

# Passion Project Proces / Gebruikerstesten

## Inhoudsopgave

Inleiding .....	2
Inspiratie .....	2
Game Design Document .....	3
Paper Prototype .....	3
Feedback Paper Prototype .....	3
Unity .....	4
Level & Character Design V1 (First Prototype) .....	4
Level Design V2 .....	4
Playable Character V2 .....	5
Feedback .....	5
Playable Character v2, Animaties .....	6
User Tests – Character Movement .....	7
Volgende Camera .....	7
Parallax Achtergrond Toevoegen .....	8
Talking NPC + Feedback Iteraties .....	8
Feedback + Iteraties .....	9
Titel Scherm .....	11
Inspiratie .....	11
Schets .....	12
Tutorial Level User Testing .....	13
Spikes/Respawn .....	13
Camera Transitions .....	14
Feedback .....	14
Attacking Enemies .....	15
Feedback/User testing .....	15
Music (Background, Jump, Punch) .....	16
Level Design V3 .....	16
Build for Html 5 + link .....	17
User Testing + Feedback .....	17

## Inleiding

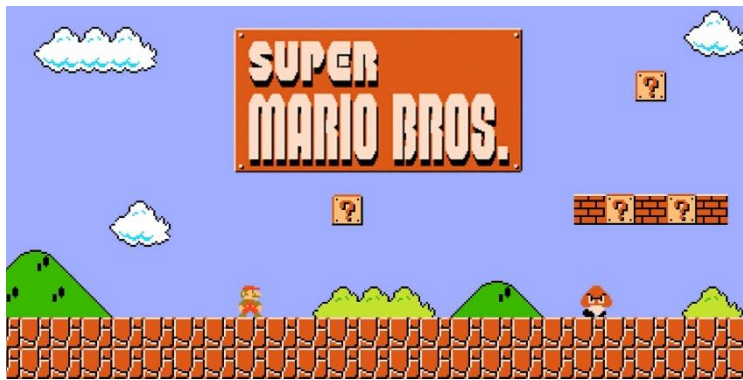
Voor mijn Passion Project maak ik een game in Unity. Hiervoor doe ik inspiratie, maak ik schetsen en keuzes op basis van feedback en gebruikerstesten. Het proces van dit project, de designkeuzes en gebruikerstesten staan beschreven in dit document. Ik heb hiervoor een game design document gemaakt met alle informatie over mijn game, deze wordt constant bijgewerkt en aangepast nadat ik feedback heb gekregen en user tests heb gedaan zodat het concept meegroeit in het proces. Ik heb elke keer nadat ik een mechanic heb toegevoegd, user tests gedaan of feedback gevraagd aan Bernd-Jan, hierdoor kon ik meteen bugs of suggesties krijgen die ik dan ook meteen kon fixen of implementeren.

## Inspiratie

Ik heb vaker in dit document inspiratie opgedaan. Voordat ik begon heb ik wat algemene inspiratie opgedaan door naar 2D platformers te kijken die populair zijn en ik zelf ook heel leuk vind. Laterna heb ik bij sommige mechanics ook gekeken naar andere games. Hier zijn wat populaire games die ik heb gebruikt als inspiratie:



Dit spel heet Celeste, het is een van de hoofdinspiraties van mijn game. In celeste beklim je ook een berg door gebruik te maken van simpele mechanics: Springen, Lopen, Klimmen en Dubbel Springen. Het mechanics zijn dus enorm simpel maar de levels worden steeds moeilijker. Ook focussen ze erg op het verhaal.



Mario Bros is ook een 2D platformer, de main mechanics hierin zijn: lopen, springen en aanvallen. Je kunt tegen een ? box aanspringen om een item te krijgen zoals een vuurbloem, hiermee kan je vervolgens aanvallen. Ook kan je vijanden verslaan door op hun hoofd te springen.



In de oude zelda games zit vaak een enorm grote verhaallijn, ook al komt deze niet direct naar voren kom je vaak veel over het verhaal te weten door borden, NPC's of geheime plekken die je vind. Ik vind vooral de geheime plekken en hoe ze het verhaal naar voren laten komen erg inspirerend in de Legend of Zelda games.

## Game Design Document

Een Game Design Document is een document waar alles van mijn game in beschreven staat. Deze groeit mee in het proces omdat er vaak wat verandert na user testing. Deze gebruik ik als leidraad voor mijn game. Deze kun je vinden in [Game Design Document \(GGD\).pdf](#)

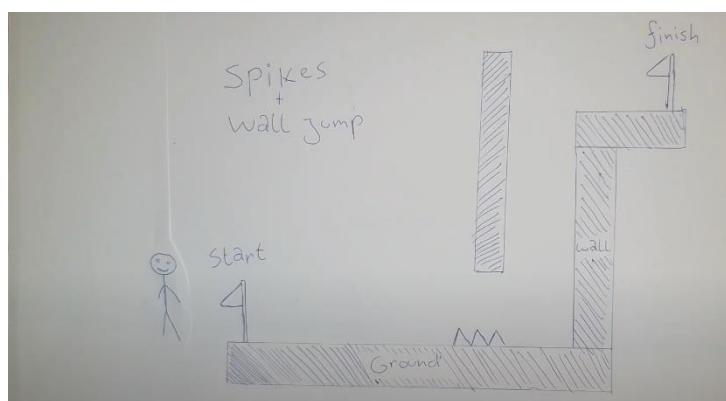
## Paper Prototype

Nu ik een beter beeld heb van mijn game maak ik een paper prototype om mijn core gameplay loop duidelijk te krijgen en te testen of dit ook leuk is. Een video van de Paper Prototype is te zien op:

[https://www.youtube.com/watch?v=1mQGULza4WU&ab\\_channel=Airblade](https://www.youtube.com/watch?v=1mQGULza4WU&ab_channel=Airblade)

## Feedback Paper Prototype

Deze heb ik voorgelegd bij Bernd-Jan, hij vertelde me dat dit er goed uitzag en het een duidelijk beeld gaf van de mechanics die ik in mijn game wou bedenken. Hiermee heb ik dus een lijstje gemaakt van mechanics die ik in mijn game wil verwerken en heb deze ook meteen in mijn Trello-bord gezet.

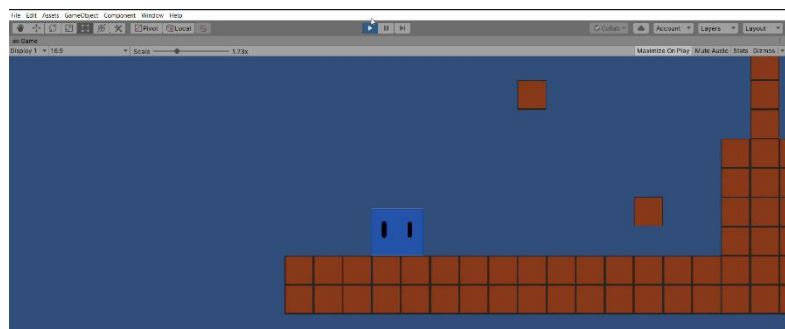


## Unity

Nadat ik mijn playable prototype heb laten testen en feedback heb ontvangen is het tijd om mijn idee up te daten in mijn game design document. Nadat de aanpassingen opgeschreven zijn kan ik eindelijk beginnen met het maken van mijn game in Unity. Omdat ik weinig ervaring heb met Unity volg ik veel tutorials en maak ik gebruik van assets uit de asset store.

