

Treball de Fi de Grau

Enginyeria en Tecnologies Industrials

**DISSENY D'UN JOC SERIÓS D'APRENTATGE DE L'ÚS
D'INSTRUMENTACIÓ D'UN LABORATORI QUÍMIC**

MEMÒRIA

Autor: Laura Torrens Quesada

Director/s: Daniela Tost Pardell, Maria Martínez Martínez

Convocatòria: Setembre 2015



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

RESUM

La finalitat d'aquest projecte és crear un joc seriós que simuli el funcionament d'un aparell del laboratori químic, mitjançant la programació.

Es tracta de fer una simulació d'un espectrofotòmetre d'absorció atòmica de forma que es pugui realitzar la pràctica de laboratori de l'assignatura de Química sense necessitat d'assistir al laboratori, simplement mitjançant l'ordinador. La simulació no pretén ser una substitució de la pràctica però sí una forma de que l'alumnat aprengui a manipular l'instrument tenint en compte totes les mesures de seguretat associades. D'aquesta manera, pot fer-se una idea del que es trobarà al laboratori fent aquesta sessió molt més productiva de cara a l'aprenentatge.

El joc pot emprar-se com una activitat avaluable o simplement com una activitat de caràcter formatiu. En algun cas podria substituir la sessió presencial però aquest no és en cap cas l'objectiu principal.

Es busca a més, que l'usuari pugui divertir-se alhora que aprèn els conceptes i coneixements que engloba aquesta pràctica.

El llenguatge de programació escollit per crear el joc en 2D ha estat Python, un llenguatge d'alt nivell que facilita la seva escriptura, lectura i correcció. També s'ha utilitzat Pygame, un conjunt de mòduls que formen part de la llibreria de Python i que permeten la creació i el desenvolupament de videojocs ja que identifiquen els diferents elements que pot tenir un joc, com són els sons, imatges, superfícies, esdeveniments o fonts de text.

Finalment, els resultats del projecte han estat els esperats. S'ha obtingut un joc seriós que motiva a l'usuari alhora que permet aprendre i assimilar els coneixements associats a la pràctica de laboratori.

INDEX

1. INTRODUCCIÓ	5
1.1 CONTEXT	5
1.2 OBJECTIUS.....	5
1.3 ESTAT DE L'ART	6
1.3.1 Python 3.3	6
1.3.2 Pygame 3.....	7
1.3.3 GIMP	7
1.4 AVANTATGES	8
1.5 DESAVANTATGES.....	8
2. ANTECEDENTS.....	9
2.1 ABSORCIÓ ATÒMICA	9
2.2 ESPECTROFOTÒMETRE D'ABSORCIÓ ATÒMICA.....	10
2.2.1 COMPONENTS FUNCIONALS	10
2.2.2 COMPONENTS ESTRUCTURALS	12
2.3 PART PRÀCTICA AL LABORATORI.....	13
2.3.1 OBJECTIUS.....	13
2.3.2 PROCEDIMENTS.....	14
2.3.3 MÈTODE D'ADDICIÓ ESTÀNDAR	15
2.4 EL VIDEOJOC.....	17
3. ESPECIFICACIÓ	19
4. PLANIFICACIÓ.....	20
4.1 TASQUES.....	20
4.2 ESTIMACIÓ ECONÒMICA	20
4.3 DEDICACIÓ EN TEMPS	21
5. DISSENY DEL JOC	23
5.1 OBJECTES	23
5.2 DIAGRAMA D'ESTATS	25
5.3 PUNTUACIÓ I VIDES.....	29
6. IMPLEMENTACIÓ DEL JOC	31
6.1 ESCRIPTURA DEL CODI DE PROGRAMACIÓ.....	31
6.2 MÒDULS DE PYGAME	32
6.2.1 PYGAME	32
6.2.2 DISPLAY	32

6.2.3 EVENT.....	33
6.2.4 SURFACE.....	33
6.2.5 RECT.....	33
6.2.6 SPRITE.....	33
6.3 MÒDULS PROPIS.....	34
6.4 ESTATS I ACCIONS DELS COMPONENTS.....	36
6.4.1 AIRE I ACETILÈ.....	37
6.4.2 ESPECTROFOTÒMETRE.....	37
6.4.3 PANTALLA.....	39
6.4.4 VASOS DE PRECIPITATS.....	40
6.5 RESTRICCIONS DEL JOC.....	41
7. VALIDACIÓ.....	42
8. CONCLUSIONS.....	44
BIBLIOGRAFIA.....	45
ANNEX.....	46
A. DESCRIPCIÓ DE FIGURES.....	46
B. MANUAL DE L'USUARI.....	47
C. PANTALLES.....	49
D. MÒDULS PYTHON.....	57
D.1 practica.py.....	57
D.2 Global.py.....	60
D.3 escena.py.....	61
D.4 intro.py.....	61
D.4 principal.py.....	62
D.5 fin.py.....	67
D.7 component.py.....	68
D.8 objecte.py.....	70
D.9 textObjecte.py.....	73
D.10 componentGrafic.py.....	73
D.11 componentGraficText.py.....	74
D.12 escenaGrafica.py.....	75
D.13 estats.py.....	76
D.14 input.py.....	77
E. QÜESTIONARI PROVES D'USUARI.....	78

1. INTRODUCCIÓ

1.1 CONTEXT

Aquest projecte consisteix, bàsicament, en la creació d'un joc seriós que simula el funcionament d'un instrument del laboratori químic: un espectrofotòmetre d'absorció atòmica amb flama.

Dins del programa d'estudis del grau en Enginyeria Química de l'ETSEIB es troba l'assignatura *Experimentació en Química*, assignatura de sis crèdits que s'ensenya al segon curs del grau. Aquesta consisteix en aprendre i aplicar els coneixements de química adquirits durant el grau i posar-los en pràctica al laboratori d'una forma dinàmica i tangible.

Una de les pràctiques que es realitzen durant el curs és "*Control de metalls tòxics en mostres aquoses per espectrofotometria d'Absorció Atòmica*", on l'aparell a utilitzar és l'espectrofotòmetre, objecte principal d'aquest projecte.

Aquest programa neix de la necessitat dels alumnes de Química de desplaçar-se al laboratori de la Universitat per repassar els conceptes i procediments de la pràctica, de cara a l'examen de pràctiques que es realitza al final del quadrimestre. L'accés al laboratori està limitat per dues raons: es necessita la supervisió d'un membre del professorat obligatòriament i l'horari d'apertura de l'aula és limitat.

Cal esmentar també que la creació d'un programa que substitueixi la pràctica reiterada dels alumnes, redueix considerablement la despesa econòmica del laboratori: material, professorat, incidències, electricitat, llum...

El programa de simulació es crea amb el llenguatge de programació d'alt nivell Python i la llibreria Pygame, un conjunt de mòduls per Python que permeten la creació de videojocs.

1.2 OBJECTIUS

Els principals objectius d'aquest projecte són:

- Aplicar i ampliar els coneixements de programació informàtica adquirits durant el grau.
- Dissenyar i desenvolupar un programa d'utilitat que cobreixi una necessitat real.
- Crear un joc seriós que permeti a l'usuari aprendre alhora que es diverteix.
- Ampliar els coneixements de química sobre l'absorció atòmica.
- Planificar i estructurar un projecte ambiciós d'un videojoc.
- Conèixer les fases de desenvolupament d'un videojoc.
- Aprendre a organitzar i optimitzar els recursos disponibles del projecte.

- Arribar a dominar el llenguatge de programació emprat així com els diferents softwares o programes que seran utilitzats al llarg del projecte.
- Planificar amb detall la distribució de feina per tal d'optimitzar el temps i poder complir al màxim les expectatives esperades.
- Crear un videojoc que compleixi amb els objectius plantejats.

1.3 ESTAT DE L'ART

1.3.1 Python 3.3



Il·lustració 1: origen a referència [12]

Python és un llenguatge de programació d'alt nivell com són C++, PHP o Java. Els programes escrits en llenguatges d'alt nivell han de ser processats abans de poder funcionar, a diferència d'aquells escrits en llenguatges de baix nivell. El gran avantatge dels llenguatges com Python és que són molt més senzills alhora de crear programes ja que són més curts i més fàcils d'escriure, llegir o corregir. A més, funcionen en diversos tipus de màquines i programes operatius amb poques o cap modificació, el que els hi dóna molta flexibilitat a diferència dels llenguatges de baix nivell.

Aquest llenguatge admet múltiples paradigmes de programació: estructurada, funcional, orientada a objectes i orientada a aspectes. A més, permet la combinació de tots aquests paradigmes dins un mateix codi dotant-lo d'una gran versatilitat. Va ser dissenyat per ser fàcilment llegible, sent la seva característica més notòria l'ús del sagnat per delimitar blocs de programa. Això el diferencia d'altres llenguatges, que utilitzen diferents signes de puntuació, com poden ser les claus {}, per realitzar aquesta funció.

Un altre dels grans trets de Python és el seu extens conjunt de llibreries, que conformen un ampli ventall d'eines aptes per a la realització de qualsevol tasca. Així com el suport per a llibreries escrites en C, que serà un dels pilars d'aquest projecte, ja que Pygame es basa en la llibreria *SLD (Simple Directmedia Layer)*, una llibreria multimèdia multi plataforma de C++ dissenyada per proporcionar accés de baix nivell a l'àudio, teclat o ratolí.

Actualment es troba a la versió 3.4.2, llançada el 13 d'octubre de 2014, però en aquest projecte s'usarà la versió 3.3.0 amb l'objectiu de tenir compatibilitat amb la biblioteca Pygame 3.

1.3.2 Pygame 3



Il·lustració 2: origen a referència [11]

Pygame és un conjunt de mòduls per Python que permeten la creació de videojocs. Està basat en la llibreria SDL comentada anteriorment. Està dividit en diversos mòduls que agrupen múltiples funcionalitats comunes per a desenvolupar videojocs o aplicacions

que utilitzen gràfics en 2D. Al estar escrit en Python, té tots els avantatges del multi paradigma i, per tant, utilitzar-se a diversos programes operatius com Windows, Linux o Macintosh.

Avantatges:

- Especialitzat en 2D
- Multi plataforma
- Proporciona tot lo necessari per crear un joc
- Programació senzilla al haver estat creat per Python
- No necessita una interfície gràfica

Desavantatges:

- No permet crear jocs en 3D
- Velocitat d'execució més lenta al haver estat creat per Python.

1.3.3 GIMP



Il·lustració 3: origen a referència [6]

GIMP és un programa d'edició i retoc d'imatges digitals. Conté eines per retallar, dibuixar, editar, fer fotomuntatges, convertir a diferents formats...

Una de les característiques més importants del programa és que aquest permet treballar les imatges en diferents capes, de forma que es poden treballar les diferents capes de manera

independent.

És el programa de manipulació de gràfics disponible en més sistemes operatius (Unix, GNU/Linux, FreeBSD, Solaris, Microsoft Windows, Macintosh...).

Serà emprat principalment pel retoc i creació de les imatges dels diferents objectes del joc.

1.4 AVANTATGES

Un cop es va plantejar la idea de crear un joc que simules una pràctica de laboratori, van estudiar-se les possibles avantatges que això tindria sobre la realització presencial d'aquesta al laboratori de l'Escola.

L'objectiu principal del joc, com s'ha comentat amb anterioritat, és suprimir la necessitat de l'alumne d'assistir al laboratori per tal de poder practicar amb l'aparell i repassar els conceptes. Aquesta necessitat neix del fet que l'assistència al laboratori està limitada per l'horari de l'Escola o bé per la falta de personal a la sala, la qual cosa es obligatòria. S'han definit però alguns impactes positius que comportaria la creació d'aquest joc:

- Estalvi econòmic a llarg termini degut a:
 - Reducció d'hores de feina del personal de laboratori
 - Reducció de les despeses de llum i electricitat del laboratori
 - Reducció de material químic emprat
- Disminució del risc al laboratori: la pràctica treballa amb flama, gasos inflamables...
- Suprimeix la necessitat de l'alumne de desplaçar-se fins l'Escola
- Suprimeix la limitació horària, la simulació pot fer-se en qualsevol moment del dia
- Suprimeix la limitació de material: el laboratori consta d'una sol espectrofotòmetre per tots els alumnes mentre que el programa pot ser personal
- Augmenta la capacitat de concentració mentre es realitza la pràctica: Al haver-hi només un aparell les pràctiques es realitzen en grup i la responsabilitat es reparteix. A la simulació és un únic usuari el que realitza tota la pràctica sencera
- Avaluació numèrica del procés mitjançant la puntuació del joc

1.5 DESAVANTATGES

Lògicament hi ha aspectes que la simulació no pot imitar de la realitat. Encara que s'han definit molts trets positius de la creació del projecte, també hi ha trets que no són tant positius o avantatjosos:

- La pràctica queda reduïda a la utilització del espectrofotòmetre, suprimint la part química de la preparació de les mostres i els patrons
- La simulació limita el camp de visió, les opcions i alguns possibles errors
- La supressió del risc pot ser un desavantatge quan degut a la seva absència, la pràctica es fa sense prestar atenció

2. ANTECEDENTS

2.1 ABSORCIÓ ATÒMICA

L'objectiu de la pràctica és l'estudi de la radiació electromagnètica absorbida per una mostra a la zona de l'ultraviolat (UV) i del visible (VIS) de l'espectre electromagnètic, per determinar la quantitat de metall que es troba en aquesta mostra.

L'espectrofotometria d'absorció atòmica es una tècnica per a determinar la concentració d'un element metàl·lic determinat en una mostra. Pot utilitzar-se per analitzar la concentració de més de 62 metalls diferents en una mateixa dissolució.

FONAMENT TEÒRIC

L'àtom de qualsevol element consisteix en un nucli i un determinat número d'electrons que omplen diferents nivells d'energia, la configuració més estable d'un àtom correspon a la de menor contingut energètic, és a dir, quan els seus electrons es troben en els nivells d'energia menors. Aquest estat és l'anomenat "estat fonamental".

Si un àtom que es troba en un estat fonamental absorbeix una determinada energia, aquest experimenta una transició cap a un estat particular de major energia. La quantitat de llum absorbida per la mostra depèn de:

- L'energia i la freqüència de la radiació utilitzada
- La naturalesa de la mostra
- El número de molècules sobre el qual incideix la radiació

Quan la mostra absorbeix la radiació a una determinada longitud d'ona, la intensitat de la radiació inicial (I_0) disminueix. Aquest efecte es pot mesurar com a Transmissió (T) o com a Absorbància (A):

- Transmissió o percentatge de radiació que travessa la mostra: $T = \frac{I}{I_0}$
- Absorbància o quantitat de radiació absorbida per la mostra: $A = -\log T$

L'absorbància en funció de la longitud d'ona es característica de cada component, i és per aquesta raó que es pot determinar quin component és present a la mostra estudiada.

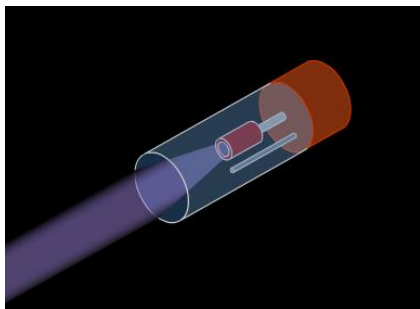
2.2 ESPECTROFOTÒMETRE D'ABSORCIÓ ATÒMICA

2.2.1 COMPONENTS FUNCIONALS

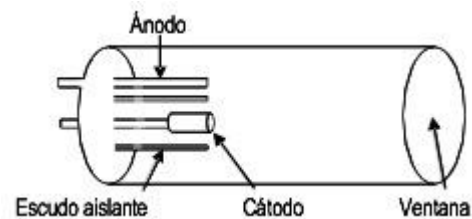
Un espectrofotòmetre és un aparell que mesura la quantitat de radiació absorbida quan una radiació d'una determinada energia travessa una mostra, és a dir, l'absorbància.

Els principals components relacionats amb el procés de funcionament del sistema que va canviant d'estat, és a dir, amb la circulació de matèria, energia i informació, són els següents:

- Font de radiació: emet radiació electromagnètica continua en les zones del visible o de l'ultraviolat. Les més utilitzades són les làmpades de càtode, constituïdes per un càtode metàl·lic capaç d'emetre radiacions de les mateixes longituds d'ona que són capaços d'absorbir els àtoms dels elements que es desitja analitzar, ja que aquest està constituït del metall en qüestió. Quan s'aplica una diferència de potencial suficient a través de l'ànode i el càtode, el gas s'ionitza i els cations gasosos són accelerats adquirint suficient energia cinètica per arrencar alguns àtoms metàl·lics del càtode. Aquests àtoms són excitats al xocar contra els ions i emeten una radiació característica al tornar al seu estat fonamental.



