UNIVERSITAT DE BARCELONA

Grau de Física

Departament de Física Quàntica i Astrofísica

Pràctiques en Empresa

POPULARITZACIÓ DE LA FÍSICA QUÀNTICA

QUANTUM SPACESHIP BATTLE

Autora

Elisabeth Llanos Pla

Supervisors

Bruno Juliá Díaz

Montserrat Guilleumas Morell

Carles Calero Borrallo

Març 2021 - Gener 2022

Data d'entrega memòria: 2 de Febrer de 2022

Índex

1. Introducció

L'objectiu d'aquestes pràctiques en empresa, realitzades del març de 2021 al gener de 2022, ha estat el desenvolupament d'un joc per ordinador dirigit a un públic amb interessos per la quàntica que els permeti començar a introduir-se en aquesta branca de la física. En aquest projecte en concret s'ha programat un joc que reprodueix el famós joc d'enfonsar la flota, en què s'han integrat conceptes de física quàntica i s'ha après a utilitzar llenguatges de programació i plataformes de desenvolupament de videojocs com Unity. Per tal de donar a conèixer aquesta iniciativa, es va exposar el projecte a la 14a edició de la Festa de la Ciència de manera divulgativa i interactiva.

2. QuantumLab UB

QuantumLab UB és un projecte iniciat per Daniel Allepuz i Jan Albert Iglesias el febrer del 2018, sota la supervisió dels professors Bruno Juliá i Montserrat Guilleumas, dirigit des del Departament de Física Quàntica i Astrofísica de la Universitat de Barcelona. És un projecte col·laboratiu entre els estudiants que cursen l'assignatura de Pràctiques d'Empresa, que els permet desenvolupar aplicacions i videojocs enfocats a la popularització de la física quàntica. Cadascuna d'aquestes aplicacions dissenyades pels alumnes tracta d'un tema diferent relacionat amb la física quàntica: des de simulacions de la funció d'ona d'una partícula en travessar una escletxa, fins a videojocs interactius.

2.1. Objectius

Els objectius d'aquest projecte es poden resumir en els següents:

- Desenvolupament de videojocs senzills per popularitzar i divulgar la física quàntica a alumnes de secundària i batxillerat amb interès científic, per tal de començar a introduir-los en aquest món i que puguin conèixer les nocions bàsiques de la quàntica d'una manera interactiva.
- Crear aplicacions realistes i fidels a les solucions numèriques de l'equació de Schrödinger, és a dir, plasmar de manera molt acurada els resultats que esperaríem obtenir d'un experiment quàntic.

2.2. Inicis del projecte

Arran del meu interès per la computació i per la física quàntica, em vaig posar en contacte amb el professor Bruno Juliá, que em va oferir participar en aquest projecte juntament amb un altre estudiant de grau, Gabriel Linares. Els dos vam contribuir cadascú amb el seu projecte individual, emmarcat dins de l'essència i el propòsit de QuantumLab UB. Originalment, el desenvolupament dels videojocs o aplicacions es duia a terme mitjançant el llenguatge de programació Python, addicionamlment fent ús de Kivy, que és una biblioteca de Python de codi obert pel desenvolupament d'aplicacions i permet crear interfícies de joc interactives. No obstant això, el meu company de pràctiques, Gabriel, estava familiaritzat amb Unity, una plataforma especialitzada en desenvolupament de videojocs, que permet ajuntar el codi i la interfície de manera senzilla i pràctica en comparació amb Python. És per això que es va proposar de fer el projecte mitjançant Unity, deixant Python de banda. D'aquesta manera, els nostres respectius projectes han estat creats amb Unity, a diferència de les aplicacions dels altres estudiants que havien participat

amb QuantumbLab UB. En un principi, el fet de canviar Unity per Python va ser un petit inconvenient per mi, ja que jo no coneixia aquesta eina i per contra sí que estava familiaritzada amb Python. A més a més, els professors que ens supervisaven tampoc havien utilitzat mai Unity, el que va fer que al llarg del desenvolupament del projecte haguéssim de treballar de manera molt autònoma. Vaig haver de llegir la documentació de Unity [2], i veure molts videotutorials per complementar el meu aprenentatge. També va ser útil la guia de programació en llenguatge C Sharp [3]. Cal mencionar que el meu company de pràctiques em va oferir la seva ajuda sempre que ho vaig necessitar.

El primer pas per encaminar el nostre projecte va ser pensar una idea a partir de la qual es pogués desenvolupar un videojoc que tingués un rerefons quàntic. El nostre propòsit era programar un joc "jugable", més interactiu que les aplicacions que s'havien fet anteriorment. Entre tots vam fer diferents propostes i van quedar definits els dos treballs: el del Gabriel es basava en el Dimoni de Maxwell i el meu en un joc d'enfonsar la flota, en què l'usuari juga contra un electró que es troba en un estat descrit per una funció d'ona que compleix l'equació de Schrödinger (explicat detalladament en l'apartat 3.2. Funcionament del joc).

2.3. Metodologia

En aquest apartat es vol exposar la forma de treball que s'ha adoptat al llarg d'aquests mesos. Per tal de fer un bon seguiment del projecte, setmanalment es van dur a terme sessions telemàtiques en què els dos estudiants exposàvem els nostres progressos i rebíem feedback dels nostres tutors: Bruno Juliá, Carles Calero i Montserrat Guilleumas. Les primeres reunions van servir per establir la idea del joc i per definir quins aspectes hi volíem incloure. En les següents sessions ens vam dedicar a tractar més a fons els problemes quàntics que fonamenten els projectes: com plasmar la quàntica de manera fidel en el nostre joc, resoldre dubtes del codi programat, o fer provatures per comprovar el bon funcionament d'aquesta primera part. En les reunions posteriors exposàvem els nostres avenços. Aquestes sessions van ser molt enriquidores, ja que van permetre intercanviar coneixements, aprendre i col·laborar, així com proposar noves idees pel projecte.

Per compartir el nostre codi amb els professors, en un primer moment vam utilitzar la plataforma Github [1], on vam crear un repositori per a cada projecte dins de QuantumLab UB. Vam penjar els nostres codis per tal que els professors poguessin analitzar-los. No obstant això, més endavant vam optar per compartir-los el nostre treball directament a través de Unity Collaborate, que està habilitat al núvol i integrat directament a Unity.

3. Quantum Spaceship Battle

La física quàntica és aquella part de la física que explica els fenòmens i el comportament de la matèria de dimensions més petites. A tan petita escala és impossible conèixer amb exactitud la posició i velocitat d'una partícula de manera simultània, de manera que la descripció de la física quàntica és essencialment probabilística. Ens permet tenir un millor enteniment del món que ens envolta, i gran part de les innovacions d'un futur pròxim estaran regides per la quàntica, com per exemple la computació: tindrem ordinadors que treballin directament d'acord amb les lleis de la quàntica. Aquesta és una de les moltes raons per les quals és important que els nois i noies que tenen interessos científico-tecnològics comencin a conèixer què és la física quàntica. És per això que nosaltres volem introduir-los en aquest món mitjançant jocs que els permetin entendre les lleis més bàsiques de manera interactiva. A continuació s'expliquen els fonaments teòrics de la física quàntica que s'han utilitzat en aquest projecte, i com els vinculem a un joc quotidià, obtenint com a resultat un videojoc senzill amb un rerefons quàntic.

3.1. Fonaments teòrics

Per explicar la quàntica que hi ha darrere del nostre joc, considerarem un electró tancat en una caixa bidimensional. Podem fer l'analogia amb la presència d'un pou de potencial quadrat infinit en les direccions x i y, tal que no permet que la partícula pugui escapar de la caixa. Aquest electró es pot trobar en diferents estats quantitzats d'energia (n=1,2,3, ...) descrits per una funció d'ona, $\psi(x, y)$, que és solució de l'equació de Schrödinger bidimensional independent del temps:

$$\frac{-\hbar^2}{2m}\left(\frac{\partial^2\psi(x,y)}{\partial x^2}+\frac{\partial^2\psi(x,y)}{\partial y^2}\right)=E\psi(x,y) \tag{1}$$

Podem resoldre aquesta equació proposant una solució del tipus:

$$\psi(x,y)=X(x)Y(y) \tag{2}$$

Introduint aquesta solució dins l'equació diferencial:

$$\frac{-\hbar^2}{2m}\left(Y(y)\frac{\partial^2X(x)}{\partial x^2}+X(x)\frac{\partial^2Y(y)}{\partial y^2}\right)=EX(x)Y(y)\implies\frac{X''(x)}{X(x)}+\frac{Y''(y)}{Y(y)}=\frac{-2mE}{\hbar^2} \tag{3}$$

On X"(x) i Y"(y) corresponen a les segones derivades de X(x) i Y(y) respectivament. Aquesta igualtat només es compleix si els dos termes per separat són iguals a una constant, en aquest cas a $\frac{-2mE}{\hbar^2}$. Per tant, ens reduïm a resoldre un problema equivalent al d'una caixa unidimensional en cada eix. La representació de les solucions en el cas unidimensional es poden veure en la Figura 1.