## Level & Character Design V1 (First Prototype)

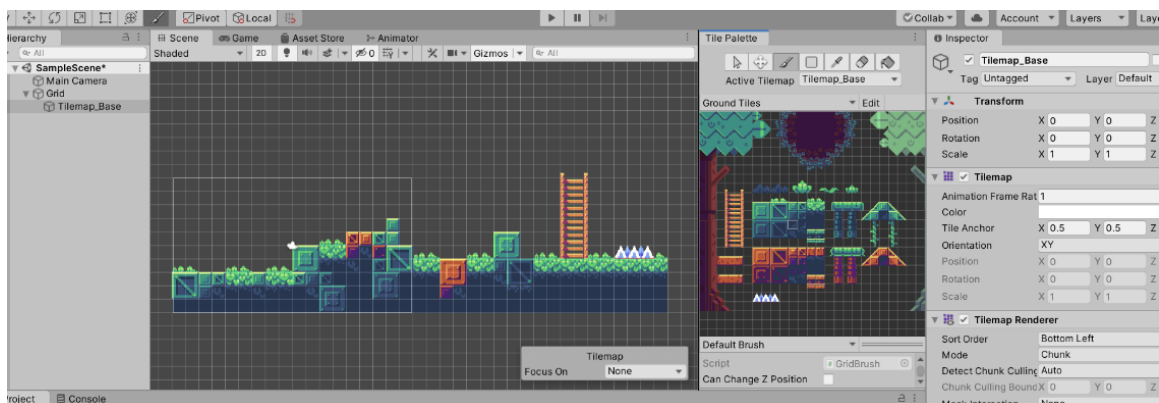
Om een speler te kunnen laten lopen heb ik eerst een grond nodig waar dit op kan. Ik heb in Photoshop snel een 32x32 pixel blokje gemaakt en deze in Unity gezet. Door deze een collider te geven kan hier interactie mee plaats vinden en zal de speler hier niet zomaar doorheen vallen. Ook heb ik een simpel blokje met twee ogen gemaakt als character zodat ik de controls kan testen. Het lukte me om het blokje van links naar rechts te verplaatsen. Met deze start kon ik steeds meer uitbreiden en nieuwe iteraties en testen doen. Hieronder zie je waar het allemaal mee begon.



## Level Design V2

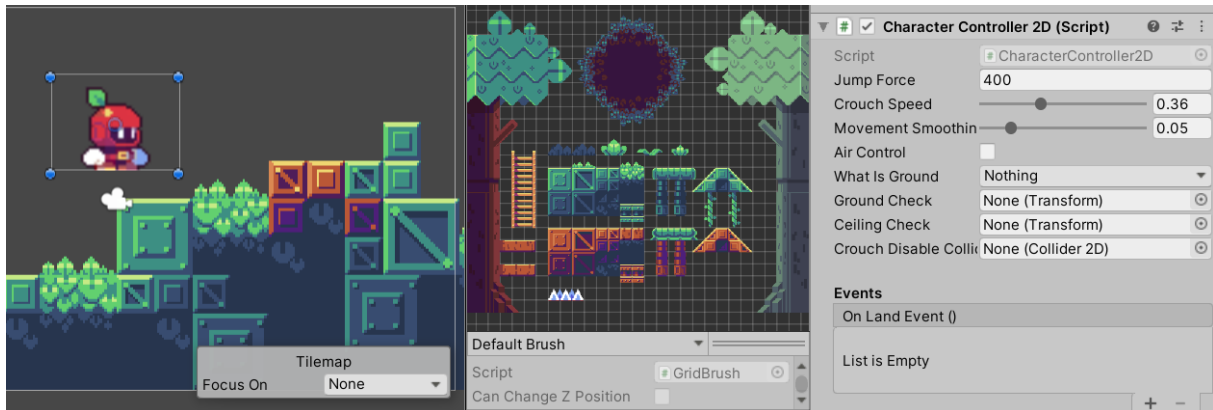
Ik ben begonnen met het downloaden van een tileset uit de asset store. Dit zijn blokjes die samen een level of wereld kunnen vormen. Door een tileset grid te maken in Unity krijg je een grid, oftewel een rooster van vierkantjes. Je kunt nu makkelijk deze blokken uit je tileset in je game plaatsen. Hiermee heb ik mijn eerste basic level gemaakt waarop ik wat mechanics zoals springen kan gaan testen. Later maak ik levels waar meer diepgang en user testing achter zit. Het is belangrijk om colliders op de objecten te zetten waar de speler interactie mee heeft, anders valt hij door de grond heen. Deze moet je juist niet op achtergrondobjecten zetten zodat hij hier gewoon voorbij loopt.

Ik heb gekozen voor een bos tileset omdat de speler begint onder aan een berg, hier is vaak een bos of veel natuur. Hier heb je dus nog geen sneeuw of grotten.



## Playable Character V2

Om het spel te kunnen spelen moet je personage kunnen lopen en springen. Hiervoor moet je een aantal dingen opzetten voordat je kunt beginnen met coderen. Je moet colliders toevoegen aan de objecten waar je speler op gaat lopen zodat hij niet door het level heen valt. Je maakt een object aan voor je speler met dezelfde colliders, dit kan een blokje zijn of pixel art die je zelf hebt gemaakt. Verder voeg je een character controller toe in je Unity file en uiteindelijk voeg je een script toe.

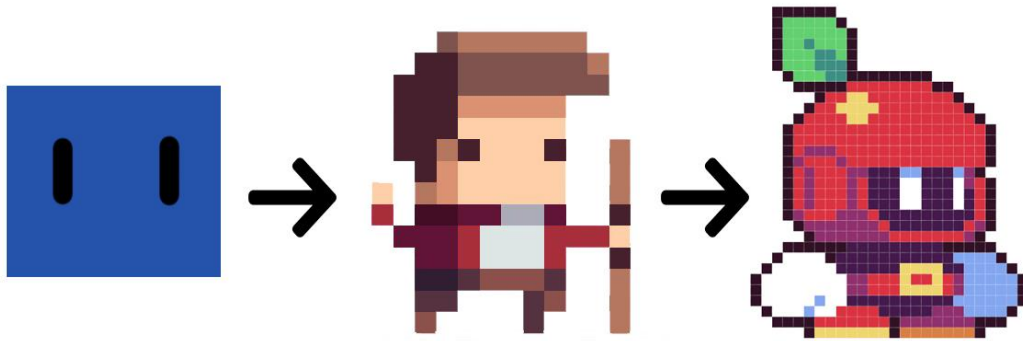


Je ziet dat je in de player controller een aantal variabele kan veranderen zoals de Jump Force en Air Control. Je speelt hiermee totdat de movement goed en smooth aanvoelt, ook als je animaties gebruikt moet je ervoor zorgen dat de snelheid overeenkomt met de loop animatie. Als je wilt dat je personage in de lucht bestuurbaar is vink je Air Control aan. Zo zijn er enorm veel variabele die je aan kan passen om het springen en bewegen beter aan te laten voel. Hierom zijn mensen vaak erg lang bezig om een player controller te fine tunen.

## Feedback

Ik kreeg als feedback te horen dat ik goed bezig was en de belangrijkste mechanics in mijn game had verwerkt. Bij het blauwe blokje als personage kreeg ik vaak te horen dat deze er erg simpel uitzag. Ik heb hierna een mannetje erin gezet. De docent gaf aan dat het personage wel bij het level moest passen dus heb ik deze uiteindelijk nog een keer aangepast.

