

# **UAB**

## **Universitat Autònoma de Barcelona**

### **Informe progrés II**

Engraginy

Biel Alavedra

Tutor: Enric Marti Godia

**Grau en Enginyeria Informàtica**

**Escola d'Enginyeria**

**Curs 2025-2026**

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	2/15

## Control del document

### Informació del document

	Informació
<b>Identificació del document</b>	Informe Progrés II
<b>Autor del document</b>	Biel Alavedra Busquet
<b>Data de creació</b>	12/12/2025
<b>Data del document</b>	14/12/2025
<b>Nom del fitxer</b>	Informe progres II.odt

### Historial del document

Versió	Data versió	Canvis
1.0	12/12/2025	Creació document, diagrama de mòduls
1.1	13/12/2025	Implementació de les mecàniques i explicació del sistema de potència
1.2	14/12/205	Explicació dels altres sistemes i acabar l'apartat

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	3/15

## Taula de continguts

1. Resum.....	4
2. Paraules clau.....	4
3. Motivació.....	5
4. Objectius del projecte.....	6
5. Planificació.....	7
5.1. Estat de l'art.....	7
5.2. Game design document.....	7
5.3. Implementació.....	8
6. Bibliografia i referències.....	13
7. Annexos.....	14

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	4/15

## 1. Resum

Aquest projecte té com a objectiu el desenvolupament d'un videojoc del gènere de l'automatització. Aquest gènere es va començar a popularitzar amb la sortida de Factorio<sup>[5]</sup>, però no va ser el primer del gènere, aquest joc està inspirat en el mod de Minecraft, IndustrialCraft<sup>[10]</sup>.

Abans d'entrar en cap altre tema vull definir una mica que és per mi aquest gènere i que el fa especial. En els jocs d'automatització jo considero que hi ha dos grans apartats, les fàbriques, encarregades de produir tots els materials necessaris per progressar en el joc, i les centrals d'energia, dedicades a subministrar tota l'energia necessària per al funcionament de les fàbriques. No té cap sentit ampliar la fàbrica si no tens energia per suplir-la, i no té sentit dedicar-te només a millorar el subministrament d'energia, primerament perquè a mesura que avances en el joc desbloqueges millors mètodes de generació, i segon perquè perds temps i recursos en ampliar una cosa que no t'ajuda a progressar. Cada joc fa una implementació diferent dels dos sistemes, a excepció del Shapez<sup>[8]</sup>, el qual no té un sistema d'energia.

L'inici de tots aquests jocs és similar, comences processant uns recursos bàsics, per fer un producte més complex, aquest producte complex l'utilitzes per poder construir noves màquines i tecnologies que a la vegada et serviran per fer un producte encara més complex. I així es va repetint el cicle fins que el joc acaba. És una idea senzilla i no és un cicle de joc massa complex, la complexitat de tots aquests jocs ve de poder balancejar totes les entrades de recursos i escalar correctament totes les parts de les fàbriques, això mentre fas un disseny no massa rebuscat perquè si es trenca alguna cosa ho hauràs d'arreglar.

En el meu treball vull desenvolupar un videojoc d'aquest gènere utilitzant el motor de codi obert Godot<sup>[1]</sup>, vull fer un videojoc en primera persona, amb una ambientació prerevolució industrial i sense elements fantàstics, l'objectiu és implementar 4 nivells tecnològics, començar utilitzant fusta i pedra → coure → ferro → acer. Els detalls d'aquesta implementació estaran explicats en el GDD. Com a llenguatge de programació utilitzaré GDscript perquè està fortament integrat amb el motor i és més senzill, l'objectiu final serà passar les parts del codi més crítiques i que necessitin una millora de rendiment a C++.

## 2. Paraules clau

Videojoc, desenvolupament, automatització

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	5/15

### 3. Motivació

La meva motivació per desenvolupar aquest projecte és principalment la d'aprendre com funciona el desenvolupament d'un videojoc. Els meus referents a l'hora de desenvolupar el projecte són videojocs com 'Factorio'<sup>[5]</sup>, 'Satisfactory'<sup>[6]</sup>, 'Dyson sphere program'<sup>[7]</sup>, 'Shapez'<sup>[8]</sup>, i 'Minecraft: Create Mod'<sup>[9]</sup> (una modificació del joc no oficial).

