

# **CONTROL No. 2**

**9 DESEMBRE 2021**

---

**ASSISTENTS: Martí, Ricard, Hernan, Sergi, Guillem, Àlex**

**HORA D'INICI: 17:00**

**LLOC: Reunió per Teams.**

---

## **1. Objectius del projecte**

Enunciat del projecte a realitzar:

Es demana desenvolupar una versió del Tetris en 3D utilitzant OpenGL. S'ha de fer de la forma més realista possible però es poden afegir tocs personals amb la idea de fer un joc original i realista a la vegada.

L'objectiu del projecte és el disseny i implementació i test d'una aplicació gràfica.

Funcionalitats o objectius més concrets:

- El taulell on es jugarà serà 3D. És a dir, tindrà, a part d'amplada i alçada, profunditat.
- Com que és un joc 3D, la càmera s'ha de poder moure. Es vol un control per poder rotar al voltant l'eix Y del món, de manera que el jugador pugui veure tots els costats del taulell quan ell vulgui.  
Per altra banda, l'alçada de la càmera i el lookAt s'ha de moure de forma automàtica a mesura que el taulell canvia. Així s'aconsegueix enfocar sempre a la zona principal del joc i a la vegada lliurar el jugador de controlar aquesta part de la càmera i permetre més immersió.  
Com a últim aspecte de la càmera, el jugador ha de poder veure el taulell des de sobre en vas que ho vulgui.
- Per mantenir una dificultat fàcil, les peces seran les mateixes que al joc original, però en 3D. És a dir, la mida de la profunditat sempre serà 1.
- Les peces del Tetris han d'anar caient una a una. A mesura que cauen s'han de poder rotar. De primeres es programarà una rotació sobre l'eix Y, però la intenció és aconseguir rotar les peces sobre els 3 eixos.
- Si el tetris en 2D elimina files quan estan completes, el nostre Tetris 3D ha de poder eliminar un "pis" del taulell quan està complet.
- A raó del punt superior, quan una peça es col·loca al taulell, aquesta deixarà de ser una peça s'apilarà al taulell amb la resta de bloc que ja el formen. Permetrà esborrar els pisos més fàcilment.
- Les peces no seran un sol objecte. Estaran formades per cubs per tal de poder eliminar part d'una quan sigui necessari.

- Els controls de la càmera seran amb el ratolí, mentre que els controls de les peces seran amb el teclat. Els controls per pausar, reiniciar, etc.. també seran amb el teclat.
  - Es vol donar una ambientació interestelar al joc, així que es jugarà a l'espai (implementar un SkyBox amb galàxies, planetes i àngels).
- També es vol posar música i efectes de so que acompanyin amb l'ambientació. Aquest és un fet que contradiu la part de realisme ja que al buit no es pot escoltar res.

Les eines de desenvolupament del projecte són:

- Visual Studio amb MFC / Qt 5.10.0
- Llibreria gràfica OpenGL 3.3+
- Llibreria GLEW
- Blender

## 2. Planificació

### Taula de disseny:

DESENVOLUPAMENT	Descripció	Responsable	Participants	Durada (setmanes)	Grau Finalització
<b>Disseny de l'estructures de Dades</b>	Portar (i evaluar) Possibles estructures de dades que continguin el projecte	Alex	<b>Discussió oberta (tots)</b>	7	<b>100%</b>
<b>Disseny de la càmera en l'entorn</b>	Valoració i decisió del moviment de la càmera per a obtenir una bona experiència de joc	Martí	<b>Discussió oberta (tots)</b>	3	<b>100%</b>
<b>Disseny models de fitxes</b>	Disseny de models en <i>blender</i>	Ricard	<b>Ricard</b>	1	<b>100%</b>
<b>Disseny i implementació del Taulell (grid)</b>	Generar en blender un taulell en forma de reixa que delimiti l'espai de joc i els seus límits	Guillem	<b>Guillem</b>	1	<b>100%</b>
<b>Disseny de controls</b>	Avaluar un "mapejat" de teclat, així com la possibilitat de fer servir controls alternatius (gamepad per exemple)	Sergi	<b>Discussió oberta (tots)</b>	1	<b>80%</b>
<b>Disseny i decisió de mecàniques secundaries</b>	Pensar i limitar les mecàniques de joc per a fer un joc únic però funcional	Hernan	<b>Discussió oberta (tots)</b>	1	<b>70%</b>

