

Leviosa

New Technologies



Ivo van Ooijen – GDS
Melvin van Berkel – GART
Roel Zwakman – GDV
Nathan Flier - GDV

Concept 1:

Voor ons eerste concept kozen wij om de Tobii Eye Tracker te gebruiken. We kozen deze technologie omdat het minder populair was dan andere opties zoals VR en we wilden graag zien wat het kon doen. Na wat testen en research om te kijken wat de Tobii kon doen hadden we een brainstorm. We hadden een redelijk aantal ideeën; Lazer ogen, Waar is Waldo en zelfs een Amazon Drone simulator. Uiteindelijk besloten we dat Wingardium Leviosa, een wizard puzzel game, de leukste applicatie van de Tobii zou zijn. Het had een sterke fantasy met zich verbonden en het was een erg natuurlijke manier van het gebruik van de Tobii (je moet toch kijken naar objecten als je ze wilt beïnvloeden).

We probeerde het concept simpel te houden; je bent een tovenaars, je kan 2 verschillende spreuken gebruiken om puzzels op te lossen. De eerste spreuk is de bekende Wingardium Leviosa (of geooon leviosa) uit de Harry Potter wereld. Met deze spreuk kun je objecten laten zweven door er naar te kijken en dan een knop ingedrukt te houden. Hoe zwaarder het object hoe moeilijker het is om te verplaatsen. De tweede spreuk is Tempus Phohibere, 'Time Lock'. Hiermee kun je een object vastzetten in de tijd waardoor het complete statisch wordt.

Ons eerste concept bestaat uit 1 level, waar de speler op verschillende manieren Leviosa en Tempus moet gebruiken om het einde te bereiken.

Concept 2:

Voor ons tweede concept hadden we verschillende ideeën, maar we wisten al redelijk snel dat wij gewoon op ons concept 1 wilden voortborduren. We hebben opnieuw gekeken naar mogelijke technologieën die we konden gebruiken om ons huidige spel te verrijken. We zaten eerst te denken aan de Leap Motion of the Kinect. Deze zouden we dan gebruiken op een manier dat de speller bepaalde hand tekens of bewegingen moest maken om zijn spreuken te gebruiken. Helaas hoe meer we hier over nadachten hoe meer movement een probleem begon te worden. Als je de Tobii gebruikt samen met een Leap Motion of een Kinect dan heb je niks meer om de camera te besturen of om rond te lopen. Als we deze technologieën zouden gebruiken dan zouden we het concept van ons spel moeten omgooien naar iets wat niet fijn beweeg/camera werk nodig zou hebben. We overwogen ook nog kort om met de linker hand de Leap Motion te gebruiken en met de rechter hand een joystick om de bewegingen te doen, mar dit bleek niet praktisch.

Uiteindelijk kwamen we op het idee om Voice Input te gebruiken. Het viel goed samen met onze fantasy element, gezien je spreuken vaak moet noemen bij hun naam om te gebruiken (ook weer iets wat je duidelijk kan zien in Harry Potter). Tevens zorgde dit ervoor dat de handen compleet vrij waren waardoor er niks aan de camera en bewegingsvrijheid van de speler hoefde te veranderen. Bovenop de Voice Input plande we om een stel nieuwe spreuken erbij te maken zodat we complexere puzzels konden maken. De spreuken die we wilden toevoegen waren een 'teleportatie' spreuk (in de vorm van iets opzuigen end an later weer uitspugen), Een bevries spreuk die iets vast kon zetten voor een paar seconde, en een push spell die een object met grote kracht kon weg schieten.

Als toevoeging op de puzzels hebben we ook een Slime NPC gemaakt dat als obstakel dient voor de speler en als eindbaas wilde we een kwaadaardige tovenaars. Helaas hadden we niet genoeg tijd om nog een eindbaas toe te voegen.

Execution:

In ons eindproduct hebben we helaas niet alles kunnen bereiken wat we wilden. Door tijdsdruk moesten we toch bepaalde spreuken weglaten en de levels wat kleiner maken dan gepland. Het grootste probleem bleek toch onze nieuwe technologie, Voice Input, te zijn. Roel had het goede idee om Windows 10 Cortana te gebruiken gezien die een plugin had voor Unity. Helaas bleek het dat Microsoft de code voor deze plugin heeft weggehaald tijdens de creator's update eerder dit jaar. Daarna hebben we gezocht naar alternatieven maar kwamen alleen bij betaalde services uit. Om onze moeite niet voor niks te laten zijn kwam Roel met een concept voor een DIY Voice Input systeem dat Autohotkey zou gebruiken. Hiermee zou hij naar google voice search gaan in een browser, automatisch op record klikken met de muis, wachten to iets is ingesproken, het resultaat van google kopiëren en deze dan vergelijken met een lijst van commands in Unity. Dit super gammele systeem zou ontzettend sloom zijn, maar een werkend systeem is beter dan geen systeem. Helaas konden we dit niet implementeren omdat Autohotkey de muis nodig zou hebben en dat zou de speler's eigen acties onderbreken.

