

2018

Apple Runner

BEN NICHOLAY GYLLENHAAL JOHANSEN
ERIK JACOBSEN
SANDER SJØTHUN

WESTERDALS OSLO SCHOOL OF ARTS, COMMUNICATION AND TECHNOLOGY | Chr. Krohgs gate 32,
0186 Oslo

Innholdsfortegnelse

Proessen.....	2
Liste - Features	3
Liste - Scripts.....	3
Spesielle utfordringer	4
Fancy features – Ting gruppen er spesielt fornøyde med.....	5
Referanser	6

Proessen

Gruppen bestemte seg først for å velge oppgave B: Helse App, og startet med en brainstorm etter idéer for ulike funksjoner vi ville implementere i applikasjonen vår. Deretter lagde vi en enkel skisse, og skrev ned hva vi kunne tenke oss å ha med av funksjonalitet og utseende. Noen av gruppemedlemmene startet på proof-of-concept, mens andre begynte med skisse og mock-up. Vi benyttet oss av Autodesk's Character Generator for å lage figuren vår, deretter lastet vi den inn i Maya for å eksportere den til riktig filformat slik at Unity kunne gjenkjenne den. Moqups og InVision sammen med Sketch ble benyttet for mock-up. Det var ikke før vi var halvveis i prosjektet at vi fant ut at vi ikke ville klare å gjennomføre oppgave B. Gruppen hadde alt snakket om å lage et spill, så derfor bestemte vi oss for å velge oppgave A: Spill. Gruppen fant bilder av spill på web til inspirasjon, og bestemte oss for å lage et enkelt quick time event type spill med en single input controller. Idéen var et infinite runner spill, hvor målet med spillet var å komme så langt som mulig uten å falle ut av plattformen mens man plukker opp objekter på veien. Når man faller ut så er spillet over, og man får en best score. Vi lagde modeller til spillet ved bruk av MagicaVoxel, og fant copyright free 8-bit musikk på web (AudioBrandName, 8-Bit retro video game sound effects 1 2016). Tanken bak spillet var at det skulle være lett å spille, og også lett å bli hektet. Grunnet tidspress så bestemte gruppen seg for å ta utgangspunkt en kilde vi fant på web (inScopeStudios 2015). Vi bygde på koden til eksempelet, og gjorde spillet til vårt eget ved å lage egne objekter i stedet for å bruke prefabs, samt at vi fant passende musikk som harmoniserte med spillets sjanger.

Liste - Features

Her en komprimert liste over ulike features som er med i spillet.

- En game scene og en menu scene.
- Et event system.
- Audio listener og audio source.
- Main camera med orthographic projection. Dette gir en 2D-platformer følelse i et 3D spill.
- Directional light i realtime mode med shadow type soft shadows.
- Rigidbody – både Player og CurrentTile har rigidbody. Fysikk som gravity og kinematic gjør at spilleren kan falle, og klossene detter ned etter at spilleren har passert de.
- Mesh filter – Det er benyttet egne mesh filtere til figurer og objekter.
- Mesh renderer – Vi har brukt egne mesh renderere og egne paletter.
- Unity UI Framework er benyttet til menyen.
- Graphic raycaster
- Particle system – objektene som plukkes opp har et particle system som gir effekten av at når spilleren plukker opp et eple så fordufter det.

Liste - Scripts

Her er en liste over forskjellige metoder og biblioteker benyttet i ulike de scriptene. Unity Scripting API er benyttet som referanse.

- `Using System.Collections;`
- `Using System.Collections.Generic;`
- `Using UnityEngine;`
- `Using UnityEngine.SceneManagement;`
- `Using UnityEngine.UI;`
- `GetComponent<>();`
- `Input.GetMouseButtonDown`
- `Transform.Rotate`
- `Transform.Translate`
- `Transform.position`
- `SceneManager.LoadScene`
- `Input.GetKey`
- `Animator.SetTrigger`
- `Object.Destroy`
- `Collider`
- `GameObject`
- `Transform.GetChild`
- `Mfl.`

Spesielle utfordringer

Gruppen hadde problemer med å erstatte default objektene som skulle plukke opp, med de objektene vi hadde selv laget. Måten vi løste problemet på var å forandre alle mesh-feltene i Pickup fra cube til apple, og materials til palett. Deretter måtte vi forandre på rotasjon og størrelse, samt forandre posisjonen slik at eplene havnet midt over tile-ene. Et annet problem som oppstod var at lydklippet som skulle spilles av når spilleren døde, ble enten ikke spilt av, eller ble konstant loopet. Problemet løste seg ved å legge til en 'bool hasPlayed = true;' og når spilleren dør så er det blitt benyttet en if-statement som sier:

```
If (hasPlayed == true) {  
    deathSound.Play();  
    hasPlayed = false;  
}
```

Dette sørger for at lydklippet blir kun spilt av én gang. For å avslutte bakgrunnsmusikken når man dør, ble denne kodesnutten benyttet:

```
If (!IsGrounded() && isPlaying) {  
    isDead = true;  
    mainTheme.Stop();  
    GameOver();  
}
```

Dette gjør at kun lydklippet som indikerer at spillet er over, spilles av, og gir en bedre spillopplevelse.

Fancy features – Ting gruppen er spesielt fornøye med

Gruppen er spesielt fornøye med det faktum at vi klarte å omstille oss og gjennomføre oppgaven med tanke på at vi skiftet oppgave halvveis i prosjektet. Vi klarte å finne en ny idé som vi var fornøye med, og vil si at vi er positive ovenfor resultatet og hva vi fikk til på så kort tid.

Referanser

AudioBrandName. *8-bit retro video game Sound Effects 2*. 15 Februar 2016.

<https://www.youtube.com/watch?v=7UZQ7NvLNgA> (funnet Januar 30, 2018).

—. *8-Bit retro video game sound effects 1*. 15 Februar 2016.

<https://www.youtube.com/watch?v=nzjtkaLCn60> (funnet Januar 30, 2018).

inScopeStudios. *Endless Runner*. 24 September 2015. https://www.youtube.com/playlist?list=PLXuZVK_OK_73EIM5VvzfrBUDqztzbARm (funnet Januar 26, 2018).

MoonflowerCarnivore. *assetstore.unity.com*. 25 Juli 2016.

<https://assetstore.unity.com/packages/vfx/shaders/lava-flowing-shader-33635> (funnet Januar 30, 2018).

Music, TeknoAXE's Royalty Free. *Royalty Free Music #288 (Hero's Day Off) Chiptune/8-bit/Eight Bit*. 5 Mars 2014. <https://www.youtube.com/watch?v=SgH3Bcs60d0> (funnet Januar 30, 2018).

UnityTechnologies. *Script Reference*. 2018. <https://docs.unity3d.com/ScriptReference/> (funnet Januar 26, 2018).

XenosmashGames. *Creating a start menu in Unity 5*. 6 Januar 2015.

<https://www.youtube.com/watch?v=pT4uca2bSgc> (funnet Januar 30, 2018).