

RESUM



En el següent projecte podrem trobar un tema poc desenvolupat en el món dels videojocs, la pluja en temps real. Veurem que el projecte permetrà al desenvolupador de videojocs incorporar a les seves aplicacions, zones de pluja amb diferents intensitats utilitzant el hardware gràfic més modern, per així evitar que aquesta pluja sigui processada per la CPU i per tant pugui alentir el videojoc que està creant. Ens vam decidir a desenvolupar el present projecte al veure:

- Que no trobàvem aplicacions que permetessin al desenvolupador escollir una zona de pluja per la seva aplicació.
- Al veure que els sistemes de pluja incorporats en els diferents videojocs o simulacions que hi ha actualment en el mercat, no eren tractats des del hardware gràfic.
- Al observar que es podia millorar el comportament i l'aparença de les gotes que hi ha en els sistemes de pluja actuals.

Tot això ens va fer crear el present projecte de simulació en temps real de pluja realista.

Per aconseguir els objectius citats, s'ha hagut de desenvolupar dos sistemes ven diferenciats, el sistema d'edició de pluja i el sistema de visualització en temps real.

- **Sistema d'edició de pluja:** Mitjançant un sistema de finestres desenvolupat amb la llibreria wxWidgets, s'ha desenvolupat un programari que permetrà al dissenyador definir, per a una escena, les àrees on plourà i amb quina intensitat plourà. Concretament hi hauran 255 intensitats diferents d'intensitats. Aquest mòdul generarà, doncs, la informació necessària per la segona part de l'aplicació. Aquestes són les funcions bàsiques que realitzarà l'editor:
 - Permetrà carregar i guardar qualsevol tipus d'imatge (tot i que l'objectiu és que es carreguin les imatges capturades de l'escena de l'aplicació en temps real, per així seleccionar les zones en què es vol fer ploure). Acceptarà una llista important de formats (com per exemple els formats jpg, gif, png, bmp, tiff etc.).
 - Observant la imatge de l'escenari escollit, aquesta, es modificarà dibuixant figures a les zones en què es vol que caigui pluja (*Veure figura 1*). S'han creat tres tipus de figures que combinant-les permetran omplir tots els racons de la imatge. Es podrà dibuixar:
 - Cercles.
 - Rectangles.
 - Polígons de x costats.

- Permetrà modificar les figures dibuixades. Per aconseguir un contrast menor entre les diferents intensitats de pluja, per així obtenir un major realisme en l'aplicació en 3D.
- Permetrà escollir el número exacte de gotes d'aigua que es desitja que caiguin en l'aplicació en temps real. Aquestes cauran en la zona dibuixada en aquest editor. Les quals es distribuiran gràcies a un algorisme que s'aprofita de les funcions de densitat de probabilitat (PDF) i de distribució acumulativa (CDF).

