

NEW TECHNOLOGY

{ DOCUMENTATION }

INLEIDING

HET EXPERIMENT STOND IN HET TEKEN VAN MOTION SICKNESS. HET DOEL VAN DIT EXPERIMENT IS DE GRENZEN WEERLEGGEN ERVAN, STIMULATIE DUS. WAAR WE VOORAL INTERESSE IN HADDEN WAS HET GEBRUIKEN VAN IMAGE EFFECTS. NATUURLIJKE EFFECTEN ZOALS DEPTH OF FIELD, FIELD OF VIEW, FISH-EYE ZIJN ELEMENTEN DIE WE ZEKER ZULLEN TOEPASSEN. ECHTER WIL IK OOK GEBRUIK MAKEN VAN SURREALISTISCHE EFFECTEN OM NOG EEN STAPJE VERDER TE GAAN.

HET LIJKT MISSCHIEN EEN WAARDELOOS EXPERIMENT, OMDAT HET EEN NEGATIEVE UITKOMST HEEFT, MAAR DAT KAN OP EEN POSITIEVE MANIER WORDEN INGEZET. BIJVOORBEELD IN-GAME TIJDENS EEN SCENARIO WAARBIJ DE SPELER SCHADE OPLOOPT EN DEZE TECHNIEKEN WORDEN GEBRUIKT OM DE IMMERSIE ERVAN TE VERGROTEN.

BUITEN HET THEMA VIRTUAL REALITY WIL IK OOK ONDERZOEK DOEN NAAR GENERATIVE PATHS. HET DOEL IS EEN EINDELOZE PATH KUNNEN MAKEN UIT VERSCHILLENDE OBJECTEN, ONGEFACHT FORMAAT/AFMETING. DAARBIJ WIL IK ERVOOR ZORGEN DAT HET GENEREREN SOEPEL VERLOOPT OP EEN EFFICIENTE MANIER, DOOR OBJECTEN VOOR HET PAD, AAN TE MAKEN IN EEN STACK. DEZE TOEPASSING HEB IK VOORAL WILLEN UITVOEREN, OMDAT WIJ MET EEN ANDERE MODULE OOK BEZIG ZIJN MET DIT FENOMEEN.

DISCLAIMER:

IN DIT VERSLAG ZAL IK MIJN **BEVINDINGEN** VERWERKEN IN HET PROCESS.