D'aquests referents amb els quals tinc més experiència són el DSP<sup>[7]</sup> i el Satisfactory<sup>[6]</sup>, són dos videojocs que m'han enganxat molt i m'han mostrat que tan entretingut i mentalment activador pot arribar a ser el gènere. El DSP<sup>[7]</sup> és un joc molt més ambiciós, comences en un petit satèl·lit dins d'un cúmul d'estrelles generat aleatòriament, i acabes fent una esfera de Dyson i viatjant entre sistemes solars, tot això amb una perspectiva isomètrica que et dona molt control i molta visió de tot allò que construeixes. En el Satisfactory<sup>[6]</sup> l'objectiu és completar un Ascensor espacial, es comença en un planeta (el qual sempre és el mateix, el mapa és prefixat). Aquest joc és en primera persona, i totes les màquines que es construeixen són gegantesques, això impedeix una visió tan completa d'allò que construeixes, en quantitat de màquines les construccions són sempre més petites i molt més lentes de construir.

Dels dos jocs vull fer una cosa més similar al DSP<sup>[7]</sup>, principalment perquè m'agrada molt més la sensació de construir amb la vista isomètrica, degut al control i a la visió general que et dona. També vull agafar una mecànica del mod Minecraft: Create<sup>[9]</sup>, en la majoria de jocs de l'estil una part molt important és la generació d'energia, no serveix de res fer fàbriques més grans si no tens els recursos per mantenir-les, tant DSP<sup>[7]</sup>, Factorio<sup>[5]</sup> i Satisfactory<sup>[6]</sup> utilitzen energia elèctrica, sigui solar, eòlica, carbó o nuclear. Tots aquests generadors s'acaben connectant a la xarxa elèctrica per donar energia a totes les màquines, en canvi, el Minecraft: Create<sup>[9]</sup> requereix que tot sigui alimentat per energia cinètica, fent ús d'eixos, engranatges grans i petits, cadenes, corretges de transmissió... Com el meu objectiu és fer un joc amb una ambientació prerevolució industrial, i no vull fer un joc fantàstic, aquesta és la millor forma de fer-ho, crec jo, també això afegeix totes les mecàniques d'haver de connectar les màquines no només a una velocitat concreta, també algunes seran direccionals, com les cintes per transportar material.

D'entre tots els gèneres de videojocs perquè he escollit automatització? Considero que aquest és un gènere que segueix molt la filosofia d'un programador, o en general la de tots els enginyers: dividir i vèncer. L'inici d'aquests jocs és senzill, hi ha unes poques màquines i cal fer processos simples. Per exemple, amb un extractor de recursos automàtic i has de processar els recursos en brut per tal d'obtenir el recurs processat (mena de ferro → lingots de ferro). Però això ràpidament canvia, agafant d'exemple el joc DSP<sup>[7]</sup>, si vols fer una placa

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	6/15

de circuits necessita un forn de fosa que converteixi el ferro en brut a plaques de ferro, un forn de fosa que converteixi coure en brut a plaques de coure i un assemblador que agafi aquests dos materials i els converteixi en la placa de circuits. Això és només un dels primers passos, on cada vegada es disposa de més i més materials diferents, processos diferents, i els objectes requerits són cada vegada més complexos. Per força no es poden afrontar tots aquests problemes simultàniament, has de dividir els problemes en mòduls que puguis replicar cada vegada que calgui resoldre aquell problema.

Dins de la comunitat de fans d'aquest gènere de videojocs hi ha molts enginyers i gent que li agrada molt optimitzar processos i fer construccions el màxim d'eficients possibles. Per tot això no només considero el gènere com a entretingut, sinó que també en certa manera és educatiu, et veus forçat tant sí com no a organitzar-te, pensar formés eficients de construir coses, com connectes les entrades i sortides de totes les fabriques, calcular que tanta entrada necessites per la teva sortida objectiva i construir d'acord amb això. Per completar un joc d'aquests has de ser organitzat, metòdic i pensar en solucions segons el problema que tinguis,