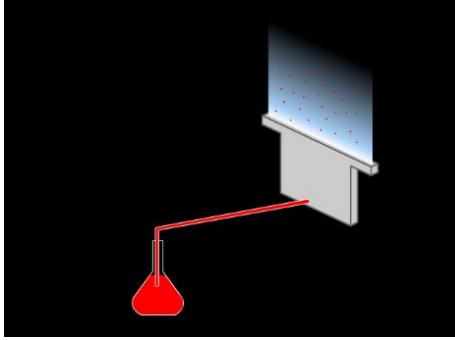
Il·lustració 5: origen a referència [4]



Il·lustració 4: origen a referència [9]

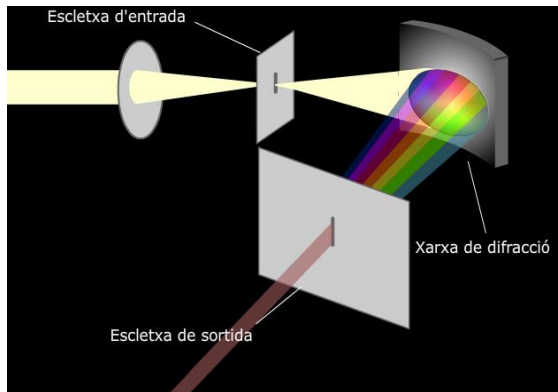
- Font tèrmica o atomitzador: Tenint en compte que les espècies absorbents han d'esser àtoms aïllats, cal disposar d'un sistema per produir aquest vapor atòmic. Això s'aconsegueix mitjançant energia tèrmica. Aquesta energia és la que proporciona un atomitzador, el tipus d'atomitzador més emprat és la flama. Aquesta tècnica requereix la nebulització de la dissolució que conté l'analit per tal de produir un aerosol que és conduït cap a la flama. El vapor atòmic que es genera és travessat per una radiació específica de l'element que és vol analitzar, generada per la làmpada de càtode buit. Una part d'aquesta radiació és absorbida pel àtoms de l'analit.

És important que la flama sigui transparent, és a dir, que no absorbeixi radiació provinent de la làmpada. La flama aire/acetilè és la més utilitzada ja que és completament transparent i solament mostra auto absorció sota els 230 nm.



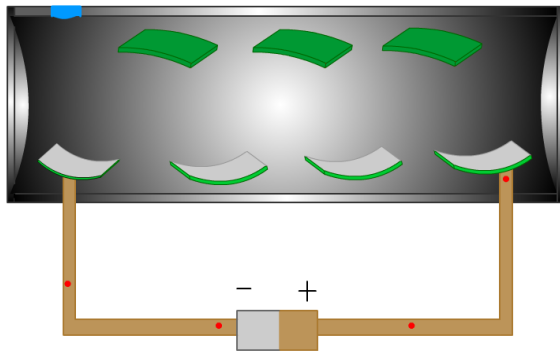
Il·lustració 6: origen a referència [4]

- Sistema òptic (monocromador) : selecciona una longitud d'ona de la radiació policromàtica que prové de la font. D'aquesta manera es pot treballar amb la longitud d'ona constant que permetrà calcular el valor de l'absorbància. Com s'ha comentat anteriorment, es treballarà amb la longitud d'ona que dona els màxims nivells d'absorbància.



Il·lustració 7: origen a referència [4]

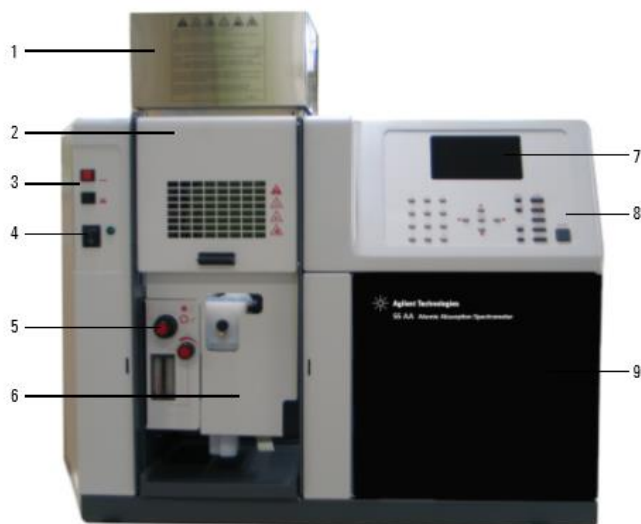
- Detector: transforma el senyal òptic que li arriba del monocromador en un senyal elèctric que permet mesurar la radiació absorbida a través d'un sistema de lectura. El detector utilitzat en l'espectrofotòmetre és un fotomultiplicador, un dispositiu que per efecte fotoelèctric produeix un fotoelectró que va sent multiplicat degut al camp elèctric creat per una tensió externa. La resposta final del dispositiu consisteix en un únic impuls d'amplitud proporcional a l'energia subministrada externament.



Il·lustració 8: origen a referència [4]

2.2.2 COMPONENTS ESTRUCTURALS

A la següent imatge es pot observar l'estructura que es troba alhora de utilitzar un espectrofotòmetre. Aquests components, visibles a l'usuari, formen part de l'organització o distribució a l'espai dels elements que componen l'estructura.



Il·lustració 9: origen a referència [1]

1. Extractor: La calor, vapors i fums generats poden ser perillosos, tòxics o perjudicials per al personal. Aquests han de ser extrets de l'instrument per mitjà d'un sistema d'escapament. L'instrument ha de ser ventilat mitjançant una campana, conductes i l'extractor d'aire cap a l'exterior de l'edifici, mai cap a dins.
2. Carcassa protectora de la flama: protegeixen l'usuari de la radiació ultraviolada emesa per la flama
3. Interruptor flama: per encendre i apagar la flama
4. Interruptor aparell: per encendre i apagar l'aparell
5. Controls de la flama: per controlar la magnitud de la flama

6. Compartiment frontal de la mostra
7. Pantalla LCD
8. Teclat
9. Compartiment de les làmpades: On es troben les dues làmpades de càtode buit

2.3 PART PRÀCTICA AL LABORATORI

2.3.1 OBJECTIUS

Com s'ha comentat anteriorment, l'objectiu principal és la determinació de metalls en una mostra aquosa mitjançant espectroscòpia d'absorció atòmica.

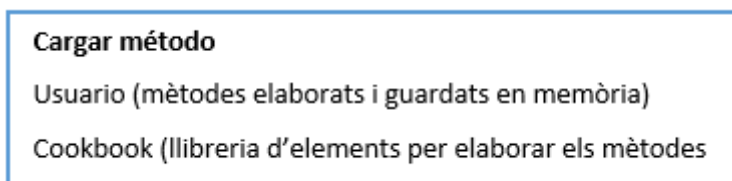
Al finalitzar aquest experiment l'estudiant ha de ser capaç de:

1. Diferenciar entre els mètodes de calibrat extern i per addició estàndard
2. Identificar els principals metalls tòxics o contaminants presents en les aigües
3. Utilitzar un espectrofotòmetre d'absorció atòmica amb flama aire/acetilè, seguint un manual
4. Identificar la mesura de l'absorbància com a paràmetre per quantificar la quantitat de metall per Absorció atòmica
5. Preparar de forma correcta dissolucions amb més d'un patró, distingint quan el mètode és d'addició estàndard
6. Construir i utilitzar una recta de calibratge pel mètode d'addició estàndard
7. Ajustar els punts experimentals per regressió lineal per el mètode de mínims quadrats
8. Reconèixer l'aplicació d'aquesta tècnica per al control de qualitat de mostres d'aigües.

Dels objectius de la pràctica, el programa cobreix el 3, 4, 6, 7 i 8. Cal dir que no és necessari cobrir-los tots ja que el joc està dissenyat per repassar els conceptes, no per substituir tota la pràctica del laboratori. Els estudiants utilitzaran el programa un cop hagin fet la pràctica a l'Escola, per repassar dubtes, conceptes que no hagin quedat clars i per interioritzar el procediment. Per un usuari que no ha fet aquesta pràctica, el programa li serveix per entendre el concepte d'absorció atòmica i aprendre a utilitzar l'espectrofotòmetre.

2.3.2 PROCEDIMENTS

1. A partir de les dissolucions patró de cada metall (100ppm) es preparen 8 dissolucions de diferent concentració en matrassos de 25 ml. Cadascuna de les dissolucions ha de contenir 10 ml de mostra i el volum corresponent de cadascuna de les dissolucions patró dels metalls.
2. S'encén l'interruptor de l'espectrofotòmetre que es troba a l'esquerra de l'aparell [4]. S'encén la llum verda i a continuació apareix a la pantalla el logotip de VARIAN. Seguidament apareix:



Amb les fletxes del teclat [8] es col·loca a sobre <<Usuario>> i es pitja INTRO.

3. A la pantalla apareixen els mètodes d'absorció que estan memoritzats (només depenen del metall de la làmpada que s'utilitza):
 - 1 Cd 228.8
 - 2 Zn 213.9
 - 3 Ni 352.5
 - 4 Cu 324.8

Es selecciona un dels mètodes mitjançant les fletxes i el botó INTRO.

4. S'obre la porta del compartiment de les làmpades [9], mitjançant una petita palanca que es troba a la part superior – esquerra, es selecciona la làmpada 1 (palanca cap a baix) o la làmpada 2 (palanca cap a dalt). La làmpada seleccionada ha d'anar d'acord amb el mètode seleccionat.
5. Es tanca la porta, es pitja el botó OPTIMIZAR [8] i apareixen a la pantalla les absorbàncies de les làmpada escollida sota aquest dibuix:

Cd
_ T _____

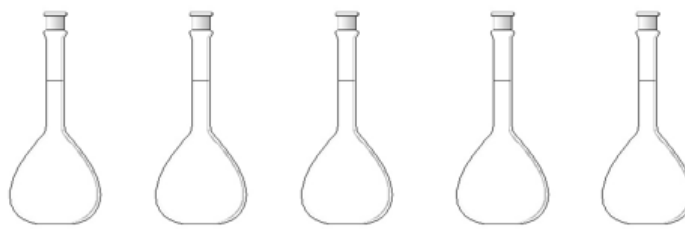
Amb les fletxes es col·loca damunt de *Lampara HC* i es pitja INTRO. El programa comença a optimitzar-se i s'ha d'esperar un mínim de quinze minuts. Les absorbàncies que apareixen inicialment han d'aproximar-se al valor màxim que és 1, per aconseguir-ho s'ha d'anar prement INTRO les vegades que calgui per tal de re-escalar el sistema.

6. Es col·loca a la pantalla amb ajuda de les fletxes damunt de <<SEÑAL>> i es prem: INTRO i seguidament ALT + LECTURA per fer el zero instrumental.
7. S'obren els gasos: primer l'aire, després l'acetilè.
8. Seguidament es pressiona durant uns segons el botó negre per encendre la flama
9. Un cop encesa la flama, es col·loca el vas de precipitats amb aigua destil·lada a la safata i s'introdueix la cànula per obtenir el zero instrumental amb l'aigua.
10. Es prem RESULTADOS i seguidament ALT+LECTURA. Un cop s'obté el zero, es prem només LECTURA.
11. Per mesurar les absorbàncies de les diferents mostres es col·loquen els diferents vasos de precipitats, comprovant que la cànula quedi ben submergida. Es torna a pitjar LECTURA per obtenir l'absorbància. És important que entre les diferents lectures es col·loqui el vas amb aigua destil·lada per tornar a fer el zero instrumental.
12. Un cop es llegeixen i s'apunten les mesures que dona l'instrument, s'apaga l'aparell en aquest ordre:
 - Apagar flama (botó vermell)
 - Apagar gasos (acetilè primer, aire després)
 - Apagar l'instrument, s'apagarà la llum verda
13. Amb els valors d'absorbància obtinguts de l'aparell es calcula la recta de regressió i mitjançant el mètode de l'addició estàndard es calcula la concentració (en ppm) del metall que hi ha a la mostra.

2.3.3 MÈTODE D'ADDICIÓ ESTÀNDARD

Com s'ha comentat anteriorment, el mètode emprat per calcular la concentració del metall a la mostra és el mètode de l'addició estàndard. El procediment experimental d'aquest mètode consisteix a preparar diferents dissolucions (en aquest cas de 25ml) amb el mateix volum de mostra, en aquest cas de 10ml, però diferents concentracions del patró del metall de la mostra.

A la Il·lustració 10 es poden observar els volums i material de preparació.



MOSTRA (mL):	10	10	10	10	10
PATRÓ C₀ (mL):	0	V₁	V₂	V₃	V₄

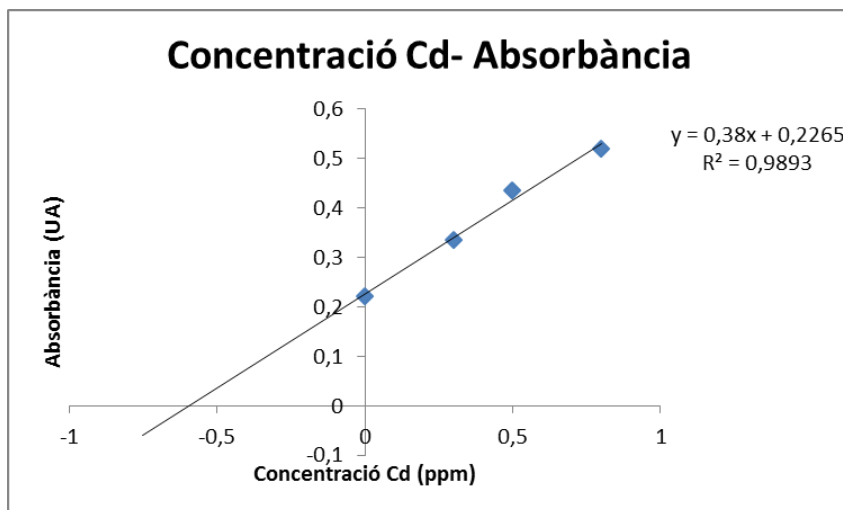
Il·lustració 10: origen a referència [15]

Un cop s'obtenen al laboratori els valors d'absorbància propis de cada dissolució, s'obté la recta de regressió "Absorbància – Concentració de metall afegida". S'extrapolen els resultats per tal d'obtenir la concentració del metall afegida quan l'absorbància és nul·la, aquest valor de concentració serà doncs la quantitat del metall que es troba a la mostra en un volum de 25 mL. Com el que es necessita saber és la concentració en els 10 mL de mostra i no en el volum total simplement caldrà multiplicar aquest valor aconseguit per volum total / volum mostra (25 mL/10 mL).

A continuació s'adjunta la taula i recta dels valors emprats en el joc de simulació:

Taula 1

Concentració Cd afegida(ppm)	Absorbància (UA)
0	0,222
0,3	0,336
0,5	0,436
0,8	0,52



Il·lustració 11

Com es pot observar a la Il·lustració 11, que mostra el gràfic de la recta, el valor obtingut en aquest cas seria 0.6 ppm de Cd en 25 mL de dissolució. Per tant, el valor final de Cd a la mostra de 10 mL seria de 1.5 ppm (**valor que hauran de donar els alumnes si han fet tot el procediment de la pràctica correctament**).

2.4 EL VIDEOJOC

Un videojoc és un joc electrònic creat generalment per l'entreteniment, mitjançant el qual una o més persones interactuen amb un dispositiu dotat d'imatges de vídeo.

Aquests poden classificar-se segons el dispositiu electrònic o plataforma que fan servir:

- Ordinador
- Màquina recreativa
- Videoconsola
- Dispositiu portàtil (mòbil, tauleta...)

A més, també poden classificar-se segons el gènere. Aquest es defineix com la categoria que s'utilitza per classificar i organitzar els videojocs per elements clau com poden ser la forma de joc, els controls utilitzats o bé els objectius d'aquest. Aquesta classificació es basa en coincidències entre els diferents jocs i un mateix videojoc pot pertànyer a dos gèneres diferents:

- Estratègia
- Lluita
- Acció o arcade
- Simulació

- Educatius
- [...]

Per últim, es poden classificar segons una sèrie de termes propis dels videojocs:

- Individual o multi – jugador: depenent si la interacció es individual o col·lectiva
- Ocasional: es juga en breus períodes de temps i té regles fàcils d'aprendre
- Seriós: són simulacions d'esdeveniments del món real o processos dissenyats amb el propòsit de resoldre un problema. El seu propòsit principal és formar o educar a l'usuari alhora que ofereix entreteniment.

Seguint aquesta pauta de classificació, el **joc objectiu** d'aquest projecte és:

- Joc d'ordinador: l'objectiu és aconseguir una pràctica online que es pugui accedir tant des de casa com des dels ordinadors de l'Escola.
- Joc de simulació: pretén simular un aparell de laboratori. La seva funció es aproximar-se al màxim a la realitat, fer sentir a l'usuari que es troba al laboratori químic, davant de l'aparell.
- Individual: encara que les pràctiques d'aquesta assignatura es fan en grup, el videojoc té un caràcter individual al igual que l'examen de pràctiques. L'usuari ha d'aprendre i interioritzar tots els coneixements i procediments de la pràctica sense ajuda.
- Seriós: com s'ha comentat anteriorment, l'objectiu principal és aprendre i aplicar els coneixements adquirits a l'assignatura d'*Experimentació en Química*. No obstant això, característiques com la limitació del temps o els premis tenen la finalitat de dinamitzar la pràctica.

3. ESPECIFICACIÓ

Un cop fixats els objectius i estudiats els antecedents s'han de definir les especificacions del joc, és a dir, el conjunt de requisits que ha de complir aquest per tal de crear un projecte coherent amb els objectius fixats.

Aquestes es dividiran en especificacions funcionals (relacionades amb el funcionament i comportament intern del programa) i les no funcionals (relacionades amb les característiques que poden limitar el programa).

