left: 0px;

}

#next{
       background-image: url("../image/next.png");
       left: 1536px;

}
```

```
#play{
	background-image: url("../image/play2.png");
	left: 1152px;

}

#stop{
	background-image: url("../image/stop.png");
	left: 384px;

}

#pause{
	background-image: url("../image/pause.png");
	left: 768px;
}




.playlist-text{
	position: absolute;
	top: 30px;
	left: 50px;
	font-weight:bold;
	font-size: 50px;
	color: #000000;
}


.playlist-1 {
	position: absolute;
	top: 110px;
	left: 50px;
	font-weight:bold;
	font-size: 50px;
	color: #000000;
}


.playlist-2 {
	position: absolute;
	top: 190px;
	left: 50px;
	font-weight:bold;
	font-size: 50px;
	color: #000000;
}


.playlist-3 {
	position: absolute;
	top: 270px;
	left: 50px;
	font-weight:bold;
	font-size: 50px;
	color: #000000;
}

.playlist-4 {
```

```
        position: absolute;
        top: 350px;
        left: 50px;
        font-weight:bold;
        font-size: 50px;
        color: #000000;

}

.playlist-5 {
        position: absolute;
        top: 430px;
        left: 50px;
        font-weight:bold;
        font-size: 50px;
        color: #000000;
}


.sel_music {

        position: absolute;
        background-image: url("../image/sel_music.png");
        left: 0px;
        top: 110px;
        height: 50px;
        width: 50px;
}


.playing {

        position: absolute;
        background-image: url("../image/music.png");
        left: 650px;
        top: 110px;
        height: 50px;
        width: 50px;
}

.paused {

        position: absolute;
        background-image: url("../image/pause - Copy.png");
        left: 700px;
        top: 110px;
        height: 50px;
        width: 50px;
}




.proj {

        position: absolute;
        left: 960px;
        top: 0px;
        height: 696px;
        width: 960px;
```

```
        border-bottom: solid;
        background-color: #000000;
}

.title-text{
        position: absolute;
        top: 30px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        left: 300px;
}

.apl-text{
        position: absolute;
        top: 110px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        left: 275px;
}

.musicplayer-text{
        position: absolute;
        top: 190px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        left: 325px;
}


.student-text{
        position: absolute;
        top: 430px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        right: 0px;
}

.group-text{
        position: absolute;
        top: 510px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        right: 0px;
}

.etti-text{
        position: absolute;
        bottom: 30px;
        font-weight:bold;
        font-size: 50px;
        color: #ffffff;
        left: 360px;
}
```

Cod sursă Javascript:

```
(function()  {

	var RETURN_BUTTON = 10009,
		CENTER_BUTTON = 13,
		LEFT_ARROW_BUTTON = 37,
		UP_ARROW_BUTTON = 38,
		RIGHT_ARROW_BUTTON = 39,
		DOWN_ARROW_BUTTON = 40,

		selector = document.querySelector(".sel_item"),
		music_sel= document.querySelector(".sel_music");

	 var playing_icon = document.querySelector(".playing"),
	 paused_icon = document.querySelector(".paused");


	 playing_icon.classList.remove("playing");		//remove playing icon
	 paused_icon.classList.remove("paused");	//remove paused icon


	var audioElement1 = document.getElementById('audio1');   // song 1
	var audioElement2 = document.getElementById('audio2');   // song 2
	var audioElement3 = document.getElementById('audio3');   // song 3
	var audioElement4 = document.getElementById('audio4');   // song 4
	var audioElement5 = document.getElementById('audio5');   // song 5



	audioElement1.load();
	audioElement2.load();
	audioElement3.load();
	audioElement4.load();
	audioElement5.load();




	//left and right buttons function

	function move_selector(keyCode)
	{
		var moveLeftLimit = 0,
	  moveRightLimit = 1536,
	  margin;

		var selectorStyle = window.getComputedStyle(selector, null),
		selectorMarginLeft = parseInt(selectorStyle.marginLeft.replace("px",
""));

	   if (keyCode === LEFT_ARROW_BUTTON)
	   {
		margin = selectorMarginLeft - 384;
		margin = margin < moveLeftLimit ? moveLeftLimit : margin;
		selector.style.marginLeft = margin + "px";
	  }
	   else if (keyCode === RIGHT_ARROW_BUTTON)
	  {
		margin = selectorMarginLeft + 384;
		margin = margin > moveRightLimit ? moveRightLimit : margin;
```