Hierna kreeg ik de feedback om animaties toe te voegen. Nu bewoog mijn poppetje nog erg stijf over het level heen. Door deze te animeren zal iemand een vele betere feel van de game krijgen en zal alles ook smooth eruit gaan zien. Ook moet je goed opletten dat je de loopsnelheid en springkracht goed aanpast op de levels die je gaat maken. Als een speler een te lage springkracht heeft zal deze niet overal op kunnen komen en vice versa.



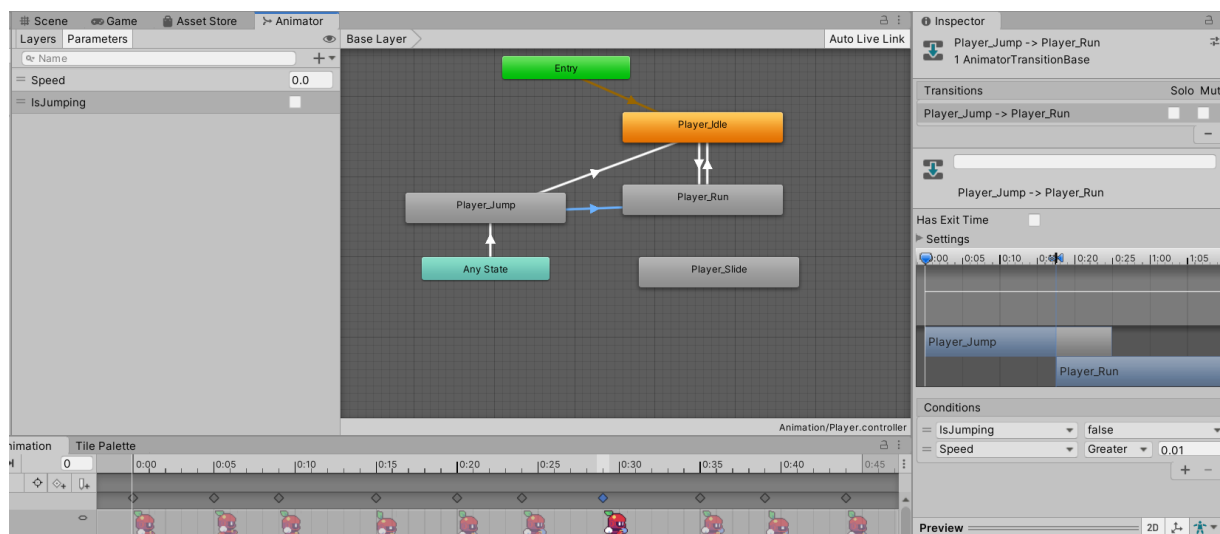
Ik begon met een simpel blokje zodat ik goed de mechanics kon testen, nadat ik het blokje kon bewegen wou ik kijken hoe dit eruit zag met een ander poppetje. Na feedback heb ik een poppetje gekozen dat beter bij het level past. Nu moet ik hier nog animaties aan toe voegen om een betere feeling te krijgen van het bewegen.

## Playable Character v2, Animaties

Nu ziet het personage er nog stijf uit, hij beweegt wel van plek maar de afbeelding beweegt niet. Om meer diepgang te krijgen in het spel maak je gebruik van animaties. Deze animaties roep je op wanneer er een bepaalde input wordt gegeven, zo begint bijvoorbeeld de loop animatie wanneer er wordt gelopen.

Dit doe je door gebruik te maken van de Unity Animator en de Animation Window. Eerst maak je een animatie aan in de Animation Window. Je maakt de naam aan, bijvoorbeeld Player\_Jump en sleept hier de juiste sprites op. Hier kun je de duratie en andere dingen aanpassen om de animatie te bewerken. Als je dit eenmaal voor elke animatie gedaan hebt kun je de Animator openen.

Hier kies je welke animaties er op welk moment afgespeeld worden (zie afbeelding hieronder). Dit doe je door transitions toe te voegen (de witte pijltjes). In deze transitions kun je aangeven wanneer deze transitie moet plaatsvinden en dus wanneer je van animatie moet switchen. We maken een parameter aan genaamd "Speed" en vertellen dat de ren animatie moet starten als de speed groter is dan 0.1. Om weer terug naar de idle animatie te gaan maken we een transitie (witte pijl) terug en vertellen dat deze animatie start wanneer de Speed kleiner is dan 0.1. Nu moeten we in het script alleen nog een waarde meegeven aan deze parameter. Gelukkig had ik al een variabele voor de snelheid gemaakt.



Je wilt vanuit elke animatie/state kunnen springen. In plaats van tussen elke animatie een transitie naar de Player\_Jump te maken kun je dit doen vanuit de Any State balk (blauwe balk) hierdoor kun je op elk moment switchen naar de spring animatie.

## User Tests – Character Movement

Ik heb mijn game in deze staat laten testen bij docenten en vrienden. Er was nog niet echt een doel of veel levels maar ik wou weten hoe het personage aanvoelde. Ik stuurde de hele Unity File zodat de gebruikers zelf ook even verschillende snelheden of springkrachten konden uittesten. Hieruit bleek dat in mijn oorspronkelijke instellingen de loopsnelheid te snel was. De gebruikers kregen niet veel mee en vlogen razendsnel door een level heen. Ook was het personage hierdoor moeilijker te beheersen.

Toen de gebruikers langs de muren heen gleden vertelde ze me dat er geen wall slide animations zijn. Deze was ik zelf ook helemaal vergeten. Ik moet dus ook meer animaties toevoegen. Nu ik weet wat fijne instellingen zijn voor het personage kan ik hier ook de levels op gaan baseren. Ik moet bijvoorbeeld obstakel maken waar de speler niet overheen kan springen nadat ik de springkracht heb verminderd.

Ook vond mijn vriend Wouter een bug waarin de spelers animaties glitchen wanneer je bukt, onder een blokje kruipt en vervolgens probeert op te staan. Dit heb ik vervolgens aangepast en verholpen.