FUNCIONALS

RF1. El joc haurà de permetre a l'usuari iniciar i acabar una partida en tot moment.

RF2. El programa haurà de ser suportat només per ordinador amb possibilitat d'extensió en un futur.

RF3. El programa haurà de ser compatible amb qualsevol sistema operatiu.

RF4. La partida del joc haurà de tenir associada una puntuació per augmentar la motivació de l'usuari.

NO FUNCIONALS

RNF1. El joc haurà de ser accessible per tot usuari de forma que es reconeguin tots els components de la pràctica fàcilment.

RNF2. El joc haurà de tenir associats tots els coneixements que aporta la pràctica de laboratori de química.

RNF3. El joc haurà de suposar un repte per l'usuari però no ha de ser impossible.

RNF4. El joc haurà de tenir temps de resposta ràpids donat que és una característica que denota qualitat en l'ús per part de l'usuari.

RNF5. L'aplicació haurà de proporcionar seguretat al sistema operatiu de forma que no comporti la instal·lació de softwares no desitjats.

RNF6. El llenguatge de programació haurà de ser fàcil d'analitzar i modificar de cara a possibles errors.

4. PLANIFICACIÓ

Com a projecte informàtic serà necessària l'elaboració d'un calendari per tal d'estructurar la feina i poder calcular els costos que suposarà.

4.1 TASQUES

Les tasques necessàries per la creació del joc són:

- Estudiar els antecedents i entendre els conceptes associats
- Definir tots els objectes que intervenen i les seves funcions
- Definir les especificacions
- Dibuixar els components necessaris
- Definir les eines necessàries pel disseny i la implementació
- Disseny del videojoc
- Implementació del videojoc
- Correcció d'errors i millora del joc
- Validació
- Documentació completa del joc

4.2 ESTIMACIÓ ECONÒMICA

Per tal de calcular els costos del projecte s'ha de tenir en compte els treballadors implicats, les hores treballades així com els softwares i materials utilitzats. El temps que s'ha utilitzat per calcular els costos són 6 mesos treballant a mitja jornada o bé les 540 hores que fan els crèdits del projecte. Més endavant es discutirà si aquest ha sigut el temps empleat o si hi ha hagut variacions respecte a la planificació.

Taula 2

TIPUS	CONCEPTE	COST
Personal	Sou enginyer programador	13.500,00 €
Material	Ordinador	125,00 €
	Material d'oficina	95,00 €
	Software	70,00 €
	Connexió internet	210,00 €
	Altres	150,00 €
TOTAL		14.150,00 €

A continuació es desglossen i es detallen les diferents despeses de la taula anterior (Taula 2) :

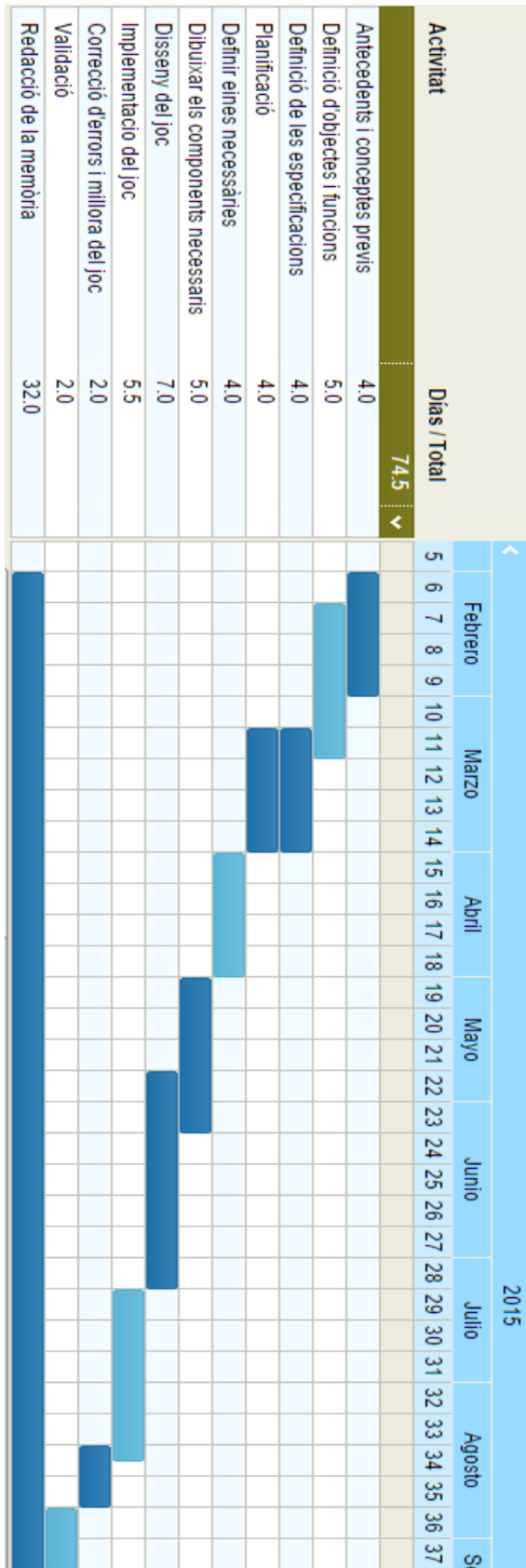
- Sou enginyer programador: $25 \text{ €/h} * 540\text{h} = 13500 \text{ €}$
- Ordinador: el cost de l'ordinador incloent el software instal·lat és de 1000 €, amb un temps de vida de 4 anys (48 mesos) i tenint en compte que s'amortitza només sis mesos: $1000\text{€} * (6/48) = 125 \text{ €}$
- Material d'oficina: Impressions de la documentació, llibretes, bolígrafs... S'estima un cost total de 95€
- Software: així com els programes principals emprats com Python o Pygame són gratis, s'ha de tenir en compte el paquet de Microsoft Office (amb el mateix temps d'amortització que l'ordinador): $560\text{€} * (4/48) = 70\text{€}$
- Connexió a internet: La quota de connexió emprada és de 35€/mes, per tant serà un cost de 210€ en els 6 mesos de treball.
- Altres: despeses com poden ser desplaçaments, electricitat, aigua, dietes...

4.3 DEDICACIÓ EN TEMPS

Tenint en compte les tasques comentades a l'apartat 3.1 s'ha realitzat un diagrama de Gantt per tal d'intentar fer una bona distribució i organització de la feina a realitzar. S'ha calculat un temps de treball de mitja jornada durant 6 mesos sense comptabilitzar els dies festius. Cal esmentar que aquest diagrama es realitza abans de començar el projecte pel que pot variar durant la realització d'aquest.

Els temps, en setmanes, estimats per a cada tasca són:

- Estudiar els antecedents i entendre els conceptes associats : 4
- Definir tots els objectes que intervenen i les seves funcions: 5
- Definir les especificacions: 4
- Dibuir els components necessaris: 5
- Definir les eines necessàries pel disseny i la implementació: 4
- Disseny del videojoc: 7
- Implementació del videojoc: 5.5
- Correcció d'errors i millora del joc: 2
- Validació: 2
- Documentació completa del joc: 32










Il·lustració 12: origen a referència [5]

5. DISSENY DEL JOC

5.1 OBJECTES

En un joc de simulació trobem diferents tipologies d'objectes : interactius, receptors, decoratius... En aquest apartat es definiran i es classificaran els diferents objectes que formen part d'aquest joc en forma de taula:

Taula 3

NOM	GRÀFIC	TIPOLOGIA	ACCIONS
Vas de precipitats		Interactiu	Agafar/Deixar
Bombona d'aire		Receptiu	Obrir / Tancar
Bombona d'acetilè		Receptiu	Obrir / Tancar
Pantalla		Interactiu	Pressionar tecles
Làmpades		Receptiu	Encendre / Apagar
Flama		Receptiu	Encendre / Apagar
Comporta làmpades		Receptiu	Obrir / Tancar

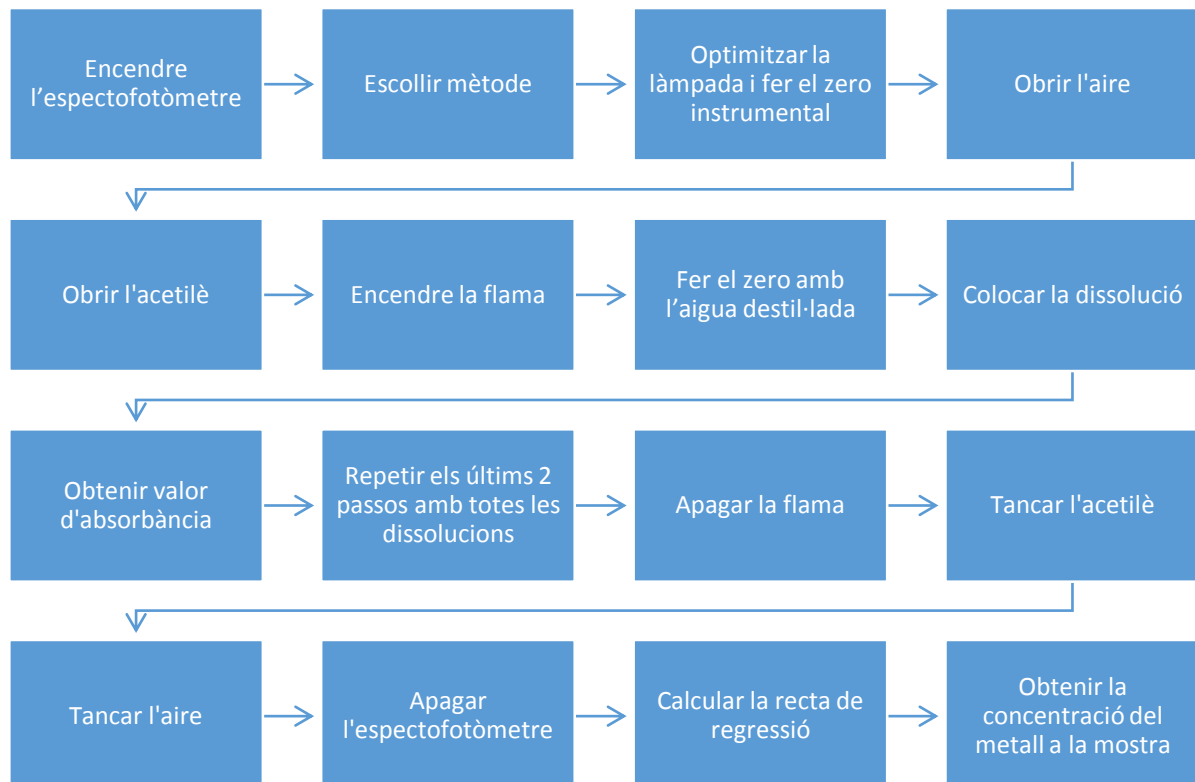
Extractor		Decoratiu	-
Espectrofotòmetre		Interactiu	Accions components
Taula		Decoratiu	-

5.2 DIAGRAMA D'ESTATS

El diagrama d'estats d'un joc permet definir els diferents estats en els que es pot trobar l'usuari durant la partida, així com les accions que es poden realitzar per tal de passar d'un estat a un altre. A més, aquest diagrama mostra el camí necessari a seguir per tal d'arribar a la final del joc (conjunt d'accions que s'han de realitzar en l'ordre correcte per tal d'acabar el joc obtenint la màxima puntuació).

Un diagrama d'estats es representa amb un conjunt de nodes i arcs o fletxes, on els primers representen els diferents estats i els segons les diferents accions.

Primerament es representa el diagrama de fases que mostra el camí per arribar a la meta del joc (esquema que resumeix la pràctica sense incloure accions o estats):



Il·lustració 13

A continuació es defineixen, a la Taula 4, els diferents estats pels quals s'ha de passar en ordre per arribar a la meta del joc. Aquests es defineixen en funció dels estats dels diferents elements i dels components de l'espectrofotòmetre:

Taula 4

ESTAT	MÀQUINA	FLAMA	AIRE	ACETILÈ	MOSTRA.A*	MOSTRA.B**	MOSTRA.C***	PANTALL A
0	OFF	OFF	OFF	OFF	-			-
1	ON	OFF	OFF	OFF	-			0
2	ON	OFF	OFF	OFF	-			1
3	ON	OFF	OFF	OFF	-			2
4	ON	OFF	OFF	OFF	-			3
5	ON	OFF	OFF	OFF	-			4
6	ON	OFF	OFF	OFF	-			5
7	ON	OFF	OFF	OFF	-			6
8	ON	OFF	ON	OFF	-			6
9	ON	OFF	ON	ON	-			6
10	ON	ON	ON	ON	-			6
11	ON	ON	ON	ON	Aigua	-		6
12	ON	ON	ON	ON	-	Aigua		6
13	ON	ON	ON	ON	-	Aigua		7
14	ON	ON	ON	ON	-			8
15	ON	ON	ON	ON	Aigua			8
16	ON	ON	ON	ON	-		Aigua	8
17	ON	ON	ON	ON	1			8
18	ON	ON	ON	ON	-	1		8
19	ON	ON	ON	ON	-	1		9
20	ON	ON	ON	ON	1			9
21	ON	ON	ON	ON	-		1	9
22	ON	ON	ON	ON	2			9
23	ON	ON	ON	ON	-	2		9
24	ON	ON	ON	ON	-	2		10
25	ON	ON	ON	ON	2			10
26	ON	ON	ON	ON	-		2	10
27	ON	ON	ON	ON	3			10
28	ON	ON	ON	ON	-	3		10
29	ON	ON	ON	ON	-	3		11
30	ON	ON	ON	ON	3			11
31	ON	ON	ON	ON	-		3	11
32	ON	ON	ON	ON	4			11
33	ON	ON	ON	ON	-	4		11
34	ON	ON	ON	ON	-	4		12

35	ON	ON	ON	ON	4		12
36	ON	ON	ON	ON	-	4	12
37	ON	OFF	ON	ON	-		12
38	ON	OFF	ON	OFF	-		12
39	ON	OFF	OFF	OFF	-		12

**Estat de la mostra quan és agafada*

***Estat de la mostra quan està a la safata de l'aparell*

****Estat de la mostra quan es torna a deixar fóra de la safata*

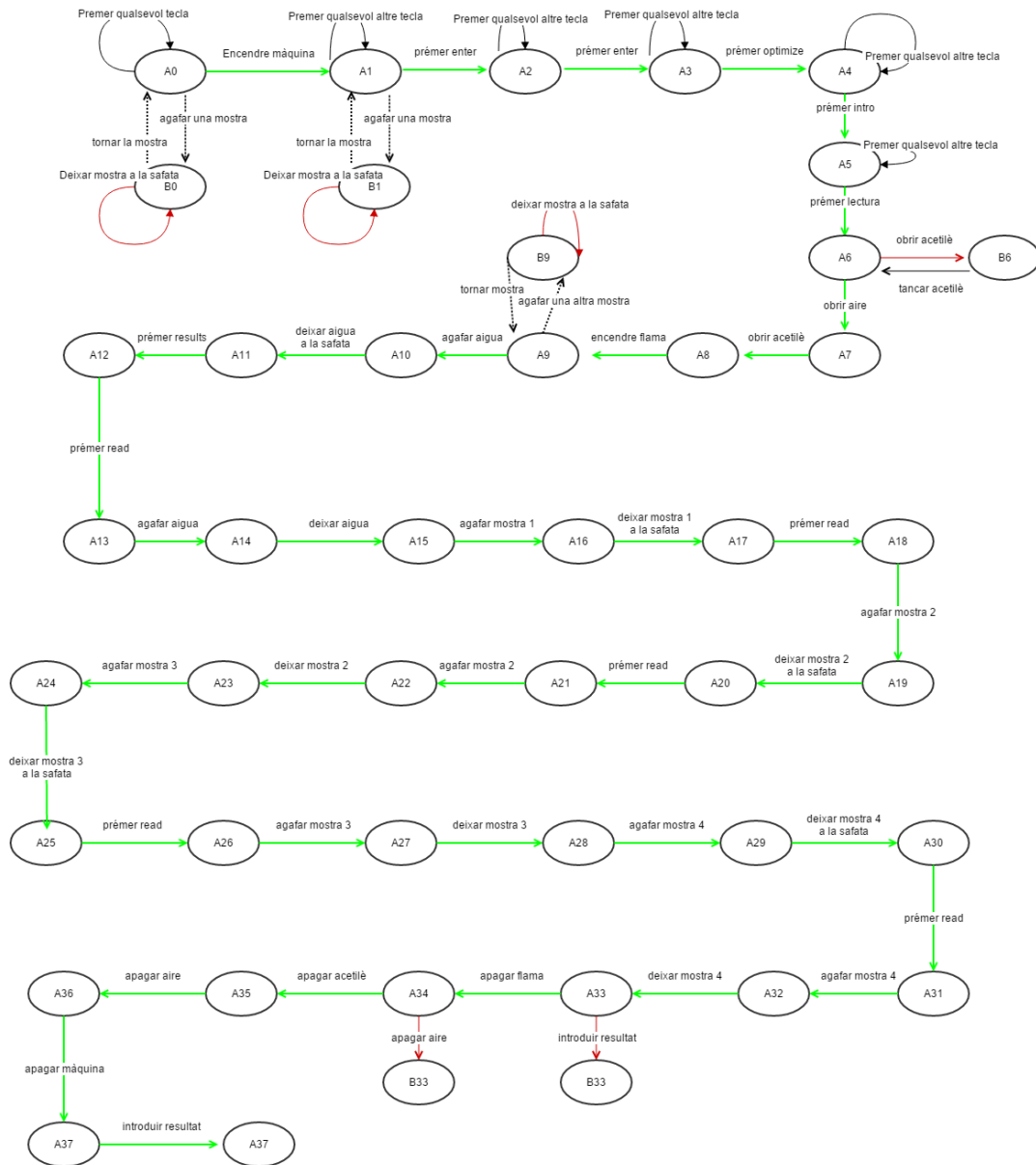
**** Les pantalles es mostren a l'annex C*

Aquesta taula no inclou cap estat erroni ni accepta cap acció equivocada, és en el diagrama d'accions i estats on es contemplen els possibles errors a més del camí correcte (Il·lustració 14).