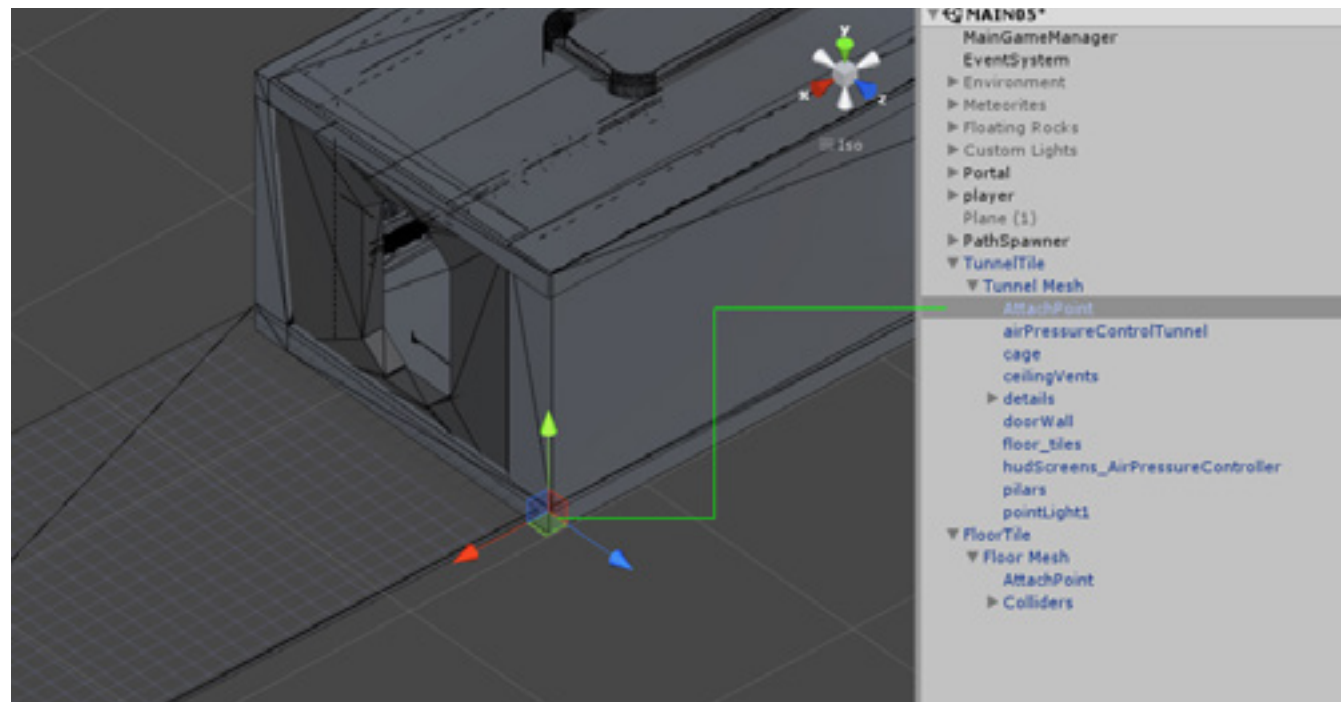
NEW TECHNOLOGY

{ DOCUMENTATION }

GENERATING PATH

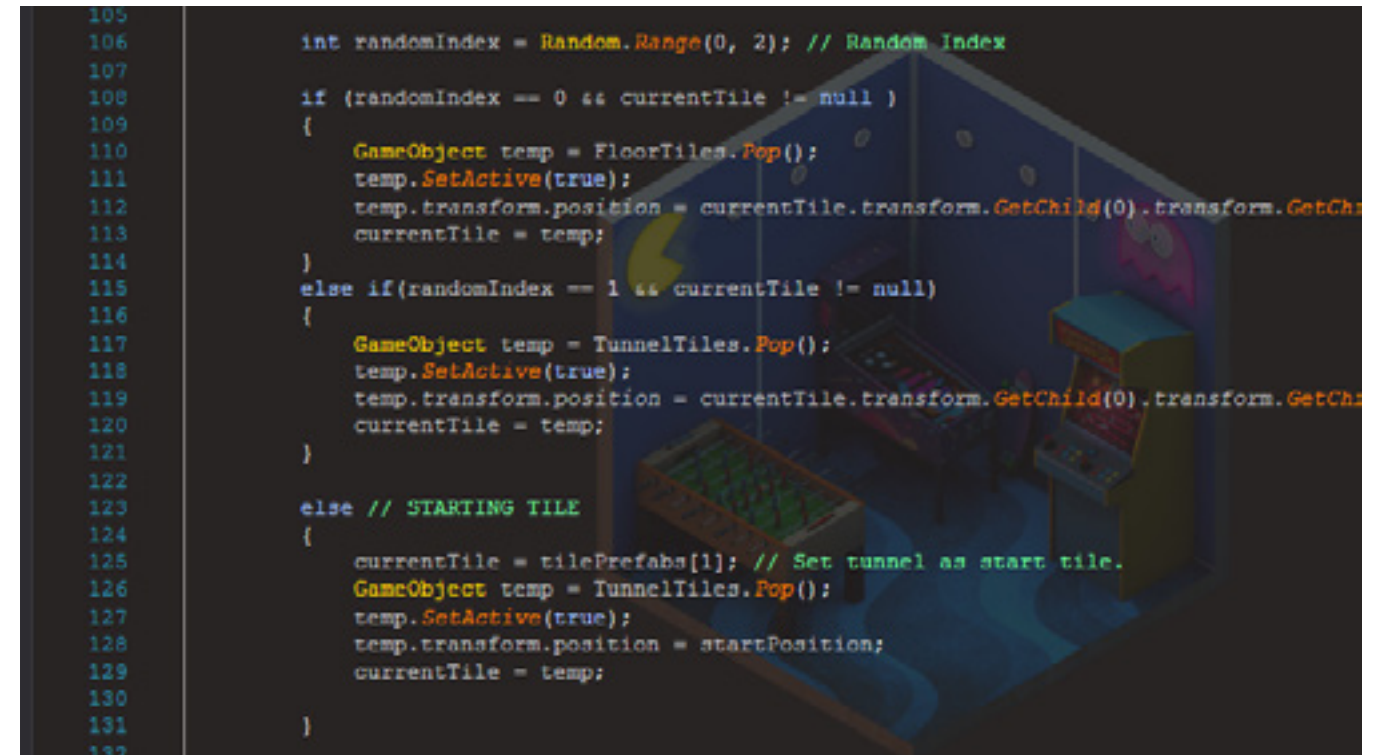
OM MIJN EFFECTEN TE TESTEN HAD IK HET IDEE OM EEN INFINITE PATH TE MAKEN DIE VANZELF WORD GEGENEREERD. DAARBIJ HAD IK EEN DOELSTELLING OM HET SYSTEEM ZO TE BOUWEN, DAT IK VERSCHILLENDE TILES KAN TOEVOEGEN, ONGEACHT HET FORMAAT/AFMETING.

DAARVOOR HAD EEN TECHNIEK NODIG MET TWEE ANCHORPOINTS. EEN VOOR HET BEPALEN DE SPAWNPOINT EN EEN OM TE AAN TE GEVEN WAAR DE NIEUWE TILE MOET WORDEN GESPAWND. DIT DOORMIDDEL VAN EMPTY GAMEOBJECTS.



BIJ HET MAKEN VAN EEN NIEUWE TILE MOET IK DAN DUS VERWIJZEN NAAR DIE ATTACHPOINT. DAT HEB IK GEDAAN DOOR MIDDEL VAN DE FUNCTIE `GETCHILD()`.

ECHTER MOEST IK OOK EEN STARTPUNT HEBBEN. DOOR EEN ELSE-STATEMENT TE GEBRUIKEN, WANNEER ER NOG GEEN CURRENT-TILE IS AANGEWEZEN, KIES IK ERVOOR OM DE TUNNEL-PREFAB TE GEBRUIKEN EN DEZE TE PLAATSEN OP DE STARTPOSITION, DIE WORDT BEPAALD DOOR DE PATHSPAWNER OBJECT.



DAARNAAST HAD IK OOK HET IDEE OM DE OBJECTEN TE LATEN VALLEN, ZODAT DE SPELER NIET MEER TERUG KAN. DAT HEB IK SIMPELWEG KUNNEN BEREIKEN DOOR `ISKENMATIC()` OP FALSE TE ZETTEN. OM TE BEPALEN WANNEER DAT MOET GEBEUREN, HEB IK DE TRIGGER TOEGEPAST IN DE `ONTRIGGEREXIT()` MET EEN COROUTINE ZORG IK ERVOOR DAT ER GENOEG TIJD TUSSEN ZIT.