## Taula d'implementació:

DESENVOLUPAMENT	Descripció	Responsable	Participants	Durada (setmanes)	Grau Finalització
<b>Càmera i espai</b>	Creació d'una Camara base i un espai on assentar-la	Marti	Marti	3	100%
<b>Subtasca: Adaptar-la</b>	Adaptar l'entorn al que necessitem. Bloquejar eixos de moviment i angle de visió		Marti	2	100%
<b>Subtasca: Distància de la càmera</b>	Distància des d'on veurem el joc i aplicar un zoom in i zoom out		Marti	2	100%
<b>Subtasca: Ajustar la càmera a les mides reals de l'entorn de joc</b>	Ajustar la càmera als paràmetres (ara ja implementats) per a augmentar la comoditat.		Marti	1	100%
<b>Fitxes</b>	Disseny dels models de les peces	Ricard	Ricard Hernan	4	100%
<b>Subtasca: Incorporar importador</b>	Incorporar/testing d'un importador d'objectes obj		Ricard Hernan	1	100%
<b>Subtasca: Passar de peça a bloc</b>	Canviar els fitxers .obj		Ricard Hernan	1	100%
<b>Texturització i entorn</b>	Disseny de les textures de les peces	Guillem	Guillem Sergi	4	100%
<b>Subtasca: Generar textures base</b>	Dibuixar/crear Textures base per a les peces		Sergi	1	100%
<b>Subtasca: Importar textures a blender</b>	Aconseguir importar les textures a <i>blender</i> i entorn		Guillem Sergi	1	100%
<b>Subtasca: "Skybox"</b>	Modificar el "Skybox" per posar un d'acord amb el tema del videojoc		Sergi	1	100%
<b>Subtasca: Sons i música</b>	Posar sons i música		Martí	1	100%

<b>Actualització Entorn</b>	Traspassar la implementació de l'entorn antic al nou	Ricard, Hernan	Dependència dinàmica	100%
-----------------------------	--	----------------	----------------------	------

<b>Programació de la funcionalitat del projecte</b>	Programar les mecàniques principals del joc	Equip Complet	Equip complet	7	95%
<b>Subtasca: Programar estructura de dades</b>	Programar i implementar l'estructura de dades base		Alex	2	100%
<b>Subtasca: Funció genera una peça</b>	Programar la funció que genera una peça en l'estructura		Ricard	1	100%
<b>Subtasca: Moviment de la peça</b>	Programar el desplaçament de la peça en el tauler per part del jugador		Ricard	1	100%
<b>Subtasca: Guardar moviment</b>	Guardar moviment a l'estructura de dades		Hernan	1	100%
<b>Subtasca: Guardar rotació</b>	Guardar la rotació a l'estructura de dades		Martí	1	100%
<b>Subtasca: calcular rotacions de cada bloc de cada peça</b>	Calcular rotacions de cada bloc de cada peça		Martí	2	100%
<b>Subtasca: Detectar límits taulell</b>	Programar funció que detecti quan una peça està fora de taulell		Martí	1	100%
<b>Subtasca: Col·lisions Rotar</b>	Programar una funció que permeti comprovar col·lisions a l'hora de rotar la peça		Marti	2	100%
<b>Subtasca: Col·lisions Moure</b>	Programar una funció que permeti comprovar col·lisions a l'hora de moure la peça		Hernan, Martí	1	100%

<b>Subtasca: Baixar peça</b>	Funció que haurà de baixar la peça a cada interval de temps	Alex	1	100%
<b>Subtasca: Creació classe bloc a partir de la importació de peces</b>	Genera blocs corresponents a partir de cada peça	Sergi	1	100%
<b>Subtasca: Eliminar fila (estructura de dades)</b>	Eliminació d'una fila quan esta plena	Alex, Hernan	2	100%
<b>Subtasca: Baixar blocs (estructura de dades)</b>	Tots els blocs superiors a una fila eliminada baixen tantes posicions com files eliminades	Hernan, Àlex, Martí	3	100%
<b>Subtasca: Rotar amb quaternions</b>	Programar la funció de rotar una peça amb quaternions	Sergi	1	100%
<b>Subtasca: rotar amb quaternions estructura de dades</b>	Programar que les rotacions a l'estructura de dades també es calculin en quaternions	Sergi	1	100%
<b>Integrar funcions de quaternions al joc</b>	Cridar i cohesionar les noves funcions amb quaternions	Martí	1	100%
<b>Subtasca: Pausa proves</b>	Funció per pausar la caiguda de peces però seguir poder movent-les	Martí	1	100%
<b>Subtasca: Pausa convencional</b>	Programar un boto per aturar el joc i que es quedi congelat	Martí	1	100%