En el següent diagrama és mostren els estats correctes (nomenclatura A + número) i les accions correctes que fan arribar a cada estat (en color verd). Els estats secundaris, son aquells anomenats amb la lletra B. Les accions en color negre descriuen aquelles accions no principals però que no suposen perdre una vida, encara que si la pèrdua de puntuació al desviar-se del camí idoni. Les accions marcades en vermell són aquelles que suposen la pèrdua d'una vida a més de la pèrdua de puntuació.

Cal esmentar que els estats A0-A8 comparteixen les mateixes possibles accions secundaries o bé errònies, és a dir, prémer qualsevol altre tecla o agafar qualsevol mostra són accions que no comporten pèrdua de vida, en canvi deixar una de les mostres a la safata si que ho fa. Per limitacions d'espai no s'ha pogut representar.

Així mateix, com s'explicarà en apartats posteriors, les diferents pantalles admeten només una sèrie d'accions activades per tecles. Les tecles que es poden clicar en un estat o pantalla són aquelles que no canvien el transcurs correcte de la pràctica i que, per tant, no activaran accions errònies o irreversibles. A cada estat correcte li correspondria la mateixa fletxa amb l'acció "prémer qualsevol altre tecla", la qual ha sigut representada només en els primers estats per falta d'espai.



Il·lustració 14

5.3 PUNTUACIÓ I VIDES

Per la definició del joc és necessària la definició de la puntuació que es pot obtenir així com la quantitat de vides disponibles i les accions on es poden perdre.

Un cop estudiada la pràctica i el seu procediment, s'estableix que les accions que perden vides són:

- Obrir l'acetilè abans que l'aire: 1 vida
- Col·locar una mostra a l'aparell abans de col·locar l'aigua i fer el zero instrumental: 1 vida
- Acabar la partida sense realitzar el procediment de tancar: apagar la flama, tancar l'acetilè, tancar l'aire i apagar l'instrument: 1 vida
- Donar malament el resultat final (concentració de Cd a la mostra: 1,5 ppm): 1 vida
- Donar com a resultat final 0,6 ppm de Cd. Aquest resultat es calcula a partir de la recta de regressió sobre un volum de 10 ml de mostra i no dels 25 ml de mostra que es tenen (explicat a l'apartat 2.3.3 *Addició Estàndard*).

Per els resultats finals s'accepta un error decimal de ± 0.1 .

Si el jugador es queda sense vides abans d'acabar la partida, aquesta s'acaba i apareix directament la pantalla final, indicant la puntuació obtinguda.

Pel que fa a la puntuació, es defineix un màxim de 5000 punts si s'executa tal i com explica el manual de l'usuari.

Per això es troben definits tots els estats pels quals es imprescindible que passi la partida per tal d'arribar a la màxima puntuació. Aquests estats es defineixen com el conjunt dels estats en els que es troben tots els elements del joc. Si s'aconsegueix arribar a un d'aquests es sumaran 100 punts. Per altra banda, si es passa per estats innecessaris del joc, encara que no siguin irreversibles, el jugador perdrà 10 punts.

Per últim, la puntuació del resultat que s'introdueixi també variarà. Com s'ha comentat en apartats anteriors, el resultat correcte és de 1.5 ppm de Cd a la solució de 25 mL. Si no s'ha fet la conversió dels 10 mL de mostra als 25 mL totals de solució, el resultat serà de 0.6 ppm aproximadament. Cal esmentar que s'ha suprimit de la pràctica el càlcul de la recta de regressió, pel que el mateix programa la mostra al jugador de forma que pugui calcular la solució si ha fet bé el procediment. La puntuació és:

- 1.5 ± 0.1 ppm: 1000 punts

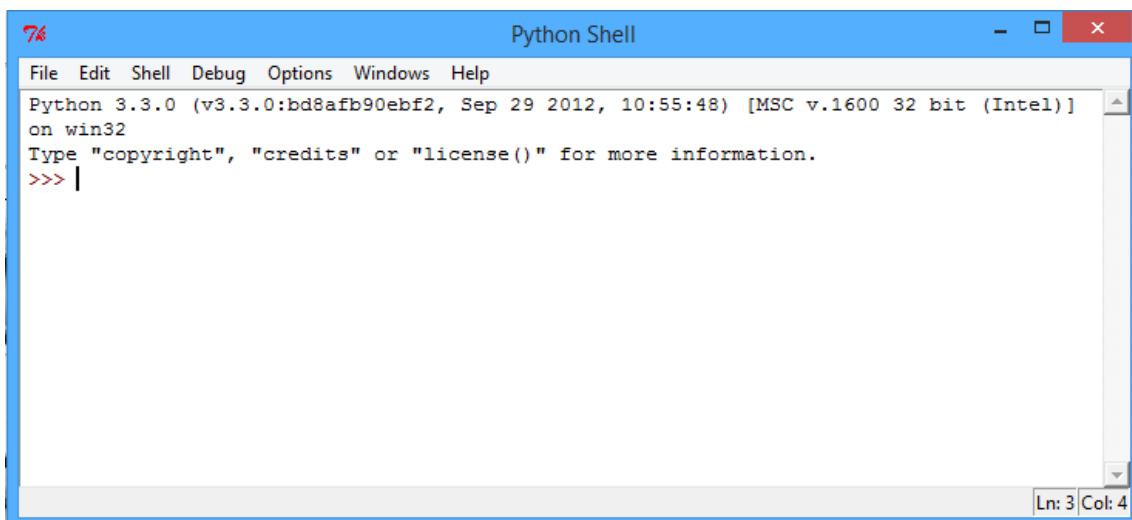
- 0.6 ± 0.1 ppm : 500 punts
- Un altre valor: 100 punts

Depenent de la puntuació final obtinguda, el joc mostrarà un missatge o un altre a la pantalla final.

6. IMPLEMENTACIÓ DEL JOC

6.1 ESCRIPTURA DEL CODI DE PROGRAMACIÓ

Tot el projecte s'ha desenvolupat des del sistema operatiu Windows8. S'utilitza un IDLE (Entorn de desenvolupament integrat exclusivament per Python). L'IDLE bàsicament és un editor de textos (on el programador escriu el codi) que destaca la sintaxis de programació. De tal manera que quan detecta que s'està fent servir una estructura determinada canvia el color de les paraules o aplica un espai o tabulador automàticament. Així es fa molt més intuïtiu el llenguatge de programació. IDLE té un depurador integrat, és a dir que un cop s'ha escrit el codi a l'editor de textos, des d'allà mateix es pot executar. Llavors, a la interfície principal del programa anomenada Python Shell, s'indica si ha trobat algun error, de quin tipus és i on l'ha detectat. Cada cop que s'executa un codi es fa en un procés separat i això s'indica a la Python Shell.



Il·lustració 15

També cal destacar que IDLE és multi-finestra, que vol dir que no cal tenir el codi escrit en un mateix arxiu, sinó que es pot organitzar per mòduls. Es tracta de tenir un arxiu principal (que serà des d'on s'executarà el codi) i organitzar la resta de la manera que més convingui en diferents mòduls. Així, en el cas de l'aplicació informàtica que ens ocupa, es té un arxiu principal (correspon a l'inici de sessió), un arxiu secundari (correspon a la pantalla de passos) i finalment un mòdul per a cada pas.

Resumint, el procés emprat per escriure el codi de l'aplicació informàtica ha sigut:

- 1- S'escriu el codi a l'editor de textos de l'IDLE.
- 2- S'executa. Si hi ha errors es corregeixen i es torna a executar.
- 3- Es comprova que el resultat sigui l'esperat.

6.2 MÒDULS DE PYGAME

Com s'ha comentat a l'*apartat 1.3*, Pygame és una llibreria de Python que permet la implementació de jocs a partir d'un conjunt de mòduls, que tracten tant gràfics, com figures, esdeveniments de tot tipus, elements multimèdia, etc.

És la llibreria perfecta per al desenvolupament de jocs de poca envergadura. És capaç d'executar-los a 30 fps ("*frames per second*"), o sigui, que no és molt potent, però és molt útil en la implementació de petits jocs, així que s'adequa perfectament a la necessitat d'aquest projecte.

La llibreria Pygame esta formada per diferents mòduls que identifiquen els diferents elements que pot tenir un joc, com són els sons, imatges, superfícies, esdeveniments, fonts... i les seves operacions pertinents. A continuació s'expliquen els que han sigut més emprats per la creació del joc.

6.2.1 PYGAME

És el mòdul pare, el que engloba tots els altres, que seran fills d'aquest, i que conté les funcions principals de la llibreria, com poden ser la d'inicialitzar tots els submòduls importants (pygame.init) i la de desactivar-los (pygame.quit). També conté les variables globals de la llibreria, ubicades a pygame.locals.

La majoria dels jocs tenen aquesta estructura: en primer lloc es fan les inicialitzacions pertinents (creació de la finestra gràfica, dels personatges si en hi ha, inicialització de la puntuació, etc.) i després hi ha una composició iterativa principal anomenada game-loop de la qual se'n surt quan s'acaba el joc, bé perquè l'usuari ha guanyat la partida, bé perquè l'ha perduda. En el game-loop o també anomenat main-loop el codi realitza tres coses:

- Gestiona els esdeveniments
- Actualitza l'estat del joc
- Dibuixa les pantalles dels diferents estats

6.2.2 DISPLAY

És el mòdul que ens permet controlar com es mostra la finestra del joc dins la pantalla de l'ordinador. Pygame permet visualitzar les aplicacions en una finestra o en pantalla completa. Aquest mòdul disposa de diferents operacions per tal de poder tractar la visualització de la superfície que ocupa l'aplicació, com pot ser inicialitzar-la, actualitzar-la o obtenir informació de les seves mides.

6.2.3 EVENT

Aquest mòdul és el que tracta els diferents esdeveniments que tindrà l'aplicació o joc creat, tant els de teclat, com els del ratolí, joystick o els botons de la finestra (tancar, maximitzar o minimitzar). En Pygame, tots el que succeeix són missatges d'esdeveniments que es van ficant en cua, i aquests són gestionats amb les operacions d'aquest mòdul. Per tant, cada cop que es prem una tecla del teclat, un botó del ratolí o bé un botó de la pantalla aquest queda registrat a la cua d'esdeveniments i podrà ser tractat de manera independent gràcies al mòdul `pygame.event`. Per exemple, la funció "`pygame.event.get`" agafa l'últim esdeveniment de la cua, és a dir, l'últim que ha quedat enregistrat.

6.2.4 SURFACE

Mòdul utilitzat per crear objectes que són imatges rectangulars en 2D. Surface fixa una resolució i un format dels diferents píxels que la contenen. Per crear una superfície només cal assignar-li unes mides, d'aquesta manera es crea una superfície de color negre per defecte, amb les mides assignades.

Aquest mòdul té moltíssimes funcions com poden ser obtenir informació de la superfície, copiar-la, omplir-la d'un color, etc.

6.2.5 RECT

Pygame utilitza aquest mòdul per emmagatzemar i manipular àrees rectangulars. Existeixen diversos constructors diferents per a l'objecte Rect, pot crear-se amb una combinació de valors d'esquerra, dreta, ample i alt o també pot ser creat des d'altres objectes de Python que siguin un objecte `pygame.Rect`. Aquest mòdul serà de gran ajuda alhora de determinar si dos objectes estan col·lisionant, és a dir, si el rectangle d'un està a sobre el rectangle de l'altre.

6.2.6 SPRITE

Dins d'aquest mòdul es defineix la classe Sprite. Aquesta crea objectes com a imatges bidimensionals que formen part d'una escena gràfica encara més gran. És molt útil i important treballar amb sprites a un videojoc ja que facilita el tractament dels objectes i sobretot, la interacció entre aquests. Hi ha moltes funcions dels sprites que permeten agrupar-los per tal de poder dibuixar-los tots alhora o bé d'altres que detecten si dos sprites estan col·lisionant. La classe més utilitzada dins del mòdul és `pygame.sprite.RenderPlain`, la qual permet gestionar i agrupar múltiples objectes de la classe Sprite.

6.3 MÒDULS PROPIS

Per la implementació del joc s'han creat diferents mòduls de Python per tal d'organitzar i jerarquitzar el codi del programa:

- `practica.py` : mòdul principal on es troba el main-loop del joc. Des d'aquí es tracta i es gestiona la cua d'esdeveniments i es defineixen i dibuixen les principals pantalles o finestres del joc amb tots els seus components. Es troben les quatre funcions principals que defineixen i gestionen les quatre etapes del joc (`introducció()`, `main()`, `final()` i `puntuació()`), així com les diferents funcions que gestionen els esdeveniments a cada una d'aquestes.
- `Global.py` : mòdul on simplement es defineixen les variables globals, és a dir, aquelles a les quals es pot accedir des de qualsevol dels diferents mòduls. Aquestes no formen part de cap classe, atribut o funció. Un bon exemple són els colors RGB (`red`, `green`, `blue`), definits com una combinació dels tres. Aquests estan definits a `global.py` de manera que cridant-los des de qualsevol mòdul es poden utilitzar.
- `escena.py` : es crea la classe amb el mateix nom on es defineix una escena com un conjunt de components i amb unes mides determinades. Cada pantalla creada durant la partida serà un objecte de la classe `escena`, cadascuna amb les seves pròpies característiques i definida en el seu propi mòdul:
 - `intro.py` : on es troba la funció que defineix la pantalla d'inici del joc. Es crea el fons, les imatges i el text que apareixen només començar el joc.
 - `principal.py` : on es troba la funció que defineix la pantalla principal del joc, on es troben tots els components necessaris per la realització de la simulació, definits sota la classe objecte que ara s'explicarà.
 - `fin.py` : on es troba la funció que defineix les dues últimes pantalles. Primer es defineix l'escena on s'introdueix el resultat obtingut de realitzar la pràctica i després l'escena final, la qual mostra el resultat de l'usuari al acabar la partida.
- `component.py` : mòdul on es troben definides dues classes:
 - `Component` : els objectes d'aquesta classe es caracteritzen per un nom, una llista d'estats, l'estat actual en el qual es troben i una llista de les seves parts o components.
 - `Estat` : on l'estat d'un objecte es caracteritza pel seu nom, la capsa mínima contenidora que ocupa en l'escena i la imatge associada. Conte també informació de l'acció que es pot fer sobre l'objecte en aquest estat (per exemple `encendre`).

Tots els elements de l'escena principal del joc seran objectes de la classe `Component` i estaran caracteritzats per diferents estats propis d'aquesta segona classe.

- `objecte.py` : mòdul on es defineix la classe amb el mateix nom. La classe `Objecte` hereta de la classe `Component`, és a dir, agafa els seus atributs i funcions i pot afegir-ne d'altres. Per tant, tindrà els atributs `nom`, `llista d'estats`, `estat actual` i `llista de parts` i a més, defineix totes les accions que pot rebre el component. Aquestes accions estan definides simplement alterant l'estat actual del component (la funció `obrir` fa que el component passi de l'estat 0 a l'estat 1).
- `textObjecte.py` : aquí es defineix la mateixa classe `Objecte` però destinada als objectes en forma de text. Aquests tindran atributs propis d'un text: `font`, `text`, `color`...
- `componentGrafic.py` : un cop creats els components, es defineixen de forma que s'adaptin a les mides de la pantalla actual re-escalant les seves mides amb les mides de l'escena principal. La classe `ComponentGrafic` agafa informació dels seus atributs com a components: `capsa de l'estat actual` i `imatge de l'estat actual` per poder dibuixar-los a la pantalla més endavant.
- `componentGraficText.py` : igual que la classe anterior, amb aquesta classe adaptem els components de text a la pantalla actual de forma que sigui més fàcil la seva representació.
- `escenaGrafica.py` : es defineix la classe `EscenaGrafica` amb els atributs `mida`, `àrea`, `color`, `components`, `escala`.. de forma que crea una superfície (`Pygame.display`) i converteix tots els components d'aquesta escena en objectes de la classe `ComponentGrafic` explicada anteriorment. A més, agrupa tots els components sota la classe `Sprite`, de manera que es puguin gestionar tots els components de forma conjunta.
- `estats.py`: en aquest mòdul es defineix la funció que gestiona els diferents estats del joc per on passa l'usuari a la pantalla principal. Es defineix la puntuació que s'obté per cada acció executada durant la pràctica, a més de les vides que es poden perdre.
- `input.py`: gestiona l'entrada de text que es demana a la tercera pantalla, on s'ha d'introduir el valor de la concentració del Cadmi a la mostra. Com aquest no és tractat com un objecte ni component, es tracta de manera independent.