NEW TECHNOLOGY

{ DOCUMENTATION }

EFFICIENCY

TIJDENS HET ‘HEAVY OBJECT ORIENTATED PROGRAMMING’ HEBBEN WE HET GEHAD OVER HEAP EN STACK. DIT ZORGT ERVOOR DAT DE OBJECTEN WORDEN AANGEWEZEN NAAR EEN SPECIFIEKE LOCATIE IN HET GEHEUGEN. HET TOEPASSEN ERVAN ZORGT ERVOOR DAT ER REFERENTIE IS, WAARDOOR ER NIET OPNIEUW EEN LOCATIE MOET WORDEN AANGEWEZEN. DIT WOU IK TOEPASSEN EN DAARVOOR HET IK EENAANTAL TUTORIAL GEVOLGD.

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=AZKRPC1SMJC](https://www.youtube.com/watch?v=AZKRPC1SMJC)

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=W-8UAPPL6JY](https://www.youtube.com/watch?v=W-8UAPPL6JY)

UITEINDELIJK HEB IK ERVOOR KUNNEN ZORGEN DAT DE OBJECTEN INACTIVE TERUG WORDEN GEPLAATST IN DE STACK (MET ISKINEMATIC TERUG OP TRUE, ZODAT ZE NIET NAAR BENEDEN VALLEN WANNEER ZE WEER TERUG WORDEN GESPAWND) WANNEER DE SPELER OVER HET OBJECT/PAD HEEN IS GELOPEN. HET OBJECT WORD DAN OPNIEUW AANGEROEPEN TIEN OBJECTEN/PADEN VERDER.

OM TE BEPALEN WELKE TILE ER MOET WORDEN GEPUSHT IN DE STACK HEB IK GEBRUIK GEMAAKT VAN DE `SWITCH()` FUNCTIE. DE CASE WORD BEPAALD DOOR DE NAAM VAN HET OBJECT.

OM DE PREFABS AAN DE PREFABS-ARRAY TOE TE VOEGEN HEB IK GEBRUIK GEMAAKT VAN `TILEPREFABS[0] = (GAMEOBJECT)RESOURCES.LOAD("PREFABS/FLOORTILE")`; IN PLAATS VAN HET PUBLIC MAKEN EN HANDMATIG TOE TE VOEGEN. ECHTER MOEST IK DAARVOOR WEL DE GROTE VAN DE STACK BEPALEN. DOOR `TILEPREFABS = NEW GAMEOBJECT[2];` TE GEBRUIKEN. BIJ PUBLIC WORD DAT ECHTER AUTOMATISCH GEDAAN WANNEER JE OBJECTEN TOEVOEGD.

OM DE SPAWNTILE FUNCTIE AAN TE ROEPEN IN ANDERE SCRIPTS HEB IK VAN DE `TILESPAWNER` EEN SINGLETON GEMAAKT. OP DEZE MANIER KON IK SIMPELWEG DE FUNCTIES AANROEPEN DIE IK NODIG HAD IN ANDERE SCRIPTS (`TILESPAWNER.INSTANCE.SPAWNTILE()`).

```
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
private Stack<GameObject> floorTiles = new Stack<GameObject>();

public Stack<GameObject> FloorTiles
{
    get { return floorTiles; }
    set { floorTiles = value; }
}

private Stack<GameObject> tunnelTiles = new Stack<GameObject>();

public Stack<GameObject> TunnelTiles
{
    get { return tunnelTiles; }
    set { tunnelTiles = value; }
}
```

```
84
85
86
87
88
89
90
91
92
93
94
95
96
97
// CREATE TILES ()
public void CreateTiles(int num)
{
    for (int i = 0; i < num; i++)
    {
        FloorTiles.Push(Instantiate(tilePrefabs[0]));
        TunnelTiles.Push(Instantiate(tilePrefabs[1]));

        FloorTiles.Peek().SetActive(false);
        TunnelTiles.Peek().SetActive(false);
    }
}
```

```
19
20
21
22
23
24
25
26
27
28
void OnTriggerExit(Collider other)
{
    if (other.Tag == "Player")
    {
        tileSpawner.Instance.SpawnTile();
        StartCoroutine(FallDown());
    }
}
```

NEW TECHNOLOGY

{ DOCUMENTATION }

IMAGE EFFECTS

HET TOEPASSEN VAN DE IMAGE-EFFECTS WAS EEN VRIJ LASTIG PROCESS. ALLEREERST WIST IK NIET DAT JE DAARVOOR EEN LIBRARY MOEST AANROEPEN. EENMAAL TOEGEVOEGD: `USING UNITYSTANDARDASSETS.IMAGEEFFECTS;` , KREEG IK HET NOG STEEDS NIET AAN DE PRAAT. IK HEB DAAROM DE ULTIEME LIBRARY TOEGEVOEGD, NAMELIJK: `USING VALENTIJNSASSETS.COCAINSTASH;`. TOT MIJN VERBAZING KREEG IK HET AANROEPEN TOEN WEL WERKENDE.

VOOR DE CAMERA HEB IK EEN APARTE SCRIPT GEMAAKT VOOR DE EFFECTEN. HET DOEL VAN DIT SCRIPT IS OM DE WAARDE VAN DE EIGENSCHAPPEN VAN DEZE EFFECTEN TE LATEN OSCILLEREN. DEZE OSCILLATIE MOET GEBEUREN BINNEN EEN SPECIFIEKE `RANGE()` PER EIGENSCHAP.

