



Madara Uchia Edo Tensei

(Infinity Runner Game)

Készítette:

- Bencze Dániel
- Fekete Dominik
- Sörös Bence

Ötletünk eredete

Az ötlet az animációs sorozat egyik epizódjából jött (Naruto Shippuuden #322), ahol Madarát feltámasztották és visszatért a jelenbe.

Egy egész shinobi hadsereg várja érkezését, ahol egymaga néz velük szembe.

Az alap játékmenet – Határtalan futójáték.

Madarát egyszer sikeresen elfogják, viszont ez csak rajtunk múlik, hogy mikor.

- Shurikeneket kell kikerülnünk,
- Sushit kell gyűjtenünk,
- Susano-vá kell változnunk.



Fejlesztés menete

GITHUB – használata.

- Külön Branch-ek használata
- 1 Master Branch

Feladatok felosztása:

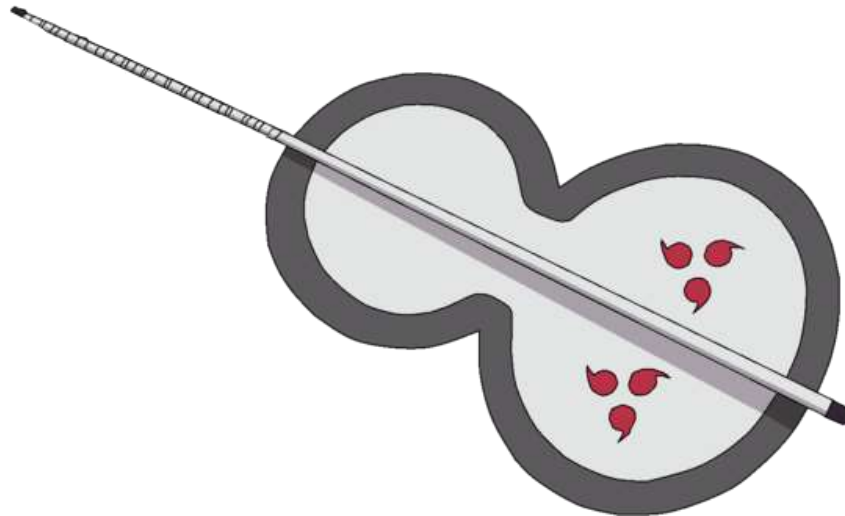
- Közös tervek
- Megegyezés alapján osztottuk szét őket
- Szempont volt, hogy ki miben jeleskedett a félév során
 - Logic? – UI?
- Órán többszöri megbeszélést tartottunk – Majd később mindenki külön dolgozott, merge előtt egy meetinget tartottunk Discord segítségével.



Fejlesztés menete

Nehézségek:

1. Oldalak közötti navigálás – (Content átadás – Hangok kikapcsolása)
2. Gombok design megalkotása – IsMouseOver etc. – Hogyan lehet megváltoztatni 1 beépített Controllert.
3. Logic megalkotása, logika felépítése. Papíron, kommentek segítségével valósult meg.



Oldalak közötti navigálás – (Content átadás – Hangok kikapcsolása)

- Működési alapelv
- Mire kell odafigyelni?

```
1 reference
private void Window_KeyDown(object sender, KeyEventArgs e)
{
    if (Gm != null)
    {
        Gm.Page_KeyDown(sender, e);
    }
}
```

```
Game Gm;
private object content;
public double volume;
0 references
public MainWindow()
{
    InitializeComponent();
    content = Content;
}

1 reference
public void ChangeToGame()
{
    Gm = new Game(this);
    this.Content = Gm;
}
```

Gombok design megalkotása

- Megkellett oldanunk, hogy ne legyen szürke háttere a gombnak, ha az egeret rávisszük.
- Ehhez új Resource-okat hoztunk létre, a meglévő gombokhoz.

- Ezek a .xaml-ben találhatóak. Régi hátteret felülírtuk 1 saját, beimportálttal.

(Nehézsége onnan ered, hogy megtaláljuk, hogy lehetséges-e ilyen és ennek alkalmazása.)

```
</Trigger>
<Trigger Property="IsMouseOver" Value="true">
    <Setter Property="Background" TargetName="border" Value="{StaticResource MenuStart }"/>
    <Setter Property="BorderBrush" TargetName="border" Value="■"{StaticResource Button.MouseOver.Border}"/>
</Trigger>
<Trigger Property="IsPressed" Value="true">
    <Setter Property="Background" TargetName="border" Value="{StaticResource MenuWaiting}"/>
    <Setter Property="BorderBrush" TargetName="border" Value="■"{StaticResource Button.Pressed.Border}"/>
</Trigger>
```



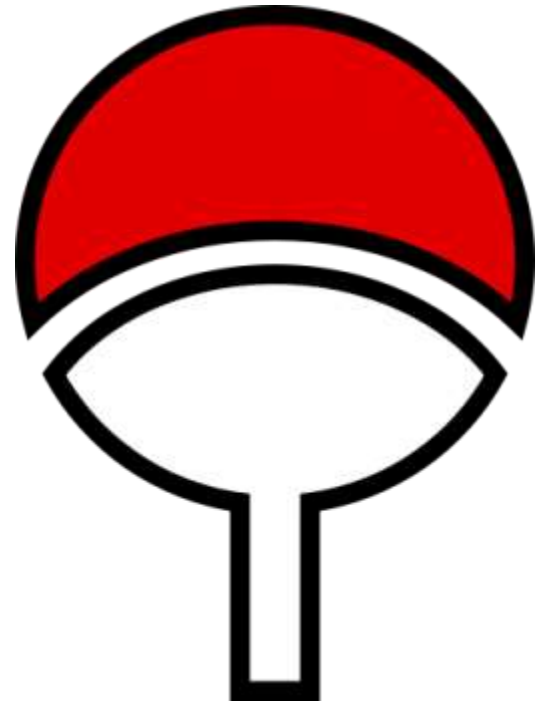
Logic megalkotása, logika felépítése.

Összesen 1 darab logic létezik a projektünkben.

- Kezeli a gombok lenyomását
- Játékmenetet
- Grafikát

Játékunkat 1 db „mesterséges intelligencia” kezeli.

- Timer – Dispatcher melynek segítségével haladunk a játékban előre
- Jönnek az enemyk
- Megjelennek a bónuszok
- Átváltozhatunk
- Zene, videó, egyéb dolgokat tud adni nekünk



GAME PRESENTATION



Köszönjük a megtisztelő figyelmeteket!