```
            selector.style.marginLeft = margin + "px";
        }
       }




       //up and down buttons function

       function move_music_selector(keyCode)
       {
             var moveUpLimit = 0,
        moveDownLimit = 320,
        margin;

             var selmusicStyle = window.getComputedStyle(music_sel, null),
             music_selMarginTop = parseInt(selmusicStyle.marginTop.replace("px",
""));

           if (keyCode === UP_ARROW_BUTTON)
           {
            margin = music_selMarginTop - 80;
            margin = margin < moveUpLimit ? moveUpLimit : margin;
            music_sel.style.marginTop = margin + "px";
        }
         else if (keyCode === DOWN_ARROW_BUTTON)
        {
            margin = music_selMarginTop + 80;
            margin = margin > moveDownLimit ? moveDownLimit : margin;
            music_sel.style.marginTop = margin + "px";
        }
       }



       //center button function


       function center(keyCode)
       {
             var selectorStyle = window.getComputedStyle(selector, null),
                   selectorMarginLeft =
parseInt(selectorStyle.marginLeft.replace("px", ""));

             var selmusicStyle = window.getComputedStyle(music_sel, null),
             music_selMarginTop = parseInt(selmusicStyle.marginTop.replace("px",
""));

              if (keyCode === CENTER_BUTTON)
                 {

                         //play button

                  if (selectorMarginLeft === 1152 && music_selMarginTop === 0)
                   {
                   playing_icon.classList.add("playing");
                   playing_icon.style.marginTop = 0;
                   audioElement1.play();

                   audioElement2.addEventListener('playing',
audioElement2.load());
```

```
                    audioElement3.addEventListener('playing',
audioElement3.load());
                    audioElement4.addEventListener('playing',
audioElement4.load());
                    audioElement5.addEventListener('playing',
audioElement5.load());

                    audioElement2.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    }

                  else if (selectorMarginLeft === 1152 && music_selMarginTop ===
80)
                    {
                    playing_icon.classList.add("playing");
                    playing_icon.style.marginTop = 80 + "px";
                           audioElement2.play();

                           audioElement1.addEventListener('playing',
audioElement1.load());
                    audioElement3.addEventListener('playing',
audioElement3.load());
                    audioElement4.addEventListener('playing',
audioElement4.load());
                    audioElement5.addEventListener('playing',
audioElement5.load());

                    audioElement1.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    }

                  else if (selectorMarginLeft === 1152 && music_selMarginTop ===
160)
                    {
                    playing_icon.classList.add("playing");
                    playing_icon.style.marginTop = 160 + "px";
                           audioElement3.play();

                           audioElement1.addEventListener('playing',
audioElement1.load());
                    audioElement2.addEventListener('playing',
audioElement2.load());
                    audioElement4.addEventListener('playing',
audioElement4.load());
                    audioElement5.addEventListener('playing',
audioElement5.load());

                    audioElement1.addEventListener('paused',
paused_icon.classList.remove("paused"));
```

```
                    audioElement2.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    }

                  else if (selectorMarginLeft === 1152 && music_selMarginTop ===
240)
                    {
                    playing_icon.classList.add("playing");
                    playing_icon.style.marginTop = 240 + "px";
                           audioElement4.play();

                           audioElement1.addEventListener('playing',
audioElement1.load());
                    audioElement2.addEventListener('playing',
audioElement2.load());
                    audioElement3.addEventListener('playing',
audioElement3.load());
                    audioElement5.addEventListener('playing',
audioElement5.load());

                    audioElement1.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement2.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    }

                  else if (selectorMarginLeft === 1152 && music_selMarginTop ===
320)
                    {
                    playing_icon.classList.add("playing");
                    playing_icon.style.marginTop = 320 + "px";
                           audioElement5.play();

                           audioElement1.addEventListener('playing',
audioElement1.load());
                    audioElement2.addEventListener('playing',
audioElement2.load());
                    audioElement3.addEventListener('playing',
audioElement3.load());
                    audioElement4.addEventListener('playing',
audioElement4.load());

                    audioElement1.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement2.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('paused',
paused_icon.classList.remove("paused"));
                    }
```

