

ACTA DE REUNIÓ No. 5

28 D'OCTUBRE DE 2021

ASSISTENTS: Sergi, Martí, Ricard, Hernan, Guillem, Alex

HORA D'INICI: 12:30

LLOC: Aula Q5/1005

1. Aprovació de l'acta anterior.

Hi ha un error en l'acta anterior, la definició de la Subtasca: Moviment de la peça es errònia, i ja ha estat corregida en aquesta acta.

2. Discussió.

Hem decidit canviar el nom de la variable interna *m_set* de la classe Bloc a *m_fixed*, booleà encarregat de marcar si el bloc està col·locat en el tauler, perquè quedi més clarificada la seva funció. A més, afegirem a Bloc una nova variable *m_color*, que guardarà el color corresponent al bloc.

La implementació de l'atribut *m_color* ens ha provocat un canvi en els fitxers .obj dels cubs a importar, ja que només n'utilitzarem un per a tots els diferents blocs en canviar-li el color dintre el nostre programa, i no important un cub de cada color com ho havíem pensat originalment.

Ajustat l'entorn perquè a l'hora d'iniciar-se, s'activi la vista perspectiva, la il·luminació de Gouraund, i la visualització del skybox (figura 1 de l'annex).

En inicialitzar les peces, el primer bloc generat és el que determinarà sobre quin punt pivota la peça en qüestió quan aquesta roti. Com està implementat actualment el pivot és el bloc més central de la peça, però per a facilitar la implementació de la futura rotació, canviarem tots els pivots al bloc central més superior. aquest canvi es pot visualitzar en la peça L. (figura 2)

Per a visualitzar les peces i blocs, utilitzarem un VAO per a cada peça i un extra que contindrà un bloc senzill.

3. Acords.

- *Merge* del nou entorn amb tota la feina feta fins el dia d'avui (que es troba a la versió antiga de l'entorn)
- Implementació de textures del bloc de forma dinàmica, en comptes d'importar-ho tot junt com es feia anteriorment
- La classe bloc necessitarà un atribut `m_color`
- Emplenar Acta de control

Objectius per a la setmana següent:

DESENVOLUPAMENT	Descripció	Responsable	Participants	Durada (setmanes)	Grau Finalització
Càmera i espai	Creació d'una Camara base i un espai on assentar-la	Marti	Marti		0%
Subtasca: Adaptar-la	Adaptar l'entorn al que necessitem. Bloquejar eixos de moviment i angle de visió		Marti	2	30%
Subtasca: Distància de la camara	Distància des d'on veurem el joc i aplicar un zoom in i zoom out		Marti	2	70%
Fitxes	Disseny dels models de les peces	Ricard	Ricard Hernan	4	50%
Subtasca: Crear models de fitxes	Disseny de models en <i>blender</i>		Ricard	1	100%
Subtasca: Incorporar importador	Incorporar/testing d'un importador d'objectes obj		Ricard Hernan	1	100%
Instanciar fitxes a l'espai	Mostrar diferents fitxes importades a l'espai i que es vegin correctament		Ricard Hernan	2	50%
Passar de peça a bloc	Canviar els fitxers .obj		Ricard Hernan	1	0%
Texturitació i entorn	Disseny de les textures de les peces	Guillem	Guillem Sergi	4	60%
Subtasca: Generar textures base	Dibuixar/crear Textures base per a les peces		Sergi	1	100%
Subtasca: Importar textures a <i>blender</i>	Aconseguir importar les textures a <i>blender</i> i entorn		Guillem Sergi	1	80%
Skybox	Modificar l'Skybox per posar un d'acord amb el tema del videojoc		Sergi, Guillem	1	100%
Actualització Entorn	Traspassar la implementació de l'entorn antic al nou		Ricard, Hernan	1	30%