<b>Subtasca: controls dinàmics</b>	Adaptar controls per anar en funció de la posició de la càmera	Sergi	1	100%
<b>Subtasca: Game Over</b>	Detectar Game Over	Martí	1	100%

<b>Visualització OpenGL</b>	Representar gràficament totes les accions que es van realitzant durant el joc	Ricard	Ricard, Sergi	Dependència dinàmica	100%
<b>Subtasca: Instanciar peces</b>	Importar i visualitzar per pantalla les peces generades a blender	Ricard	Ricard	2	100%
<b>Subtasca: Printar Moviment</b>	Crear una funció que mogui les peces mostrades per pantalla per a sincronitzar el moviment amb l'estructura de dades	Ricard	Ricard	1	100%
<b>Subtasca: Convertir peça a blocs del taulell</b>	Un cop ens trobem amb una col·lisió s'hauran de passar els blocs que formen la peça al taulell	Sergi	Sergi	2	100%
<b>Subtasca: Rotació OpenGL</b>	Rotació dels blocs de la peça a nivell visual	Sergi	Sergi	1	100%
<b>Subtasca: Mostrar taulell</b>	Mostrar el taulell(i tots els canvis que es facin en aquest) per pantalla	Sergi	Sergi	1	100%
<b>Antialiàsing</b>	Implementar antialiàsing	Guillem	Guillem	1	50%
<b>Integració de funcions</b>	Integrar els elements per a que funcionin en un entorn ben cohesionat	Sergi	Sergi, Marti	3	100%

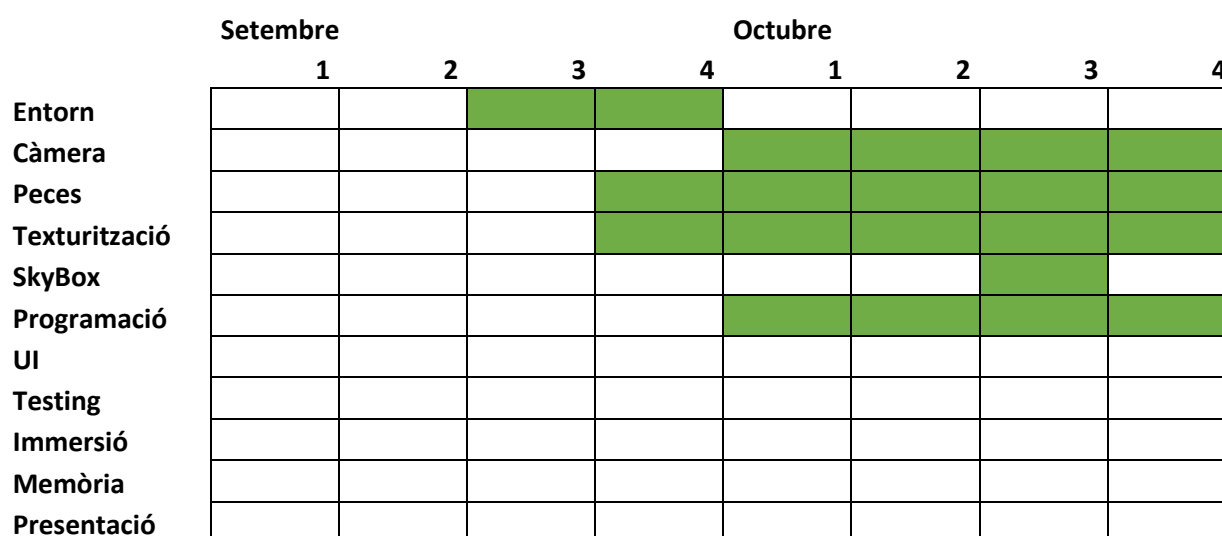
UI	Disseny de la interfície d'usuari	Ricard	Ricard	1	20%
----	-----------------------------------	--------	--------	---	-----

GRAU DE MADURESA	95%
------------------	-----

## Diagrama de Gantt

Estat actual:

- Verd: realitzat quan estava previst.
- Vermell: intenció de dedicar-hi temps en un futur.
- Groc: S'ha hagut de dedicar temps tot i no estar previst.
- Lila: Tot i estar previst, no s'hi ha pogut dedicar temps.