6.4 ESTATS I ACCIONS DELS COMPONENTS

Com s'ha comentat a l'apartat anterior, al mòdul `component.py` és defineix la classe que porta el mateix nom. Aquesta defineix cada component d'una escena pel seu nom, la seva llista d'estats i l'índex del seu estat actual (com a mínim tenen un estat). Tots els elements de l'escena seran objectes de la classe `Component`.

Els estats dels diferents components, són alhora, objectes de la classe `Estat`. Aquests tenen atributs com el nom, la capsa mínima contenidora que ocupa en l'escena, la imatge associada a aquell estat i les possibles accions que es pot fer sobre l'objecte en aquest estat.

En el mòdul `principal.py` és on es defineix l'escena principal del joc, amb tots els components que formen part d'aquesta i els seus estats característics.

Per canviar d'un estat a un altre es necessiten accions, que en ser efectuades durant la partida produeixen un canvi d'estat en el joc, o en aquest cas, en els components.

En aquest cas, totes les accions és realitzen mitjançant el ratolí. Un component canviarà d'estat quan al clicar amb el ratolí es produeixi una col·lisió entre aquest i el seu atribut de capsa mínima. S'ha de tenir en compte que la capsa està representada per un tuple de quatre elements que indiquen la posició on es troba i les mides del rectangle.

La classe `Objecte` és la que gestiona els canvis d'estat. Heretant els atributs de la classe `Component`, defineix també totes les possibles funcions que es poden realitzar sobre els diferents components del joc. Així per exemple, la funció `obrir` canvia de l'estat actual zero a l'estat actual 1 mentre la funció `tancar` ho fa al revés. Els estats es tractaran per l'índex que tenen a la llista d'estats del component, el primer en definir-se serà l'estat zero.



A continuació es definiran els diferents estats de tots els components de l'escena principal, definint els seus atributs: nom, imatge i accions. La capsa contenidora no es definirà ja que depèn simplement de la mida que es vulgui donar a la imatge d'aquell estat i de la seva posició a l'escena.

Hi ha components, com poden ser la taula o l'extractor, que no s'esmentaran ja que tenen només un estat i no interactuen. Quan és així, l'objecte té una sola imatge i no es realitza cap acció sobre aquest.

6.4.1 AIRE I ACETILÈ

Les bombones d'aire i acetilè que serveixen per encendre la flama tenen dos estats: tancat i obert.

Taula 5

Nom estat	Imatge	Accions
Tancat		Obrir
Obert		Tancar

Com es pot observar a la Taula 5, la diferència gràfica entre ambdós estats és la clau que obra i tanca la bombona. A més, per remarcar la diferència, quan la bombona s'obre, el color d'aquesta s'intensifica.

6.4.2 ESPECTROFOTÒMETRE

L'espectrofotòmetre o component principal del joc és una mica més complex. Format per diferents elements que interactuen, el seus estats es definiran a partir dels estats d'aquests elements. Així l'aparell es pot encendre o apagar mitjançant un botó, es pot obrir i tancar la porta de les làmpades i també es pot encendre i apagar la flama.




Cal esmentar que si l'aparell està apagat, la flama no es podrà encendre de cap manera.




Quan aquest s'encén o s'apaga, apareix i desapareix una llum verda que es troba al costat del botó que realitza les accions. En canvi, quan s'encén la flama, aquesta apareix per darrera de la porta de seguretat situada al mig de l'aparell.

Per realitzar l'acció d'encendre o apagar, simplement es detecta si el ratolí, a més d'estar clicant sobre l'espectrofotòmetre, ho està fent sobre el botó. Si es així, al processar l'esdeveniment només es realitzaran sobre l'aparell les accions d'obrir o tancar.

El mateix passa amb la flama o la porta de les làmpades. Al clicar sobre el botó que encén la flama, només es realitzarà l'acció d'encendre la flama i si es fa sobre la porta, aquesta només s'obrirà o es tancarà.

Taula 6

Nom estat	Imatge	Accions
'Apagada i tancada'		Encendre, obrir porta
'Encesa i tancada'		Apagar, obrir porta, encendre flama
'Apagada i oberta'		Encendre, tancar porta

'Encesa i oberta'		Apagar, tancar porta, encendre flama
'Encesa, tancada, flama'		Apagar, obrir porta, apagar flama
'Encesa, oberta, flama'		Apagar, tancar porta, apagar flama

6.4.3 PANTALLA

Aquest és el component amb més estats definits. A més de tenir múltiples tecles que realitzen accions diferents, hi ha una gran quantitat de possibles pantalles que apareixen al joc.

Les tecles que intervenen a la partida i que formen part del component són:

- fletxes up, down : serveixen per posicionar-se a la pantalla
- enter: serveix per seleccionar
- optimize: serveix per optimitzar les làmpades
- read: es fa servir per realitzar les lectures d'absorbància o bé per fer el zero instrumental de l'aparell
- results: introdueix la pantalla on es mostren els resultats d'absorbància

Depenent de la tecla que es premi es realitzarà una acció o un altre. Aquestes accions són simplement passar d'una pantalla a una altra. A la vida real es podrien realitzar moltes més accions o intervindrien moltes altres tecles, però només s'han considerat les que són importants per l'objectiu de la pràctica.

En aquest cas, les imatges s'adjuntaran a l'annex C: PANTALLES.

Taula 7

Nom estat	Accions
p0	Up, down, intro
p2	Up, down, intro
p3	Up, down, intro
p4	Up, down, intro
p5	Up, down, intro
p6	Up, down, intro
p7	Up, down, intro, optimitzar
p8	Up, down, intro, optimitzar
p9	Up, down, intro, optimitzar
p10	Up, down, intro, optimitzar
p11	Up, down, intro, optimitzar
p12	Up, down, intro, optimitzar
p13	Up, down, intro
p14	Up, down, intro
p15	Up, down, read
p16	Up, down, read
p17	Up, down, results
p18	Up, down, results
p19	Up, down, intro, read
p20	Up, down, intro, read
p21	Up, down, intro, read
p22	Up, down, intro, read
p23	Up, down, intro, read
p24	Up, down, intro

6.4.4 VASOS DE PRECIPITATS

Aquests cinc components (les quatre mostres i l'aigua destil·lada) estan definits de manera molt similar. De fet, els atributs que varien són la capsa mínima (es troben a diferents posicions de l'escena) i la imatge (les quatre solucions es diferencien per la concentració de cadmi representada).

En aquest cas, els vasos tenen dos possibles estats: 'estàtic' o 'en moviment'. L'objectiu d'aquests components és portar-los a la màquina per tal de mesurar la seva absorbància.

Si es pitja sobre un d'aquests, el ratolí es mou arrossegant el vas de precipitats fins que es torna a pitjar i es deixa allà on es troba el ratolí.

Com la imatge dels dos estats no canvia i té molta més importància la capsa que indica la posició de l'element, els següents estats es definiran amb el nom, la capsa i les possibles accions.

Taula 8

Nom estat	Capsa	Accions
'estàtic'	(2.9, 3.05, 0.35, 0.35)	Up, down, intro
'en moviment'	(x, y, 0.35, 0.35) On x,y = x,y = pygame.mouse.get_pos()	Up, down, intro

6.5 RESTRICCIONS DEL JOC

Com a la vida real, hi ha algunes accions que no es poden realitzar sense realitzar-ne unes altres. Encara que els components no estiguin connectats entre sí a la seva definició amb el codi, hi ha alguns que inevitablement estan connectats i s'ha de reflectir. Al definir alguns elements com elements independents (per exemple la pantalla i l'espectrofotòmetre) s'han de tenir en compte algunes restriccions en la implementació:

- No es pot encendre la flama si l'espectrofotòmetre està apagat
- No es pot encendre la flama si l'aire o l'acetilè estan tancats
- No es pot interactuar amb la pantalla si l'espectrofotòmetre està apagat
- No es poden efectuar les lectures d'absorbància si no hi ha una mostra a l'aparell. A més, les lectures s'han de fer en l'ordre establert: primer l'aigua destil·lada i després les mostres en ordre de menys concentrada a més concentrada.

7. VALIDACIÓ

Les proves d'usuari, és una de les fases prèvies obligatòries abans de la publicació oficial de qualsevol videojoc professional.

A partir de l'experiència d'aquests jugadors, es pot obtenir un feedback sobre els errors del joc no detectats prèviament o les millores que es poden realitzar a última hora.

En aquest projecte, s'ha considerat oportú simular la fase de proves d'usuari en una menor escala, per a detectar possibles errors no previstos i a més a més obtenir una valoració del joc per parts dels jugadors. Per a obtenir la valoració, cadascun dels jugadors ha completat el qüestionari de l'apartat E de l'apèndix amb preguntes estructurades.

En el qüestionari es pregunta sobre la impressió general del joc i sobre altres aspectes com: l'aspecte visual, l'entreteniment, la complexitat del joc...

La mostra, formada per 10 enquestats, conté persones d'entre 20 i 45 anys. Dins d'aquest grup, només la meitat comptava amb coneixements de química relacionats amb la pràctica i quatre eren aficionats dels videojocs.

Els resultats de les enquestes són els següents (les preguntes amb resposta numèrica es basen en una escala on el número 1 és el més negatiu i el 5, el més positiu. Es fa una mitja per obtenir la resposta final):

- Impressió general del joc: Molt bona (80%)
- Facilitat per entendre el joc: 3.9
- Aspecte visual del joc: 4.2
- Oportunitat per aprendre els conceptes associats: 4.5
- Entreteniment: 4
- Motivació per tornar a jugar: 3
- Serveix com a formació prèvia a la pràctica: Sí (90%)
- Aspecte que es podria millorar: llibertat de moviments i/o d'opcions (60%)

A partir d'aquests resultats es treuen les següents conclusions:

- Les impressions sobre el joc són bastant bones per part de l'usuari i a la gran majoria li serveix com a formació prèvia de la pràctica (objectiu principal del projecte).
- L'aspecte més valorat és la oportunitat d'assimilar o aprendre els conceptes i el que menys agrada és la motivació que aporta el joc per tornar a jugar una partida.

- Aquest últim aspecte segurament és degut a la falta de llibertat de moviments o d'opcions que l'usuari considera que es pot millorar.

8. CONCLUSIONS

L'objectiu principal del projecte era la creació d'un joc seriós mitjançant la programació. Com a joc seriós aquest havia de complir dos requisits: educar i divertir.

La primera part de l'objectiu es pot dir que s'ha complert ja que mitjançant el llenguatge de programació Python i el conjunt de mòduls de Pygame s'ha creat un videojoc de simulació per recrear una pràctica de Química al laboratori. La segona part també s'ha complert segons la valoració de l'usuari, que afirma trobar-lo entretingut a més d'instructiu.

En quant a la planificació inicial del temps pot dir-se que s'ha vist significativament modificada. La majoria de les tasques han ocupat més temps del previst o s'ha afegit alguna tasca que no estava prevista en el primer moment. Cal esmentar que la gran majoria d'aquestes tasques eren realitzades per primer cop i la duració d'aquestes pot ser molt variable. La fase d'implementació ha sigut la principal causa del des ajustament final, aquesta pot arribar a ser molt enutjosa ja que un videojoc pot abastar moltes àrees de desenvolupament.

També, gràcies a la realització del projecte, s'han pogut aplicar els coneixements tant de química com de programació adquirits durant els cursos del Grau. A més, s'ha pogut desenvolupar un grau de coneixement del llenguatge de programació Python al qual no s'hauria arribat amb els coneixements bàsics de les assignatures d'informàtica.

Aquest projecte es deixa obert de cara a possibles ampliacions i millores com l'augment de llibertat de moviments o d'opcions que remarca l'usuari enquestat.

Cal esmentar que els projectes de programació solen ser bastant complexos ja que encara que et trobis en una fase avançada pots trobar-te amb obstacles que et fan retrocedir, modificar i tornar a començar. A part d'això, la programació es una tasca molt dinàmica que pot suposar un repte molt estimulant.

BIBLIOGRAFIA

- [1] *Agilent | Chemical Analysis, Life Sciences, and Diagnostics* [en línia]. Disponible a: <http://www.agilent.com/> [consulta: febrer-març 2015]
- [2] Daniel Fuentes. *Python Mania* [en línia]. Disponible a: <https://pythonmania.wordpress.com/> [consulta: 20 de maig 2015]
- [3] Downey B, A., Elkner, J., Meyers, C. (2009). *How to Think Like a Computer Scientist: Learning with Python [versió electronica]*. SoHo Books
- [4] *Espectrofotometria d'Absorció atòmica* [en línia] . Disponible a: <http://upcommons.upc.edu/ocw/diposit/material/33762/46337.swf> [consulta: febrer-abril 2015]
- [5] Eric Chiang. *GLIFFY.COM* [en línia]. Disponible a: <https://www.gliffy.com/>. [consulta: juliol 2015]
- [6] *GIMP* [en línia]. Disponible a: <http://www.gimp.org/es/> [consulta: abril-setembre 2015]
- [7] Harris, Daniel C.(2016). *Análisis químico cuantitativo*. 3a ed. Barcelona: Reverté. ISBN 8429172246
- [8] Informàtica [en línia]. Disponible a <http://gie.lsi.upc.edu/mediawiki/> [consulta febrer-març 2015]
- [9] Javier Aller, A. (1987). *Espectroscopia de absorción atómica analítica*. León: Universidad. Servicio de Publicaciones, D.L
- [10] Lie Hetland, M. (2005). *Beginning Python: from novice to professional*. New York: Apress, cop
- [11] *Pygame* [En línia]. Disponible a: <http://www.pygame.org> [consulta: febrer-juliol 2015]
- [12] *Python* [en línia]. Disponible a: <http://www.es.python.org/> [consulta: març-juliol 2015]
- [13] *SDL* [En línia]. Disponible a: <http://www.libsdl.org/> [consulta: març 2015]
- [14] Shaw Zed, A. (2014). *Aprenda a programar con Python*. Madrid: Anaya Multimedia, cop
- [15] *Simulació: espectrofotometria d'absorció en l'UV-visible* [en línia]. Disponible a: <http://upcommons.upc.edu/ocw/diposit/material/34078/46714.swf> [consulta: febrer-abril 2015]
- [16] Sweigart, A. (2012). *Making Games with Python & Pygame*. CreateSpace Independent Publishing Platform
- [17] *Wallpaperscraft* [en línia]. Disponible a: <https://wallpaperscraft.com/> [consulta: 10 de maig 2015]

ANNEX

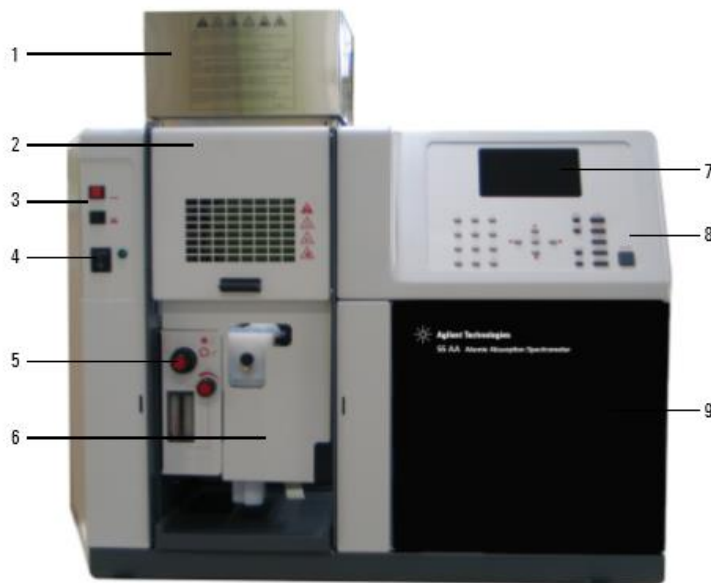
A. DESCRIPCIÓ DE FIGURES

Il·lustració 1: Logo de Python	6
Il·lustració 2: Logo de Pygame	7
Il·lustració 3: Logo de Gimp	7
Il·lustració 4: Aspectre d'absorció característic d'un metall	10
Il·lustració 5: Dibuix d'una làmpada de càtode buit	10
Il·lustració 6: Esquema d'una làmpada de càtode buit	11
Il·lustració 7: Dibuix d'un atomitzador de flama	11
Il·lustració 8: Esquema d'un monocromador.....	12
Il·lustració 9: Dibuix d'un fotomultiplicador (detector òptic).....	12
Il·lustració 10: Volums en mL de les diferents dissolucions	16
Il·lustració 11: Taula de concentracions (ppm) i absorbàncies (UA) del joc.....	17
Il·lustració 12: Diagrama de Gantt de les etapes del projecte	22
Il·lustració 13: Diagrama de fases del joc	25
Il·lustració 14: Diagrama d'accions i estats del joc	28
Il·lustració 15: Exemple d'una Python Shell	31
Il·lustració 16: espectrofotòmetre de la marca "Agilent"	48
Taula 1: Concentracions [ppm] i absorbàncies [UA] emprades a la simulació.....	16
Taula 2: Descripció de les despeses econòmiques estimades	20
Taula 3: Descripció dels diferents elements del joc	23
Taula 4: Descripció dels estats correctes del joc	26
Taula 5: Descripció dels estats i les accions de l'aire i l'acetilè	37
Taula 6: Descripció dels estats i les accions de l'espectrofotòmetre	38
Taula 7: Descripció dels estats i les accions de la pantalla.....	40
Taula 8: Descripció dels estats i les accions de l'aigua i les dissolucions	41