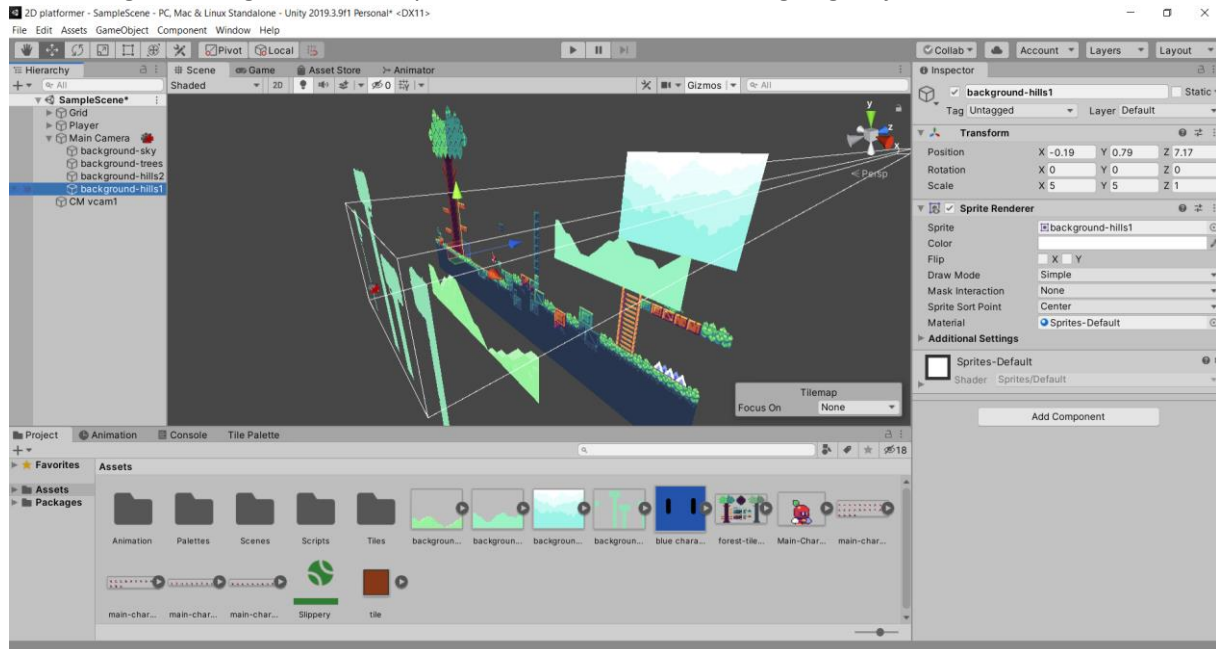
## Volgende Camera

Om ervoor te zorgen dat het personage niet uit het zicht verdwijnt moet je ervoor zorgen dat de camera het personage volgt. Dit kan heel makkelijk door de Main Camera op de speler te slepen en de x,y coördinaten op 0 te zetten. Echter zal de camera niet zo natuurlijk aanvoelen als in de meeste games omdat hij heel blokkerig en direct volgt. Eerst moest je al deze instellingen coderen maar nu heeft Unity daar een package voor die je kunt installeren genaamd CineMachine. Hiermee kun je een CineMachine Camera toevoegen en veel variabelen veranderen. Je kunt bijvoorbeeld de camera laten bewegen wanneer de speler buiten een bepaald gebied gaat of de camera alvast bewegen naar de plek waar de speler naartoe loopt. Ik ga met deze instellingen spelen totdat ik een camera krijg die voor mij goed en natuurlijk aanvoelt. Daarnaast maak ik een aantal opties die ik uiteindelijk laat testen. Door deze feedback en usertesting weet ik uiteindelijk wat goede instellingen zijn voor de Player Camera.



## Parallax Achtergrond Toevoegen

Om toch een gevoel van diepte te krijgen verdeel je de achtergrond op in stukken. Deze zet je vast aan de camera maar wel op een ander punt op de Z-as. Als je met een script de voorwerpen die verder weg staan langzamer laat bewegen, zal het eruit zien alsof deze ook verder weg staan. Net als in het echt lijkt het alsof voorwerpen die dichterbij staan sneller verplaatsen. Dit geeft een parallax effect en geeft een gevoel van diepte. De code hiervoor staat uitgelegd bij Transferable Code.



## Talking NPC + Feedback Iteraties

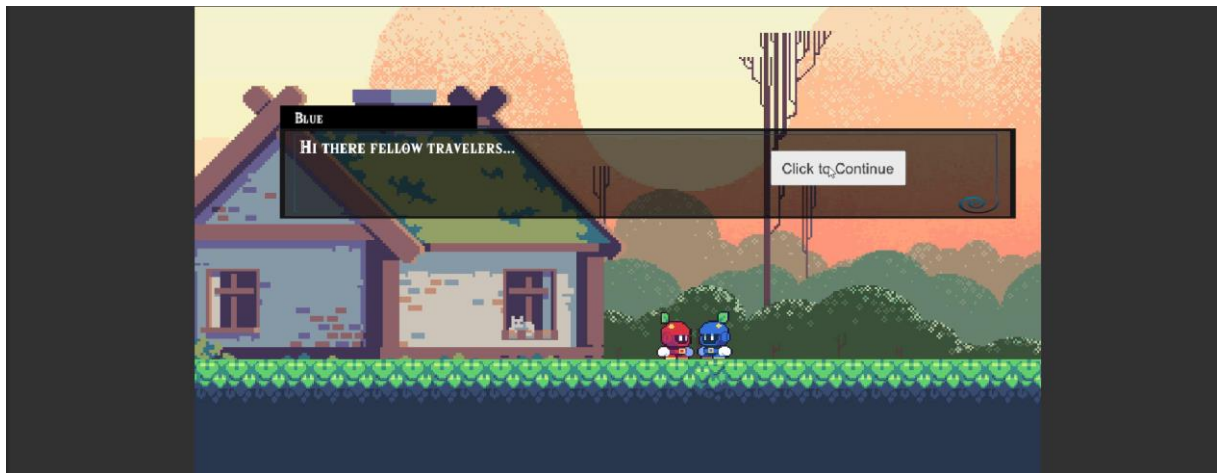
Ik wil in mijn game een ander personage hebben waarmee je kunt praten en die je handige tips geeft. Hiervoor moet ik een dialogue system maken. Hiervoor gebruik je Unity's canvas object met daarin wat tekst en een dialogue box. Ik heb uiteindelijk voor de simpele dialogue box hieronder gekozen.



Hierin zet je een tekst object voor de NPC naam en de tekst die hij gaat zeggen. Nu je dit klaar hebt is het tijd om te coderen en ervoor te zorgen dat deze tekst elementen worden aangepast. De comments bij de code leggen alles goed uit. <https://git.fhict.nl/I437759/semester-3-media/-/tree/master/UnityScripts> (Script DialogueTrigger.cs, DialogueManager.cs en Dialogue.cs)

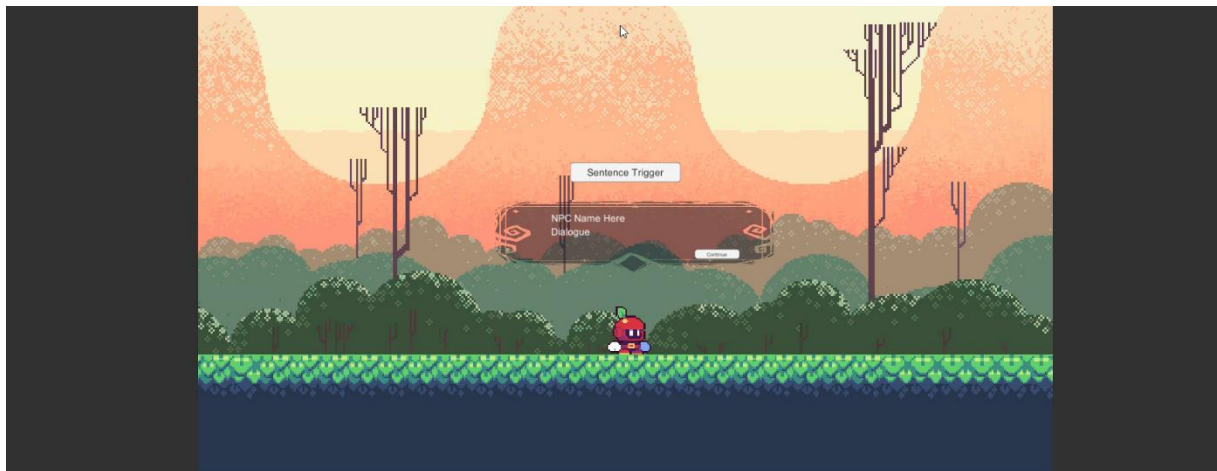
Er zijn veel variabelen die ik kan aanpassen zoals: de snelheid van de tekst, de grootte, de font en de plek van de dialoguebox. Ik heb na veel codeerwerk eindelijk mijn eerste versie klaar.