Per l'eix x:

$$X''(x)=\frac{-2mE}{\hbar^2}X(x) \tag{4}$$

Que correspon a l'equació que descriu al moviment d'un oscil·lador harmònic. Té per solució:

$$X(x)=Asin\left(\frac{\sqrt{2mE}}{\hbar}x\right)+Bcos\left(\frac{\sqrt{2mE}}{\hbar}x\right) \tag{5}$$

Imposem les condicions de contorn: La funció d'ona s'anul·la en x=0 i x=L_x. Per tant:

$$X(0) = 0 \implies B = 0 \implies X(x) = Asin\left(\frac{\sqrt{2mE}}{\hbar}x\right) \tag{6}$$

$$X(L_x) = 0 \implies sin\left(\frac{\sqrt{2mE}}{\hbar}L_x\right) = 0 \implies \frac{\sqrt{2mE}}{\hbar}L_x = n_x\pi \implies \frac{\sqrt{2mE}}{\hbar} = \frac{n_x\pi}{L_x} \tag{7}$$

Per tant obtenim:

$$X(x) = Asin\left(\frac{n_x\pi}{L_x}x\right) \tag{8}$$

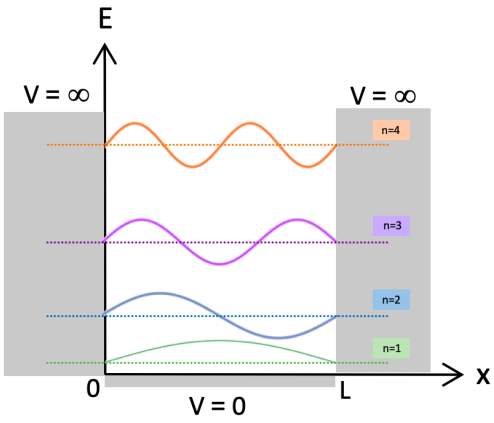


Figura 1: Funcions d'ona (línies contínues) i nivells d'energia (línies discontínues) d'un electró en una caixa unidimensional, és a dir, confinat per un potencial V = 0 per $0 < x < L$, V = ∞ per $x < 0$ i $x > L$.

Procedim de manera anàloga per l'eix y, obtenint:

$$Y(y) = Csin\left(\frac{n_y\pi}{L_y}y\right) \tag{9}$$

Per tant, la funció d'ona que descriu l'estat de l'electró tancat en una caixa bidimensional té la forma:

$$\psi(x,y) = ACsin\left(\frac{n_x\pi}{L_x}x\right)sin\left(\frac{n_y\pi}{L_y}y\right) = \lambda sin\left(\frac{n_x\pi}{L_x}x\right)sin\left(\frac{n_y\pi}{L_y}y\right) \tag{10}$$

On hem definit la constant $\lambda = AC$. Normalitzant la funció d'ona, podem obtenir el valor d'aquesta constant.

$$\int_0^{L_x}\int_0^{L_y}|\psi(x,y)|^2dxdy = \int_0^{L_x}\int_0^{L_y}\lambda^2sin^2\left(\frac{n_x\pi}{L_x}x\right)sin^2\left(\frac{n_y\pi}{L_y}y\right)dxdy = \lambda^2\frac{L_x}{2}\cdot\frac{L_y}{2} = 1 \tag{11}$$

$$\implies \lambda = \sqrt{\frac{2}{L_x}\cdot\frac{2}{L_y}} \tag{12}$$

Finalment:

$$\psi(x,y) = \sqrt{\frac{2}{L_x}\frac{2}{L_y}}sin\left(\frac{n_x\pi x}{L_x}\right)sin\left(\frac{n_y\pi y}{L_y}\right) \tag{13}$$

Sent L_x, L_y, les dimensions de la caixa i n_x, n_y, els nivells d'energia en la direcció x i y de l'electró.
Podem determinar la probabilitat de trobar l'electró en una certa posició a partir de la densitat de probabilitat, que la podem obtenir fent el quadrat de la funció d'ona. Aquesta probabilitat ens dependrà del nivell energètic en què es trobi l'electró.

3.2. Funcionament del joc

Quantum Spaceship Battle ha estat el resultat d'aquests mesos de treball: un joc inspirat en el conegut Enfonsar la Flota, amb la diferència que l'enemic que vol destruir les naus de l'usuari, no és un segon jugador, sinó un electró en estat quàntic. En la pantalla de joc es mostra una quadrícula (taulell de joc) de dimensions L_x x L_y, que simula una caixa bidimensional en la qual es troba confinat un electró (aquest electró no apareix per pantalla, només la quadrícula).

L'usuari disposa de 5 naus que ha d'arrossegar fins a col·locar sobre la quadrícula, cada una en una casella diferent. Un cop situades les naus, s'habilita un botó amb el qual es fan mesures simbòliques de la posició de l'electró. Tal com hem explicat en l'apartat 3.1. Fonaments teòrics, la posició de l'electró dins la caixa no es pot saber de manera exacta, però sí que es pot determinar la probabilitat que l'electró estigui en una certa posició calculant la densitat de probabilitat (el quadrat de la funció d'ona que descriu l'estat de l'electró en funció de les dimensions de la caixa i el nivell energètic n_x i n_y en què es trobi). En prémer aquest botó es generen, a partir del mètode d'acceptació i rebuig, 100 nombres o "posicions"(en 2D, coordenades x,y) distribuïts segons la densitat de probabilitat donada. A continuació, sobre la quadrícula apareixeran punts vermells que simulen la posició de l'electró en haver fet les mesures (és a dir, mostrem sobre el taulell de joc els punts distribuits segons la densitat de probabilitat, que hem obtingut a partir del mètode acceptació-rebuig). Així doncs, si es prem diverses vegades el botó, apareixeran noves marques sobre la quadrícula, i a mesura que en tinguem més, es podran distingir regions determinades on es concentren més punts vermells i altres zones on no en trobem cap. Estarem veient de manera il·lustrativa la densitat de probabilitat del nostre electró. En les figures 2 i 3 es mostren exemples de la conversió de la densitat de probabilitat als punts sobre la pantalla de joc, per a dos nivells energètics diferents.

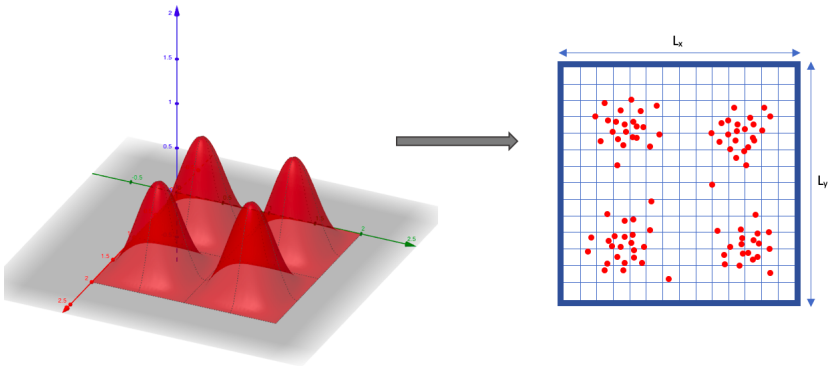


Figura 2: Representació gràfica de la densitat de probabilitat de la funció d'ona que descriu l'estat de l'electró, quan aquest es troba en el nivell energètic $n_x = n_y = 2$. Aquesta densitat transportada al taulell de joc queda representada per concentracions de punts en les regions en què és més probable trobar l'electró.

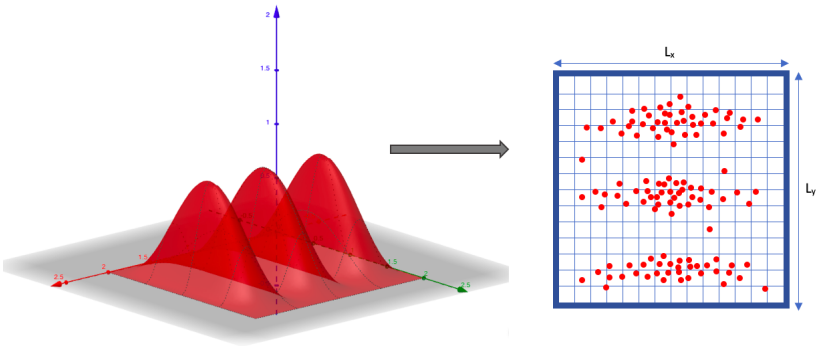


Figura 3: Representació gràfica de la densitat de probabilitat de la funció d'ona que descriu l'estat de l'electró, quan aquest es troba en el nivell energètic $n_x = 1, n_y = 3$. Aquesta densitat transportada al taulell de joc queda representada per concentracions de punts en les regions en què és més probable trobar l'electró.

L'objectiu del jugador és col·locar les seves naus en una posició estratègica tal que no coincideixin amb la posició dels punts vermells. Per tant, per tal de guanyar el joc, l'usuari haurà de conèixer quina distribució seguiran els punts, i per tant on serà més probable que aquests apareguin, per tal d'evitar aquelles posicions.

A l'inici l'usuari disposa de 10 vides, que anirà perdent a mesura que es destrueixin les seves naus. L'objectiu és superar totes les rondes del joc sense haver perdut totes les vides. Comença el joc, primera ronda. En la pantalla se li proporciona a l'usuari una pista sobre el nivell energètic, n_x i n_y, en què es troba l'electró (generat aleatòriament pel nostre programa). Seguidament, el jugador col·loca les seves naus en 5 de les caselles de la quadrícula, i a continuació prem el botó per fer les mesures de la posició de l'electró. Com hem explicat, apareixeran per pantalla punts vermells que simulen la posició de l'electró cada cop que es fan mesures. Si algun d'aquests punts coincideix amb la posició d'alguna de les naus, aquesta és destruïda i l'usuari perd una vida. Després d'haver premut el botó per primer cop i havent vist com ha quedat la distribució de punts, el jugador pot canviar de posició les naus segons convingui, disposant-les de manera més estratègica. L'usuari ha de prémer el botó de mesures 5 vegades (cada vegada la distribució serà més òbvia i més fàcil ho tindrà el jugador per recol·locar les naus). Si després d'haver repetit el procés 5 vegades, l'usuari encara té vides, es passa a la següent ronda. A cada nova ronda, l'usuari torna a tenir totes les naus actives (són naus amb una nova aparença) i el nivell energètic de l'electró ha canviat, essent ara més complicat d'intuir. Podríem dir que cada nova ronda és com reiniciar l'escena original, adoptant aquests canvis, però conservant les vides. Si a l'usuari se li acaben les vides abans d'haver superat totes les rondes (5 rondes), perd la partida i obté la puntuació de la ronda a la qual ha arribat. Si aconsegueix superar totes les rondes sense haver perdut totes les vides, guanya el joc. A l'apartat 5. Resultats es poden veure figures de la pantalla de joc.