### 3. Desenvolupament del projecte

**Controls:** Els controls són pràcticament semblants a com s'havia pensat des d'un principi. El ratolí mou la càmera i el teclat serveix per moure i rotar les peces. Durant la implementació ens hem trobat amb un problema d'orientació al món del jugador. Per exemple, la tecla 'W' serveix per moure una peça cap endavant. El problema és que quan es mou la càmera, la peça, tot i moure's endavant, sembla que es mogui cap un altre costat. Com que el joc requereix estar controlant els 4 costats en tot moment, és molt difícil mantenir un seguiment de quina és l'orientació inicial del joc. Per resoldre el problema s'ha implementat uns controls dinàmics que canvien segons la posició de la càmera. D'aquesta manera, la tecla 'W' sempre mourà la peça cap endavant respecte la càmera. Per aconseguir-ho s'ha hagut de modificar alguns aspectes de la càmera que es detallen al seu respectiu apartat

**Càmera:** De moment permet moure's al voltant del taulell, fer zoom per abarcar tot el taulell o no, veure la part superior del taulell enforçant cap a baix. Està codificat el moviment automàtic seguint la posició de les peces actuals, resta implementar-ho un cop estiguin les peces implementades i comencin a posicionar-se al taulell.

Una decisió que ens va costar va ser si permetre al jugador moure la càmera lliurement sobre tots els eixos. S'ha decidit no deixar que això es pugui fer per facilitar el joc i augmentar la immersió.

El punt que s'ha hagut de modificar per adaptar els controls dinàmics és la distància respecte el taulell. En un principi, estava implementat que la càmera s'apropés al taulell de manera que la distància horitzontal canviés, però no la vertical. Així es podia aconseguir tenir una vista des de dalt bastant bona. El problema trobat és que arribava un moment que els controls dinàmics no podien saber quina orientació de la càmera tenia un jugador. Ara, doncs, s'ha eliminat el moviment en horitzontal respecte el centre del taulell, sempre es troba a la mateixa distància. Es segueix podent veure la part superior (des de més lluny però) i s'aconsegueix tenir controls dinàmics.

**Peces:** Modelades amb Blender i exportades a .obj. També s'importen al nostre entorn i ja es poden instanciar de forma automàtica amb el codi en comptes de fer-ho manualment. Com a decisió important era si les peces havien de ser un sol objecte o per altra banda estar formades per blocs. Després d'analitzar els requisits del joc, vam veure clar que havien d'estar formades per blocs per tal de poder esborrar part de la peça o també per poder fusionar-les amb la pila de peces que ja formen part del taulell.

Un problema relativament senzill però que ha portat bastant temps a trobar és que Blender i OpenGL no utilitzen per defecte els mateixos eixos de coordenades. Un cop solucionat a Blender no hi ha hagut més problemes.

**Texturització:** Mitjançant blender s'han aplicat les textures a la peça. S'ha decidit que sigui metàl·lica.

**SkyBox:** S'ha modificat per tal de simular un joc a l'espai. Més endavant s'ha vist que el aplicat inicialment era molt fosc i dificultava veure la profunditat d'algunes peces respecte el taulell. S'ha canviat per un més clar que no generi aquest problema.

**Programació:** Estructura principal creada. Tenim els objectes taulell, peça i bloc.

El taulell ha donat bastant a pensar i discutir. Al final s'ha decidit que s'organitzi com una llista de llistes de llista de blocs. La primera dimensió és el nivell vertical del taulell, la segona la profunditat al nivell en qüestió, i per cada profunditat es té els diferents blocs. S'ha decidit fer així per facilitar després la programació de les restriccions de les posicions de les peces i comprovar quan un nivell està complet i es pot esborrar. Cada posició guarda un objecte de tipus bloc amb la variable lliure. El joc està programat de manera que OpenGL mostri per pantalla tots els blocs del taulell que tenen lliure = false. D'aquesta manera es facilita molt la manipulació de cada bloc del taulell en tot moment, només cal modificar el valor a l'estructura de dades.