```
                    //pause button

                    else if (selectorMarginLeft === 768 &&
playing_icon.style.marginTop === 0 + "px")
                     {
                     audioElement1.addEventListener('playing',
audioElement1.pause());
                     audioElement1.addEventListener('playing',
paused_icon.classList.add("paused"));
                     audioElement1.addEventListener('playing',
paused_icon.style.marginTop = 0 + "px");
                     }

                    else if (selectorMarginLeft === 768 &&
playing_icon.style.marginTop === 80 + "px")
                     {
                     audioElement2.addEventListener('playing',
audioElement2.pause());
                     audioElement2.addEventListener('playing',
paused_icon.classList.add("paused"));
                     audioElement2.addEventListener('playing',
paused_icon.style.marginTop = 80 + "px");
                     }


                    else if (selectorMarginLeft === 768 &&
playing_icon.style.marginTop === 160 + "px")
                     {
                     audioElement3.addEventListener('playing',
audioElement3.pause());
                     audioElement3.addEventListener('playing',
paused_icon.classList.add("paused"));
                     audioElement3.addEventListener('playing',
paused_icon.style.marginTop = 160 + "px");
                     }

                    else if (selectorMarginLeft === 768 &&
playing_icon.style.marginTop === 240 + "px")
                     {
                     audioElement4.addEventListener('playing',
audioElement4.pause());
                     audioElement4.addEventListener('playing',
paused_icon.classList.add("paused"));
                     audioElement4.addEventListener('playing',
paused_icon.style.marginTop = 240 + "px");
                     }

                    else if (selectorMarginLeft === 768 &&
playing_icon.style.marginTop === 320 + "px")
                     {
                     audioElement5.addEventListener('playing',
audioElement5.pause());
                     audioElement5.addEventListener('playing',
paused_icon.classList.add("paused"));
                     audioElement5.addEventListener('playing',
paused_icon.style.marginTop = 320 + "px");
                     }
```

```
                    //stop button

                    else if (selectorMarginLeft === 384)
                     {
                     audioElement1.load();
                     audioElement2.load();
                     audioElement3.load();
                     audioElement4.load();
                     audioElement5.load();
                     playing_icon.classList.remove("playing");
                     paused_icon.classList.remove("paused");
                     }




                    //next button

                    else if (selectorMarginLeft === 1536 &&
playing_icon.style.marginTop === 0 + "px")
                     {
                     audioElement1.addEventListener('playing',
audioElement1.load());
                     audioElement1.addEventListener('playing',
paused_icon.classList.remove("paused"));
                     audioElement1.addEventListener('playing',
playing_icon.classList.add("playing"));
                     audioElement1.addEventListener('playing',
playing_icon.style.marginTop = 80 + "px");
                     audioElement1.addEventListener('playing',
audioElement2.play());
                     }

                    else if (selectorMarginLeft === 1536 &&
playing_icon.style.marginTop === 80 + "px" )
                     {
                     audioElement2.addEventListener('playing',
audioElement2.load());
                     audioElement2.addEventListener('playing',
paused_icon.classList.remove("paused"));
                     audioElement2.addEventListener('playing',
playing_icon.classList.add("playing"));
                     audioElement2.addEventListener('playing',
playing_icon.style.marginTop = 160 + "px");
                     audioElement2.addEventListener('playing',
audioElement3.play());
                     }

                    else if (selectorMarginLeft === 1536 &&
playing_icon.style.marginTop === 160 + "px" )
                     {
                     audioElement3.addEventListener('playing',
audioElement3.load());
```

```
                    audioElement3.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement3.addEventListener('playing',
playing_icon.style.marginTop = 240 + "px");
                    audioElement3.addEventListener('playing',
audioElement4.play());
                    }

                  else if (selectorMarginLeft === 1536 &&
playing_icon.style.marginTop === 240 + "px" )
                    {
                    audioElement4.addEventListener('playing',
audioElement4.load());
                    audioElement4.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement4.addEventListener('playing',
playing_icon.style.marginTop = 320 + "px");
                    audioElement4.addEventListener('playing',
audioElement5.play());
                    }

                  else if (selectorMarginLeft === 1536 &&
playing_icon.style.marginTop === 320 + "px" )
                          {
                    audioElement5.addEventListener('playing',
audioElement5.load());
                    audioElement5.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement5.addEventListener('playing',
playing_icon.style.marginTop = 0 + "px");
                    audioElement5.addEventListener('playing',
audioElement1.play());
                          }