HET OSCILLERENDE EFFECT HEB IK KUNNEN BEHALEN DOOR HET GEBRUIKEN VAN DE `PINGPONG()` METHODE ONDER `MATHF`. ECHTER LEEK HET MIJ GAAF ALS IK DEZE RANGE OOK `RANDOM` ZOU KUNNEN MAKEN BINNEN EEN MAXIMALE RANGE.

DAARVOOR HEB IK PER EIGENSCHAP, EEN RANDOM VALUE GEGENEREERD. DEZE WAARDE VERMENIGVULDIG IK MET EEN STANDAARD FLOAT WAARDE. DEZE STANDAARD WAARDE WORD BEPAALD DOOR DE MAXIMALE OUTPUT VAN DE RANDOM VALUE, VERMIGVULDIGD MET DE STANDAARD WAARDE. OP DEZE MANIER ZIT DE RANGE ALTIJD TUSSEN DE MAXIMALE WAARDE IN, MAAR WEL RANDOM.

ECHTER LIEP IK WEL TEGEN EEN PROBLEEM AAN. OP EEN GEGEVEN MOMENT RAAKT DE WAARDE CORRUPT EN KRIJG IK '`ISNAN`' TERUG UIT DE BEREKENING. DAAR HEB IK OPLOSSING VOOR GEVONDEN.