## 4. Objectius del projecte.

Llistat d'objectius del projecte:

- Primera persona
- Prerevolució industrial
- Un sol mapa, no generació automàtica
- Tres nivells de tecnologia
- Sistema d'energia basat la rotació

Eines de desenvolupament:

- Godot<sup>[1]</sup>
- C++<sup>[2]</sup> for Godot<sup>[3]</sup>
- Visual Studio Code<sup>[4]</sup>
- Blender<sup>[11]</sup>

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	7/15

## 5. Planificació

Aquesta taula s'utilitzarà de guia per fer un seguiment del projecte i seu el grau de maduresa d'aquest. Aquest grau es calcula com un promig ponderat.

Tasca	Descripció	Durada (setmanes)	Grau de finalització
Estat de l'art	Buscar productes similars, caracteritzar-los i comparar-los	1	100%
GDD	Creació del GDD	3	80%
Implementació	Implementació i creació del videojoc a partir del GDD	10	50%
Test	Test intern i extern	2	20%
Memòria i Presentació	Escriure la memòria i la presentació oral	2	0%
GRAU DE MADURESA			30%

*Taula 1. Planificació de tasques del projecte*

### 5.1. Estat de l'art

Aquest apartat està completat en el document Informe Inicial1.5.pdf

### 5.2. Game design document

Aquest apartat està completat en el document Informe progres I.pdf

## 6. Implementació

En aquest informe en centraré a explicar el funcionament de tot el sistema de potència del videojoc, com funcionen totes les rotacions, connexions d'eixos i engranatges, com el sistema és indiferent a la direcció del món i com es manté un control de l'energia total utilitzada. També esmentaré com funciona el sistema de construcció i la GUI actual.

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	8/15

## 6.1. Implementació de les mecàniques

El sistema de potència és una adaptació del sistema existent a Minecraft: Create, per tant, he volgut simular el seu funcionament, a continuació faré un llistat de les especificacions del sistema, i com s'hauria de comportar sota cada cas específic.

- Tota connexió d'un eix ha de mantenir direcció i velocitat.
- Tota connexió d'un engranatge ha d'invertir la direcció.
- Connectar un engranatge petit a un gran a través de les dents de l'engranatge ha de duplicar velocitat, a l'invers la velocitat es divideix.
- La xarxa no pot superar el límit d'energia subministrada per tot el conjunt de generadors connectats.
- En cas de superar el límit tot el sistema s'ha d'aturar a l'instant.
- En cas de tornar a estar per sota del límit el sistema ha d'entrar en funcionament a l'instant.
- En cas que un node tingui incoherències en els seus ports connectats aquell node s'ha d'eliminar automàticament.

Pel que fa al sistema de construcció l'objectiu és tenir una barra d'accés directe (hotbar) on pugui vincular fins a nou objectes diferents, abans de col·locar qualsevol objecte es veurà un objecte fantasma que serà de color vermell en cas de no poder col·locar l'objecte i verd quan la seva col·locació és possible. També es podran girar els objectes abans de col·locar-los, alguns en qualsevol direcció i d'altres només es podran girar en l'eix vertical.

En el menú principal de construcció han d'estar els objectes que el jugador pot construir organitzats en pestanyes segons la categoria i s'han de poder arrossegar a la barra d'accés inferior per modificar els accessos directes.

## 6.2. Implementació tècnica

En aquest apartat entraré principalment en detall sobre tot el sistema de potència, ja que és la part més interessant i més complexa. Però primer introduiré el funcionament del joc utilitzant el diagrama de mòduls de la figura 1. El jugador controla tres parts importants, del joc, el mateix objecte de jugador, encarregat del moviment, la càmera, encarregada d'allò que veu el jugador, i la GUI que conté el menú amb els objectes i la hotbar. Aquesta última és l'encarregada de gestionar la construcció de nous objectes. Tot objecte té un node anomenat «ContextComponent», aquest node permet saber a quin objecte mira el jugador. Per últim, tenim els «PowerNodes», aquests són tots els objectes que pertanyen al sistema de potència, tenim els mateixos elements de la xarxa com eixos i engranatges, el cervell



Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	9/15

central que controla tot el sistema («PowerGridManager») i els ports que gestionen les connexions entre elements («PowerNodePort»).