DESENVOLUPAMENT	Descripció	Responsable	Participants	Durada (setmanes)	Grau Finalització
Programació de la funcionalitat del projecte	Programar les mecàniques principals del joc	Alex	Alex	7	15%
Subtasca: Pensar l'estructures de Dades	Portar (i evaluar) Possibles estructures de dades que continguin el projecte		Alex	1	100%
Subtasca: Programar estructura de dades	Programar i implementar l'estructura de dades		Alex	2	50%
Subtasca: Funció genera una peça	Programar la funció que genera una peça		Sergi	2	70%
Subtasca: Moviment de la peça	Programar el desplaçament de la peça en el tauler per part del jugador		Alex	2	40%
Subtasca: Main loop	Execució del programa		Alex	3	0%
Subtasca: Creació classe bloc a partir de la importació	Instancia el bloc en la classe		Sergi	2	20%
Subtasca: Eliminar pis	Eliminació d'un pis quant esta ple		Guillem, Alex	2	70%

Feina encara no assignada					
Texturització del entorn Shaders UI Testing Producte	Disseny de les textures secundaries	---			
	Programació de la il·luminació				
	Disseny de la interfície d'usuari				
	Assegurar-se de que el producte funciona sota qualsevol circumstancia			2	

TESTING	Descripció	Responsable	Durda (setmanes)	Grau finalització

Timeline global:

Progrés actual:40%	Simultaneitat de objectes	Animacions i main loop	Complementació	Testing	Demo
--------------------	---------------------------	------------------------	----------------	---------	------

ANNEX ESTRUCTURA DE DADES

Como guardar las piezas y el tablero

- **Bloques:** elemento principal que conforma el tablero y las piezas. De cada bloque, necesitamos guardar su posición, “si es movable”, tamaño (fijo) y si es candidato de recibir una colisión.
 1. Posición (Vector3/matrix)
 2. Movable (bool) ?
 3. Tamaño (cons int)
 4. Colisionable (bool)
- **Piezas:** guardamos las piezas como conjuntos de bloques. De cada pieza guardamos la forma y la estructura de bloques que la forman.
 1. Forma (char)
 2. Bloques (block[])
- **Tablero:** Se puede definir el tablero como un conjunto de bloques sin colisionador, se deberán definir los limitadores del tablero (costados).
 1. Bloques (Vector3/matrix)
 2. Limitadores[] (Vector3) ?
 3. Origen (Vector3)

Como definir la rotación

Utilizar un bloque como pivote de giro y definir un mapeado con la forma de la pieza tras cada rotación (rotación discreta y no animada).

Como comprobar las colisiones

Si se utiliza la variable “es movable”, se puede comparar la posición del bloque más bajo de la pieza que se está moviendo actualmente con la posición de los bloques “no movibles”, que harían de bloques ya colocados(límites). En el momento en que se detecta una colisión entre los diferentes tipos de bloque, se considerará el bloque actual como colocado y se modificará la variable de “es movable”.

Definir una variable que indique si la pieza es candidata de recibir una colisión, de esta manera reduciríamos la cantidad de bloques a comprobar.

Diseñar el bucle principal del juego

- Se genera una pieza de forma aleatoria (rotación incluida) en la posición origen del tablero. Para ello se puede hacer un mapeado de piezas identificadas con un número en concreto, después seleccionar una pieza a partir de una función de randomización de números, y por último, seleccionar su rotación de las diseñadas o calcularla.
- A cada segundo, se desplaza la pieza una posición hacia abajo, cualquier rotación de la pieza se hace al momento, por lo que se podrá mover la pieza varias veces por cada desplazamiento.
- Con cada desplazamiento se comprobará si se produce una colisión. Si no hay colisión de la pieza, se desplazará y se volverá al paso anterior. De lo contrario, la pieza no se desplazará y ésta pasará a formar parte de los límites de colisión.
- Se deberá comprobar si se ha rellenado algún piso del tablero, en caso afirmativo se deberá eliminar el piso y desplazar todos los bloques superiores una posición hacia abajo.