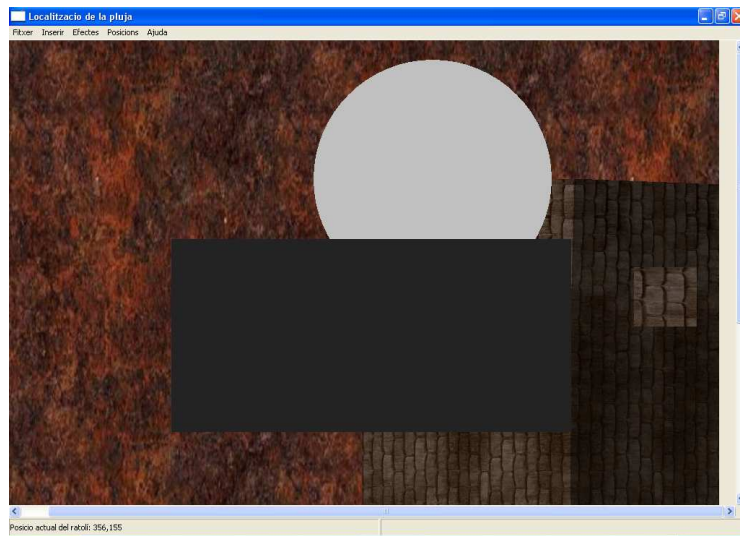


Figura 1. Podem observar l'editor amb una imatge modificada

- Permetrà generar una sola imatge on s'hi guardarà un conjunt d'imatges realistes de gotes d'aigua amb una estructura lògica que ens permetrà recuperar les imatges que pertoquen segons els paràmetres que es vagin obtenint durant l'execució de l'aplicació. Aquesta imatge s'anomenarà atlas, a la figura 2 es pot observar l'atles reduït generat per l'editor.
- **Sistema de visualització en temps real:** Aquest és el mòdul que s'executarà al videojoc, permetent al jugador percebre una pluja realista. Per això, gràcies a les dades obtingudes de l'anterior mòdul, es processaran les dades generades per aquest (posicions de les gotes d'aigua), i es processaran a la targeta gràfica en dos àrees diferents: **Control del moviment** de les gotes d'aigua mitjançant l'ús de vèrtexs shaders i **control de l'aparença** visual mitjançant píxels shaders. Per executar l'aplicació que s'està presentant s'ha treballat sobre el motor de 3D Ogre3D.

Podem resumir les funcions que realitza el sistema presentat:

- **Captura des del cel de l'escena:** Com s'ha vist, en el primer sistema es modificarà la imatge de l'escenari. Aquesta imatge serà obtinguda en aquesta aplicació, serà obtinguda des del cel, per oferir una visió general de l'escena.

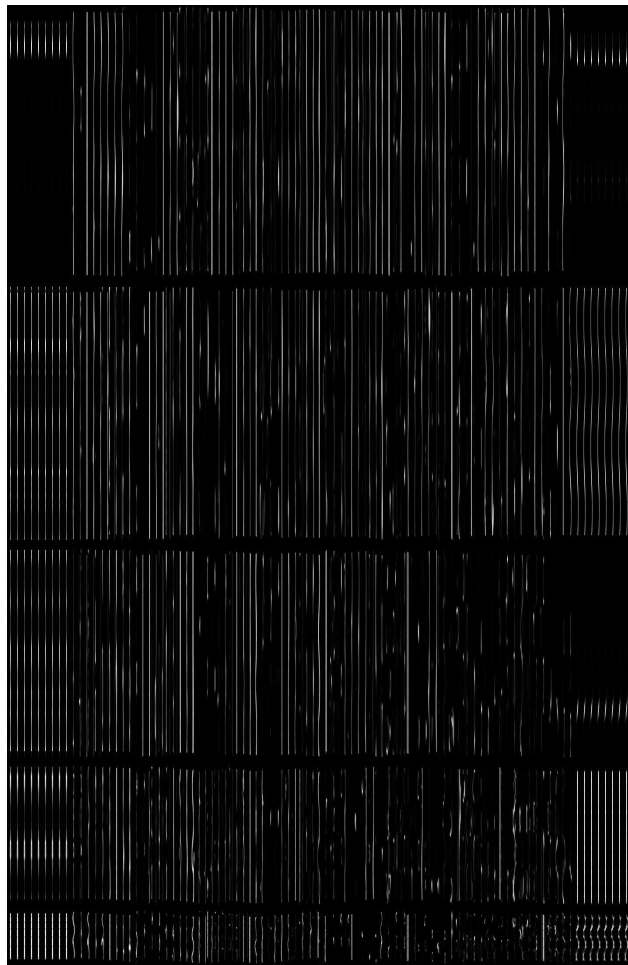


Figura 2, atlas reduït de gotes de pluja de llums puntuals.

- Captura del mapa de profunditats: Per que la pluja sigui més real, l'aplicació captura un mapa de profunditats (escala de grisos, el qual com més a prop del negre estiguin els colors, voldrà dir que l'objecte dibuixat està a una alçada més elevada). Aquesta imatge ens servirà per evitar que les gotes d'aigua impactin amb els objectes que hi puguin haver a l'escena. A la figura 4 es pot veure un mapa de profunditats generats per l'aplicació.