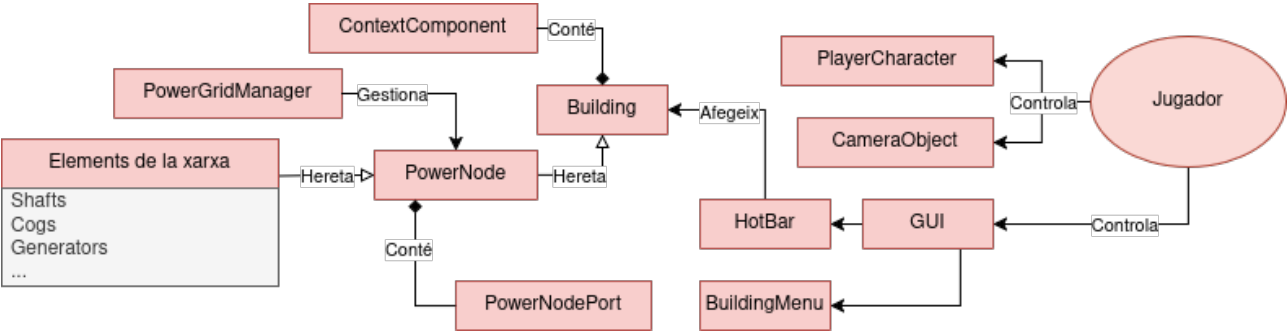


Figura 1. Diagrama de mòduls.

• Sistema de potencia

La classe principal d'aquest sistema són els «PowerNodes», d'aquesta classe hereten tots els elements de la xarxa que afegirem, cada element tindrà mínim un «PowerNodePort», aquest és un cub ubicat allà on es vol que el node tingui un punt de connexió, aquest port té quatre propietats molt importants, la direcció de gir respectivament al node, el modificador de velocitat, el tipus de port que és, i a quins altres ports es pot connectar.

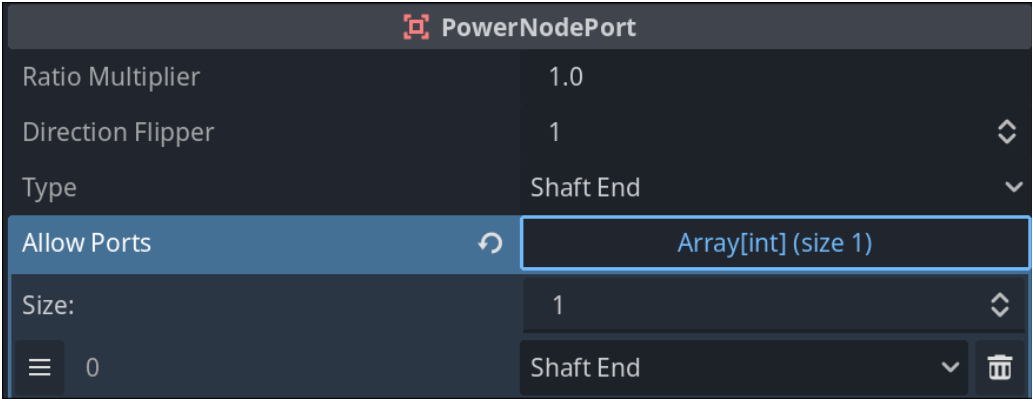
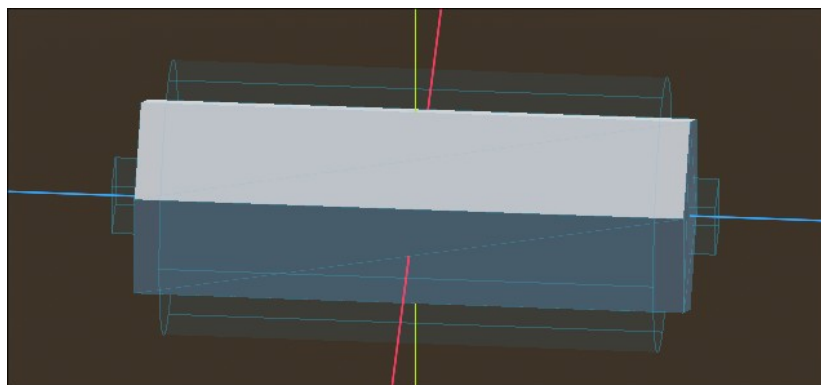


Figura 2. Exemple de port d'un eix

En la figura 2 podem veure com estan configurats els ports d'un eix bàsic, en cas d'implementar un inversor de gir hauríem de tenir dos ports, un com el de la figura 1 i l'altre amb el «Direction Flipper» a -1, per a un duplicador el mateix però amb el «Ratio Multiplier» a 2.

