

Multiplatform Development

Documentatie

Celine de Wijs

13/04/2018

Het Spel

Deze opdracht heb ik gecombineerd met Project Context. Voor dit project heb ik samen met mijn groep een 2D/3D platformer gemaakt. Ik had zelf besloten dat ik deze game multiplatform wilde maken. Uiteindelijk is hij te spelen op PC en Android.

Multiplatform Design

De Input code valt onder een abstract class genaamd "InputSystem". Deze class heeft een static functie: `GetInputSystem()`, die een bepaalde instantie returned op basis van je platform.

```
public static InputSystem GetInputSystem()
{
    #if UNITY_IPHONE || UNITY_ANDROID
        return new InputMobile();
    #elif UNITY_STANDALONE || UNITY_EDITOR
        return new InputPC();
    #endif
}
```

Verder zijn er twee abstract functies `GetAxis(GameAction action)` en `CheckInput()`. `GetAxis` checkt of de speler naar links of rechts loopt, bukt of springt.

In de game zijn twee classes die van `InputSystem` deriven. Dit zijn `InputMobile` en `InputPC`. Deze classes regelen de platform specifieke code in `GetAxis` en `CheckInput`.

In de `PlayerController` wordt er gebruik gemaakt van deze input systemen. In de `Awake` functie wordt het inputsysteem gedefiniëerd:

```
public InputSystem inputSystem;

private void Awake()
{
    inputSystem = InputSystem.GetInputSystem();
}
```

In de `Update` worden de axes `inputHorizontal` en `inputVertical` gecheckt:

```
inputHorizontal = inputSystem.GetAxis(GameAction.MOVE_HORIZONTAL);  
inputVertical = inputSystem.GetAxis(GameAction.JUMP);
```

Ook wordt de functie `CheckInput` gecheckt. Hiermee wordt er gekeken of een speler een interactable object aanklikt.

```
inputSystem.CheckInput();
```

Reflectie

`InputMobile` heeft nog een extra public functie: `GetTouch()`. In deze functie word gecheckt of de speler het scherm aanraakt op een android. In de `PlayerController` wordt deze functie op een “hard coded” manier aangeroepen:

```
if (InputMobile.instance != null) // TEMP HACK!  
{  
    InputMobile.instance.GetTouch();  
}
```

Hier ben ik niet trots op, omdat dit betekent dat de platform specifieke code niet helemaal gescheiden is van de game code (deze code slaat namelijk alleen op `InputMobile` en kan niet gebruikt worden voor andere input systemen). Ook dit zou ik volgende keer onder een abstracte functie willen zetten.

Wat ik de volgende keer nog meer anders zou willen doen is de manier waarop die touches geregeld worden. Ik heb nu twee functies: `DetectTouch1()` en `DetectTouch2()`. Dit is om twee vingers op het scherm tegelijk te kunnen checken.

Ik heb geprobeerd deze twee functies samen te voegen en de touch als parameter door te geven, maar dit werkte niet, omdat ik dezelfde `firstTouchPosition` en `lastTouchPosition` gebruikte voor beide touches. Ik heb helaas niet genoeg tijd gehad om dit te kunnen oplossen.

Input Systems en `PlayerController`:

<https://github.com/CdeWijs/Multiplatform-Development>

Volledige project:

<https://github.com/CdeWijs/Project-Context-Poppa-Milk>