DOOR DE FUNCTION `IF(FLOAT.ISNAN(VALUE))` TE GEBFRUIKEN, KAN IK CONTROLEREN OF DE WAARDE CORRUPT IS. OMDAT DIT REGELMATIG GEBEURD HEB IK, INPLAATS VAN EEN `COROUTINE`, DEZE FUNCTIE GEBRUIKT OM DE RANDOM VALUES OPNIEUW TE GENEREREN.

```
void Update ()
{
    vortexEffect.angle = Mathf.PingPong(Time.time * 20, randomRange1 * 10.0f) + 0.01f ;
    fisheyeEffect.strengthX = Mathf.PingPong(Time.time * 40, randomRange2);
    fisheyeEffect.strengthY = Mathf.PingPong(Time.time * 40, randomRange3);

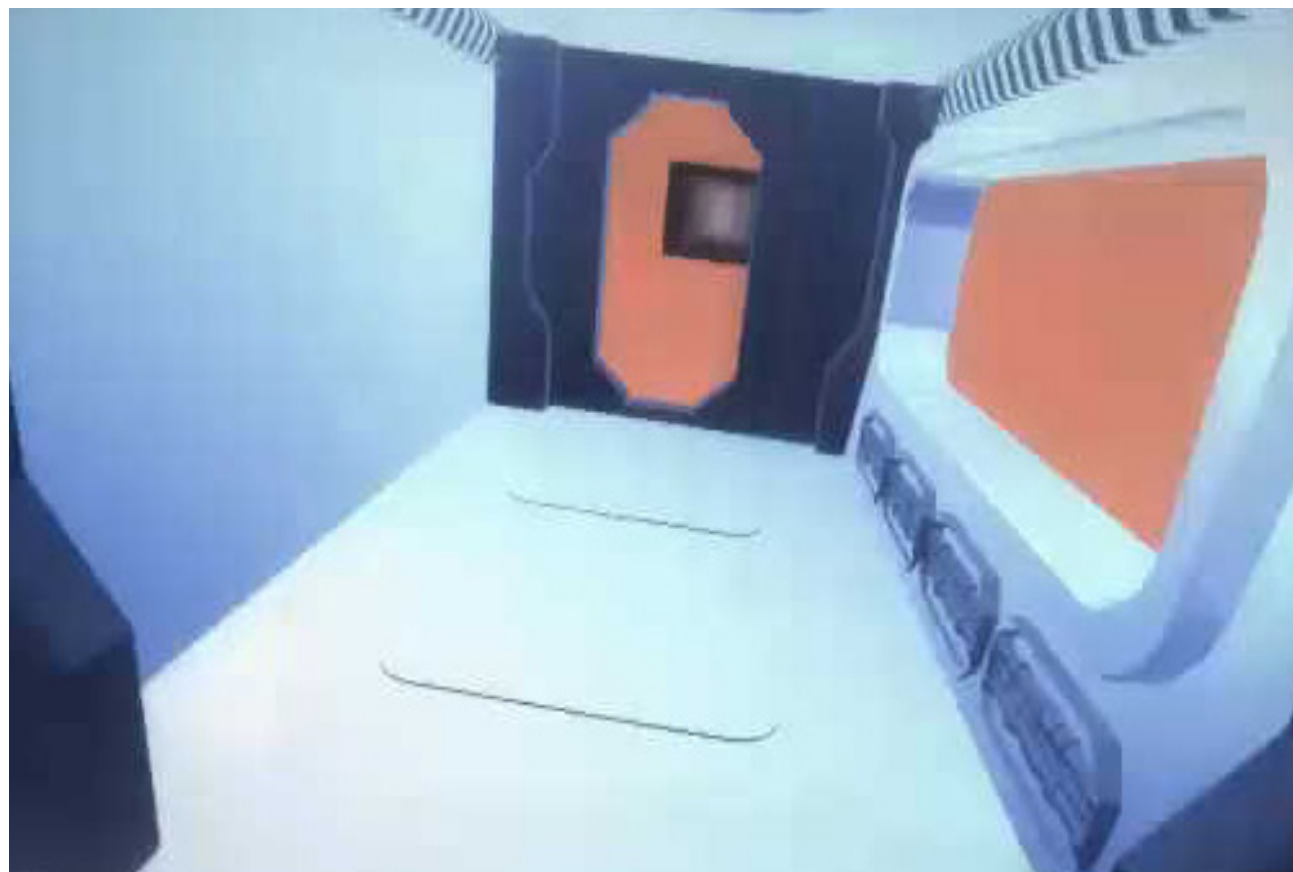
    dof.focalLength = Mathf.PingPong(Time.time * 20, dofFocalLenght + randomRange4 * 10.0f); ;
    dof.focalSize = Mathf.PingPong(Time.time * 20, dofFocalSize + randomRange4 * 10.0f); ;
    dof.aperture = Mathf.PingPong(Time.time * 20, dofAperture + randomRange4 * 10.0f); ;

    bloom.bloomThreshold = Mathf.PingPong(Time.time * 0.5f, randomRange5 * 1.2f) + 0.01f;
    bloom.bloomIntensity = Mathf.PingPong(Time.time * 0.5f, randomRange6 * 1.42f) + 0.01f;

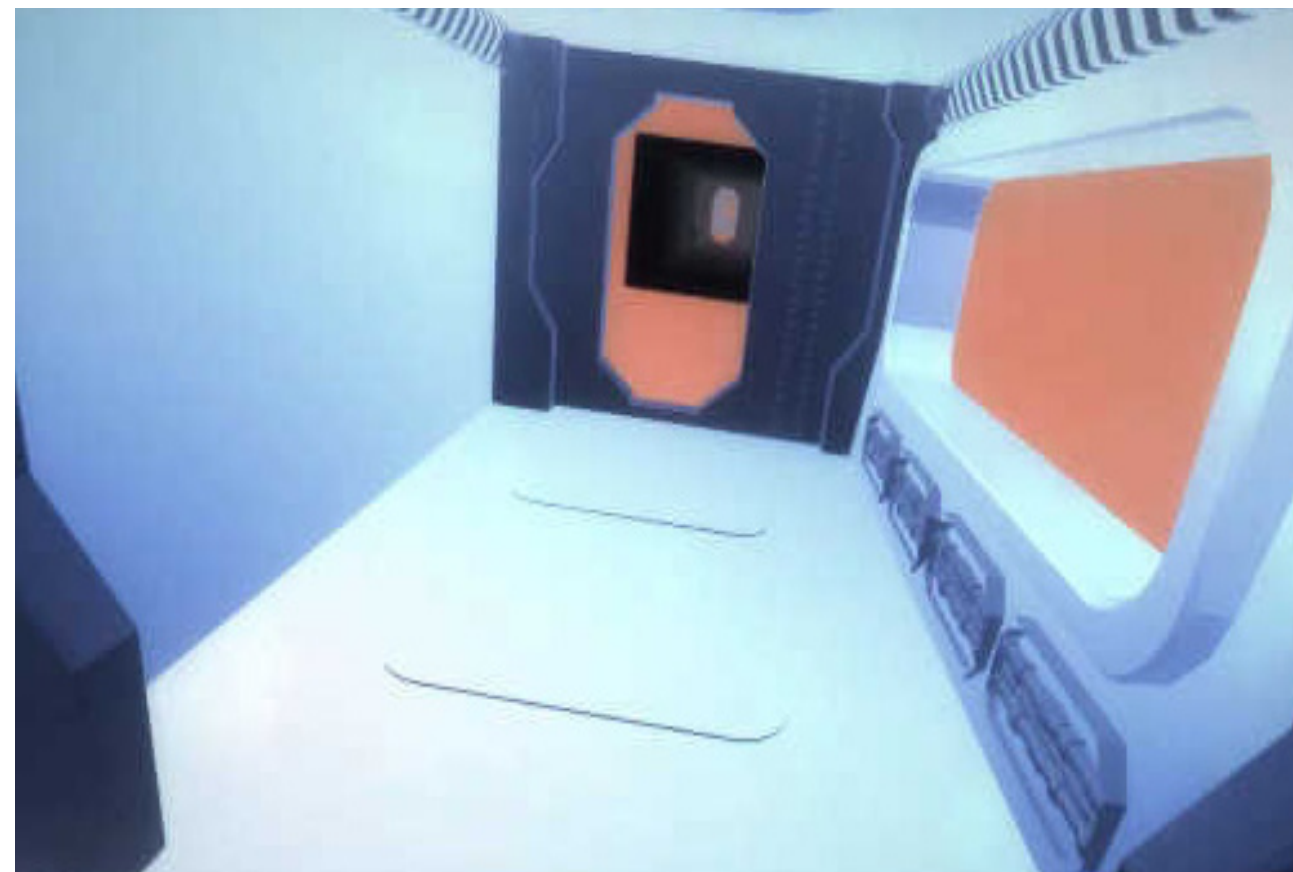
    if (float.IsNaN(vortexEffect.angle) || float.IsNaN(bloom.bloomIntensity))
        // Check if isNaN! Which happens randomly!
        // Set new randomRange values and
    {
        randomRange1 = Random.Range(-5, 5);
        randomRange2 = Random.Range(0, 0.2f);
        randomRange3 = Random.Range(-0.4f, 0.4f);
        randomRange4 = Random.Range(-5, 5);
    }
}

public void DestroyValentijn()
{
    /*
    if(currentplayer.name == "Valentijn")
    {
        Application.Quit();
    }
    */
}
```