En la figura 3 veiem l'exemple de l'eix, amb els dos ports col·locats un a cada extrem de l'objecte.

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	10/15



*Figura 3. Exemple de l'escena d'un eix*

Aquests ports hereten de la classe base de Godot «Area3D», això és perquè aquesta classe base té ja té implementat un mètode que en cas d'un altre «Area3D» entri en contacte podem connectar una funció que com argument ens doni l'àrea que ha entrat.

Quan detectem aquesta connexió comprovem que l'altra àrea sigui un «PowerNodePort» i que la connexió sigui vàlida. En cas de connexió vàlida llancem el senyal «network\_changed», aquest senyal indica al «PowerGridManager» que hi ha hagut un canvi en la xarxa a la qual pertany el node, i per tant s'ha de recalculer aquella xarxa.

Aquest procés de càlcul comença amb un algoritme «BFS», es podria utilitzar també un «DFS» en aquest cas, per tal de trobar tots els nodes connectats, una vegada tenim tots aquests nodes busquem quins són els generadors que hi ha a la xarxa, i és a partir d'aquests generadors que comencem a propagar tots els canvis. Per propagar els canvis fem un altre «BFS», en aquest cas si ha de ser aquest algoritme, ja que interessa que els canvis es propaguin capa a capa, que es calculin abans els nodes més pròxims per així detectar els possibles conflictes en els extrems. Per cada node definim la seva velocitat com la calculada en el moment d'afegir-lo a la llista de «per visitar», i iterem per les seves connexions per tal de calcular la velocitat que haurien de tenir per no generar conflictes, en cas de no tenir la connexió una velocitat ja assignada li assignem la calculada, si ja tenia una velocitat assignada ens assegurem que no hi hagi conflictes, en cas de conflicte eliminem la peça i tornem a començar el procés de càlcul de la xarxa, si no hi ha conflicte afegim la connexió al llistat de «per visitar». Finalment, quan hem acabat amb el càlcul de les velocitats recorrem tota la xarxa per comprovar que la potència és prou per mantenir la xarxa activa, tots els generadors sumen les seves potències i la resta d'elements resten.

La part més complexa d'aquest procés és el càlcul de velocitats, els nodes poden estar situats en qualsevol punt del món i en qualsevol direcció, per exemple un eix sense girar connectat a un girat 180 graus semblen el mateix, però si es fa una assignació directa de

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	11/15

velocitat el segon eix girarà en sentit contrari. Per solucionar això fem una sèrie de càlculs amb els vectors de direcció de cada element per comprovar com s'ha de comportar aquella connexió. Per fer aquest càlcul seguirem una sèrie de passos.

$$W_{input} = W_{conn} \times R_{conn} \times D_{conn}$$

*Fórmula 1. Càlcul de velocitat d'entrada*

En la fórmula 1 tenim la velocitat d'entrada aparent, sense tenir en compte l'alineació dels dos nodes, on «W» és la velocitat de rotació, «R» és el multiplicador de velocitat d'aquell port i «D» és la direcció del port.

$$d = \vec{A} \cdot \vec{B}$$

*Fórmula 2. Càlcul de l'alineació dels eixos de rotació*

En la fórmula 2 calculem l'alineació de l'eix de rotació propi, «A», i l'eix de rotació del port connectat, «B». Aquesta alineació es calcula utilitzant el producte escalar entre els dos eixos de rotació. A partir d'aquest punt poden passar dues coses, el valor «d» podrà ser 0 o 1/-1.