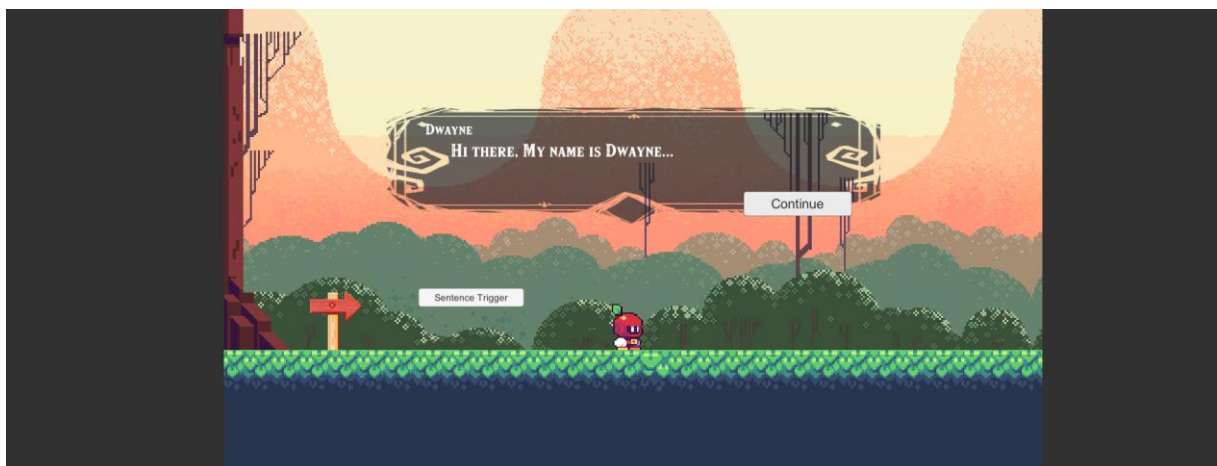


### Feedback + Iteraties

Nu ik alles werkend heb gekregen wil ik graag feedback van Tom en mijn mede studenten. Deze vertelde me dat het er erg goed uitzag maar dat ik misschien iets met de dialogue box kan doen omdat deze er nu erg blokkerig en simpel uitziet. Ik heb deze aangepast zoals hieronder te zien is.



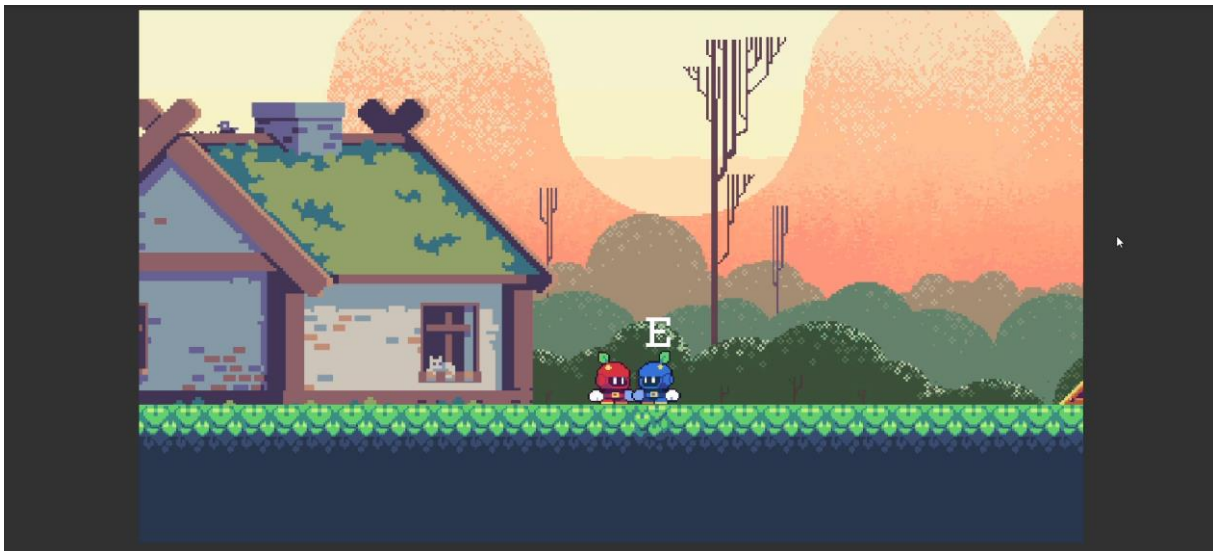
Nu ik een nieuwe dialogue box heb, heb ik heb deze meteen weer voorgelegd bij studenten en Tom. Hieruit bleek dat deze eigenlijk te laag in het beeld stond en dat de dialoguebox wat donkerder mag. Op sommige plekken in het level zie je de tekst niet goed. Ook kan ik de font veranderen voor meer duidelijkheid.



Nu de dialoguebox goed was vertelde Tom dat ik de Sentence Trigger button moest veranderen door een echte NPC. Het zou dan natuurlijk mooi zijn als deze ook een animatie heeft en ergens staat te lopen of te wachten. Dit heb ik vervolgens gedaan door een gameobject voor de NPC te maken. Ik moest in de code een aantal dingen aanpassen. Zo moest ik er bijvoorbeeld voor zorgen dat de speler kan praten door op E te klikken. Dit lukte maar je kon nu overal in het level op E klikken. Ik moest de afstand tussen de speler en de NPC berekenen en zodra de speler binnen een distance van 1.5 zit én hij op E klikt verschijnt de tekst dialogue. Nu kun je dus alleen met de NPC praten als je in de buurt staat. Ik heb een blauwe variant van de main-character gemaakt en deze ook een idle animatie gegeven.



Nu kon ik weer laten testen of alles werkte. Ik kwam er al snel achter dat de gebruikers niet wisten hoe ze met de NPC moesten praten, er staat namelijk ook nergens aangegeven dat dit met de E knop moet. Daarom heb ik er meteen voor gezorgd dat er een animatie begint wanneer je in de buurt van de NPC komt. Dit kan je hieronder zien.



## Titel Scherm

In de meeste games begin je niet meteen in een level. Je hebt vaak een startscherm. Ik ben erachter gekomen dat je in Unity in verschillende scenes kunt werken. Zo heb ik een scene gemaakt voor het startscherm. In dit startscherm heb ik een knop gezet en een titel. Door op de knop te drukken wordt de volgende scene ingeladen. Ik heb hier een klein scriptje voor moeten schrijven.

## Inspiratie

Ik heb wat inspiratie opgedaan. Ik vond de startschermen van oude arcades vet, dit past ook goed bij mijn game omdat dit een pixel stijl game is. Deze oude startschermen laten ook vaak de highscore bovenin de hoek zien.