A més a més, s'ha introduït un apartat en què s'exposen gràfics interactius de les densitats de probabilitat per a diferents nivells energètics. L'usuari pot seleccionar un valor discret de n_x i n_y i apareix el comportament corresponent. El jugador té l'opció de consultar aquest apartat en qualsevol moment de

la partida. S'explica amb més detall en l'apartat 5. Resultats.

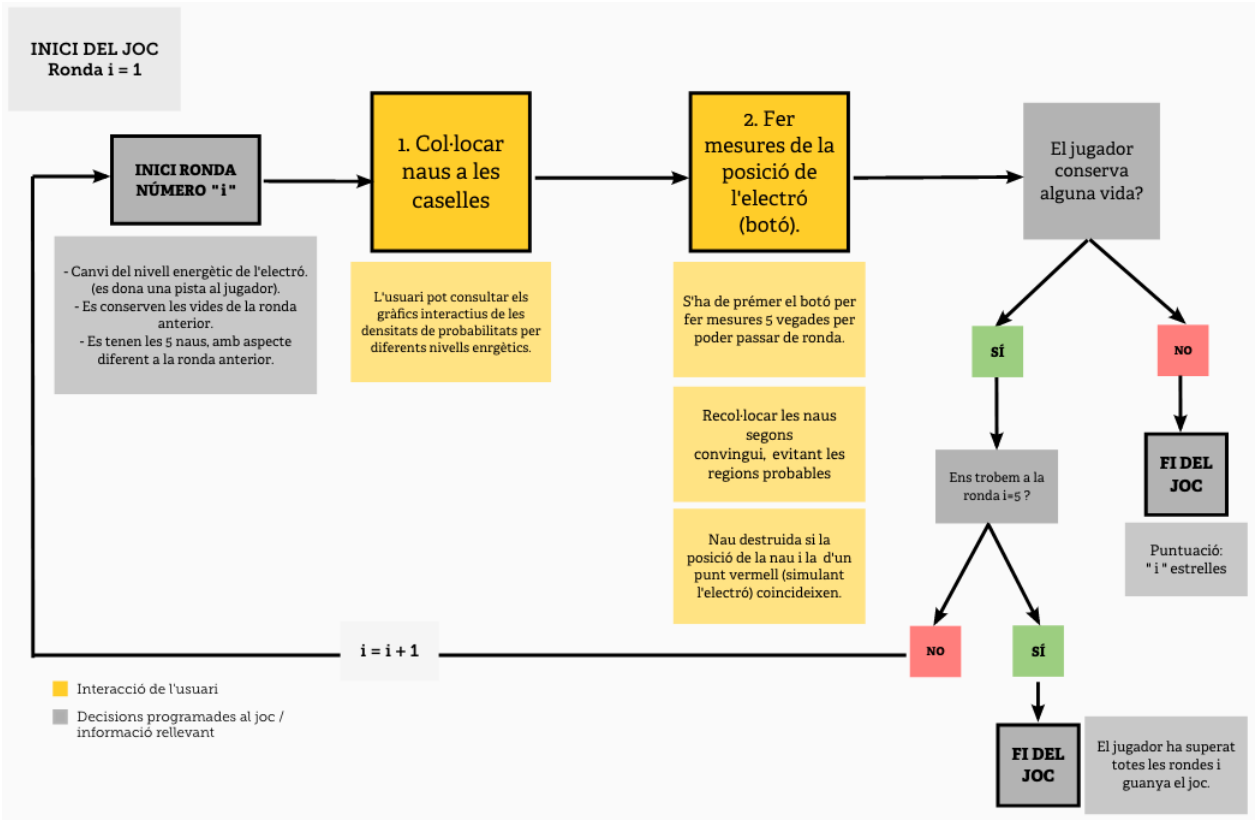


Figura 4: Esquema de la seqüència de joc.

Més endavant es va afegir l'opció multijugador, en què la dinàmica del joc és la mateixa, però els dos usuaris competeixen; qui aconsegueixi superar més rondes, guanya. Aquesta opció s'ha desenvolupat per jugar off-line. Els dos usuaris comparteixen ordinador i juguen per torns.

4. Desenvolupament del joc

4.1. Eines i programes utilitzats

- **Fortran:** Llenguatge de programació que s'ha utilitzat per fer els càlculs inicials de les funcions de densitat de probabilitat. Com que jo estava familiaritzada amb aquest llenguatge, Fortran juntament amb Gnuplot em van servir per comprovar si el meu codi generava nombres correctament distribuïts segons la funció donada. Aquest codi va ser adaptat a C Sharp posteriorment per al desenvolupament del joc. Per tant, Fortran i Gnuplot van ser una eina útil per fer provatures i anàlisis de certes parts del codi.

- **Gnuplot:** Programa d'interfície de línia de comandes per generar gràfics de dues i tres dimensions. S'ha utilitzat per comprovar si els càlculs de les funcions de densitat de probabilitat en 2D fets amb Fortran eren correctes. Posteriorment s'ha usat per fer animacions en 3D d'aquestes mateixes funcions, per a diferents valors de n_x i n_y, per tal d'introduir-les al joc i que l'usuari pugui inter-

accionar amb elles i entendre de manera visual les funcions que regeixen les probabilitats de trobar l'electró en una certa posició de l'espai x,y, delimitada per la caixa de dimensions L_x x L_y.

- **C Sharp:** Llenguatge de programació que s'ha utilitzat per escriure tot el codi corresponent a la programació del joc. Aquests codis van vinculats a la plataforma Unity.
- **Unity:** Plataforma que permet la programació (via scripts de C Sharp), creació, disseny i funcionament de videojocs. El fet que Unity estigui disponible per diferents sistemes operatius, com també les facilitats i comoditats que dona a l'hora d'unir les diferents parts d'un joc, el converteix en la plataforma líder de desenvolupament de videojocs, tant per amateurs com per professionals.

4.2. Components del joc

Per tal d'obtenir el joc Quantum Spaceship Battle s'ha hagut de seguir un procés llarg, tractant diferents punts importants per separat, per tal de comprovar que aquests funcionaven i s'adequaven als resultats que esperaríem del nostre sistema quàntic.

A continuació s'exposa el procés que s'ha dut a terme per a desenvolupar els tres pilars principals del joc: la programació i comprovació de la distribució de probabilitats, la creació d'aquests gràfics interactius, i el disseny de la interfície i la cohesió de tots els elements per crear un joc jugable.

4.2.1. Mesures de la posició de l'electró

El primer pas per començar a crear el joc d'enfonsar la flota, va ser la programació de generació de nombres distribuïts segons la densitat de probabilitat de la funció d'ona que descriu l'estat de l'electró. Tal com hem vist a l'apartat 3.1. Fonaments teòrics, aquesta densitat de probabilitat la podem obtenir fent el quadrat de la funció d'ona.

$$f = |\psi(x,y)|^2 = \left| \sqrt{\frac{2}{L_x}\frac{2}{L_y}} sin\left(\frac{n_x \pi x}{L_x}\right) sin\left(\frac{n_y \pi y}{L_y}\right) \right|^2 = \left| \frac{2}{L} sin\left(\frac{n_x \pi x}{L_x}\right) sin\left(\frac{n_y \pi y}{L_y}\right) \right|^2 \qquad (14)$$

Anomenarem f a aquesta densitat de probabilitat, que serà una funció que depèn del nivell energètic de l'electró n_x, n_y, i de les dimensions de la caixa quadrada $L_x = L_y =$ L (L està fixada, mentre que el nivell energètic variarà a cada ronda del joc).

Per tal de generar nombres distribuïts segons aquesta funció f, s'ha utilitzat el mètode d'Acceptació i Rebuig [4] en dues dimensions. Per tal de poder utilitzar aquest mètode, les variables aleatòries (x,y) han d'estar definides en un domini finit: x, y, $\in [a,b]$. Per altra banda, la seva densitat de probabilitat ha d'estar acotada $f(x,y) < M$. En el nostre cas es compleixen aquests requisits, ja que la funció f és acotada i x, y, $\in [0,L]$ amb L finit.

El funcionament del mètode és el següent:

1. Generem dos nombres aleatoris, x, y distribuïts uniformement entre a i b.
   Els podem obtenir generant dos nombres aleatoris, x_{10}, x_{11} distribuïts uniformement entre el 0 i 1,

i a continuació fer el canvi de variable:

$$x = (b - a)x_{10} + a$$

$$y = (b - a)x_{11} + a$$

2. Generem un tercer nombre a l'atzar, x_2, distribuït uniformement entre 0 i 1. A continuació definim una variable p = Mx_2.
   El valor de M és un cota que establim tal que la funció f estigui per sota d'aquest valor. Determinem M "a ull", segons el comportament de la funció. Com més baixa sigui la cota, més precís i ràpid serà el mètode.

3. Avaluem la funció f en els dos nombres aleatoris x, y. Si f(x,y) $\geq$ p, acceptem els valors de x, y. En cas contrari, comencem el procés de nou.