En el primer cas, «d» és igual a 1/-1, els dos ports són paral·lels, si els dos ports són del tipus «SHAFT\_END» la velocitat del port serà la «W<sub>input</sub>» multiplicada pel signe de «d», en cas que els ports siguin del tipus «COG» la velocitat serà «W<sub>input</sub>» multiplicada pel negatiu de signe «d», per tal d'invertir la direcció de gir.

En el cas que «d» sigui 0 voldrà dir que els dos ports són perpendiculars, aquest cas actualment només es dona quan dos engranatges grans estan connectats perpendicularment, en aquest cas hem de comprovar cap a on hem de girar, aquesta direcció dependrà de la posició relativa (esquerra, dreta, dalt, baix) de l'engrenatge connectat.

$$\vec{V} = \vec{P}_{conn} - \vec{P}_{local}$$

$$\vec{T}_{local} = \vec{A} \times \vec{V}$$

$$\vec{T}_{conn} = \vec{V} \times \vec{B}$$

$$Alignment = sgn(\vec{T}_{local} \cdot \vec{T}_{conn})$$

*Fórmula 3. Resolució del conflicte*

En la fórmula 3 per tal de calcular la direcció hem de fer diversos passos, comencem calculant el vector de connexió «V» que va des de la meua posició a la posició del node connectat, amb aquest vector calculem la tangent pròpia, que ens indica cap a on es mouen les dents de l'engrenatge en el punt de contacte, després calculem la tangent de la connexió,

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	12/15

en aquest càlcul els valors estan invertits per tal d'obtenir la perspectiva correcta des de l'altre node. Per últim, obtenim la direcció obtenint el signe del producte escalar entre els dos vectors tangent, aquesta direcció la multipliquem a la velocitat « $W_{input}$ ».

$$W_{res} = \frac{W_{input} * D_{local}}{R_{local}}$$

*Fórmula 4. Càlcul de la velocitat interna*

Finalment, aquesta velocitat « $W_{input}$ », que representa la velocitat del port la convertim a la velocitat interna que ha de tenir el node, utilitzant la fórmula 4 desfem el càlcul fet a la fórmula 1, però utilitzant els valors del nostre port. Aquest és el valor que retornem perquè el «PowerGridManager» comprovi si hi ha conflicte.

- **Sistema de construcció**

L'objectiu d'aquest sistema és que el jugador pugui posar els objectes allà on està mirant, els objectes han de tenir una textura diferent per indicar si la posició es valida, han d'estar sempre lligats a la quadrícula del món, s'han de poder girar lliurement i els objectes no han d'interactuar amb el món fins que sigui col·locat. Per aconseguir tot això el procés segueix els següents passos.

Primerament, en aparèixer un objecte desactivem totes les seves propietats que li permeten interactuar amb el món i el jugador.

A continuació calculem la posició a la qual ha d'aparèixer l'objecte, des del centre de la càmera fem un «RayQuery» per obtenir la posició a la qual el jugador està mirant, des d'aquest punt busquem el vector normal de la superfície que coincideix amb el raig i sumem aquest vector normalitzat al nostre punt d'impacte, així evitem que un objecte aparegui dins d'un altre, finalment aquest punt l'arrodonim a nombres enters per fer l'efecte de quadrícula. Amb la posició ja obtinguda comprovem que aquell és un lloc vàlid per col·locar l'objecte, això és una funció pròpia de l'objecte que ens retorna un booleà indicant si és possible, aquesta funció també s'encarrega d'assignar a l'objecte una textura verda si la posició és vàlida o vermella si no ho és. Aquest procés de càlcul de posició i de comprovació de posició vàlida es repeteix a cada fotograma del joc fins que el jugador no colloqui l'objecte.

Quan el jugador envii l'acció de col·locar objecte primerament comprovarem que l'edifici estigui en una posició vàlida, si ho està cridarem a la funció pròpia de l'objecte per tal de col·locar-lo i farem aparèixer un altre objecte igual amb la mateixa rotació, ja que és molt

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	13/15

possible que el jugador vulgui col·locar més d'un objecte del mateix tipus, per evitar tornar a seleccionar-lo i rotar-lo ja en fem aparèixer un altre directament.