Per les peces tenim constructors diferents per cada tipus. Aquests generen un seguit de blocs a les posicions corresponents. També guardem quin és el bloc pivot i es tenen creades les funcions per rotar sobre l'eix X.



S'ha tingut molts problemes respecte les rotacions i col·lisions de les peces. En un principi, al tenir només 4 peces, s'havien programat les rotacions de forma manual, calculat les noves posicions relatives. Més endavant, al implementar rotacions sobre l'eix Y i Z, hem vist que fer-ho de forma manual no és possible. Un altre problema trobar, aquest cop a la visualització amb OpenGL, és que al rotar sobre l'eix X i llavors moure cap endavant, la peça es mou cap a l'esquerra. Aquests dos problemes s'ha solucionat fent us de quaternions. Ara les noves posicions de les rotacions es calculen automàticament i no es té el problema de moviment al fer rotacions sobre diferents eixos.

Per últim, els blocs tenen ja els atributs necessaris per saber on estan posicionats amb les coordenades món.

Un altre aspecte de la programació que és bastant complexa és detectar files i baixar-les. En un principi es volia eliminar plans, en comptes de files. És a dir, fins que no hi hagi un nivell del taulell completat, aquell nivell seguiria allà. Més endavant vam veure que, al aparèixer peces aleatòries, no sempre era possible completar un pla sencer. Així doncs, vam haver de fer eliminacions per files.

UI: S'ha començat a treballar, però es té moltes dificultats. El principal problema és que totes les llibreries que hem trobat per mostrar GUI a OpenGL no són compatibles amb MFC.

Testing: S'han realitzat 3 casos de test per comprovar col·lisions i baixar files. És molt necessari per tal de poder assegurar-nos que tot funciona correctament. Intentar provocar aquestes situacions mentre es juga és molt costós i pot comportar 30 minuts. Amb els tests ràpidament es pot aconseguir una disposició de blocs i peces que permetin comprovar el bon funcionament.

Sons i música: Ara s'utilitza diferents sons a cada tipus d'acció (moure, rotar, colisionar, etc...) i una música de fons basada en l'original de tetris però amb més ritme per crear una sensació d'estrès al jugador.

### 3.1. Estat de l'art

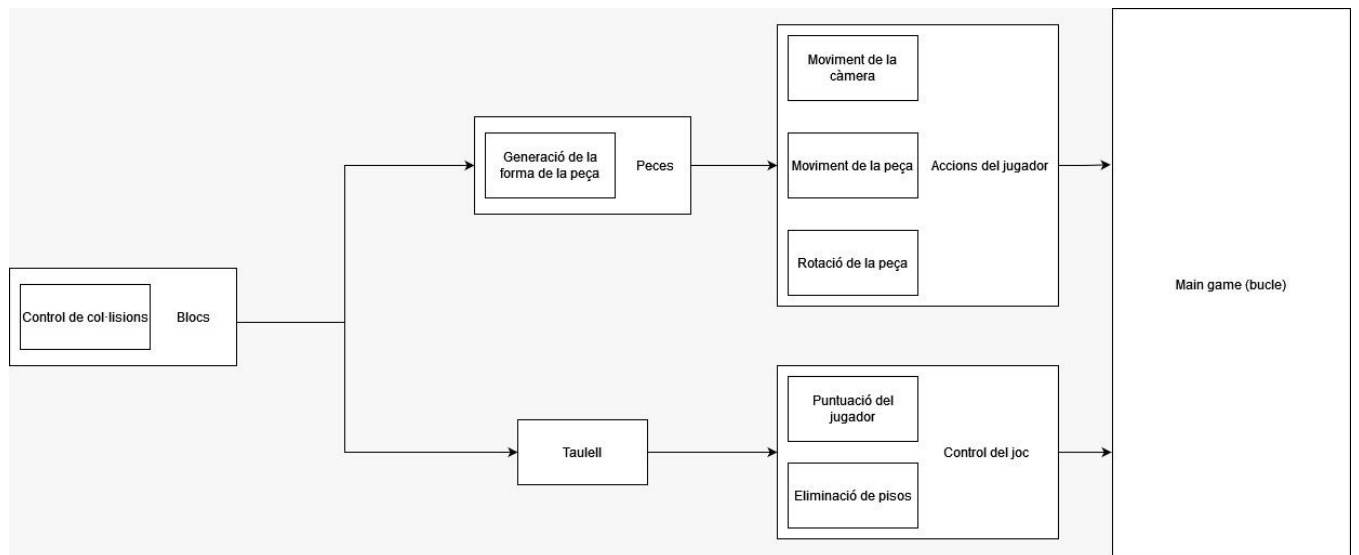
Com a principal i més reconeguda aplicació similar al nostre joc hi ha el 3D Tetris. Es va llançar als anys 90 per la Virtual Boy. La principal diferència amb el nostre, a part de ser VR, és la vista de la càmera. El nostre és des del costat del taulell, mentre que el joc ja creat és amb una vista d'ocell. L'altre aspecte a destacar és que el joc de la VirtualBoy té gràfics sense textures, simplement mostrant les línies de les geometries. Enllaç: [https://en.wikipedia.org/wiki/3D\\_Tetris](https://en.wikipedia.org/wiki/3D_Tetris)