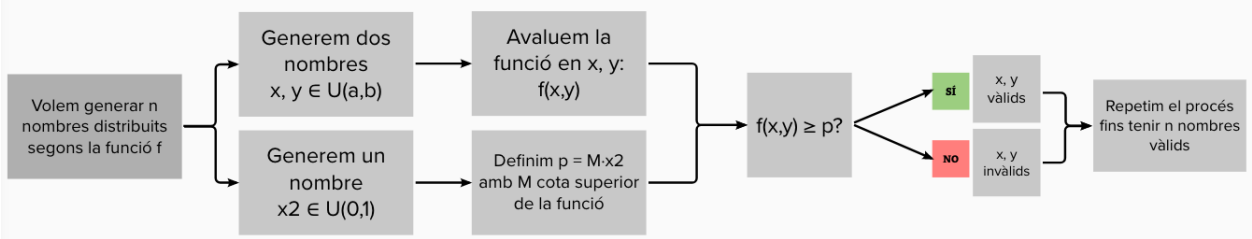


Figura 5: Seqüència del mètode d'Acceptació i Rebuig.

En el nostre cas, a=0, b=L, i hem pres M=0.21 per la cota superior.

Si es volen, per exemple, 100 nombres distribuïts segons aquesta densitat, s'haurà de repetir tot el procés fins a trobar 100 nombres vàlids.

Per tal de comprovar el correcte funcionament d'aquest mètode, es va programar aquest algorisme amb Fortran i es van fer histogrames mitjançant Gnuplot, per veure si els diferents punts creats s'adequaven al perfil de la densitat de probabilitat. Es van fer histogrames en x i y separadament per veure els talls en cada eix.

Un cop comprovat el bon funcionament del mètode, es va procedir a implementar-ho a Unity, a partir de la programació de l'algorisme en C Sharp. A partir d'aquest moment, es van necessitar certs coneixements de Unity per lligar el programa amb la pantalla de joc. En primer lloc, es va crear una quadrícula de 14x14 posicions, simulant la caixa en què l'electró està tancat. L'objectiu era mostrar per pantalla punts vermells simulant la posició de l'electró. Aquests punts són els nombres aleatoris, x, y obtinguts a partir del mètode d'Acceptació i Rebuig. Per tant, en el codi de C Sharp es va haver d'incorporar una ordre per tal que es creessin punts en la posició (x,y) sobre la quadrícula de joc.

```
Random _Random = new Random ();
M=0.21;
comptador=0;
ndat=100;
do
{
    double x10 = _Random.NextDouble();
    double x11 = _Random.NextDouble();
    x = L*x10;
    y = L*x11;

    double x2 = _Random.NextDouble();
    p = M*x2;
    funcio = f(x,y);

    if(funcio>p)
    {
        xnumsx[i]=x;
        xnumsy[i]=y;
        Instantiate(punt, new Vector3((float)x, -(float)y, 10), Quaternion.identity);

        comptador=comptador+1;
    }

} while(comptador<ndat);
```

Figura 6: Mètode d'Acceptació i Rebuig amb la implementació del codi pel dibuix dels punts en la quadrícula de joc.

En la Figura 6 s'exposa el codi utilitzat per programar l'algorisme d'Acceptació i Rebuig, per tal de generar ndat nombres distribuïts segons la funció f. Prèviament s'han hagut de definir les variables x, y, L, p, i els vectors on es guarden els nombres vàlids, xumsx[i], xnumsy[i]. Per tal que es mostri a la pantalla de joc el punt corresponent a la posició (x, y), hem d'utilitzar la ordre: Instantiate(punt, new Vector3((float)x, -(float)y, 10), Quaternion.identity). Aquesta línia fa possible col·locar un objecte "punt" creat anteriorment en la plataforma Unity, a la posició que volem. El concepte "Instantiate", fa referència a "duplicar" un objecte ja creat a la plataforma, i poder situar aquesta còpia a qualsevol posició. Tal com veiem en el codi, apareixeran a la pantalla els 100 primers punts vàlids obtinguts amb Acceptació i Rebuig. Des de Unity, vincularem un botó a aquest codi: cada cop que es premi el botó, es generaran 100 nombres distribuïts segons la densitat de probabilitat f, i apareixeran per pantalla. Això equivaldrà a fer mesures de la posició dels electrons.

4.2.2. Interfície, disseny i cohesió dels elements

Després d'aconseguir fer mesures de la posició de l'electró i mostrar-la per pantalla mitjançant els punts vermells, es va procedir a implementar la dinàmica del joc. La gran avantatge de Unity és que permet enllaçar els objectes i elements creats a la plataforma amb el codi a través del qual manipulem aquests objectes.

4.2.2.1. Creació d'escenes i objectes amb Unity

En primer lloc, s'han de crear diferents escenes per estructurar el nostre joc. Entenem per escenes les

diferents pantalles que li donaran forma, com per exemple, l'escena del menú, l'escena de joc o l'escena de l'explicació. Cadascuna d'elles es desenvolupa per separat i conté uns elements i objectes específics, així com scripts de codi. En una escena sempre tindrem l'apartat "Main Camera" i "Canvas", assenyalats en groc i verd respectivament en la Figura 8. "Main Camera" correspon al què es veu en la pantalla de joc en executar-lo, mentre que "Canvas"ha de contenir tots els objectes, botons i comptadors dels quals podem modificar les propietats mitjançant el nostre codi. Per tant, és important que els elements de "Canvas"quedin dins del marc de "Main Camera" per tal que es puguin visualitzar en la pantalla de joc. Via script, es pot carregar una escena en particular, mitjançant la comanda següent:

SceneManager.LoadScene(SceneManager."NomEscena");

Un cop creada l'escena principal del joc, el següent pas va ser la creació d'objectes en la plataforma Unity, simulant les naus del jugador. A aquests objectes se'ls hi pot carregar una imatge, per tal que tinguin l'aparença que vulguem; en el nostre cas, naus de Star Wars. Cada objecte del nostre joc té unes certes propietats: posició, color, sprite (imatge), animacions... que podem modificar a través de la pestanya "Inspector"que apareix en fer clic a l'objecte en qüestió de l'escena que estem desenvolupant, o bé via script. Paral·lelament a la plataforma Unity, escrivim el nostre codi en llenguatge C Sharp, definint els objectes que hem creat a Unity de la manera següent:

public GameObject "NomObjecte"

El terme "public" fa referència a què des de diferents scripts podem modificar les propietats d'aquest objecte, mentre que el terme "GameObject" ens indica de quin tipus d'element es tracta. Aquests scripts s'importen a Unity i s'incorporen dins de l'apartat "Main Camera". Per altra banda, dins la pestanya de les propietats de l'objecte, hi ha l'opció d'enllaçar-lo amb l'element que hem definit via script. D'aquesta manera, l'objecte de Unity estarà controlat pel nostre codi.

En un primer moment, col·loquem aquestes naus fora del taulell de joc per tal que posteriorment el jugador les pugui arrossegar a la posició que vulgui, de manera que cada lloc de la quadrícula només pugui estar ocupada per una nau. Per programar aquestes ordres, utilitzem unes funcions intrínseques de Unity, les quals es basen en els termes "Parent" i "Daughter", fent referència a un element sobre el qual se'n pot situar un altre, respectivament. En el nostre cas, els "Parents" són cada una de les caselles de la quadrícula, mentre que els "Daughters" són les naus.

```
void Start()
{
    dragParent = GameObject.FindGameObjectWithTag("DragParent").transform;  //que el joc trobi quin es el parent inicial
}

public void OnBeginDrag(PointerEventData eventData)
{
    itemDragging = gameObject; // associa l'objecte que arrosseguem al gameObject
    startPosition = transform.position; //guardem en un vector la posicio inicial de l'objecte que arrosseguem
    startParent = transform.parent; // assignem quin es el parent inicial
    transform.SetParent(dragParent); //li assignem a l'objecte un parent
}
public void OnDrag(PointerEventData eventData)
{
    transform.position = Input.mousePosition; //canviem la posicio de la nau, a on l'hem arrossegat
}
public void OnEndDrag(PointerEventData eventData)
{
    itemDragging = null;

    if(transform.parent == dragParent) //si la casella ja esta ocupada, es retorna la nau a la posicio original
    {
        transform.position = startPosition;
        transform.SetParent(startParent);
    }
}
```

Figura 7: Codi per arrossegar les naus sobre les caselles de la quadrícula.

4.2.2.2. Comptador de vides

Un cop el jugador ha situat totes les naus, s'habilita el botó corresponent per a poder fer les mesures de la posició de l'electró. Aquest botó també és un objecte del Canvas que podem enllaçar amb el nostre codi. En la pestanya Inspector corresponent, podem assignar que s'executi una funció definida en el nostre script quan es premi el botó, de manera que es posi en marxa el procés explicat a l'apartat 4.2.1. (apareixen per pantalla els punts vermells que simulen la posició de la partícula en haver fet una mesura). Per tant, sobre el taulell de joc hi haurà les naus del jugador i els punts vermells distribuïts d'una manera determinada. El botó per fer mesures s'ha de prémer fins a 5 vegades, fent que apareguin nous punts cada cop. Hem programat el joc de manera que el jugador pugui moure les naus segons convingui, veient la distribució que adopten els punts.

A continuació es va implementar la variable "comptador de vides". Es va crear un lliscador visible en la pantalla de joc que indica al jugador les vides que té en tot moment. A l'inici del joc es disposa de 10 vides. Mentre el jugador en conservi alguna, la partida segueix. Hem creat una dinàmica senzilla en la qual el jugador només pot perdre vides, és a dir, no hi ha cap manera de recuperar-les. El funcionament és el següent: després d'haver col·locat totes les naus sobre la quadrícula i d'haver premut el botó de fer mesures, si la posició d'algun dels punts vermells que han aparegut al taulell de joc coincideix amb la posició d'alguna de les naus, aquesta queda destruïda. En aquest cas, modifiquem la imatge de l'objecte en qüestió: se substitueix la nau per una calavera. D'aquesta manera, el jugador perd una vida, que es tradueix en el desplaçament del lliscador.