### 6.2.1. Dificultats i solucions

En la implementació tècnica el més complicat ha sigut sense dubte el sistema de potència, en les fases inicials de la implementació trobava tota l'estona errors i coses que no es comportaven com un esperava. Els objectes girats 180 graus sempre es comportaven a l'inrevés, els engranatges no invertien el gir correctament, els nodes no es trencaven en cas de conflicte, separar un conjunt de nodes del generador no actualitzava el seu comportament, el sistema no parava i s'encenia seguint els límits de potència.

Tots aquests problemes s'havien de solucionar sense requerir un sistema que s'estigués actualitzant constantment, volia tant sí com no una solució que només es recalculés quan fos necessari. En un inici no tenia cap solució que fos independent de la direcció del món, i utilitzava tècniques com girar les peces 180 graus en el moment de col·locar-les perquè no estiguessin col·locades a la inversa, i altres mètodes similars que ni eren fàcils d'escalar ni implementar, i que no eren robustos, sempre trobava un cas que m'obligava a afegir més condicions al codi perquè funcionés bé. Al final la solució utilitzant càlcul vectorial ha sigut la més robusta i directa d'implementar, només he de tenir en compte que en el moment de construir les escenes tots els objectes he de mantenir les mateixes normes d'orientació, o els objectes no connectaran bé perquè un dels ports podria estar girat dins de l'escena.

El sistema de construcció també tenia anteriorment un mètode que no funcionava correctament, les col·lisions de l'objecte no estaven desactivades i l'objecte de previsualització col·lidia amb el jugador i el motor de físiques no sabia què fer en aquell cas, cosa que provocava que el jugador sortís disparat fins a sortir del mapa. Sense fer tampoc el càlcul del vector normal de la superfície d'impacte l'objecte apareixia dins de l'altre i només era possible construir mirant el terra.

### 6.2.2. Estat actual i passos futurs

Actualment, he completat ja el que considero l'apartat més complicat, junt amb altres parts del videojoc més senzilles però necessàries, com la GUI i el sistema de construcció. A continuació he de treballar en les màquines per a la producció i processament de materials, junt amb el sistema de cintes transportadores per connectar-ho tot, per a l'entrega, l'objectiu és tenir un conjunt de nivells que actuïn com a tutorial per mostrar totes les mecàniques implementades.

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	14/15

## 7. Bibliografia i referències

- [1] <https://godotengine.org/> Pàgina oficial del motor Godot (últim accés setembre 2025)
- [2] <https://isocpp.org/> Pàgina sobre el llenguatge C++ (últim accés setembre 2025)
- [3] [https://docs.godotengine.org/en/4.4/tutorials/scripting/gdextension/gdextension\\_cpp\\_example.html](https://docs.godotengine.org/en/4.4/tutorials/scripting/gdextension/gdextension_cpp_example.html) Documentació oficial Godot per crear plugins utilitzant C++ (últim accés octubre 2025)
- [4] <https://code.visualstudio.com/> Pàgina oficial Visual Studio Code (últim accés setembre 2025)
- [5] <https://www.factorio.com/> Pàgina oficial de Factorio (últim accés octubre 2025)
- [6] <https://www.satisfactorygame.com/> Pàgina oficial de Satisfactory(últim accés setembre 2025)
- [7] [https://store.steampowered.com/app/1366540/Dyson Sphere Program/](https://store.steampowered.com/app/1366540/Dyson_Sphere_Program/) Pàgina d'Steam de Dyson sphere program (últim accés setembre 2025)
- [8] <https://github.com/tobspr-games/shapez.io> Pàgina de Github de Shapez (últim accés setembre 2025)
- [9] <https://github.com/Creators-of-Create/Create> Pàgina de Github de Create (últim accés setembre 2025)
- [10] <https://www.curseforge.com/minecraft/mc-mods/industrial-craft> Pàgina oficial del mod (últim accés octubre 2025)
- [11] <https://www.blender.org/> Pàgina oficial de Blender (últim accés novembre 2025)

Engraginy	Versió 1.0
Biel Alavedra Busquet	12/12/2025
Informe progrés II	15/15

## 8. Annexos

- El Game design document (GDD) complet s'adjunta com a Annex 1 en un document separat, titulat «Engraginy\_GDD.pdf».