

INSTITUTIONEN FÖR TEKNIK OCH NATURVETENSKAP

LINKÖPINGS UNIVERSITET

TNM061: 3D-GRAFIK, VT 2019

Interaktiv 3D-miljö i webbläsaren

20 maj 2019

Författare

PHILIP NGO: *phing272@student.liu.se*

JOEL PAULSSON: *joepa811@student.liu.se*

SAMUEL LARSSON: *samla949@student.liu.se*

DAVID SUNDBERG: *davsu041@student.liu.se*

Examinator

STEFAN GUSTAVSON: *stefan.gustavson@liu.se*

1 Inledning

I det här projektet har en interaktiv 3D-miljö i *low-poly* stil utvecklats. Programkoden är skriven i JavaScript tillsammans med biblioteket THREE.js. Genom att använda THREE.js kunde mycket låg-nivå programmering abstraheras bort till förmån för det visuella, vilket var önskvärt då projektgruppen var intresserade av hur själva 3D-miljön skulle programmeras.

För att skapa olika modeller användes Blender, en gratis programvara för modellering och animering.

I 3D-miljön kan användaren navigera ett flygplan i en "oändlig" värld bestående av vatten och öar som har träd, stenar och vindkraftverk.

2 Metod

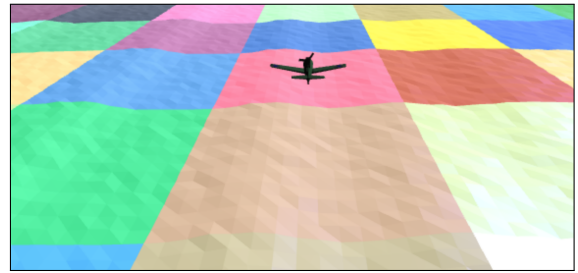
Projektgruppen delade upp arbetet i två delar, efter intresse. En person ansvarade för modellering medan övriga jobbade med implementeringen. Gruppen har vid flera tillfällen arbetat tillsammans vilket gav en god översikt av projektets status. Arbetsuppgifter har även delats ut för individuellt arbete och arbete i mindre grupper. I projektet har kod-editorn Visual Studio Code använts för att på ett smidigt sätt kunna arbeta tillsammans i samma filer genom tilläggs paketet Live Share när det har behövts. Det tillåter användarna att förutom att programmera i samma fil, dela filstrukturen, lokala portar och terminalen. Projektet har versionshanterats med systemet Git och lagrats på Linköping Universitets GitLab-server. Det har strävats efter att alltid ha en fungerande *master branch*. När ny funktionalitet utvecklats har det skapats en ny (*branch*) från *master* och senare synkroniserats med *master branch*-en. Arbetssättet har lätt till att utveckling i olika delar har kunnat genomföras parallellt med få kodkonflikter.

Utvecklingsmiljön är uppsatt med hjälp av pakethanteraren npm och verktyg som webpack och gulp. Detta för att på ett effektivt sätt strukturera upp koden i olika filer, alltså klasser/moduler.

2.1 Implementering

En central del i den skapade miljön är flygplanet. Kamerans position och vad som ska visas i världen ändras beroende på flygplanets position. Hur planet ska flyttas bestäms genom att kontrollera vilka knappar som är nedtryckta och sedan räkna ut hur mycket i x ; y ; z -led flygplanet ska förflyttas.

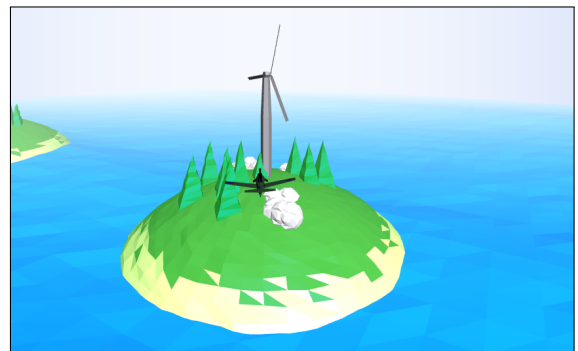
Världen som flygplanet åker i är slutet av en så kallad *sky-box* där väggarna alltid är lika långt bort från planet och flygplanet kommer aldrig nå fram till väggarna. Vattenytan består av ett rutnät där rutorna flyttas beroende åt vilket håll planet rör sig. Nya plattor skapas aldrig, utan plattornas position och egenskaper ändras beroende på planets position. En bild över planet och plattorna i förminskad storlek visas i Figur 1.



Figur 1: Flygplanet och plattorna som flyttas i planets färdriktning.

Vågorna skapades med hjälp av en endimensionell *noise*-funktion som applicerades på alla vertexpunkter i x -led. Varje platta har 16 vertexpunkter i varje led. För att få vågorna att se "bättre" ut applicerades även *noise* på alla vertexpunkter på varje platta, vilket simulerar störningar på vattnet. *Noise*-funktionen som används är inte utvecklad av gruppen, utan är en lånad implementation [1].

Öarna tillsammans med träden och stenarna är slumpmässigt genererade genom att använda färdiga funktioner i THREE.js som sfärer, koner och dodekaeder. Själva öarna är avklippta sfärer där vertexpunkterna flyttades med ett slumpmässigt tal vilket gjorde att sfären fick en mer ojämn yta för att efterlikna en ö snarare än ett klot. Se Figur 2.



Figur 2: Ett exempel på en slumpvis genererad ö.

2.2 Modellering

För att skapa flygplanet och vindkraftverken användes Blender. Att förstå sig på Blender var en utav de sakerna som tog mycket tid av projektet, men gruppen ville fortfarande använda sig av Blender då det var *open source* och enligt olika forum lätt att exportera objekt till Three.js. För att skapa planet så användes en *blueprint* som hittades online. *Blueprint*-en och det färdiga planet visas i Figur 3 & 4.

Från början så skapades hela flygplanet för hand men när gruppen såg att planet inte riktigt såg ut som önskat. Istället användes *mirror* funktionen vilket gjorde planet helt symmetriskt. Flygplanet skapades och importerades som två separata delar; flygplanskroppen och propellern. Eftersom ett vindkraftverk har en lite mer simpel design och kunskapen