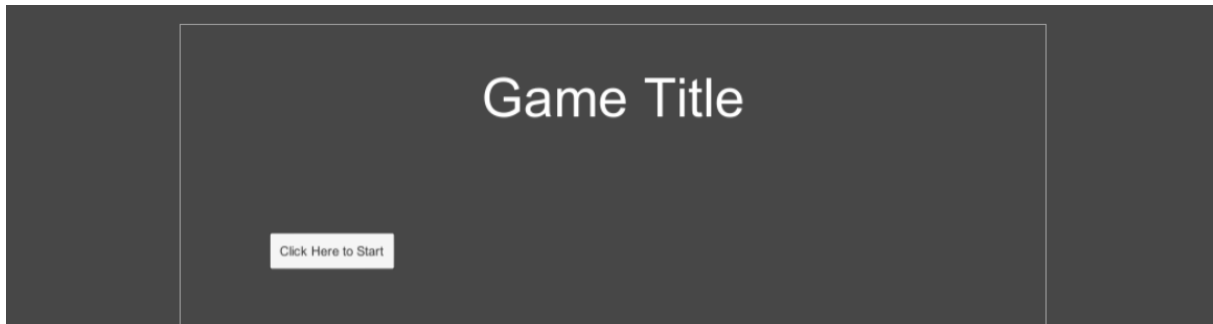


## Schets

Ik heb een snelle schets gemaakt van hoe ik mijn startscherm ongeveer wil hebben. Ik wil het liefste een scherm zonder start button maar ik weet niet wat mogelijk is dus heb ik voor beide een schets gemaakt.



Nu ik een beeld had van wat ik wou kon ik dit in Unity gaan maken.



De knop zorgt ervoor dat de tutorial level wordt geladen, ik vond het echter vetter om dit niet met een knop te doen maar door spatie in te drukken. Zo krijg je meer een arcade style startscherm waarop je "Press Space to Continue" kunt zetten. Ik heb naar fonts gekeken en een achtergrond.

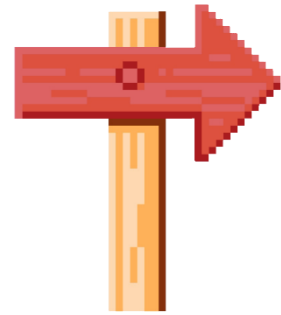
Ik heb advies gevraagd over de kleur, titel en achtergrond. Als ik er maar voor zorgde dat alles duidelijk te lezen is en de achtergrond iets met de game zelf te maken heeft zat ik wel goed. Ik heb ervoor gekozen om een geanimeerde achtergrond te maken, deze achtergrond is dezelfde als het beginlevel maar dan geanimeerd. Verder heb ik een rode titel gekozen omdat de hoofdpersoon zelf ook rood is en heb ik voor de overige tekst de Font Joystix gekozen omdat deze een arcade pixel game vibe geeft. Hieronder zie je het resultaat. (Press Space To Start is ook geanimeerd)





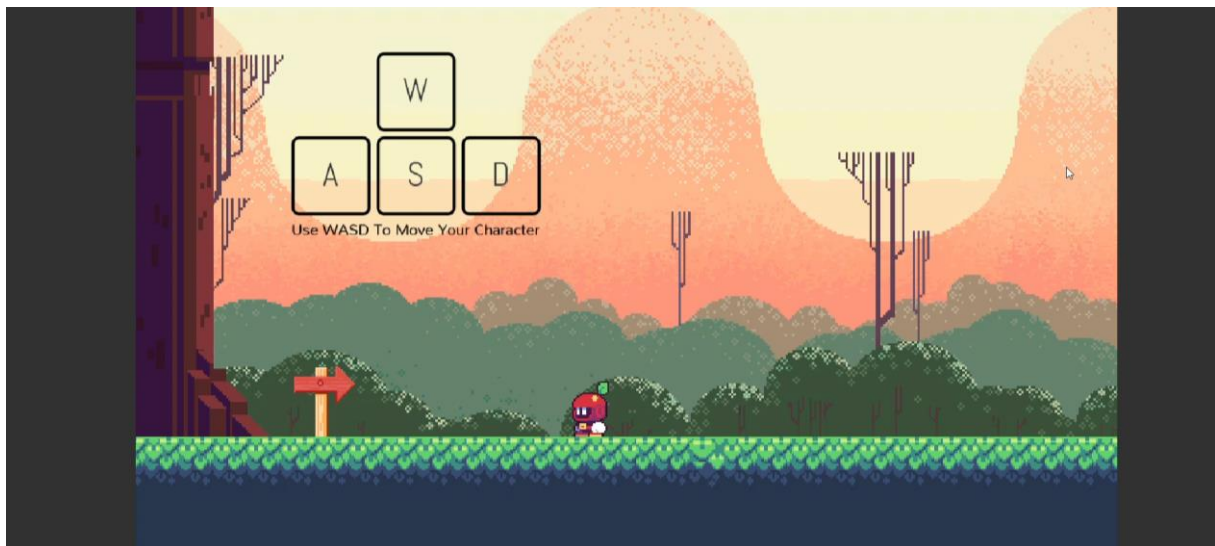
## Tutorial Level User Testing

Ik liet mijn tutorial level testen zodat ik weet of iedereen het spel en de uitleg goed begrijpt. Een paar van de gebruikers liepen eerste instantie naar de linkerkant terwijl je normaal rechts door een level heengaat. Hierdoor heb ik een bord moeten plaatsen met een pijl naar rechts en een obstakel waardoor je niet links van het level af kunt lopen.



Vaak weten mensen al dat je met WASD kunt lopen maar in sommige gevallen wist niet iedereen het meteen. Ik moet dus in het begin van mijn level even laten weten dat je WASD gebruikt om te bewegen. Hiervoor heb ik een afbeelding geanimeerd en die gehangen op de plek waar je meteen begint.

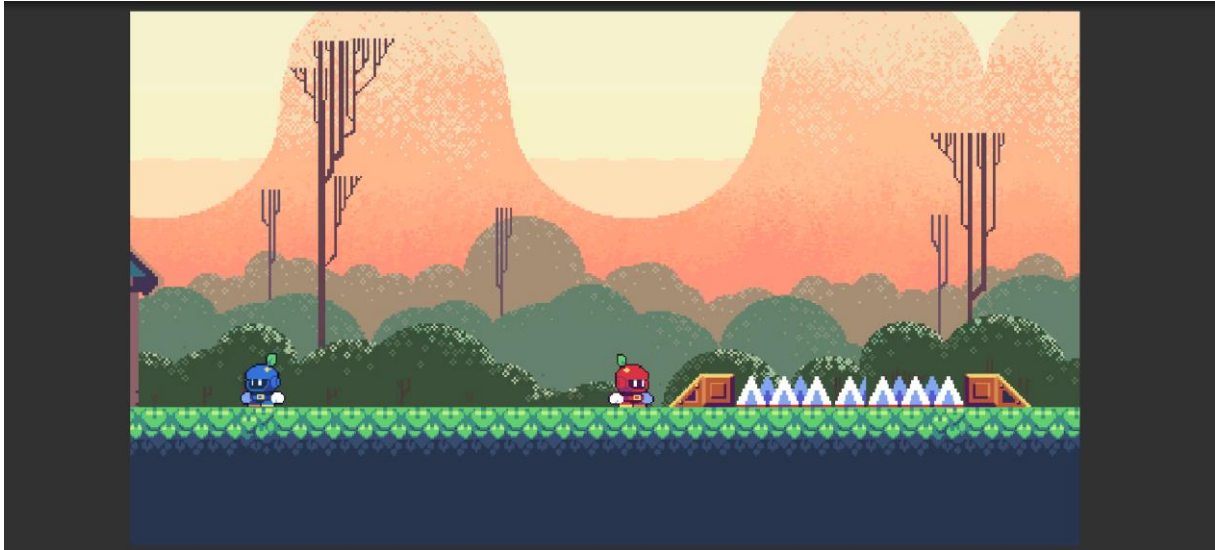
Ook wisten spelers niet met welke knop ze konden praten met de NPC. Ik heb er dus voor gezorgd dat er een E boven het poppetje verschijnt zodra je in de buurt komt. Hierdoor wisten de gebruikers meteen dat je met E kunt praten.