També hi ha el Play Tetris 3D. És molt semblant al joc de la VirtualBoy i es pot jugar al navegador. A continuació el link: <https://www.mathsisfun.com/games/3d-tetris-game.html>

Per últim hi ha varis projectes personals publicats per fòrums però cap d'ells és seriós i ben desenvolupat amb un objectiu comercial.

Com a conclusió podem dir que el gènere ja existeix des de fa temps però no sembla que hagi tingut gaire bona recepció, doncs no hi ha cap producte mínimament actual que hagi estat desenvolupat per grans empreses amb molt pressupost.

### 3.2. Disseny de l'aplicació



### 3.3. Disseny de la interfície d'usuari

De moment no hi ha cap interfície d'usuari implementada. Si més no, es té intenció de mostrar una UI similar a la del Tetris original: un comptador de la puntuació, el temps, la següent peça a aparèixer, etc.. La imatge inferior mostra cop ens agradaria que quedés la GUI

També volem crear un menú de pausa que et permeti sortir o continuar la partida.

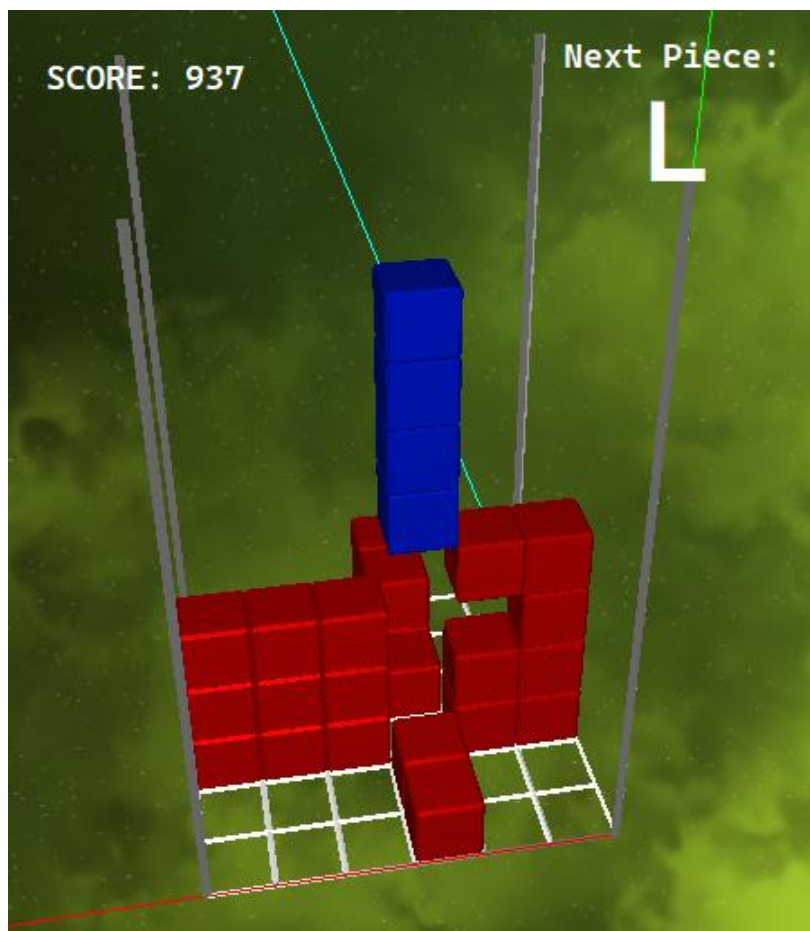


Figura 0: proposta de GUI

### 3.4. Implementació

Moviment automàtic de la càmera: es calcula la posició intermitja entre el pis més alt del taulell i el nivell màxim del taulell que és 10. Aquesta alçada és el valor de la Y que es passa al LookAt. Així aconseguim que la càmera enfoqui automàticament a la posició més adient. Per l'alçada de la càmera es calcula nivell màxim + nivell pis més alt. D'aquesta manera quan hi ha pocs nivells la càmera està més a prop ja que no ha de mostrar tot el taulell però seguim veient la peça que apareix a dalt de tot. En canvi, quan hi ha molts nivells mostra la part de dalt del taulell que és la que interessa.

