

Documentatie Multi Platform Development  
Jelle Dekkers GDV2

## CONTENTS

---

Inleiding.....	3
Het Concept .....	3
Wat ik heb gemaakt .....	3
Input.....	3
Achievements.....	4
Het Proces .....	5
Bijlage.....	7

## INLEIDING

---

Dit is het document van Jelle Dekkers voor de keuzemodule MultiPlatform Development in blok 2 van schooljaar 17/18. In dit document wordt onder andere uitgelegd wat mijn concept was, wat ik uiteindelijk heb gemaakt en het proces.

## HET CONCEPT

---

Voor dit vak waren er 2 dingen die ik graag wou uitzoeken:

- Multiplatform input voor Android en PC\*
- Multiplatform Achievements, met name Android

\*Niet onderscheidend tussen steam, origin etc.

## WAT IK HEB GEMAAKT

---

Ik heb de input en achievements toegepast op een project waar ik al aan werkte. Het is een simpel Android spelletje waarbij de speler een blokje bestuurt over een grid. Het enige wat de speler kan doen is 4 kanten op lopen.

### INPUT

De player class heeft een variabele input van interface IInputSystem. Via een switch verkrijgt player de correcte input a.d.h.v. het platform met de functie GetPlatformDependentInputSystem();

```
public interface IInputSystem {  
    float GetAxisRawHorizontal();  
    float GetAxisRawVertical();  
}
```

Voor de PC input wordt de standaard Unity Input.GetAxis("horizontal") en Input.GetAxis("vertical") returned. Voor de Android versie wordt iets meer uitgevoerd. In de scene staat namelijk een object genaamd Mobile Touch Input met de Mobile Touch Manager component. Onder dit object staan 4 UI Buttons. Deze kan de speler dus indrukken om een richting op te lopen. Ik heb hiervoor gekozen omdat ik volledige controle wil hebben over de positie de grootte en de vorm van de knoppen. Deze knoppen beïnvloeden de InputMobile axis(horizontaal en verticaal).

Als de huidige platform niet Android is dan wordt bij het maken van een nieuwe build en bij het starten van de scene in de editor het object met de Mobile Touch Input verwijderd door de OnPostProcessScene() functie. Hierdoor worden de Android specifieke objecten verwijderd uit de scene.

```
[PostProcessScene]  
public static void OnPostprocessScene() {  
    MobileTouchManager mobileManager = (MobileTouchManager)GameObject.FindObjectOfType(typeof(MobileTouchManager));  
    GameObject.DestroyImmediate(mobileManager.gameObject);  
}
```

Overigens zitten op alle platform specifieke scripts platform define directives om ervoor te zorgen dat platform specifieke classes niet cross platform aan te roepen zijn en er geen errors zijn.






```
#if UNITY_ANDROID || UNITY_IOS
```

## ACHIEVEMENTS

Voor de achievements heb ik een developer account aangeschaft bij Google(zie bijlage 1) voor 25 euro. Hier heb ik de app aangemaakt en de apk van de build toegevoegd. Om het linken van de apk en de app op de Google Play Service te voltooien heb ik een keychain gegenereerd, wat gelukkig grotendeels door Unity zelf wordt gedaan(voor een uitgebreidere uitleg over alles, zie bijlage 3).

Hierna heb ik een Play Service aangemaakt voor de achievements en er 5 dummy achievements ingezet.

Nieuwe prestatie toevoegen

Nr.	Naam	ID ?	Punten	% ontgrendeld Totaal aantal/Tijd ?	Status
1	 Baby Steps		5	—	— Niet gepubliceerd
2	 Almost there		25	—	— Niet gepubliceerd
3	 Complete First Level		30	—	— Niet gepubliceerd
4	 Try Again		10	—	— Niet gepubliceerd
5	 Insert Cake Joke Here		30	—	— Niet gepubliceerd

Bronnen ophalen

Totaal aantal punten:  
100 ?

Lees alles over het implementeren van gameprestaties in de [ontwikkelaarsdocumentatie](#).

Om deze achievements te testen heb ik de Google Play Games Service plugin gedownload van Google(bijlage 2). Dit is nodig om gebruikt te maken van de Google Play Services in Unity(, en dus ook achievements). Hierna wordt via de knop 'Bronnen ophalen' alle achievement ID's in een XML gezet. Deze moet dan worden gekopieerd naar de Setup window van Google Play Service in Unity en dan genereert de plugin de static class GPGSID.cs waar alle achievements makkelijk via code te vinden zijn.

Ook bij de achievements heb ik een interface en een switch gemaakt om de platform correcte achievement systeem te verkrijgen. Het achievement systeem voor PC heb ik voor nu leeg gelaten omdat ik me heb gefocust op de Android achievements en Google Play Service(voor meer informatie over Android Achievements, zie bijlage 4).

```
public interface IAchievementSystem {  
  
    void UnlockAchievement(string id);  
    void IncrementAchievement(string id, int stepsToIncrement);  
    void ShowAchievementsUI();  
}
```

Om achievements op Android daadwerkelijk te gebruiken en te unlocken moet PlayGamesPlatform worden geïnitialiseerd en moet via Unity's Social class de localUser worden ge-authenticate. Als dit allemaal lukt dan zijn er functies die ik aan kan roepen, namelijk:

- IncrementAchievement: dit telt een waarde bij een achievement op tot aan de max daarna wordt deze unlocked
- UnlockAchievement: dit unlocked direct de achievement, zonder te incrementen
- ShowAchievementsUI: dit opent een standard Android window waar alle achievements op te zien zijn

```

public class AndroidAchievementSystem : IAchievementSystem {

    public AndroidAchievementSystem() {
        Debug.Log("Constructor for AndroidAchievementSystem");
        PlayGamesPlatform.DebugLogEnabled = true;
        PlayGamesClientConfiguration config = new PlayGamesClientConfiguration.Builder().Build();
        PlayGamesPlatform.InitializeInstance(config);
        PlayGamesPlatform.Activate();
        Social.LocalUser.Authenticate(ProcessAuthentication);
    }

    void ProcessAuthentication(bool success) {
        if (success)
            Debug.Log("Sucesfully Authenticated");
        else
            Debug.Log("Failed to authenticate");
    }

    public void IncrementAchievement(string id, int stepsToIncrement) {
        Debug.Log("Trying to increment achievement");
        PlayGamesPlatform.Instance.IncrementAchievement(id, stepsToIncrement, success => { Debug.Log("Sucesfully incrementend achievement progress"); });
    }

    public void UnlockAchievement(string id) {
        Debug.Log("trying to unlock achievement");
        //PlayGamesPlatform.Instance.UnlockAchievement(id, success => { Debug.Log("Achievement unlocked, id: " + id); });
        Social.ReportProgress(id, 100f, success => { Debug.Log("Achievement unlocked, id: " + id); });
    }

    public void ShowAchievementsUI() {
        Debug.Log("Trying to show Achievements UI");
        //PlayGamesPlatform.Instance.ShowAchievementsUI();
        Social.ShowAchievementsUI();
    }
}

```

Om dit allemaal dus te testen moet er via de Releasebeheer in de Google Play Console een nieuwe alfa build worden aangemaakt met een bruikbare apk(alle informatie moet correct worden ingevuld bij de Player Settings in Unity en de versienummer moet hoger zijn dan de vorige). Dan moet je je android account bij alfa testers toevoegen en via een link kan dan de alfa release worden gedownload.

## HET PROCES

---

Bij de input ging alles redelijk soepel en had ik geen problemen. Het uitzoeken van OnPostProcessScene() vereiste wel het een en ander onderzoek omdat er naast deze functie ook een OnPostProcessBuild en een PostProcess attribute is, na een paar keer testen was het verschil me wel duidelijk.

Voor de Android achievements was wel veel werk en moeite nodig. Het onderscheid tussen de App en de Play Service was mij niet volledig duidelijk, dit staat ook niet goed aangegeven op de website en het is mij nu nog steeds niet 100% duidelijk. Het implementeren van de Play Service naar het project was ook veel werk omdat ik bijvoorbeeld verouderde SDK's had en ik heb een keer tijdens het project mijn Unity geüpdatet waardoor alles niet meer werkte, de oplossing was om nogmaals een nieuwe SDK te installeren.

Om mijzelf tijd te besparen heb ik Nox, een Android emulator gedownload(zie bijlage 5) om daarop te testen zodat ik niet steeds naar mijn telefoon hoeft te builden. Omdat ik toch geen gebruik maken van multi touch of ingebouwde hardware zoals de gyroscoop was dit perfect om te testen. Debuggen heb ik gedaan via de Logcat van Eclipse, die werkte gewoon met Nox.

Het meest vervelende van dit project was het aan de praat krijgen van de PlayGamesPlatform. Deze werkte steeds maar niet en wanneer ik achievements wou testen dan gebeurde er maar niets. Later kwam ik erachter dat het PlayGamesPlatform alleen wordt aangeroepen als de build van de PlayStore komt, wat testen heel vervelend maakt omdat je bij elke kleine verandering een nieuwe build moet toevoegen. Hierna werkte achievements nog steeds niet. Door het aanzetten van "PlayGamesPlatform.DebugLogEnabled = true" en naar het kijken van Logcat kwam ik erachter dat de PlayGamesPlatform niet goed werd geïnitieerd. Ik had namelijk in de AchievementManager class een property die, als hij null is, a.d.h.v. het huidige platform de

correcte AchievementSystem aanmaakt, zodat ik hem nergens zelf hoeft aan te maken. Helaas wordt de PlayGamesPlatform asynchroon geïnitieerd en dit duurt een paar seconden. Wanneer ik dus een achievement wou unlocken en de AchievementManager voor het eerst werd aangeroepen dan had hij nog eventjes nodig om te initialiseren waarop het dus fout ging. Het initialiseren gebeurt nu in een aparte Setup() functie die in Start() wordt aangeroepen.

```
namespace Achievements {  
  
    public static class AchievementManager {  
  
        private static IAchievementSystem achievementSystemInstance;  
        public static IAchievementSystem AchievementSystemInstance {  
            get {  
                if (achievementSystemInstance == null)  
                    achievementSystemInstance = GetPlatformDependentAchievementSystem();  
                return achievementSystemInstance;  
            }  
        }  
  
        public static void Setup() {  
            Debug.Log("Achievement manager setup()");  
            if(achievementSystemInstance == null)  
                achievementSystemInstance = GetPlatformDependentAchievementSystem();  
        }  
  
        public static IAchievementSystem GetPlatformDependentAchievementSystem() {  
            Debug.Log("GetPlatformDependentAchievementSystem()");  
#if UNITY_ANDROID || UNITY_IOS  
            return new AndroidAchievementSystem();  
#else  
            return new CustomAchievementsSystem();  
#endif  
        }  
    }  
}
```

## BIJLAGE

---

1. Google Play Console:  
[www.play.google.com](http://www.play.google.com)
2. Google Play Service Unity Plugin:  
<https://github.com/playgameservices/play-games-plugin-for-unity>
3. Aanmaken Google Play Service:  
<https://developers.google.com/games/services/console/enabling> 4
4. Informatie over Android Achievements:  
<https://developers.google.com/games/services/common/concepts/achievements>
5. Nox Android Emulator:  
<https://www.bignox.com/>