B. MANUAL DE L'USUARI

1. Un cop s'inicia el joc apareix la pantalla de presentació, per començar la partida cal prémer qualsevol tecla del teclat.
2. A la pantalla principal trobareu les quatre dissolucions ja preparades (cadascuna amb la seva concentració de Cd afegida) i l'aigua destil·lada.
3. Per començar enceneu l'espectrofotòmetre mitjançant el botó negre que es troba a l'esquerra de l'aparell [4], s'encendrà la llum verda del costat.
4. A la pantalla apareixen dues opcions : Usuario i Cookbook. Amb les fletxes [10] col·loqueu – vos a <<usuario>> i pitgeu <<ENTER>> .
5. Apareixen els mètodes memoritzats a l'aparell, amb l'ajut de les fletxes i <<ENTER>> escolliu el primer: Cd 228.8.
6. Seguidament apareix la pantalla "Parámetros del instrumento", no toqueu res.
7. Pitgeu el botó <<OPTIMIZE>> i a la pantalla sortiran les absorbàncies de la làmpada de Cd (el mètode escollit).
8. Amb les fletxes col·loqueu-vos damunt de <<Lampara HC>> i pitgeu <<ENTER>>. El valor de l'absorbància s'ha optimitzat i ha de ser molt pròxim a 1.
9. Si es així, poseu-vos a <<señal>> i seguidament pitgeu <<READ>>. Ara ha d'aparèixer un valor molt pròxim a zero ja que heu fet el zero instrumental per calibrar l'aparell.
10. Un cop fet el zero obriu els gasos prement sobre les bombones, però MOLT IMPORTANT:
 - Primer l'aire
 - Després l'acetilè
11. Enceneu ara la flama, pressionant el botó negre [3] que es troba a sobre del botó principal.
12. A la cànula on es mesura l'absorbància [12] col·loqueu el vas de precipitats amb l'aigua destil·lada. Si pitgeu sobre l'aigua veureu que l'arrossegueu amb el moviment del ratolí, només cal que torneu a pitjar quan estiguen a sobre de la safata de l'aparell.
13. Pitgeu <<RESULTS>>, apareix una nova pantalla sense cap lectura.
14. Pitgeu <<READ>> per fer el zero instrumental. A la pantalla apareixerà la lectura de l'absorbància de l'aigua. Aquesta és el tercer valor de la fila que apareix en pantalla i ha de ser molt pròxim a zero.
15. Un cop feta la primera lectura, agafeu el vas amb l'aigua destil·lada i deixeu-lo fóra de l'aparell.
16. Agafeu la primera mostra (sempre de menys a més concentració) i la deixeu a la safata. Torneu a pitjar <<READ>> i us sortirà la segona lectura sota l'anterior.

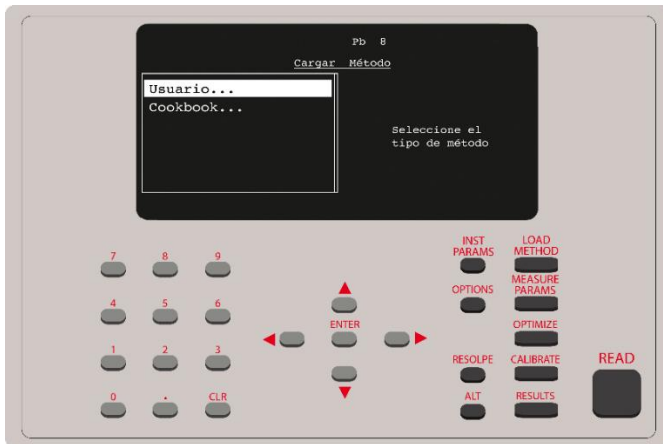
17. Agafeu la següent mostra més concentrada i torneu a pitjar <<READ>>. Feu això fins acabar les mostres.
18. Un cop fetes totes les lectures es treu la última mostra i s'apaga l'aparell:
 - Apagar flama (botó vermell [3'])
 - Apagar gasos (acetilè primer, aire després)
 - Apagar l'instrument, s'apagarà la llum verda (botó negre[4])
19. Un cop acabada la part de laboratori pitgeu la fletxa verda que apareix a la part superior-esquerra de la pantalla.
20. Us apareixerà la recta de regressió obtinguda amb les lectures que heu obtingut. A partir d'aquesta, calculeu mitjançant el mètode de l'addició estàndard, la quantitat de Cd que es troba a la mostra de 25 ml. Recordeu que en la recta de regressió 'Absorbància – concentració de Cd afegida' es busca el valor on l'absorbància és nul·la.
21. Un cop introduït el valor en ppm, pitgeu la tecla INTRO del teclat.



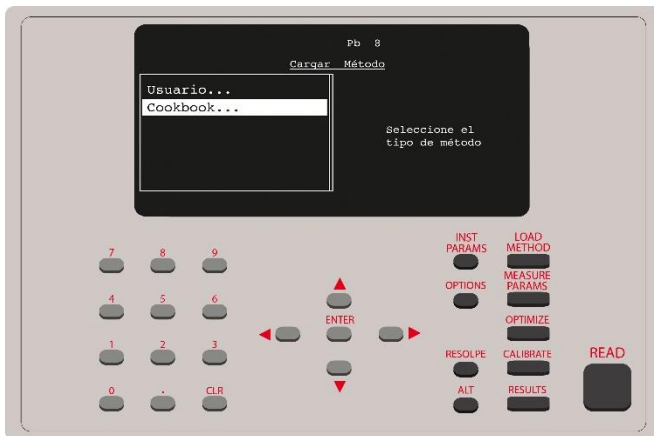
Il·lustració 16: origen a referència [1]

C. PANTALLES

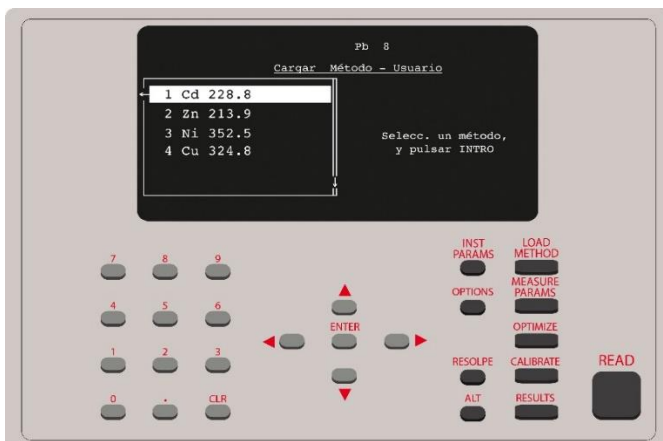
En aquest annex es mostren les pantalles que formen part dels estats essencials del joc, descrites a la Taula 7 de l'apartat 6.4.3 PANTALLA:



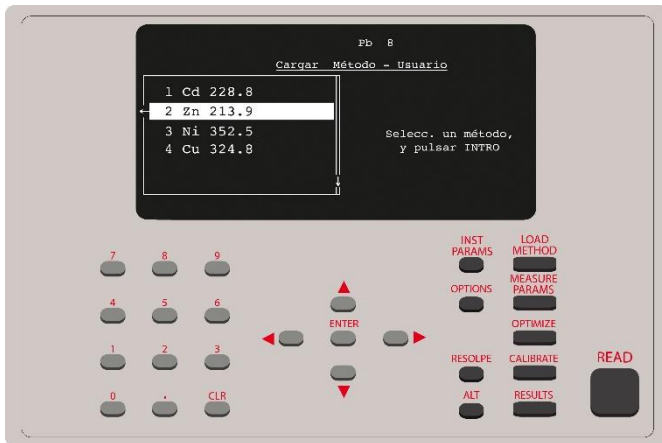
Pantalla 0



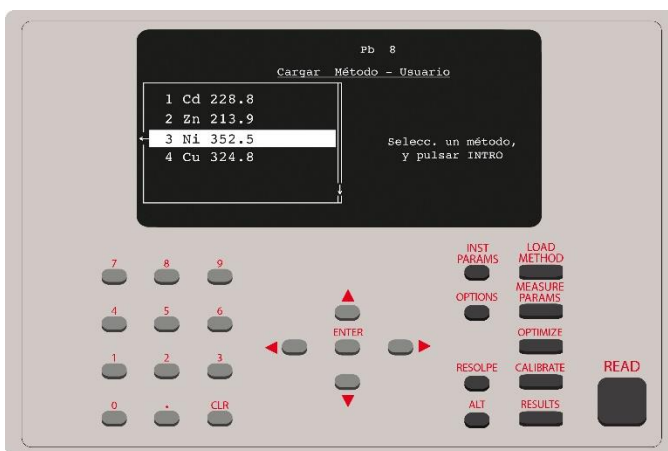
Pantalla 1



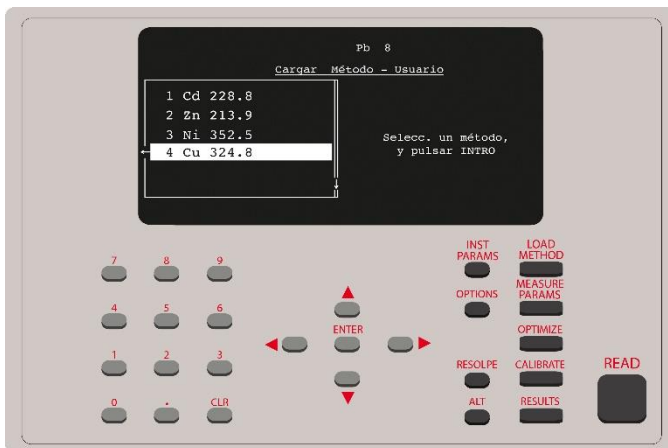
Pantalla 2



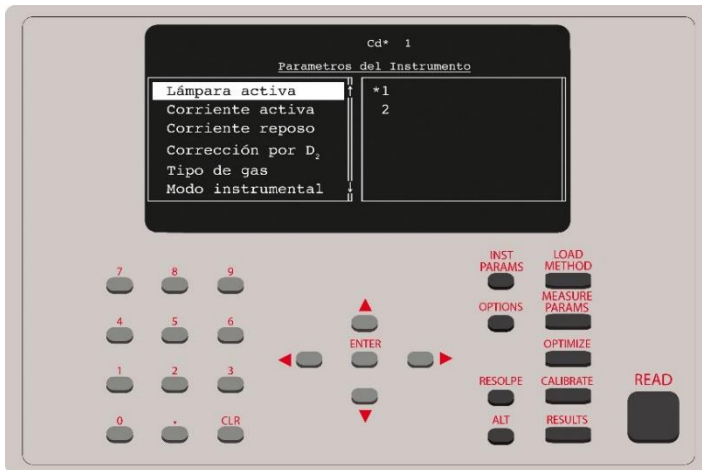
Pantalla 3



Pantalla 4



Pantalla 5



Pantalla 6



Pantalla 7



Pantalla 8



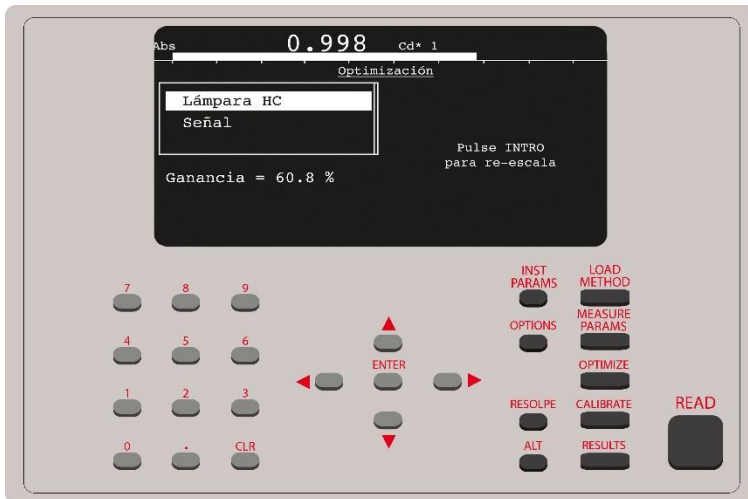
Pantalla 9



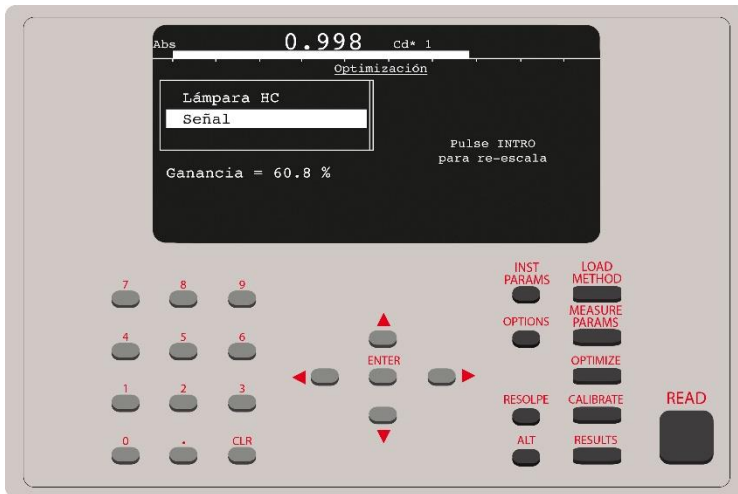
Pantalla 10



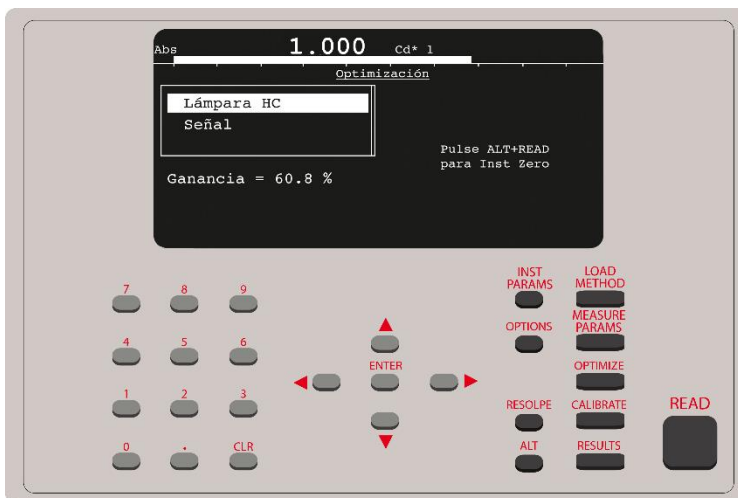
Pantalla 11



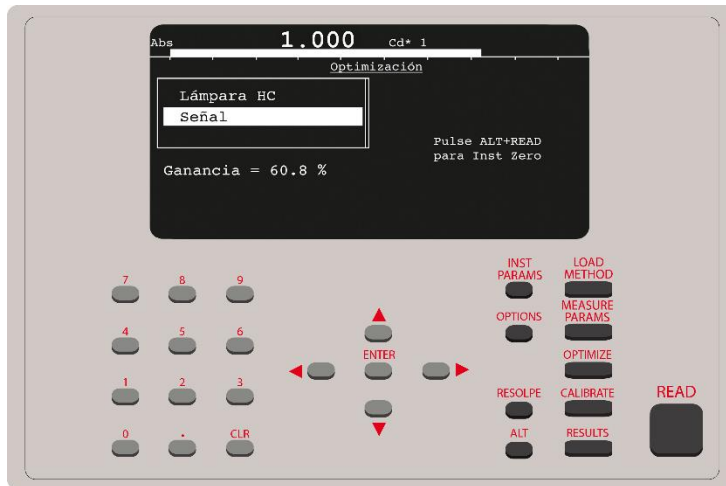
Pantalla 12



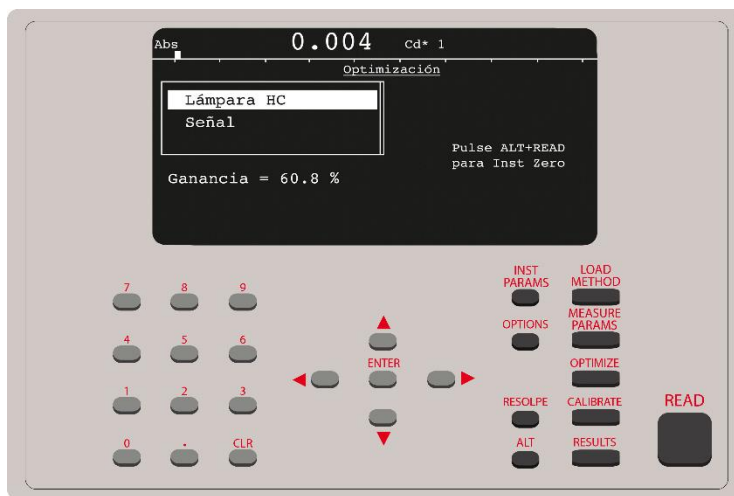
Pantalla 13



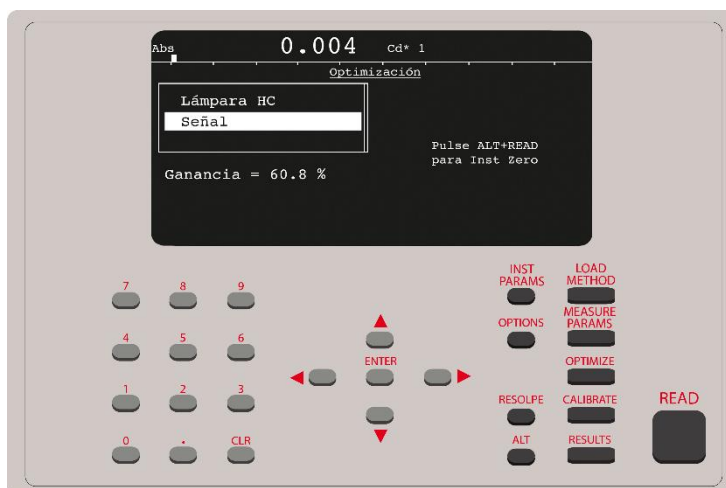
Pantalla 14



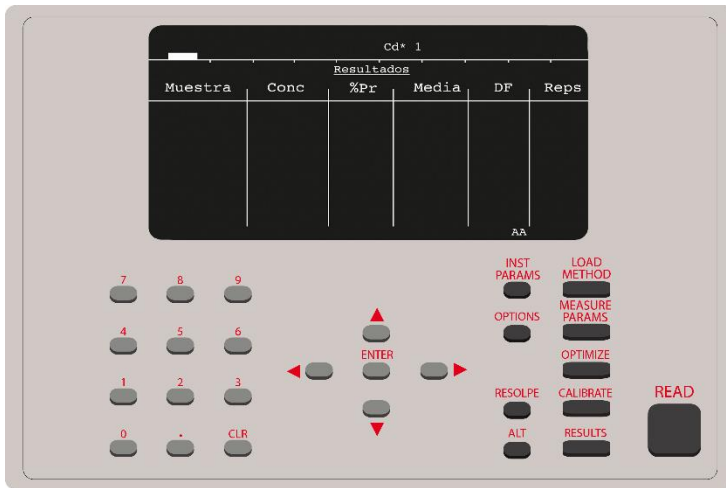
Pantalla 15



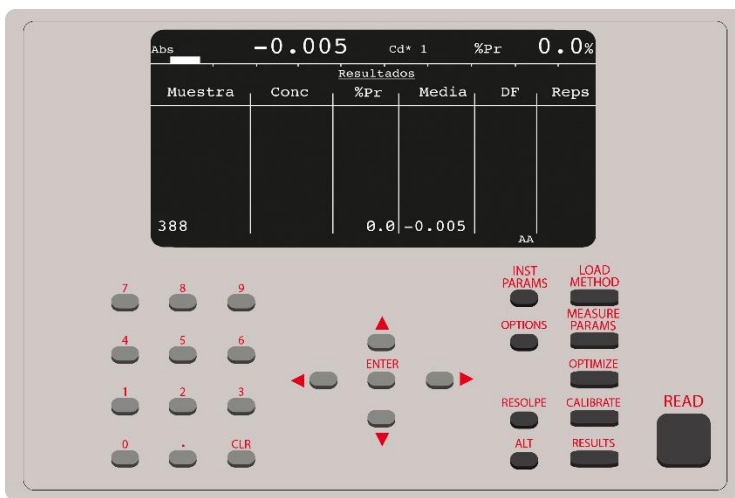
Pantalla 16



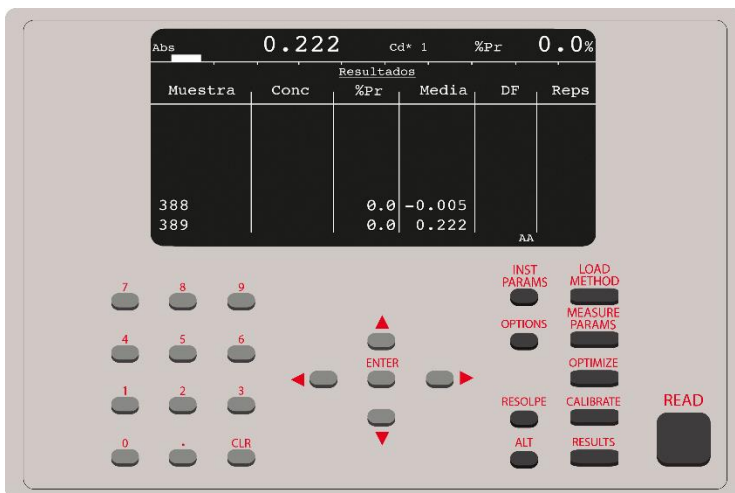
Pantalla 17



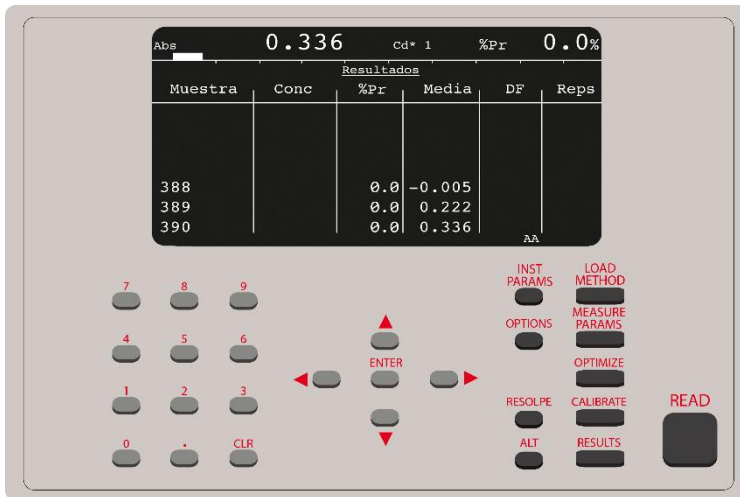
Pantalla 18



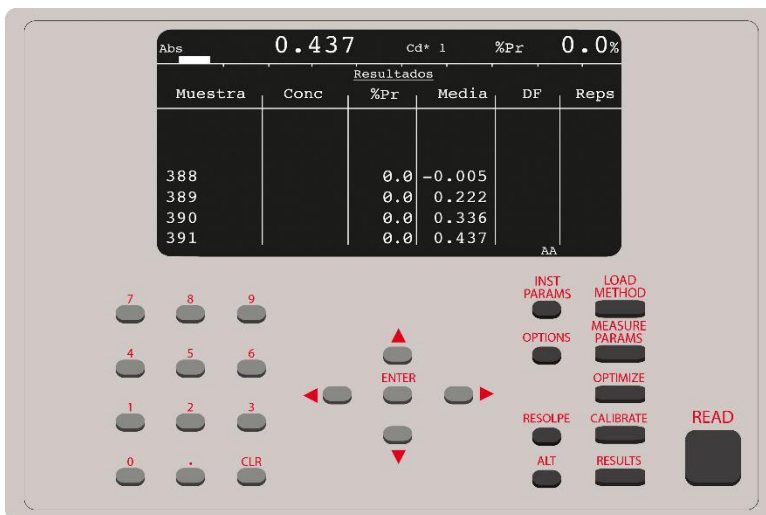
Pantalla 19



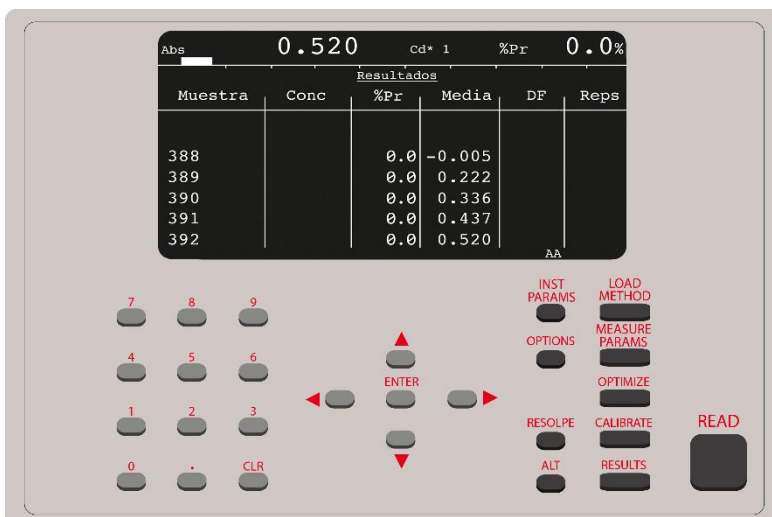
Pantalla 20



Pantalla 21



Pantalla 22



Pantalla 23

D. MÒDULS PYTHON

D.1 practica.py

```
#!/usr/bin/python
import pygame, sys

from pygame.locals import *
import component
import escena
import escenaGrafica
import principal
import intro
import Global
import objecte
import stats
import fin
import input
import datetime

def process_events_intro( g ):
    for event in pygame.event.get() :
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN and event.key != K_ESCAPE:
            g.end = True
        elif event.type == pygame.VIDEORESIZE:
            g.resize(event.size)
        elif event.type == pygame.KEYDOWN and event.key == K_ESCAPE:
            pygame.quit()
            sys.exit()

def process_events( g ):
    elements = []
    for event in pygame.event.get() :
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN and event.key == K_ESCAPE:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN and event.button ==
1:
            c = getClickedComponent(g.components, event.pos)
            if c!= None:
                c.processClick()
                elements.append (c)
                g.actualitza ()
                p,v= stats.process_stats()
                print (p,v)
                if v == 0:
                    Global.end = True
                    Global.end2 = True

        elif event.type == pygame.VIDEORESIZE:
            g.resize(event.size)
```

```

def process_events_final( g ):
    pygame.font.init()
    input.display_box(g.area, "".join(Global.current_string))
    for event in pygame.event.get() :
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == K_ESCAPE:
                pygame.quit()
                sys.exit()
            elif event.key == K_BACKSPACE:
                Global.current_string = Global.current_string[0:-1]
            elif event.key == K_RETURN:
                p,v = input.evalua (Global.current_string)
                print (p,v)
                g.end = True
            elif event.key <= 127:
                Global.current_string.append(chr(event.key))

        input.display_box(g.area, "".join(Global.current_string))

    elif event.type == pygame.VIDEORESIZE:
        g.resize(event.size)

def getClickedComponent(components, pos) :
    """
    Determina sobre quin element grafic s'ha fet el click
    """
    for c in components :
        if c.nom == 'maquina' or c.nom == 'pantalla':
            if c.isOnTopOfWhat(pos):
                return c

    else:
        if c.isOnTopOf(pos):
            return c

    return None

def introduccio():
    e = intro.setEscenaInicial() # Es crea l'escenapygame.init()
    mida = (1250,750)
    color = Global.BLACK
    nom_fons = 'inici.jpg'
    tick = 20
    crono = pygame.time.Clock()
    g = escenaGrafica.EscenaGrafica(e, mida, color,nom_fons)
    while not g.end :
        process_events_intro(g)
        g.draw()
        pygame.display.flip()
        crono.tick(tick)

```

```
def main():
    """
    main loop del joc
    """
    e = principal.setEscena() # Es crea l'escena
    info = pygame.display.Info()
    mida = (info.current_w, info.current_h)
    color = Global.WHITE
    nom_fons = 'principal.jpg'
    tick = 20
    crono = pygame.time.Clock()
    g = escenaGrafica.EscenaGrafica(e, mida, color, nom_fons)
    h_inici = datetime.datetime.now() #inici timer
    tmax = 120
    while not Global.end:
        process_events(g)
        g.draw()
        pygame.display.flip()
        h_actual = datetime.datetime.now() #temps actual
        temps = h_actual - h_inici #temps transcorregut
        t = temps.seconds
        if t > tmax:
            Global.end= True
            Global.end2 = True
            crono.tick(tick)

def final():
    e= fin.setEscenaFinal()
    info = pygame.display.Info()
    mida = (info.current_w, info.current_h)
    color = Global.BLACK
    nom_fons = 'fons.jpg'
    tick = 20
    crono = pygame.time.Clock()
    g = escenaGrafica.EscenaGrafica(e, mida, color, nom_fons)
    while not g.end:
        process_events_final(g)
        g.draw()
        crono.tick(tick)

def puntuacio():
    e = fin.setEscenaPuntuacio()
    info = pygame.display.Info()
    mida = (info.current_w, info.current_h)
    color = Global.BLACK
    nom_fons = 'fons_punt.jpg'
    tick = 20
    crono = pygame.time.Clock()
    g = escenaGrafica.EscenaGrafica(e, mida, color, nom_fons)
    while not g.end:
        process_events_intro(g)
        g.draw()
        pygame.display.flip()
        crono.tick(tick)
```

```

if __name__ == '__main__':
    pygame.init()
    introduccio()
    main()
    if Global.end2 != True:
        final()
    puntuacio()
    pygame.quit()
    sys.exit()

```

D.2 Global.py

```
import copy
```

```

WHITE      = (255, 255, 255)
BLACK      = (  0,   0,   0)
RED        = (255,   0,   0)
NBLUE     = (  0,   0, 128)
GREEN      = (  0, 192,   0)
DARKGREEN  = (  0, 155,   0)
DARKGRAY   = ( 40,  40,  40)
SILVER     = (192, 192, 192)
AQUA      = (  0, 255, 255)

```

```

ok_aire = False
ok_acetile = False
ok_maquina = False
m= [False]*5
end = False
end2 = False
l = []
t = []
estat_actual = []
actius = []
current_string = []
vides = 3
punts = 0

```

```

a1 = [0,0,1,0,0,0,0,0,0] #enciendo máquina
a2 = [0,0,1,2,0,0,0,0,0] # selecciono usuario
a3 = [0,0,1,6,0,0,0,0,0] # selecciono método
a4 = [0,0,1,12,0,0,0,0,0] # optimizar
a5 = [0,0,1,14,0,0,0,0,0] #re-escalar
a6 = [0,0,1,15,0,0,0,0,0] #me situo en señal
a7 = [0,0,1,17,0,0,0,0,0] # cero instrumental
a8 = [0,1,1,17,0,0,0,0,0] #abro aire
a9 = [1,1,1,17,0,0,0,0,0] #abro acetile
a10= [1,1,4,17,0,0,0,0,0] #enciendo llama (estat4 máquina)
a11= [1,1,4,17,0,0,0,0,1] #cojo agua
b11= [1,1,4,17,0,0,0,0,3] #dejo agua bandeja
c11= [1,1,4,18,0,0,0,0,3] #aprieto resultados --> cambio pantalla
d11= [1,1,4,19,0,0,0,0,3] #lectura del agua
e11= [1,1,4,19,0,0,0,0,1] #saco agua
f11= [1,1,4,19,0,0,0,0,2] #dejo agua fuera
a12= [1,1,4,19,1,0,0,0,2] # cojo m1
b12= [1,1,4,19,3,0,0,0,2] #dejo m1 bandeja
c12= [1,1,4,20,3,0,0,0,2] # lectura m1
d12= [1,1,4,20,1,0,0,0,2] # saco m1
e12= [1,1,4,20,2,0,0,0,2] #dejo m1 fuera
a13= [1,1,4,20,2,1,0,0,2] # cojo m2

```

```

b13= [1,1,4,20,2,3,0,0,2] # dejo m2 bandeja
c13= [1,1,4,21,2,3,0,0,2] # lectura m2
d13= [1,1,4,21,2,1,0,0,2] # cojo m2
e13= [1,1,4,21,2,2,0,0,2] # dejo m2 fuera
a14= [1,1,4,21,2,2,1,0,2] # cojo m3
b14= [1,1,4,21,2,2,3,0,2] # dejo m3 bandeja
c14= [1,1,4,22,2,2,3,0,2] # lectura m3
d14= [1,1,4,22,2,2,1,0,2] # cojo m3
e14= [1,1,4,22,2,2,2,0,2] # dejo m3 fuera
a15= [1,1,4,22,2,2,2,1,2] # cojo m4
b15= [1,1,4,22,2,2,2,3,2] # dejo m4 bandeja
c15= [1,1,4,23,2,2,2,3,2] # lectura m4
d15= [1,1,4,23,2,2,2,1,2] # cojo m4
e15= [1,1,4,23,2,2,2,2,2] # dejo m4 fuera
a16= [1,1,1,23,2,2,2,2,2] # apago llama (estat1 maq)
a17= [0,1,1,23,2,2,2,2,2] #apago acetile
a18= [0,0,1,23,2,2,2,2,2] #apago aire
a19= [0,0,0,23,2,2,2,2,2] #apago todo

```

```

s =
[a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,b11,c11,d11,e11,f11,a12,b12,c12,d1
2,e12,a13,b13,c13,d13,e13,a14,b14,c14,d14,e14,a15,b15,c15,d15,e15,a16,
a17,a18,a19]