Een andere uitdaging die we tegenkwamen was tooltips. We wilde dat als je naar een object keek er een klein textboxje tevoorschijn kwam die je een paar eigenschappen van het object vertelde. De grootste uitdaging was deze op het veld krijgen i.p.v. op de HUD omdat als de tooltip op de HUD komt, en de speler kijkt ernaar, zal de tooltip weer verdwijnen omdat de speler niet naar het object keek. Het was voor onze applicatie dus belangrijk dat de tooltip binnen de detection area van het object blijft zodat de speler de tekst kan lezen zonder zijn ogen van het object af te halen.

Task Distribution:

Ivo:

- Concept
- Level Design
- Spell Design
- Presentations
- Documentation

Melvin:

- 3D models
- Textures
- Animations

Nathan:

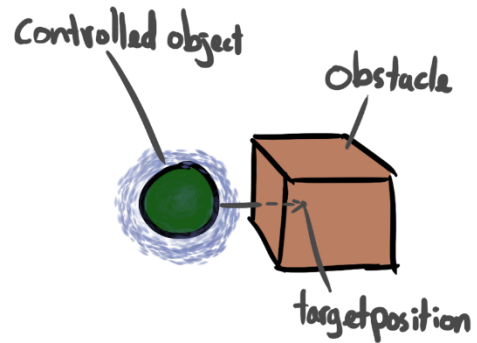
- Game mechanics Framework
- Tobii implementation
- AI Pathfinding
- Art Implementation
- Documentation

Roel:

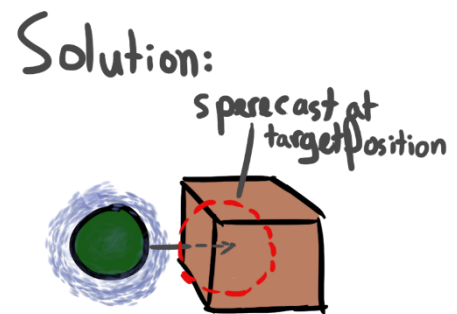
- Particle Effects
- Voice Input research & implementation
- Tech Support

Findings (Problems, Solutions, Exploration):

Zoals al eerder gezegd wouden we maken dat objecten kunnen bewegen door te kijken naar waar ze heen zouden moeten bewegen. De grootste uitdaging waar we tegenaan liepen was dat objecten niet door andere objecten heen zouden mogen bewegen:



Dit hebben we als volgt opgelost: voordat het object bewogen wordt naar de targetPosition (= de positie die de speler aanvraagt om dit object heen te bewegen) voeren we hier een spherecast op uit, en als deze een ander object raakt, dan wordt het object niet naar de targetPosition verplaatst:



Requirements om dit werkend te krijgen:



Hoe dit er in code uit ziet:

```
public void Levitate(Vector3 newLoc, out bool validLevitation){
    if (myInfo.canLevitate) {
        Vector3 offset = newLoc - transform.position;
        Vector3 renderCenter = myRenderer.bounds.center + offset;

        Collider[] colls = Physics.OverlapSphere (renderCenter, radius, staticLM);
        if (colls.Length > 0) {
            foreach (Collider c in colls) {
                if (!c.isTrigger) {
                    validLevitation = false;
                    return;
                }
            }
        }

        transform.position = newLoc;
    }

    validLevitation = true;
}
```

Wat we nog graag had willen toevoegen:

-werkende voice commands, al hebben we geen idee hoe

-een blur over het scherm als je aan het levitaten bent. Het enige deel van het scherm dat niet geblurred is, is op het Movable Object en een beetje hier omheen. Wij zijn erg benieuwd of dit het magische gevoel zou kunnen versterken.

-meer AI's en vooral ook meer dat het sense maakt, bij het level dat we nu hadden speelde de AI nou eenmaal geen belangrijke rol en dit is jammer aangezien het ervoor zorgen dat deze AI goed werkte veel tijd heeft gekost. Het zou veel toe kunnen voegen om dan ook een andere soort AI erin te verwerken omdat dit de diversiteit en de spanning van het spel enorm kan vergroten.

-dat je AI's kunt damagen door objecten tegen ze aan te gooien.

-audio. Het liefst zou de muziek dan ook veranderen als je bij een ander deel van het spel komt, zodat het beter bij de sfeer kan passen en zo de immersion kan vergroten. Sound effects zijn natuurlijk ook altijd goed!