Si després d'aquest procés el jugador encara té vides, es procedeix a passar a la següent ronda. Es va fer mitjançant la comanda:

SceneManager.LoadScene(SceneManager.GetActiveScene().name);

Es recarrega l'escena, és a dir, tornem a tenir totes les naus actives fora de la quadrícula, i els punts

vermells ja no hi són. És com tornar a començar de nou, amb la diferència que hem programat que per a cada ronda l'aparença de les naus sigui diferent, i que el nivell energètic en què es troba l'electró canvia. Hem fet que el programa esculli de manera aleatòria n_x i n_y i aparegui a la pantalla una indicació o pista per guiar a l'usuari. No obstant això, la variable comptador de vides no es veu alterada, de manera que es conserven les vides que quedaven a la ronda anterior.

Per últim, es va haver de programar el missatge de fi del joc, diferent per a les dues possibles situacions: l'usuari havia aconseguit completar totes les rondes del joc, o bé havia perdut totes les vides.

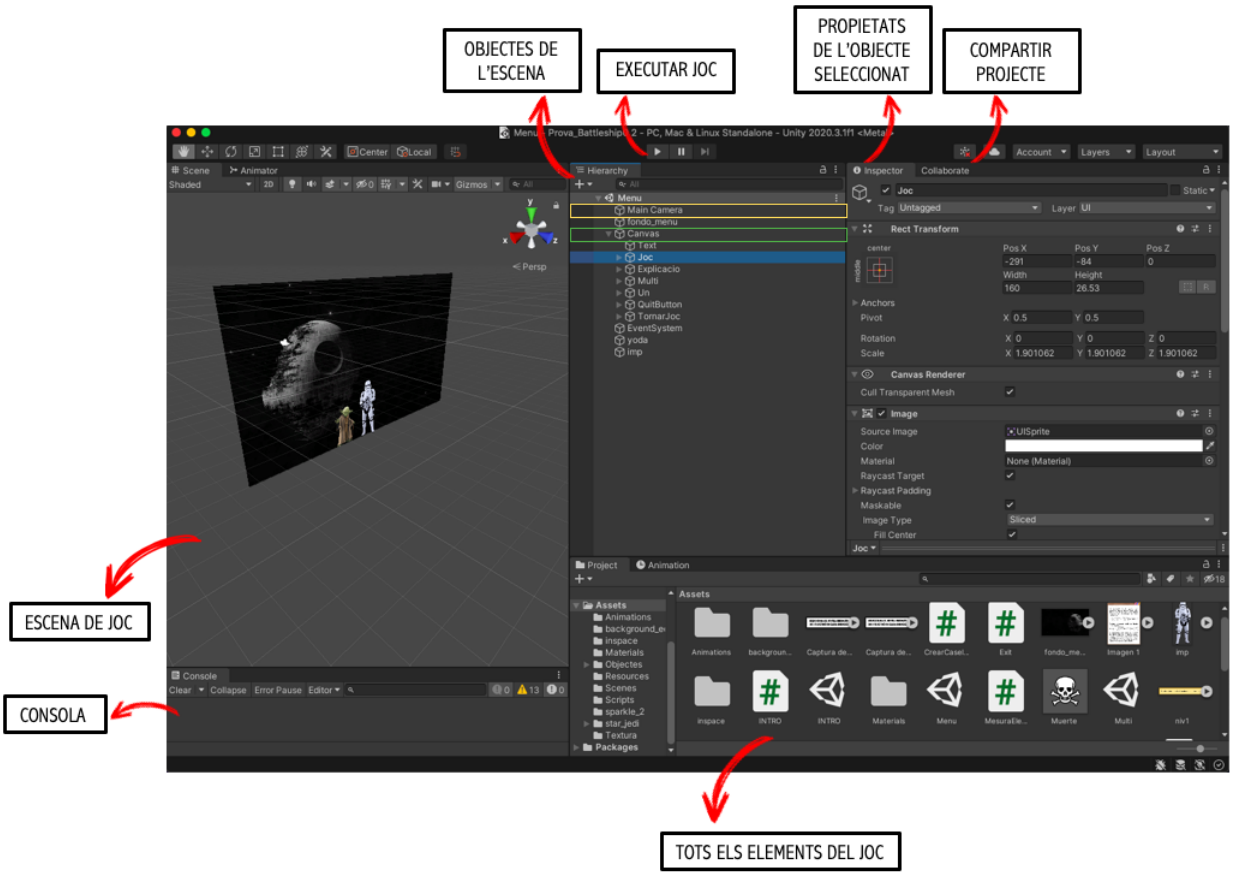
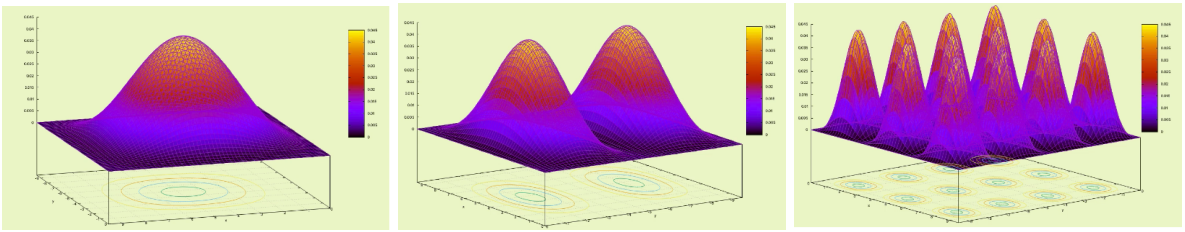


Figura 8: Imatge de la plataforma Unity amb l'escena del Menú carregada. S'indiquen les diferents parts del programa.

Aquest projecte es va voler ambientar en la saga de Star Wars: els gràfics i figures utilitzades són les que apareixen en les famoses pel·lícules. Per acabar de polir l'estètica, es va afegir un fons de pantalla adient a la temàtica del joc, així com imatges de Star Wars, un comptador de les mesures que s'han fet, i un indicador de la ronda en què es troba el jugador (vegeu apartat 5. Resultats). A part de l'escena principal del joc, es va crear una escena d'introducció, una pel menú, i una altra d'explicació de la teoria quàntica que hi ha al darrere.

4.2.3. Gràfics interactius

Aquesta última escena d'explicació de la teoria quàntica utilitzada al joc, va sorgir de la necessitat que el jugador pogués visualitzar la densitat de probabilitat de la funció d'ona que descriu l'estat de l'electró. En aquesta escena apareixen uns gràfics interactius, que es van crear amb l'eina Gnuplot. Es van fer figures en 3D que mostraven la densitat de probabilitat, per a diferents valors de n_x i n_y, que es van implementar al nostre joc de Unity. L'usuari disposa de lliscadors per controlar n_x i n_y, i segons els valors escollits, es carrega el gràfic animat en qüestió.



(a) $n_x = 1, n_y = 1$ (b) $n_x = 1, n_y = 2$ (c) $n_x = 3, n_y = 4$

Figura 9: Exemples de gràfics animats (en rotació) introduits en l'escena teòrica del joc.

Per acabar de lligar el joc, vam introduir un menú i diferents botons en cada escena per tal que el jugador pogués accedir a cadascuna d'elles fàcilment, en qualsevol moment del joc.

Més endavant es va incorporar l'opció multijugador. La programació duta a terme era molt semblant a la desenvolupada fins ara, amb la diferència que la majoria de variables es van haver de duplicar. Les naus de cada jugador s'havien de tractar per separat i això va comportar certa dificultat afegida, ja que en aquest cas era imprescindible diferenciar els objectes per tal de fer una bona comptabilització de les vides de cadascú.

4.3. Generar un executable

L'últim pas va ser generar un executable. Des de la pestanya File > Build Settings, escollim entre Windows/Mac. També tenim l'opció de configurar la resolució i la qualitat dels gràfics, així com quina serà la primera escena en executar el joc. En el cas del meu projecte, el primer executable que vam generar va ser una versió inacabada que no tenia incorporada l'opció multijugador. Aquesta versió, que constava de la part principal del joc per a un únic usuari i de la part teòrica-interactiva, va ser presentada a la Festa de la Ciència.

5. Resultats

Després d'haver seguit el procés explicat, obtenim un joc amb cara i ulls que consta de cinc escenes diferents, a través del qual hem pogut divulgar conceptes bàsics de la física quàntica. A continuació es mostren imatges de la pantalla de joc, i s'exposa el contingut de les diferents escenes:

Escena 1

En primer lloc, quan l'usuari executa el joc apareix per pantalla un text introductori per posar el jugador en context. Donant un toc divertit al joc, s'exposa la situació i l'objectiu que ha de tenir l'usuari, tot seguint la forma dels crèdits de les pel·lícules de Star Wars, afegint la música característica de la saga. Aquesta introducció s'ha creat mitjançant l'aplicació "Star Text", que permet personalitzar el famós text de les pel·lícules. Es pot saltar la introducció fent clic al botó "MENÚ"localitzat en la part dreta de la pantalla.