ANNEX IMATGES I FIGURES

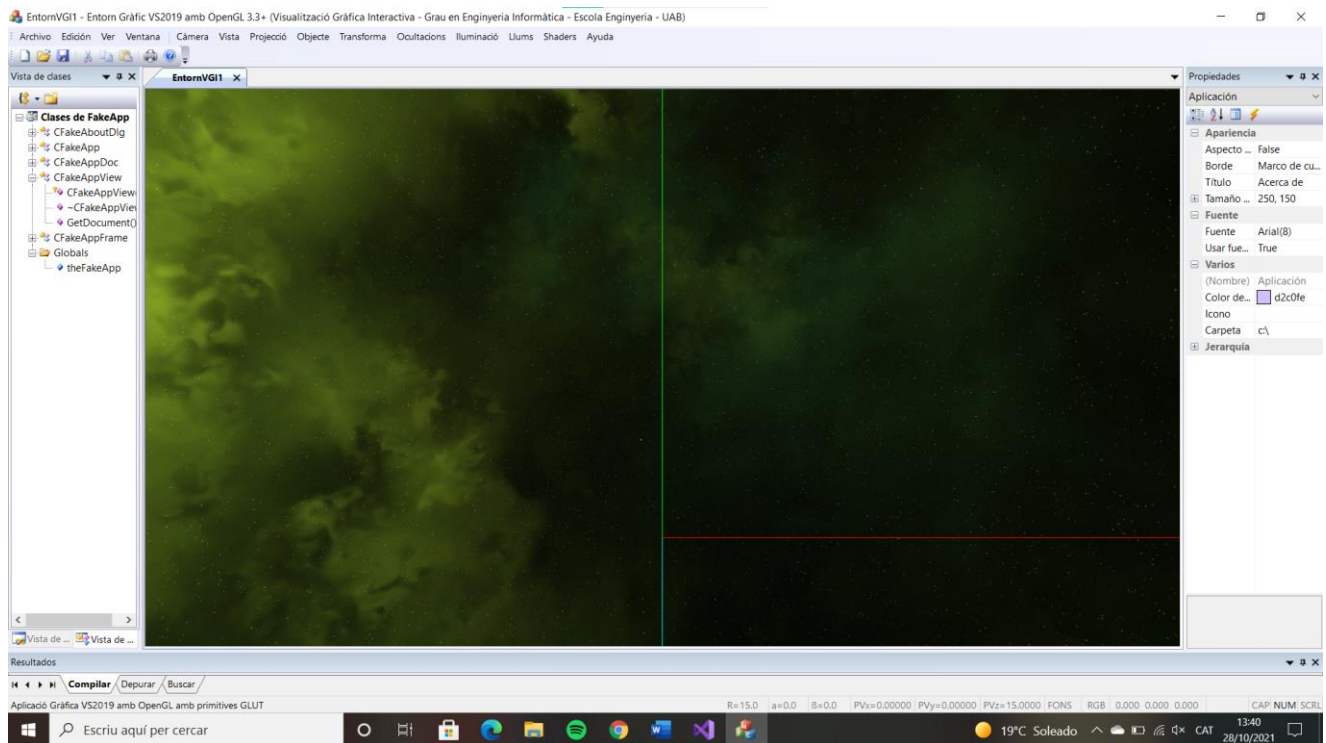


Figura 1 : Escena inicial quan s'obre l'entorn

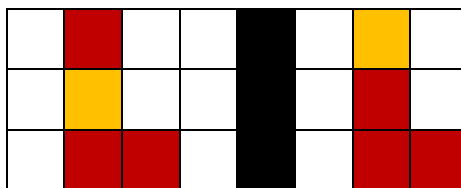


Figura 2: Canvi del pivot (groc) de la peça L, Abans/Després

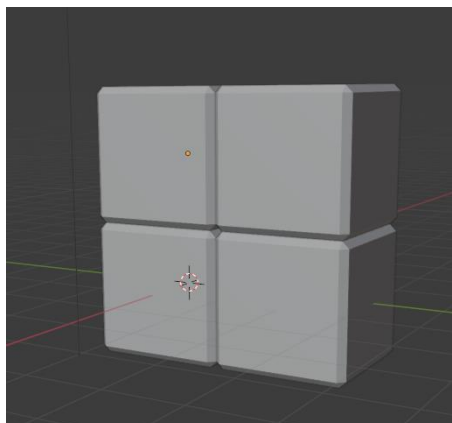


Figura 3: Peça CUB a vista en blender

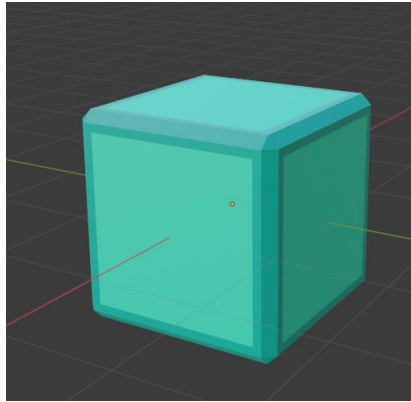