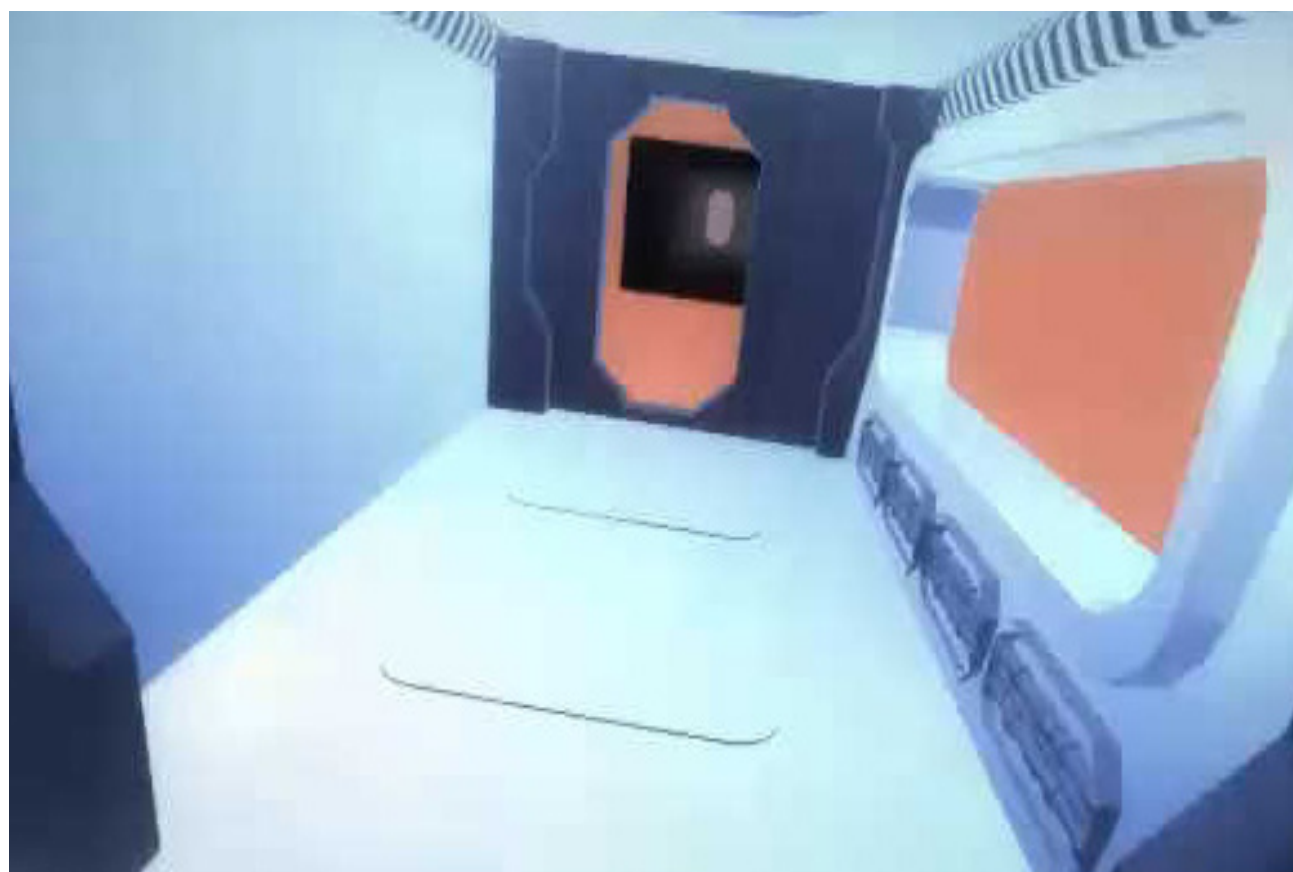

BLOOM



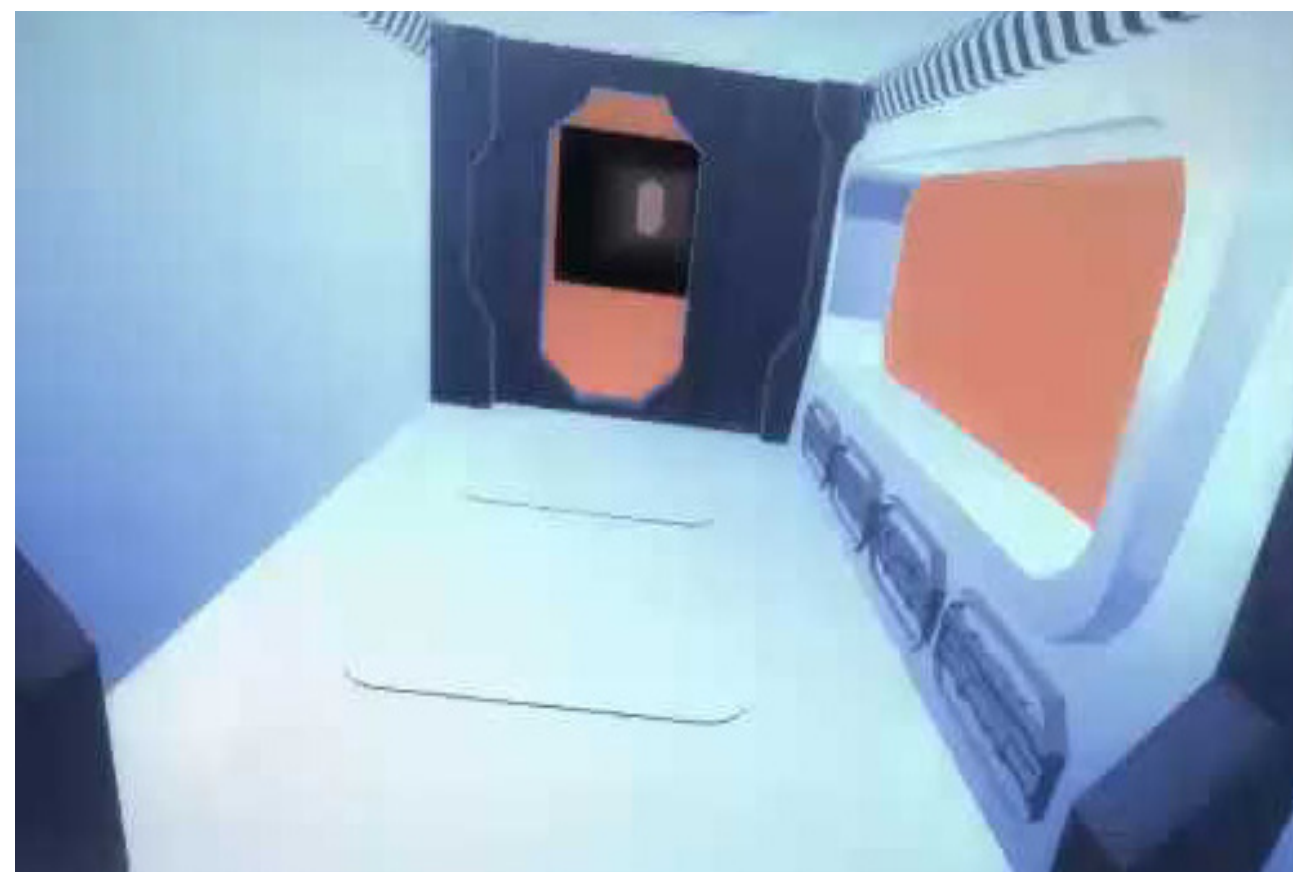
DEPTH OF FIELD MOTION



FISHEYE



VORTEX



NEW TECHNOLOGY

{ DOCUMENTATION }