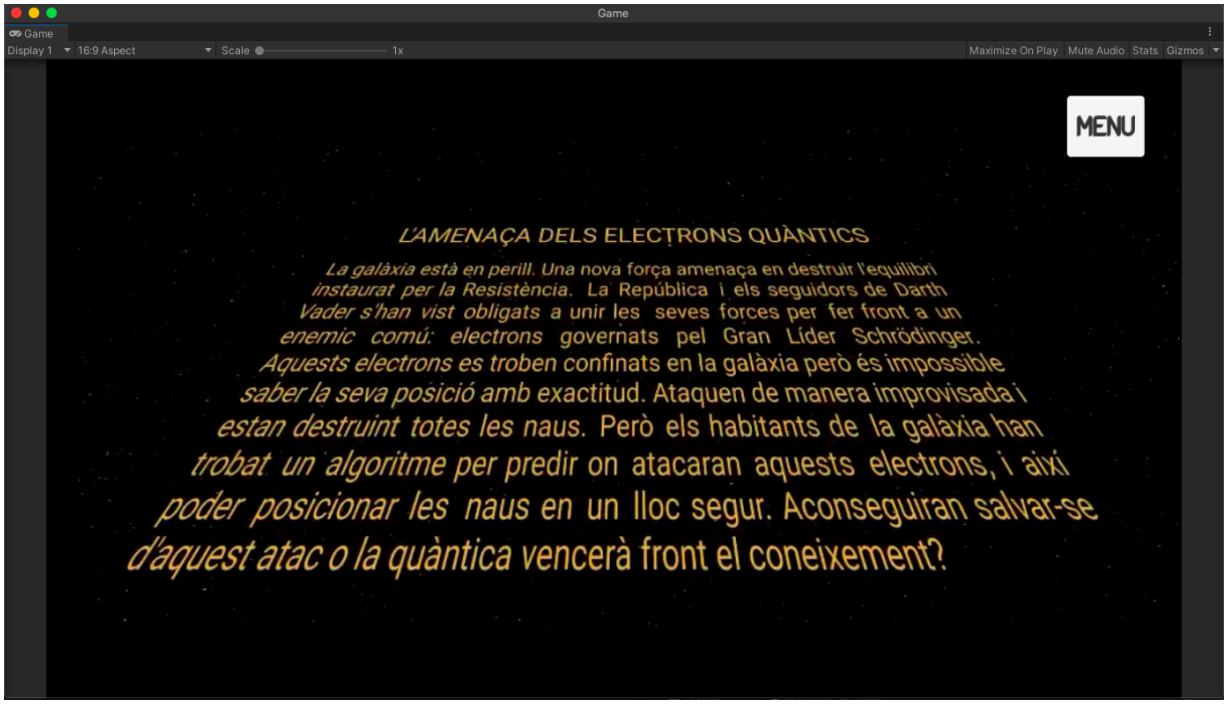


Figura 10: Imatge de la pantalla de joc. Escena 1: es posa en context al jugador.

Escena 2

Un cop vista la introducció, apareix automàticament la segona escena, que correspon al menú del nostre joc. Podem escollir entre dues opcions. Tal com podem veure en la Figura 11, el botó superior correspon a l'"entrenament". Si l'usuari fa clic en aquesta opció, se'l portarà a l'escena 3. Si per contra es vol començar a jugar, s'ha de prémer el botó inferior. L'usuari pot escollir entre jugar contra la màquina, o competir amb un company. En fer clic en el corresponent botó, s'obre la quarta escena (un sol jugador) o la cinquena (multijugador). En totes les escenes hi ha disponible un botó que porta directament al menú, al qual s'hi pot accedir en qualsevol moment. En mig d'una partida es pot retornar al menú, canviar d'opció i entrar a l'escena d'entrenament sense perdre les dades de la partida, que posteriorment es pot reprendre. Només des del menú és possible sortir del joc prement el botó "$EXIT$".

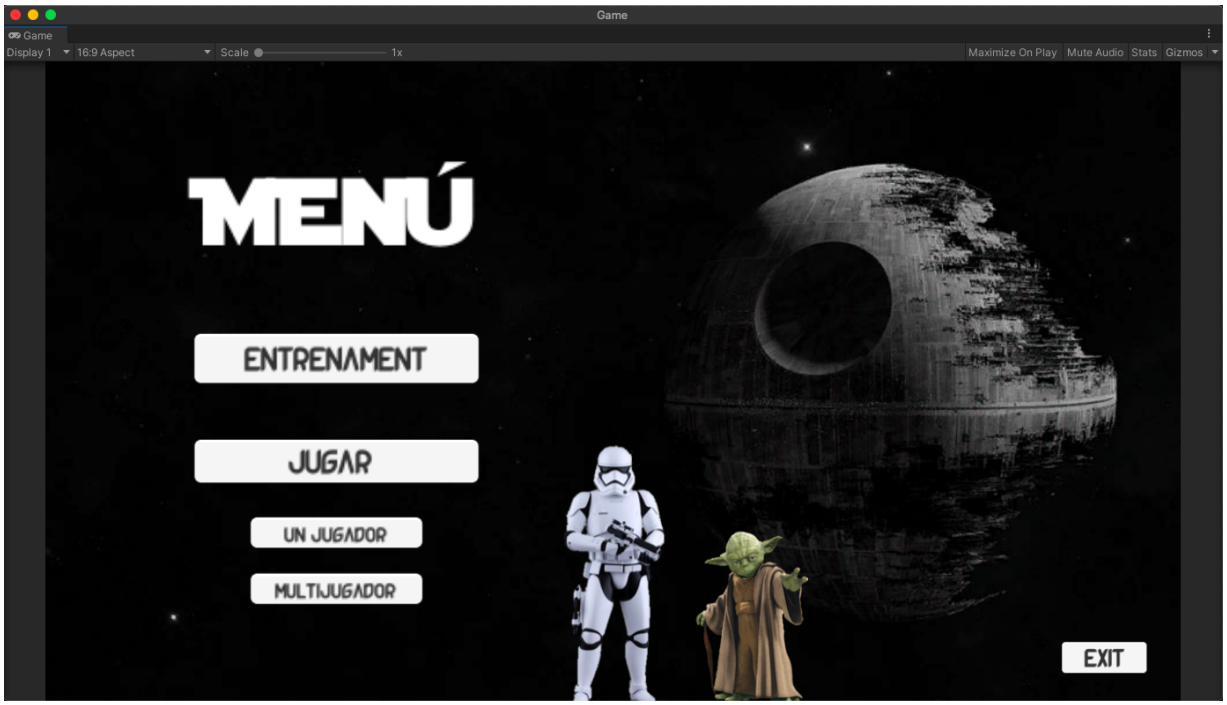


Figura 11: Imatge de la pantalla de joc. Escena 2: menú del joc.

Escena 3

En aquesta escena *d'"entrenament"* hi trobem l'explicació teòrica de la física quàntica que hi ha darrere del nostre joc. Serveix per fer entendre a l'usuari de manera il·lustrativa i interactiva el principi d'incertesa quàntic en què es basa el nostre programa per determinar la posició de l'electró tancat en una caixa bidimensional. A la part dreta de la pantalla s'exposa l'equació de Schrödinger i la funció d'ona que n'és solució. L'objectiu d'aquesta escena és fer entendre a l'usuari que la probabilitat de trobar l'electró en una certa posició queda determinada pels nivells energètics (en les direccions x i y) en què es trobi l'electró. Aquesta escena també permet a l'usuari veure gràficament la densitat de probabilitat de la funció d'ona, i en conseqüència on és més probable trobar l'electró en fer una mesura de la seva posició. S'han incorporat dos controls lliscants, també anomenats "sliders", un per a la direcció x, i l'altra per a la y, en què es pot controlar el valor del nivell energètic de l'electró dels dos eixos per separat (fins n_x, $n_y = 4$). Segons els nivells indicats per l'usuari, apareix per pantalla el gràfic animat corresponent a la densitat de probabilitat (la figura rota per tal que es pugui tenir una visió més completa i entenedora).