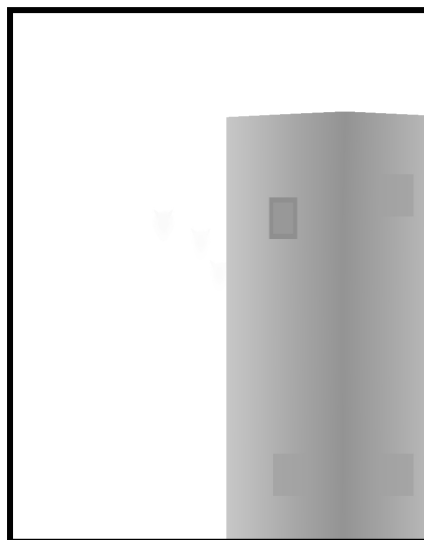


Figura 4, mapa de profunditats de l'escena.

- Captura de les posicions de l'escenari: Al mateix temps de generar el mapa de profunditats de l'escenari i també la fotografia de l'escenari, es calcularan totes les posicions que es poden veure a l'escenari en 3D. Per tant es generarà un fitxer perquè posteriorment el pugui utilitzar l'editor. Aquestes posicions seran utilitzades per l'editor al moment de generar el fitxer de les gotes distribuïdes segons els dibuixos creats per l'usuari.
- Creació dels vèrtexs (gotes): Per a simular les gotes d'aigua, s'utilitzarà un objecte del motor Ogre3D anomenat billboard. Es crearan tantes gotes com hagi decidit l'usuari des de l'editor. A les quals se les hi assignarà les posicions que s'han guardat en el fitxer al aplicar la distribució de gotes.
- Moviment i aparença de les gotes d'aigua: Per aconseguir una simulació de pluja el màxim de real, primerament s'ha fet un estudi del moviment que realitzen al moment de la caiguda lliure. És a dir, quina velocitat solen agafar, i si aquesta es va accelerant o no a causa de l'atracció de la gravetat. Després de fer aquest estudi s'ha arribat a la conclusió que la major part de la caiguda lliure de les gotes d'aigua es fa a una velocitat constant, o també anomenada velocitat terminal. Resumint-t'ho, les gotes, al anar augmentant de velocitat també augmenten el seu fragment amb l'aire. Quan la força del fregament és igual a la força que genera el pes d'aquesta, no hi ha cap força d'atracció cap a la superfície i per tant, la velocitat serà constant (veure figura 5).

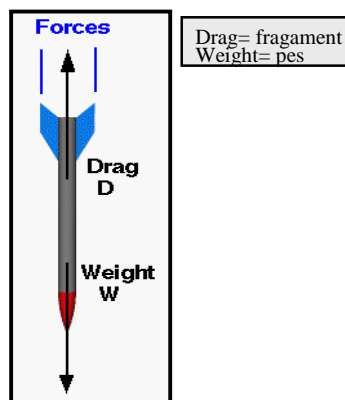


Figura 5, sistema de forces d'una gota en caiguda lliure

Per controlar el moviment de les gotes i la seva aparença, s'ha creat un material que incorporarà el vèrtex i el píxel shader.

- Vèrtex shader: Controlarà el moviment que descriuran les gotes, és a dir, la velocitat en què aquestes cauran, controlarà que no caiguin més avall de la superfície i finalment controlaran que no travessin altres objectes que estan a l'escenari de l'aplicació. Calcularà els angles que formen les gotes d'aigua amb els diferents punts de llum que hi hagi a l'escena. Per cada gota es calcularà l'angle d'elevació i azimuth dels punts de llum respecte la

gota, i l'angle que forma l'observador respecte la gota d'aigua. Aquests angles són: θ_{light} , ϕ_{light} i θ_{view} (veure figura 6). Segons els angles resultants assignarà les coordenades de l'atles per obtenir la textura que li pertoca a la gota. Aquesta informació es passarà al píxel shader ja que aquest la necessitarà per aconseguir visualitzar les gotes de forma realista.