## Spikes/Respawn

In veel games heb je obstakels die je niet aan mag raken. Ik wil spikes in mijn game hebben, als de speler deze aanraakt gaat hij dood en respawnt hij op de laatste save point. Hiervoor moet ik dus 2 mechanics toevoegen: het doodgaan en het opnieuw respawnen.

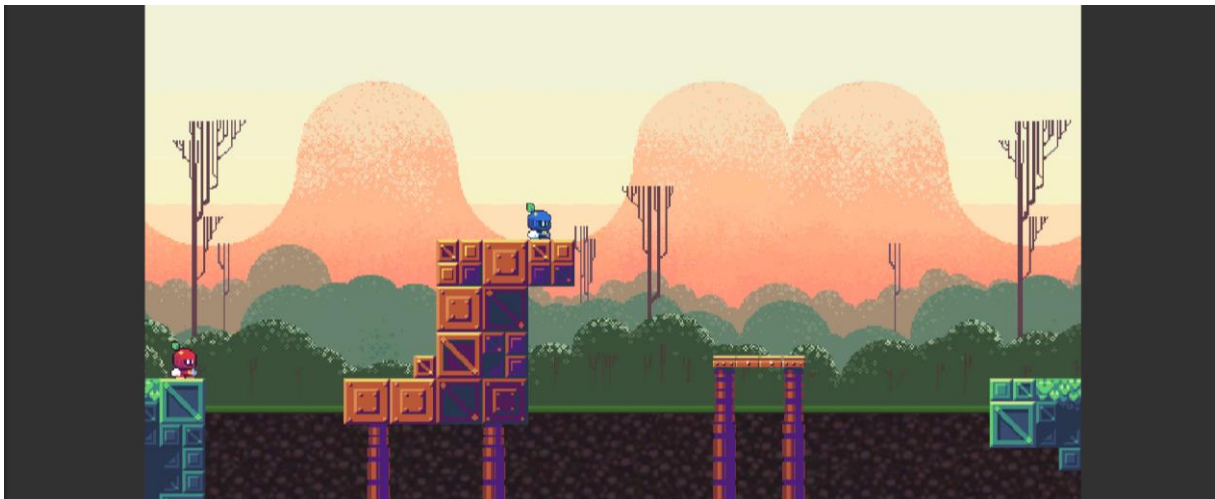
Hiervoor heb ik colliders gezet op de spikes en heb ik ervoor gezorgd dat de Death\_animation wordt afgespeeld zodra de speler de spikes aanraakt. Ik kan blijkbaar een event toevoegen aan een animatie, dat betekent dat ik gewoon een functie kan oproepen tijdens of na een animatie. Ik heb een die(); functie gemaakt die ervoor zorgt dat de scene opnieuw wordt geladen. Dit gebeurt nadat de speler zijn doodanimatie ziet. Ik heb hieronder een klein stukje spikes toegevoegd aan het level zodat de speler hier op een makkelijke manier aan kan wennen.



Het is belangrijk om verschillende mechanics geleidelijk aan je gebruiker voor te stellen. Als ik meteen vijanden, spikes, springen, aanvallen etc. tegelijk doe kan het lastig worden. Ik heb hierom een paar losse spikes gezet zonder andere moeilijke obstakels. Hierdoor went de speler aan het springen en komt hij er al snel achter dat spikes slecht zijn en je deze dus niet aan mag raken.

## Camera Transitions

Uit de user testing bleek ook dat sommige levels niet goed te zien zijn. Dit komt omdat de camera erg ingezoomd is op de speler. Ik wil dat de camera switch naar een andere camera op sommige stukken in het level. Hierdoor krijg je een beter beeld van de omgeving en kan je makkelijker navigeren. Met een script heb ik ervoor gezorgd dat je switcht van camera als je een bepaald punt in het level raakt.



## Feedback

Bernd-Jan zei dat dit echt enorm goed bedacht is, het geeft een mooi vloeiend effect en zorgt ervoor dat de speler focust op het level. Hierdoor krijg je meer diepgang in je game. Ook is het nu inderdaad makkelijker om alles goed te zien en te navigeren door je level heen.

## Attacking Enemies

Ik wil dat mijn character kan aanvallen en damage doet tegen een enemy. Hiervoor maak ik een aantal scripts aan. Eerst zorg ik ervoor dat de attack animatie werkt en vervolgens dat er damage van de enemies afgaat wanneer er in de buurt wordt aangevallen. Eerst zag dit er nog stijfjes uit maar uiteindelijk door animaties toe te voegen kreeg ik een best goed werkende punch mechanic.



## Feedback/User testing

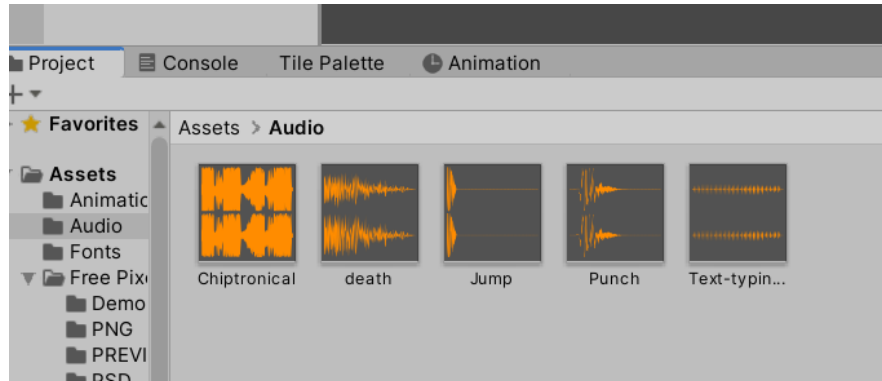
Ik had een rijtje aan enemies achter elkaar gezet zodat mijn medestudenten de punch mechanic konden testen. Hieruit bleek dat je enorm dicht tegen de vijand aan moet staan en dat je heel snel kunt klikken waardoor de enemy binnen een seconde al dood kan zijn.

Daarom heb ik een timer toegevoegd die eerst verlopen moet zijn voordat je opnieuw kunt slaan, ook heb ik de attack range van de slag ietsjes groter gemaakt.



## Music (Background, Jump, Punch)

Nu ik mijn game weer aan Bernd-Jan liet zien zei hij dat alles er erg goed uitzag. Om net wat betere Aesthetics te krijgen kon ik wat muziek toevoegen. Dit kon bijvoorbeeld achtergrond muziek zijn of muziek bij de acties die worden uitgevoerd. Ik moest wel opletten dat dit goed bij de game past en dat ze ook correct zijn qua volume en niet te overheersend zijn.