Comprovació de fila completa: s'itera per totes les files del taulell, primer avançant per l'eix X i després per l'eix Z. Per detectar una fila es mira que tots els seus blocs tinguin la variable lliure = false. Quan es troba una fila, es crea un objecte que més tard s'utilitza per baixar els blocs superiors. La figura inferior mostra com és l'objecte que guarda les files a esborrar.

```
struct aEsborrar {  
    int xInici = 0;  
    int yInici = 0;  
    int zInici = 0;  
    int xFinal = 0;  
    int zFinal = 0;  
};
```

Figura 1: Struct que guarda una fila a esborrar

Cada una d'aquestes files es guarda a un vector, que s'envia per paràmetre al mètode de baixar files.

Baixar files: Un cop obtingudes les files esborrades, s'itera a través dels blocs del taulell des de l'alçada màxima a la mínima. D'aquesta manera cada cop que ens trobem amb una de les fileres esborrades, baixarem totes les fileres que tingui a sobre. Per tal de no esborrar un mateix bloc varies vegades durant una intersecció entre fileres esborrades, s'utilitza una variable de control que ens indica si el bloc ha sigut mogut, en cas afirmatiu aquest bloc no es torna a moure fins que no es baixa al següent pis del taulell.

Cada cop que baixem de pis, la variable de control dels blocs torna al seu valor d'inici per tal que si ens trobem fileres esborrades més avall es puguin moure els blocs de sobre.

Sistema de puntuació: suma 100 punts per cada filera esborrada i si s'esborra més d'una filera a la vegada t'afegeix un multiplicador equivalent al número de fileres esborrades de cop.

### 3.5. Test

Es tenen 3 tests implementats:

1. Eliminació simple d'una fila. Resultat satisfactori.
2. Eliminació amb dos nivells d'una fila. Resultat satisfactori.
3. Eliminació amb dos nivells d'una fila interseccionant una altra que també s'ha d'esborrar. Resultat satisfactori. Es pot veure el test a la figura inferior.