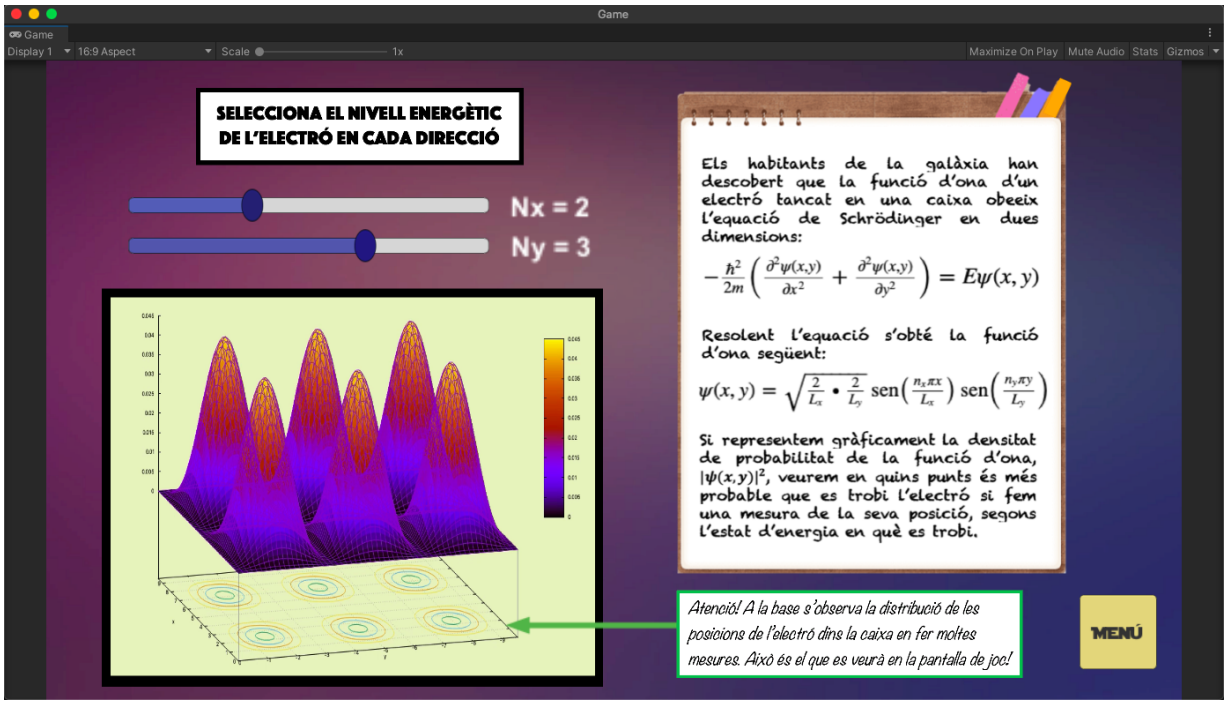
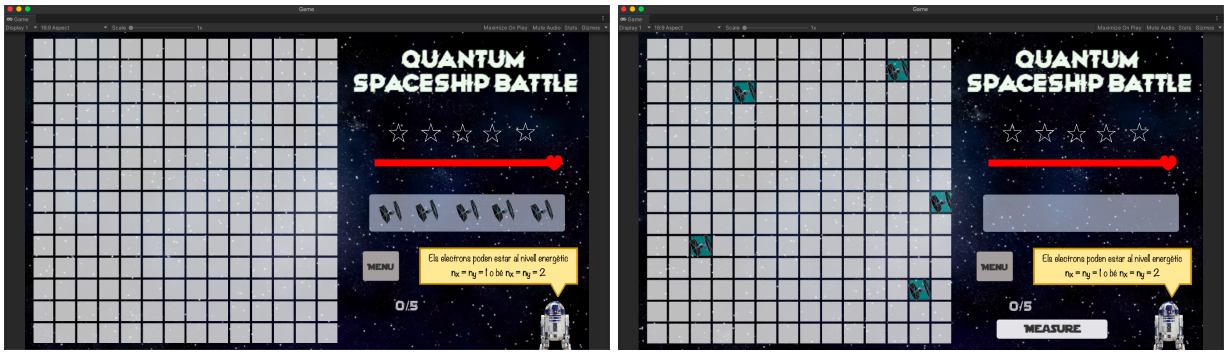


Figura 12: Imatge de la pantalla de joc. Escena 3: Explicació teòrica i gràfics interactius per tal que l'usuari entengui la quàntica que hi ha darrere del joc.

Escena 4

L'escena 4 és la corresponent al joc d'enfonsar la flota, per a un sol jugador. Al començament, les 5 naus espacials es troben a la part dreta de la pantalla, i l'usuari les ha d'arrossegar en 5 caselles diferents de la quadrícula. Un cop situades, s'habilita un botó que permet començar a fer mesures de la posició de l'electró. Sobre d'aquest botó s'ha afegit un comptador que ens indica el nombre de mesures que s'han realitzat (quantes vegades s'ha premut el botó "MEASURE"). Tal i com podem veure en la Figura 13, tenim altres elements de l'interfície del joc, com per exemple el slider vermell, que correspon a un comptador de les vides del jugador. Al començament del joc, l'usuari té 10 vides, quantitat que va disminuint quan una nau està situada en la mateixa posició de l'electró (posició que, recordem, no és exacta, tot el joc es basa en probabilitats). També disposem d'un robot (R2D2 pels fans de Star Wars), que ens indica en quin nivell energètic està l'electró. Un cop fetes 5 mesures, si l'usuari encara té vides (indicat pel slider vermell) es passa a la següent ronda, conservant les vides actuals, però reiniciant l'escena. Les naus canvien d'aspecte i s'han de tornar a col·locar tenint en compte que la probabilitat canvia, ja que, tal i com ens indica el robot R2D2, els nivells energètics de l'electró són diferents a cada ronda. En la Figura 13 es mostra la pantalla abans i després de col·locar les naus a les respectives caselles en la primera ronda del joc, encara no havent fet mesures.



(a) Abans de col·locar les naus

(b) Després de col·locar les naus

Figura 13: Imatge de la pantalla de joc. Escena 4: Primera ronda del joc d'enfonsar la flota, no havent fet mesures.

Tal i com s'ha explicat en l'apartat "3.2.Funcionament del joc", quan l'usuari prem el botó "MEASURE", apareixen per pantalla punts vermells que simulen la posició (més probable) de l'electró. Quan un d'aquests punts es troba en la mateixa casella que alguna de les naus, aquesta és destruia i se substitueix per una calavera, i el comptador de vides disminueix en una unitat. Podem veure-ho il·lustrat en la Figura 14.

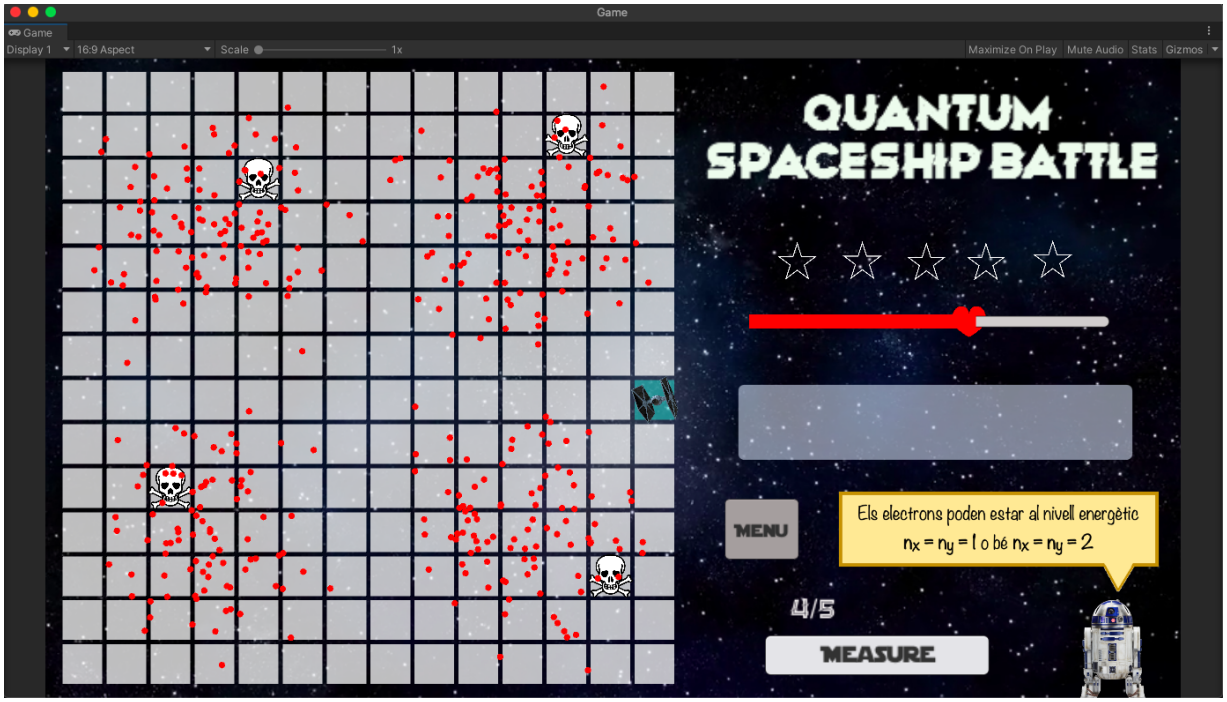


Figura 14: Imatge de la pantalla de joc. Escena 4: Primera ronda del joc d'enfonsar la flota, havent fet mesures.