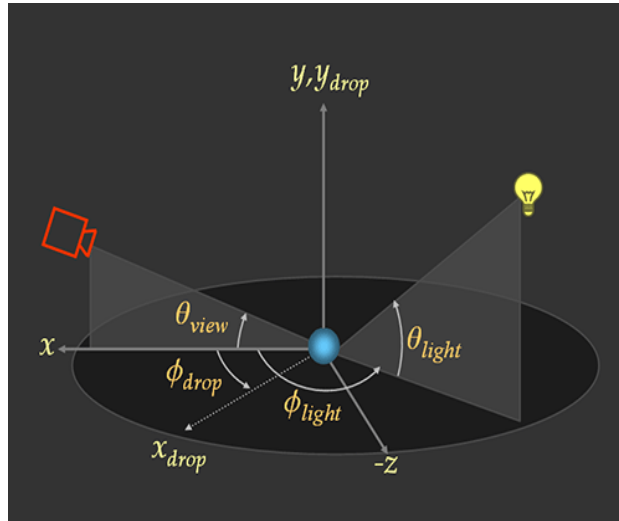


Figura 6, sistema de coordenades utilitzat per la base de dades d'imatges del projecte.

- Píxel shader: Controlarà l'aparença de cada fragment de la gota. Agafant la informació que ens proporciona el vèrtex shader obtindrem les textures adequades i controlant la llum dels diferents punts de llum obtindrem l'aparença que li pertoca a cada fragment de la gota. A més a més per tots els fragments de cada gota, controla si rep o no la llum dels diferents punts de llum de l'escenari, ja que així es diferenciarien les gotes que rep informació per exemple dels fanals i les que no.

A la figura 7 podem observar un exemple del resultat final.

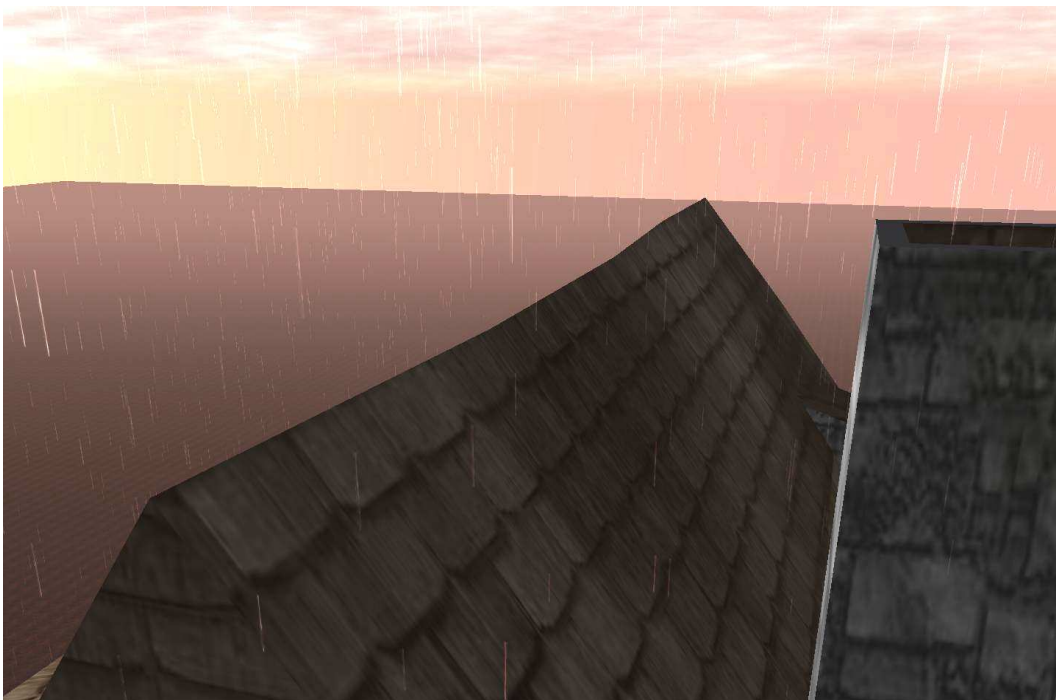


Figura 7, observem la visualització de la pluja quan el sol es pon a l'escenari.