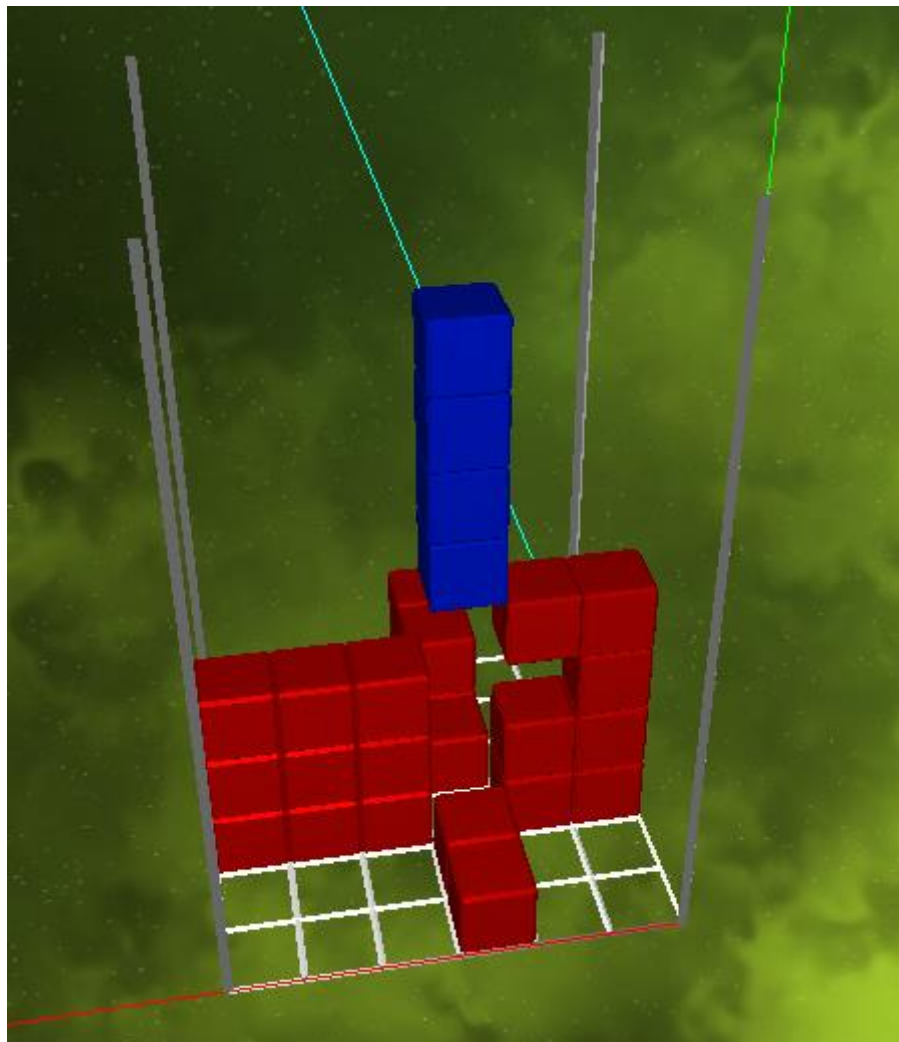


Figura 2: Test 3

Per una altra banda, s'ha fet jugar a familiars i amics per tal d'utilitzar-los com a "beta testers". A continuació hi ha una llista dels comentaris més importats que han fet:

- Fons molt fosc, dificulta la visió d'algunes peces.
- Controls molt complicats quan es mou la càmera (solucionat amb els controls dinàmics)
- Molt fàcil

Havent fet els tests i rebut les opinions, creiem que el joc està en bones condicions per ser jugat. Sí que és veritat que es pot fer millores, però és plenament funcional. En un futur ens agradaria seguir desenvolupant-lo per tal de fer-lo més "eye-friendly". El principal punt que ens hauria agradat realitzar però no s'ha pogut fer per falta de temps és transicions quan una peça es mou de posició. Ara simplement es "teletransporta", però ens hauria agradat poder fer una animació per mostrar la peça movent-se.

### 3.6. Memòria i presentació

Tant la memòria com la presentació s'ha començat a treballar. Per ara es té una estructura principal amb els apartats i subapartats que es volen incloure.

## 5. Bibliografia i referències

OpenGL	<a href="https://learnopengl.com/">https://learnopengl.com/</a> Web que ensenya OpenGL 3.0+ amb molts exemples i teoria fàcil de comprendre.	Link web
OpenGL	<a href="https://www.youtube.com/playlist?list=PLPaoO-vpZnumdcb4tZc4x5Q-v7CkrQ6M-">https://www.youtube.com/playlist?list=PLPaoO-vpZnumdcb4tZc4x5Q-v7CkrQ6M-</a> Playlist de youtube que cada vídeo aborda de forma ràpida i per donar unes pinzellades un tema diferent d'OpenGL. Va molt bé per tenir una idea bàsica de que es pot fer amb cada aspecte d'OpenGL	Playlist
OpenGL	<a href="https://github.com/JoeyDeVries/LearnOpenGL">https://github.com/JoeyDeVries/LearnOpenGL</a> Repositori que conté codi d'exemple basat en els articles oficials de LearnOpenGL.	Repositori GIT
OpenGL	<a href="https://www.udemy.com/course/graphics-with-modern-opengl/">https://www.udemy.com/course/graphics-with-modern-opengl/</a> Curs llarg i extens que explica OpenGL amb transparències de teoria i després mostra algun que altre exemple per implementar el que s'ha explicat. Va molt bé per entendre temes que no tenim suficientment clars.	Curs en línia
Textures	<a href="https://tools.wwwtyro.net/space-3d/...">https://tools.wwwtyro.net/space-3d/...</a> Generador de cubemaps per a skyboxs ambientats en l'espai,	Aplicació en línia

**Signatures:**

Martí

Ricard

Sergi

Hernan

Àlex

Guillem