Figura 4: Block amb color fet a blender

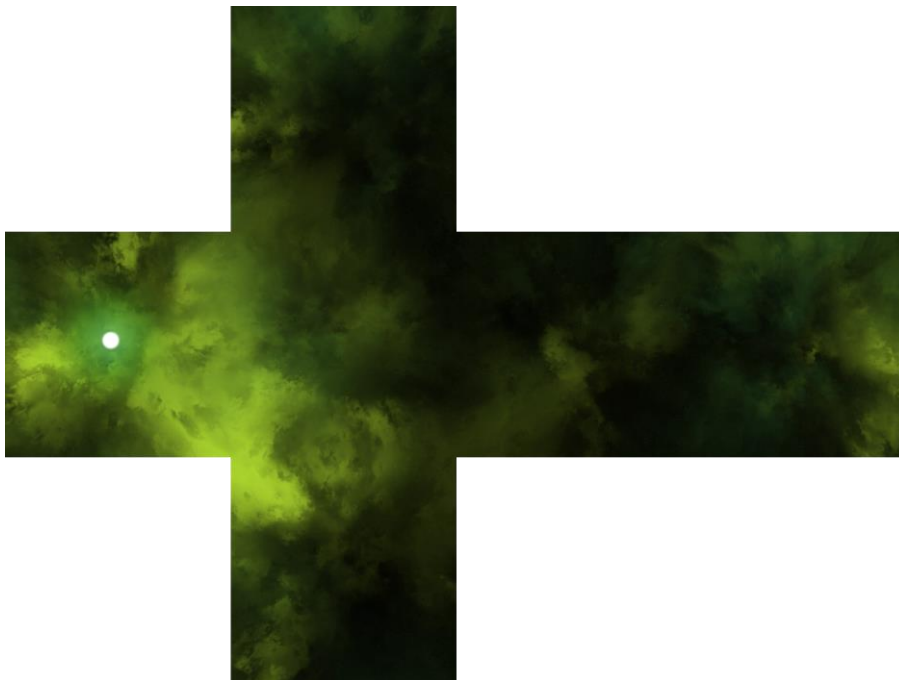


Figura 5: Cubemap del skybox

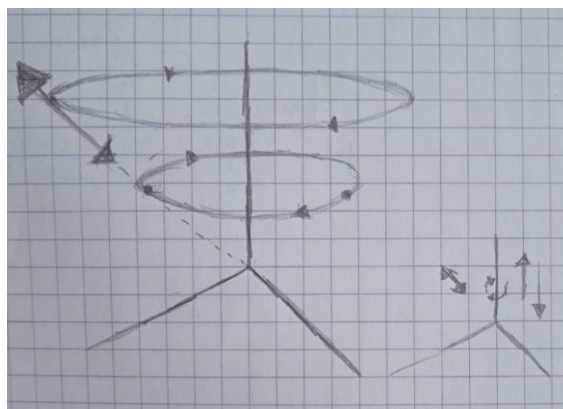


Figura 6: Esquema de moviment de la camara

S'acaba la reunió a les 14:30.

Signatures:

Alex Castro Gastón

Sergi Bons Fuses

Guillem Centelles Pavon

Martí Caixal Joaniquet

Ricard Lopez Olivares

Hernán Capilla