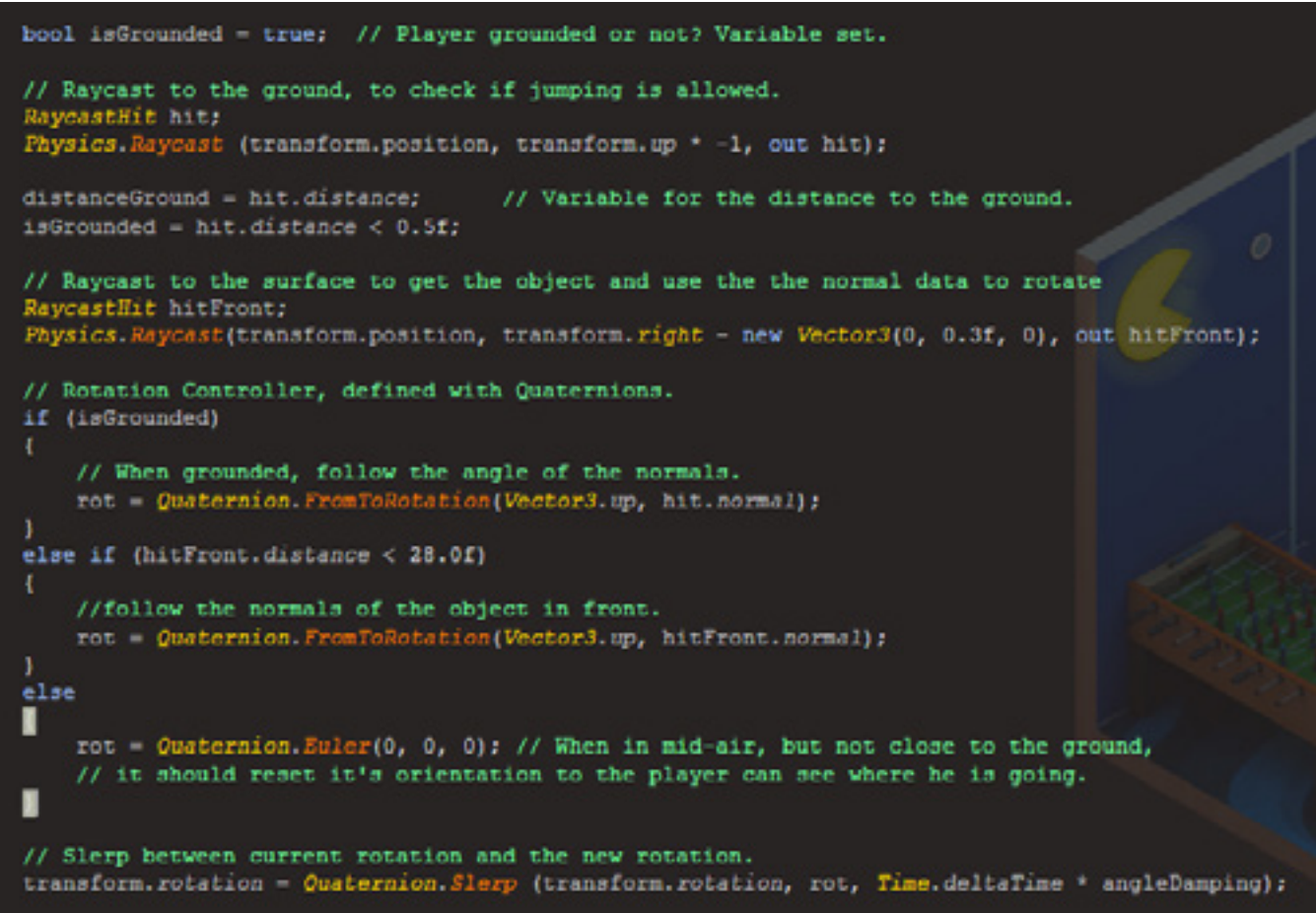
MOTION

VOOR DE BEWEGING IN DE VIRTUELE RUIMTE HEB IK EEN CONTROLLER GEBRUIKT DIE IK HEB GESCHREVEN TIJDENS EEN ANDER PROJECT. HET CONCEPT IS ALS VOLGT:

- ER IS EEN CONSTATE FORCE NAAR VOREN.
- DE SPELER KAN DE FORCE VERGROTEN DOOR OP DE 'W' TOETS TE DRUKKEN.
- DE ROTATIE WORD BEPAALD DOOR MIDDEL VAN DE MUISPOSITIE, IN HET GEVAL VAN VR IS DAT DE LOOK ROTATION
- DE SPELER IS GELOCKT IN DE X EN Z AXIS, ZODAT DE ROTATIE ALLEEN OP DE HORIZONTALE AS MOGELIJK IS.
- DE FIELD OF VIEW KAN WORDEN AANGEPAST MET DE RIGHTMOUSE KEY.

MOTION SICKNESS ONTSTAAT O.A. DOOR VERPLAATSING IN DE RUIMTE ZONDER EEN REFERENTIE OF HANDELING WAAR JE CONTROLLE OVER HEBT. WAT IK HEB GEDAAN IS DE CONSTATE FORCE HEEL LAAG ZETTEN, ZODAT ER ALTIJD BEWEGING IS. OM JEZELF ECHT TE VERPLAATSEN MOET DE 'W' TOETS DRUKKEN. OP HET MOMENT DAT JE STOPT MET VERPLAATSEN, IS ER NOG STEEDS EEN MINIMALE VERPLAATSING WAARDOOR HET LIJKT ALSOF JE DOORZWEEFT. DAT NEEMT HET GEVOEL VAN 'CONTROLE' WEG. EEN ANDERE STIMULUS VOOR MOTION SICKNESS.

DAARNAAST HEB IK EEN RAYCAST TECHNIEK GEBRUIKT WAARDOOR JE ORIENTATIE VERANDERD AAN DE HAND VAN WAAR JE HEEN KIJKT. DOOR MIDDEL VAN RAYCAST HIT ZORG IK ERVOOR DE CAMERA ZIJN ORIENTATIE PROBEERT GELIJK TE MAKEN AAN DIE VAN DE OBJECT NORMALS TEGENOVER.



```
bool isGrounded = true; // Player grounded or not? Variable set.

// Raycast to the ground, to check if jumping is allowed.
RaycastHit hit;
Physics.Raycast (transform.position, transform.up * -1, out hit);

distanceGround = hit.distance; // Variable for the distance to the ground.
isGrounded = hit.distance < 0.5f;

// Raycast to the surface to get the object and use the the normal data to rotate
RaycastHit hitFront;
Physics.Raycast(transform.position, transform.right - new Vector3(0, 0.3f, 0), out hitFront);

// Rotation Controller, defined with Quaternions.
if (isGrounded)
{
    // When grounded, follow the angle of the normals.
    rot = Quaternion.FromToRotation(Vector3.up, hit.normal);
}
else if (hitFront.distance < 28.0f)
{
    //follow the normals of the object in front.
    rot = Quaternion.FromToRotation(Vector3.up, hitFront.normal);
}
else
{
    rot = Quaternion.Euler(0, 0, 0); // When in mid-air, but not close to the ground,
    // it should reset it's orientation to the player can see where he is going.
}

// Slerp between current rotation and the new rotation.
transform.rotation = Quaternion.Slerp (transform.rotation, rot, Time.deltaTime * angleDamping);
```

NEW TECHNOLOGY

{ DOCUMENTATION }

TESTING

HET TESTEN HEB IK LATEN DOEN DOOR VERSCHILLENDE MEDE STUDENTEN EN MIJN VRIENDIN. NA IEDERE TOEVOEGING HEB GEVRAAGD AAN DE GEBRUIK OF ZIJ HET WEL OF NIET ALS VERVELEND OF MISSELIJKMAKEND ERVAREN. VOORAL DE VORTEX EN FISHEYE EFFECT HADDEN EEN GROTE IMPACT. WAT OOK OPVIEL ZIJN DE CONTROLS, DIE MAKEN HET HEEL VERVELEND. BIJ IEDERE GEBRUIKER VIEL HET MIJ OP DAT ZIJ IEDER TEGEN DE MECHANICS VOCHTEN.

STIEKEM IS ER EEN MANIER OM DIT GEVECHT TE VOORKOMEN, NAMELIJK WACHTEN RECHT VOORUIT BLIJVEN KIJKEN IN HET EINDELOZE EN JE NIET LATEN AFLEIDEN DOOR WAT JE ZIET OP HET SCHERM. MAAR DAT VERTEL JE NATUURLIJK NIET, DAT ZOU DE ERVARING VERPESTEN.

SPECIAL THANKS

HIERBIJ WIL IK NOURA EN BASTIAAN BEDANKEN VOOR DE STEUN EN LIEFDE. ZONDER HUN WAS DIT PROJECT NIET ZO VER GEKOMEN. OOK WIL IK SHOUT OUT DOEN NAAR CODY, ZONDER HEM ZOU IK VECTORS, QUATERNIONS EN NORMALS NIET ZO GOED HEBBEN BEGREPEN.