Ik heb toen wat geluidjes toegevoegd voor het springen, slaan, en doodgaan. Ook had ik een achtergrond muzikje over de hele game heen gezet. Uiteindelijk bleek dat het punch geluidje een beetje irritant werd dus heb ik die weggehaald.

## Level Design V3

Nu ik de meeste mechanics in mijn game heb zitten en deze ook goed getest heb, kan ik levels gaan maken. Ik moet hierbij opletten dat de speler snapt wat hij moet doen en ook geleidelijk de mechanics leert.

### Introductie praten

De speler leert in het begin meteen hoe hij moet lopen en springen. Ook weet hij welke richting hij op moet, hierna zal hij een NPC tegen komen. Hierdoor leert de speler hoe hij moet praten met NPC's. Deze NPC's geven handige tips, zo zegt de eerste NPC dat hij naar de dojo moet gaan die aan de rechterkant is.

### Introductie Spikes

Hierna komt de speler de eerste spikes tegen, er zijn geen andere moeilijke obstakels dus kan de speler rustig bevinden wat deze doen. Het kan zijn dat hij niet ver genoeg springt waardoor hij het springen ook beter onder de knie krijgt. De speler weet nu dat je de spikes niet mag aanraken.

### Introductie springen

Nu de speler voorbij de spikes is zal hij zijn eerste grotere obstakels krijgen, dit is een kloof met wat platformpjes. Deze is nog niet moeilijk waardoor de speler kan wennen aan het springen.

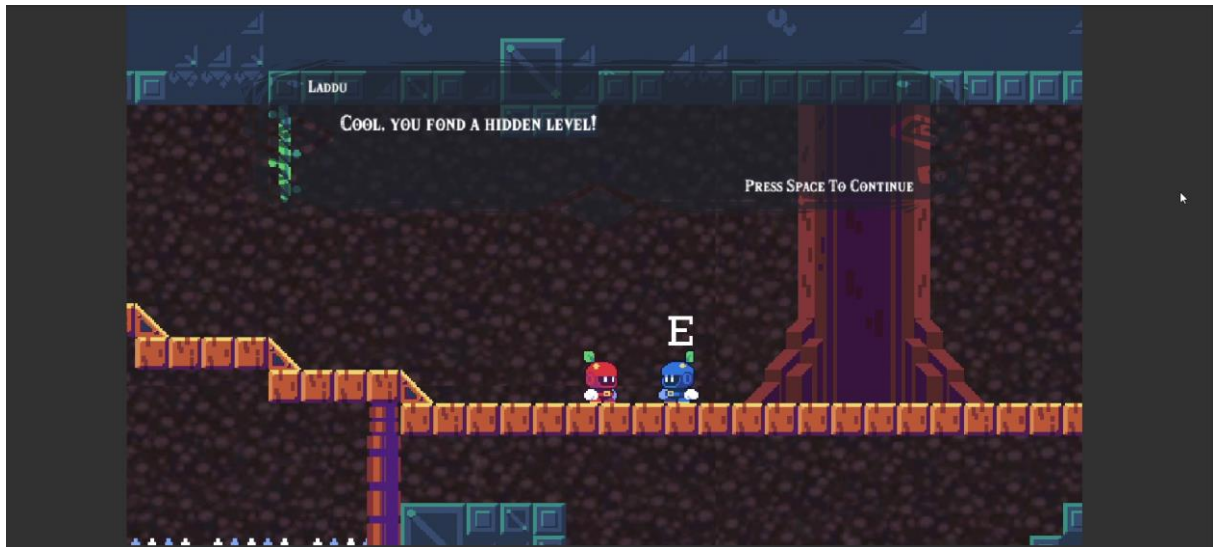
### Introducing enemies

De speler komt nu aan bij de dojo. Hier verteld de NPC dat de speler kan slaan met een muisclick. Ook zegt hij dat rechts wat vijanden zijn om dit op te oefenen. De speler leert zo rustig hoe hij kan slaan en hoe vijanden werken.

### Secret Level

Nu komt de speler bij een moeilijkere kloof met spikes. De sprong die je moet maken kan je amper

halen. Ook heb ik hier een geheim level gemaakt die net uit het beeld is. Later in de game zegt een NPC dat er een geheim level is waardoor spelers misschien terug willen om verder te ontdekken.



### Ending

Op het einde van het level zijn nog wat vijanden die je kan verslaan en een NPC die je bedankt dat je haar gered hebt. Nu het level compleet is kan ik deze gaan testen.



### Build for Html 5 + link

Om het spel meer toegankelijk te maken heb ik er een HTML5 build van gemaakt, hierdoor kan ik hem makkelijk online zetten en kan je hem in je browser spelen. De link naar mijn game is <https://joepvandijk.com/2dplatformerbuild/> (Het wordt aangeraden om hardware acceleratie aan te zetten bij je chrome instellingen)

### User Testing + Feedback

Nu ik mijn game erg makkelijk kon verspreiden heb ik deze naar al mijn vrienden gestuurd en natuurlijk Tom en Bernd-Jan. Ik kreeg enorm veel positieve feedback en zag een lach op het gezicht van mijn vrienden tevoorschijn komen terwijl ze elkaar probeerde te verslaan en het geheime level vonden. Bernd-Jan zei dat het proces en eindproduct er erg goed uitzagen en dat ik veel geleerd had, ook buiten het coderen en maken in Unity om heb ik veel geleerd over Game Design.

Wel vonden sommige vrienden nog een paar kleine foutjes, zo ging het geluid van het praten door als je heel snel er doorheen skipte en hoor je heel vaak het spring geluidje als je snel op W blijft klikken. Verder dan dat heb ik alle gevonden bugs

## Reflectie

Ik had amper ervaring met Unity voordat ik hieraan begon, wel had ik in semester een klein beetje met C# gewerkt maar hier was ik al veel van vergeten. Het was erg fijn om dit op te kunnen frissen voordat ik de specialisatie Game Design ga doen.

Ook weet ik nu dat ik het erg leuk vind om mijn eigen ideeën in een game te kunnen verwerken, ik ben nu erg positief over mijn keuze om Game Design te gaan doen.

Bernd-Jan heeft was enorm enthousiast waardoor ik ook erg gemotiveerd werd, hij stuurde meteen allemaal presentaties uit de specialisatie Game Design, hierin stonden moeilijke begrippen die ik daarna opzocht en verwerkte in mijn Game Design Document. Hierdoor heb ik enorm veel geleerd over het hele proces van een game ontwerpen. Ook wou Bernd-Jan altijd heel graag mijn game testen en gaf uitgebreid feedback en suggesties die ik daarna meteen kon toepassen.

Ik had al vaker user tests gedaan maar niet super vaak, dit keer heb ik bij bijna elke toevoeging wel testen gedaan of feedback gevraagd, hierdoor kreeg ik een steeds beter product die mensen ook fijn en leuk vonden om te gebruiken.

Voor de volgende keer wil ik graag wat meer tijd stoppen in het maken van de game in plaats van het ontwerpen, ik weet nu goed hoe het proces in elkaar zit en hoe belangrijk het is om een goed plan te hebben en onderzoek te doen, maar toch had ik graag wat meer mechanics in mijn game willen stoppen of langere levels maken.

Ik heb vaak vastgezeten met de C# code en Bernd-Jan gaf me dan tips waardoor ik uiteindelijk tijd kon besparen in het maken van mijn levels. Hij was uiteindelijk enorm tevreden over het proces en eindresultaat.