                //previous button

                  else if (selectorMarginLeft === 0 &&
playing_icon.style.marginTop === 0 + "px")
                    {
                    audioElement1.addEventListener('playing',
audioElement1.load());
                    audioElement1.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement1.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement1.addEventListener('playing',
playing_icon.style.marginTop = 320 + "px");
                    audioElement1.addEventListener('playing',
audioElement5.play());
```

```
                    }

                else if (selectorMarginLeft === 0 &&
playing_icon.style.marginTop === 80 + "px")
                    {
                    audioElement2.addEventListener('playing',
audioElement2.load());
                    audioElement2.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement2.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement2.addEventListener('playing',
playing_icon.style.marginTop = 0 + "px");
                    audioElement2.addEventListener('playing',
audioElement1.play());
                    }

                else if (selectorMarginLeft === 0 &&
playing_icon.style.marginTop === 160 + "px")
                    {
                    audioElement3.addEventListener('playing',
audioElement3.load());
                    audioElement3.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement3.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement3.addEventListener('playing',
playing_icon.style.marginTop = 80 + "px");
                    audioElement3.addEventListener('playing',
audioElement2.play());
                    }

                else if (selectorMarginLeft === 0 &&
playing_icon.style.marginTop === 240 + "px")
                    {
                    audioElement4.addEventListener('playing',
audioElement4.load());
                    audioElement4.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement4.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement4.addEventListener('playing',
playing_icon.style.marginTop = 160 + "px");
                    audioElement4.addEventListener('playing',
audioElement3.play());
                    }

                else if (selectorMarginLeft === 0 &&
playing_icon.style.marginTop === 320 + "px")
                    {
                    audioElement5.addEventListener('playing',
audioElement5.load());
                    audioElement5.addEventListener('playing',
paused_icon.classList.remove("paused"));
                    audioElement5.addEventListener('playing',
playing_icon.classList.add("playing"));
                    audioElement5.addEventListener('playing',
playing_icon.style.marginTop = 240 + "px");
                    audioElement5.addEventListener('playing',
audioElement4.play());
                    }
            }
```

```
}


//end of each track

audioElement1.onended = function()
{
	audioElement1.load();
	playing_icon.classList.add("playing");
	playing_icon.style.marginTop = 80 + "px";
	audioElement2.play();
};

audioElement2.onended = function()
{
	audioElement2.load();
	playing_icon.classList.add("playing");
	playing_icon.style.marginTop = 160 + "px";
	audioElement3.play();
};

audioElement3.onended = function()
{
	audioElement3.load();
	playing_icon.classList.add("playing");
	playing_icon.style.marginTop = 240 + "px";
	audioElement4.play();
};

audioElement4.onended = function()
{
	audioElement4.load();
	playing_icon.classList.add("playing");
	playing_icon.style.marginTop = 320 + "px";
	audioElement5.play();
};

audioElement5.onended = function()
{
	audioElement5.load();
	playing_icon.classList.add("playing");
	playing_icon.style.marginTop = 0 + "px";
	audioElement1.play();
};







//keyEventHandler

   function keyEventHandler(e)
   {
        if (e.keyCode === RETURN_BUTTON)
        {
            tizen.application.getCurrentApplication().exit();
        }

         else if (e.keyCode === LEFT_ARROW_BUTTON)
```

```
            {
                move_selector(e.keyCode);
            }

             else if (e.keyCode === RIGHT_ARROW_BUTTON)
            {
                  move_selector(e.keyCode);
              }

             else if (e.keyCode === CENTER_BUTTON)
            {
                  center(e.keyCode);
              }

             else if (e.keyCode === UP_ARROW_BUTTON)
            {
                  move_music_selector(e.keyCode);
              }

             else if (e.keyCode === DOWN_ARROW_BUTTON)
            {
                  move_music_selector(e.keyCode);
              }

             else
            {
                tizen.application.getCurrentApplication().exit();
              }
        }


    //bindDefaultEvents

     function bindDefaultEvents()
     {
         document.addEventListener('keydown', keyEventHandler);
     }


           /**
         * Initiates the application.
         * @private
         */

     function init()
     {
           bindDefaultEvents();
     }

        window.onload = init;

}());
```