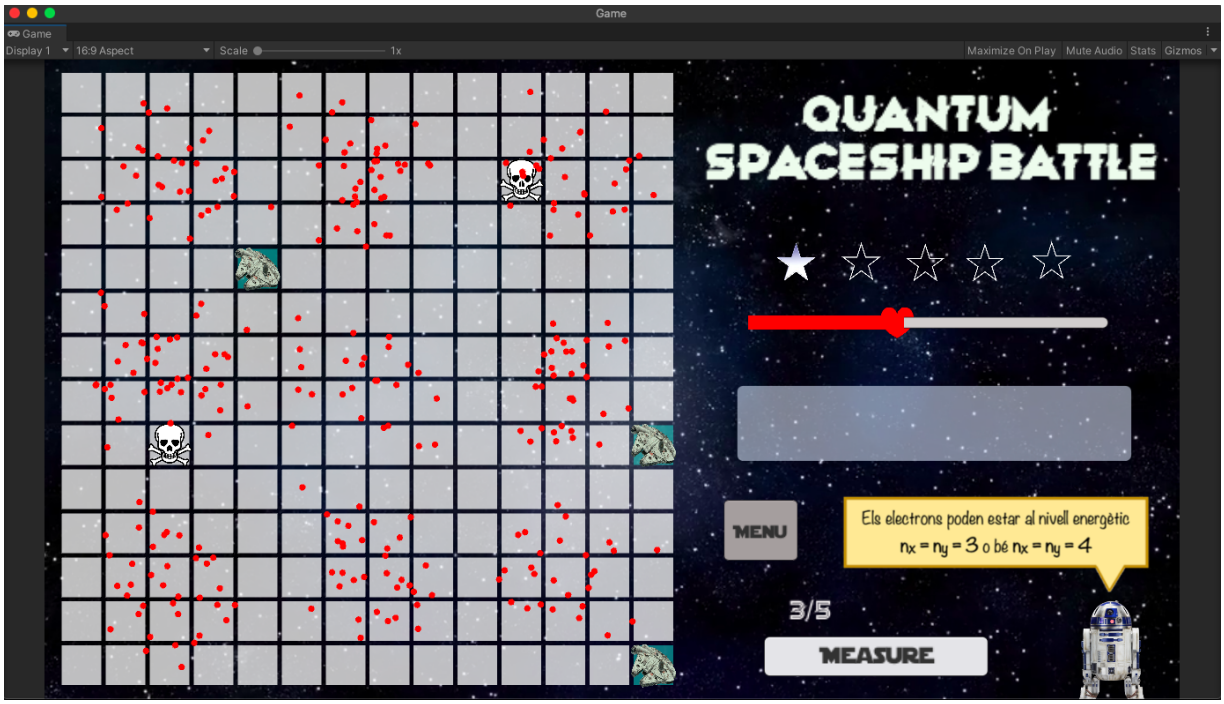
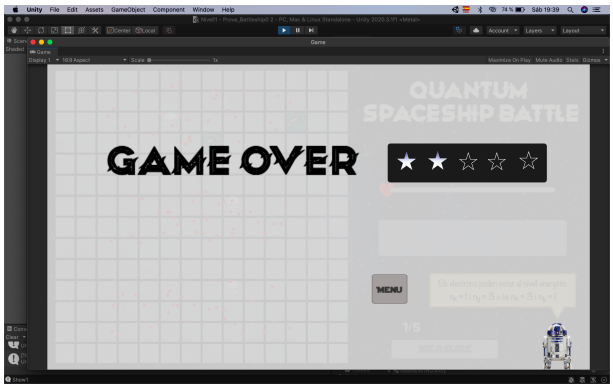
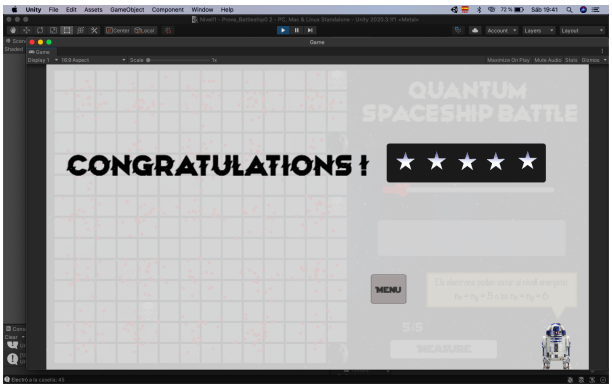


Figura 15: Imatge de la pantalla de joc. Escena 4: Segona ronda del joc d'enfonsar la flota, havent fet mesures.

Si el jugador aconsegueix superar 5 rondes sense haver perdut totes les vides al llarg del joc, l'usuari guanya la partida. Per contra, si perd totes les vides, s'acaba el joc, obtenint la qualificació de la ronda a la qual ha arribat. A la Figura 16 es pot veure la pantalla de finalització del joc.



(a) L'usuari ha perdut totes les vides. Ha completat les dues primeres rondes



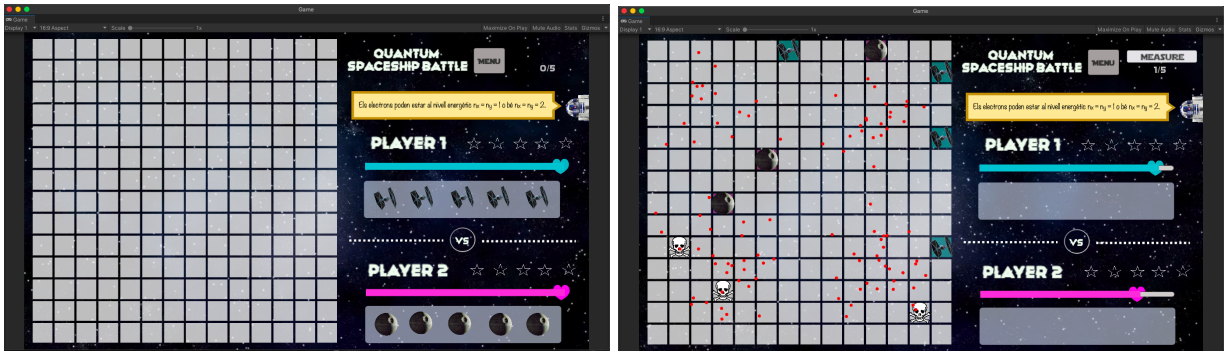
(b) L'usuari ha superat totes les rondes sense haver perdut totes les vides.

Figura 16: Imatge de la pantalla de joc. Escena 4: Finalització del joc.

Escena 5

L'escena 5 també correspon al joc d'enfonsar la flota, però en l'opció multijugador. Tal i com es mostra en la Figura 17, els dos jugadors han de col·locar les seves naus en el taulell de joc, i ambdos competiran contra l'electró tancat en una caixa. El jugador que aconsegueixi superar més rondes sense perdre totes les vides, serà el guanyador. Els elements que apareixen en l'escena són els mateixos que en l'apartat d'un

sol jugador, amb la diferència que en aquest cas tenim un comptador de vides diferent per a cada jugador i les naus de cadascú són diferents.



(a) Abans de col·locar les naus (b) Després de col·locar les naus, havent fet mesures

Figura 17: Imatge de la pantalla de joc. Escena 5: Primera ronda del joc d'enfonsar la flota en l'opció multijugador.

6. Divulgació del projecte

Com hem comentat anteriorment, el veritable objectiu d'aquest projecte consistia en la divulgació d'aquests jocs i simulacions a nois i noies amb interessos per la física quàntica. És per això que vam participar a la 14a edició de la Festa de la Ciència [9] organitzada per l'Ajuntament de Barcelona el passat mes de juny, duta a terme al Parc d'Investigació Biomèdica de Barcelona (PRBB). Dins del marc d'aquest event, juntament amb el professor Bruno Julià i un altre alumne, Lluc Vayreda, vam fer un taller en què exposàvem les nocions bàsiques de la física quàntica i permetíem als assistents interactuar amb les simulacions i jocs que havíem desenvolupat mentre els explicàvem els conceptes que hi havia al darrere.

(a) (b)

Figura 18: Imatges de la 14a edició de la Festa de la Ciència. Juny 2021, Barcelona.

7. Discussió

Després d'aquests mesos de treball, puc fer una valoració positiva del projecte QuantumLab UB. He aconseguit crear des de zero un joc senzill per ordinador amb un rerefons de física quàntica, per tant, aquestes pràctiques m'han permès combinar dues disciplines que m'interessen com són la quàntica i la computació. Al llarg d'aquest projecte he après un nou llenguatge de programació i el funcionament de la plataforma Unity, i he pogut treballar de manera autònoma i enfrontar-me a situacions que en un primer moment no sabia com resoldre. També m'ha fet veure la importància de compartir informació i coneixement amb els companys, que poden aportar un punt de vista diferent i idees interessants.

Tot i haver aconseguit desenvolupar una versió acabada del videojoc, es podrien haver afegit certs aspectes per tal d'obtenir un joc més complet, però per falta de temps no s'han pogut incorporar. Algunes d'aquestes millores serien les següents:

- Utilitzar naus que ocupessin diverses caselles com en el joc d'Enfonsar la Flota original. Això hauria donat més vitalitat i dinàmica a l'hora de fer un comptador de punts.
- Utilitzar diferents potencials per tal que la funció d'ona fos diferent, i en conseqüència també la densitat de probabilitat.
- Posar més restriccions en la possibilitat de moure les naus actives segons convingui. Per exemple, restar punts cada cop que es canviïn de posició o si es consulta l'escena amb els gràfics interactius, ja que dona pistes addicionals.
- Incorporar l'opció de multijugador online, de manera que els dos jugadors no hagin de compartir ordinador.
- A més a més de l'executable, generar una aplicació per tal que els usuaris puguin descarregar-la i jugar des dels seus dispositius.

Per altra banda, el fet d'haver creat aquest projecte amb Unity (i no amb Python com s'havia fet fins ara), obre les portes a altres alumnes a treballar amb aquesta plataforma especialitzada en la creació de videojocs. A més a més, fruit de la implementació de Unity en aquestes pràctiques, es podria plantejar la col·laboració amb entitats de videojocs.

Per últim, esmentar que han estat unes pràctiques molt enriquidores que m'han permès fer un primer pas dins el món de l'empresa i la investigació fora de l'àmbit acadèmic. A més a més, ha estat reconfortant treballar en un projecte que té un propòsit i una repercussió tan positiva en la societat científica, i estic contenta d'haver pogut aportar el meu granet de sorra a la divulgació de la Física Quàntica.

8. Referències

[1] Git Hub - QuantumLab UB.
https://github.com/brunojulia/quantumlabUB

[2] Documentació de Unity.
https://docs.unity3d.com/es/2019.4/Manual/index.html

[3] Guia de programació en C Sharp.
Introducción a C Sharp: Manual del estudiante, Miguel Muñóz Serafín

[4] Apunts de l'assignatura de Física Computacional (UB)
Física Computacional, Bruno Juliá i Marta Ibañes

[5] Unity Learn
https://learn.unity.com/tutorials

[6] Gnuplot Documentation by Thomas Williams and Colin Kelley
*http://www.gnuplot.info/docs*5,2/*Gnuplot*5,2.*pdf*

[7] Generación de variables aleatorias continuas, Georgina Flesia
$http://www2.famaf.unc.edu.ar/\ jgimenez/ModelosySimulacion/2013/clase10.pdf$

[8] Star Wars image stock
$https://www.istockphoto.com/es/search/2/image?phrase=star+wars$

[9] Festa de la Ciència, 14a edició
https://www.biennalciutaticiencia.barcelona/ca/programa/simulacions-de-mecanica-quantica