```

```

tasca = copy.copy(s)
estat_esperat = 0

```

D.3 escena.py

```

import objecte
import component
class Escena:
    """
    Una escena es un conjunt de components
    """
    def __init__(self, mides) :
        self.mides = mides #mides totals de l'escena
        self.components = [] #els components de l'escena: gas,
pantalla...

    def addComponent(self, c) :
        self.components.append(c)

    def delComponent(self, c) :
        for i in range(len(self.components)) :
            if self.components == c:
                del(self.components[i])
                return self.components

    def __len__(self) :
        return len(self.components)

    def __getitem__(self, i) :
        return self.components[i]

```

D.4 intro.py

```

import escena

```

```
import pygame
import sys
import objecte
import component
import textObjecte
```

```
WHITE      = (255, 255, 255)
BLACK      = (  0,   0,   0)
RED        = (255,   0,   0)
NBLUE     = (  0,   0, 128)
GREEN     = (  0, 192,   0)
DARKGREEN = (  0, 155,   0)
DARKGRAY  = ( 40,  40,  40)
SILVER    = (192, 192, 192)
AQUA      = (  0, 255, 255)
```

```
def setEscenaInicial():
```

```
    e = escena.Escena((70, 50))
    titleFont2 = pygame.font.Font('FreeSans.ttf', 25)
    titleFont2.set_bold(True)
    text2 = textObjecte.TextObjecte ('PREM UNA TECLA PER COMENÇAR EL
JOC', titleFont2, RED, (35, 28.5))
    e.addComponent (text2)

    return e
```

D.4 principal.py

```
import pygame
import objecte
import component
import escena
import textObjecte
import Global
```

```
def setEscena() :
```

```
    """
    Funcio que crea una escena: defineix tots els components, el seus
    estats..
    """
```

```
    e = escena.Escena((10, 5))
    x,y = pygame.mouse.get_pos()
    caps_aac = (0.1, 1.6, 0.8, 3.2)
    caps_aire = (0.9, 1.6, 0.8, 3.2)
    caps_h20 = (3.35, 3.07, 0.3, 0.3)
    caps_maq = (4.1, 1, 2.4, 2.4)
    caps_m1 = (2.9, 3.05, 0.35, 0.35)
    caps_m2 = (2.5, 3.05, 0.35, 0.35)
    caps_m3 = (2.1, 3.05, 0.35, 0.35)
    caps_m4 = (1.7, 3.05, 0.35, 0.35)
    caps_mov = (x, y, 0.35, 0.35)
    caps_mov1 = (x, y, 0.3, 0.3)
    caps_ext = (4.15, 0, 1.3, 1.1)
    caps_tau = (1.7, 3.39, 4.8, 0.2)
    caps_pant = (6.7, 0.2, 2.8, 1.8)
    caps_next = (1, 0.1, 0.55, 0.55)
```

```

    acetile= objecte.Objecte('acetile')
    estat1 = component.Estat('Tancat', caps_a_ac,'gris.png' ,
acetile.encendre)
    acetile.addEstat (estat1)
    estat2 = component.Estat('Ences', caps_a_ac, 'gris1.png',
acetile.apagar)
    acetile.addEstat (estat2)
    e.addComponent (acetile)
    Global.actius.append (acetile)

    aire= objecte.Objecte('aire')
    estat1 = component.Estat('Tancat', caps_a_aire,'azul.png' ,
aire.encendre)
    aire.addEstat (estat1)
    estat2 = component.Estat('Ences', caps_a_aire, 'azul1.png',
aire.apagar)
    aire.addEstat (estat2)
    e.addComponent (aire)
    Global.actius.append (aire)

    maquina = objecte.Objecte ('maquina')
    estat0 = component.Estat ('Apagada, tancada', caps_a_maq,
'foto0.png', [maquina.encendre, maquina.obrir, maquina.doNothing,
maquina.doNothing])
    maquina.addEstat (estat0)
    estat1 = component.Estat ('Encesa, tancada', caps_a_maq,
'foto1.png', [maquina.apagar, maquina.obrir, maquina.encendreFOC,
maquina.doNothing])
    maquina.addEstat (estat1)
    estat2 = component.Estat ('Apagada, oberta', caps_a_maq,
'foto2.png', [maquina.encendre,maquina.tancar, maquina.doNothing,
maquina.doNothing] )
    maquina.addEstat (estat2)
    estat3 = component.Estat ('Encesa, oberta', caps_a_maq,
'foto3.png', [maquina.apagar, maquina.tancar, maquina.encendreFOC,
maquina.doNothing])
    maquina.addEstat (estat3)
    estat4 = component.Estat ('Encesa, tancada, flama', caps_a_maq,
'foto4.png', [maquina.apagar, maquina.obrir, maquina.doNothing,
maquina.apagarFOC])
    maquina.addEstat (estat4)
    estat5 = component.Estat ('Encesa, oberta, flama', caps_a_maq,
'foto5.png', [maquina.apagar, maquina.tancar, maquina.doNothing,
maquina.apagarFOC])
    maquina.addEstat (estat5)
    e.addComponent (maquina)
    Global.actius.append (maquina)

    extractor = objecte.Objecte ('extractor')
    estat= component.Estat ('estatic', caps_a_ext, 'extractor.png',
extractor.doNothing)
    extractor.addEstat (estat)
    e.addComponent (extractor)

    taula = objecte.Objecte ('taula')
    estat= component.Estat ('estatic', caps_a_tau, 'mesa.png',
taula.doNothing)
    taula.addEstat (estat)
    e.addComponent (taula)

```



```
    pantalla = objecte.Objecte ('pantalla')
    estat1= component.Estat ('e1', caps_pant, 'P1.png', [pantalla.up,
pantalla.down, pantalla.intro,
pantalla.doNothing,pantalla.doNothing,pantalla.doNothing ])
    pantalla.addEstat (estat1)
    estat2= component.Estat ('e2', caps_pant, 'P2.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing] )
    pantalla.addEstat (estat2)
    estat3= component.Estat ('e3', caps_pant, 'P3.png', [pantalla.up,
pantalla.down, pantalla.intro,
pantalla.doNothing,pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat3)
    estat4= component.Estat ('e4', caps_pant, 'P4.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat4)
    estat5= component.Estat ('e5', caps_pant, 'P5.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat5)
    estat6= component.Estat ('e6', caps_pant, 'P6.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat6)
    estat7= component.Estat ('e7', caps_pant, 'P7.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat7)
    estat8= component.Estat ('e8', caps_pant, 'P8.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat8)
    estat9= component.Estat ('e9', caps_pant, 'P9.png', [pantalla.up,
pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat9)
    estat10= component.Estat ('e10', caps_pant, 'P10.png',
[pantalla.up, pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat10)
    estat11= component.Estat ('e11', caps_pant, 'P11.png',
[pantalla.up, pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat11)
    estat12= component.Estat ('e12', caps_pant, 'P12.png',
[pantalla.up, pantalla.down, pantalla.intro, pantalla.opt,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat12)
    estat13= component.Estat ('e13', caps_pant, 'P13.png',
[pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat13)
    estat14= component.Estat ('e14', caps_pant, 'P14.png',
[pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat14)
    estat15= component.Estat ('e15', caps_pant, 'P15.png',
[pantalla.up, pantalla.down, pantalla.doNothing, pantalla.doNothing,
pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat15)
```

```

    estat16= component.Estat ('e16', caps_pant, 'P16.png',
    [pantalla.up, pantalla.down, pantalla.doNothing, pantalla.doNothing,
    pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat16)
    estat17= component.Estat ('e17', caps_pant, 'P17.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.doNothing,pantalla.result])
    pantalla.addEstat (estat17)
    estat18= component.Estat ('e18', caps_pant, 'P18.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.doNothing,pantalla.result])
    pantalla.addEstat (estat18)
    estat19= component.Estat ('e19', caps_pant, 'P19.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.reading ,pantalla.doNothing])
    pantalla.addEstat (estat19)
    estat20= component.Estat ('e20', caps_pant, 'P20.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat20)
    estat21= component.Estat ('e21', caps_pant, 'P21.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat21)
    estat22= component.Estat ('e22', caps_pant, 'P22.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat22)
    estat23= component.Estat ('e23', caps_pant, 'P23.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.reading,pantalla.doNothing])
    pantalla.addEstat (estat23)
    estat24= component.Estat ('e24', caps_pant, 'P24.png',
    [pantalla.up, pantalla.down, pantalla.intro, pantalla.doNothing,
    pantalla.doNothing,pantalla.doNothing])
    pantalla.addEstat (estat24)
    e.addComponent (pantalla)
    Global.actius.append (pantalla)

    mostral = objecte.Objecte ('mostral')
    estat0= component.Estat ('estatic', caps_m1, 'vaso1.png',
    mostral.update)
    mostral.addEstat (estat0)
    estat1= component.Estat ('agafat', caps_m1, 'vaso1.png',
    mostral.update)
    mostral.addEstat (estat1)
    estat2= component.Estat ('estatic2', caps_mov, 'vaso1.png',
    mostral.update)
    mostral.addEstat (estat2)
    estat3= component.Estat ('safata', caps_mov, 'vaso1.png',
    mostral.update)
    mostral.addEstat (estat3)
    e.addComponent (mostral)
    Global.l.append (mostral)
    Global.actius.append (mostral)

    mostra2 = objecte.Objecte ('mostra2')
    estat= component.Estat ('estatic', caps_m2, 'vaso2.png',
    mostra2.update)
    mostra2.addEstat (estat)

```

```
    estat= component.Estat ('agafat', capsam2, 'vaso2.png',
mostra2.update)
    mostra2.addEstat (estat)
    estat= component.Estat ('deixat', capsamov, 'vaso2.png',
mostra2.update)
    mostra2.addEstat (estat)
    e.addComponent(mostra2)
    Global.l.append(mostra2)
    Global.actius.append (mostra2)

    mostra3 = objecte.Objecte ('mostra3')
    estat= component.Estat ('estatic', capsam3, 'vaso3.png',
mostra3.update)
    mostra3.addEstat (estat)
    estat= component.Estat ('agafat', capsam3, 'vaso3.png',
mostra3.update)
    mostra3.addEstat (estat)
    estat= component.Estat ('deixat', capsamov, 'vaso3.png',
mostra3.update)
    mostra3.addEstat (estat)
    e.addComponent(mostra3)
    Global.l.append(mostra3)
    Global.actius.append (mostra3)

    mostra4 = objecte.Objecte ('mostra4')
    estat= component.Estat ('estatic', capsam4, 'vaso4.png',
mostra4.update)
    mostra4.addEstat (estat)
    estat= component.Estat ('agafat', capsam4, 'vaso4.png',
mostra4.update)
    mostra4.addEstat (estat)
    estat= component.Estat ('deixat', capsamov, 'vaso4.png',
mostra4.update)
    mostra4.addEstat (estat)
    e.addComponent(mostra4)
    Global.l.append(mostra4)
    Global.actius.append (mostra4)

    aigua = objecte.Objecte ('aigua')
    estat= component.Estat ('estatic', capsah20, 'agua.png',
aigua.update)
    aigua.addEstat (estat)
    estat= component.Estat ('agafat', capsah20, 'agua.png',
aigua.update)
    aigua.addEstat (estat)
    estat= component.Estat ('deixat', capsamov1, 'agua.png',
aigua.update)
    aigua.addEstat (estat)
    e.addComponent(aigua)
    Global.l.append(aigua)
    Global.actius.append (aigua)

    boto_next = objecte.Objecte ('next')
    estat= component.Estat ('estatic', capsanext, 'next.png',
next.prem)
    boto_next.addEstat (estat)
    e.addComponent (boto_next)

    return e
```

D.5 fin.py

```
import pygame
import escena
import objecte
import component
import Global
import textObjecte

def setEscenaFinal():
    e=escena.Escena((10,5))

    title = pygame.font.Font('FreeSans.ttf', 25)
    title.set_bold(True)
    text = textObjecte.TextObjecte ('Introdueix la concentració de Cd
a la mostra [ppm]:', title, Global.NBLUE, (5, 1))
    e.addComponent (text)
    text = textObjecte.TextObjecte ('Recta de regressió:  $y=0.38x + 0.2265$ ', title, Global.NBLUE, (5, 1.5))
    e.addComponent (text)

    return e

def setEscenaPuntuacio():
    e=escena.Escena((10,5))

    title = pygame.font.Font('FreeSans.ttf', 25)
    title.set_bold(True)
    text = textObjecte.TextObjecte ('Puntuació:'+ str(Global.punts),
title, Global.WHITE, (5, 1))
    e.addComponent (text)
    text = textObjecte.TextObjecte ('Vides:'+ str(Global.vides),
title, Global.WHITE, (5, 1.5))
    e.addComponent (text)

    if Global.vides == 0:
        text = textObjecte.TextObjecte ('Que malament...un altre cop
serà...', title, Global.WHITE, (7, 2.5))
        e.addComponent (text)

    else:
        if Global.punts >= 4600:
            text = textObjecte.TextObjecte ('Ets un GENI!!!', title,
Global.WHITE, (7, 2.5))
            e.addComponent (text)

            elif Global.punts >= 3500 and Global.punts < 4600:
                text = textObjecte.TextObjecte ('Bastant bé crack!!!',
title, Global.WHITE, (7, 2.5))
                e.addComponent (text)

            elif Global.punts >= 2000 and Global.punts < 3500:
                text = textObjecte.TextObjecte ('Se que encara pots
més!!!', title, Global.WHITE, (7, 2.5))
                e.addComponent (text)

            elif Global.punts >= 700 and Global.punts < 2000:
```

```

        text = textObjecte.TextObjecte ('Encara podem millorar
bastant...', title, Global.WHITE, (7, 2.5))
        e.addComponent (text)

    elif Global.punts < 700:
        text = textObjecte.TextObjecte ('Aquesta no és la teva
assignatura, eh?', title, Global.WHITE, (7, 2.5))
        e.addComponent (text)

    return e

```

D.7 component.py

```

class Estat:
    """
    Un estat d'un objecte es caracteritza pel seu nom, la capsa minima
    contenidora que ocupa en l'escena i l'imatge associada.
    Conte tambe informacio de l'accio que es pot fer sobre l'objecte
    en aquest estat.
    """
    def __init__(self, nom_estat, capsa, imatge, accio) :
        self.nom = nom_estat
        self.capsa = capsa
        self.imatge = imatge
        self.accio= accio

class Component:
    """
    Un component de l'escena es caracteritza pel seu nom, la seva
    llista
    d'estats i l'index de l'estat actual. Com a minim hi ha un estat.
    """
    def __init__(self, nom):
        self.nom = nom
        self.estats = []
        self.estat_actual = 0
        self.parts =[]

    def __len__(self):
        return len(self.estats)

    def __getitem__(self, i) :
        return self.estats[i]

    def addPart(self, e) :
        self.parts.append(e)

    def addEstat(self, e) :
        self.estats.append(e)

    def delEstat(self, e) :
        for i in range(len(self.estats)) :
            if self.estats == e :
                del(self.estats[i])
                return self.estats

    def getNomEstat(self):
        return self[self.estat_actual].nom

    def getEstat (self):
        return self.estat_actual

```

```
def canviaEstat (self,e):
    self.estat_actual = e

def getCapsaActual(self) :
    return self[self.estat_actual].capsa

def getImatgeActual(self):
    return self[self.estat_actual].imatge

def getAccioActual(self):
    if self.nom == 'maquina':

        if self.parts[-1] == 0: # botó
            return self[self.estat_actual].accio[0]

        elif self.parts[-1] == 1: #porta
            return self[self.estat_actual].accio[1]

        elif self.parts[-1] == 2: # boto flama ON
            return self[self.estat_actual].accio[2]

        elif self.parts[-1] == 3: #boto flama OF
            return self[self.estat_actual].accio [3]

    elif self.nom == 'pantalla':

        if self.parts[-1] == 0: # UP
            return self[self.estat_actual].accio[0]

        elif self.parts[-1] == 1: # DOWN
            return self[self.estat_actual].accio[1]

        elif self.parts[-1] == 2: # boto INTRO
            return self[self.estat_actual].accio[2]

        elif self.parts[-1] == 3: #boto OPT
            return self[self.estat_actual].accio [3]

        elif self.parts[-1] == 4: #boto READ
            return self[self.estat_actual].accio [4]

        elif self.parts[-1] == 5: #boto READ
            return self[self.estat_actual].accio [5]

    else:
        return self[self.estat_actual].accio

def doAccioActual(self) :
    return self.getAccioActual() ()

def getText (self):
    return self.text

def getFont (self):
    return self.font

def getColor (self):
    return self.color
```

```
def getPos (self):  
    return self.pos
```

D.8 objecte.py

```
import component  
import pygame  
import Global  
  
class Objecte(component.Component) :  
    def __init__(self, nom) :  
        super().__init__(nom)  
  
    def encendre(self) :  
        if self.estat_actual == 0:  
            self.estat_actual = 1  
  
            if self.nom == 'aire':  
                Global.ok_aire = True  
            elif self.nom == 'acetile':  
                Global.ok_acetile = True  
            else:  
                Global.ok_maquina= True  
  
        elif self.estat_actual == 2:  
            self.estat_actual = 3  
            Global.ok_maquina= True  
  
        print(self.nom, 'ON', self.estat_actual)  
  
    def apagar(self) :  
        if self.estat_actual in [1,4]:  
            self.estat_actual = 0  
  
            if self.nom == 'aire':  
                Global.ok_aire = False  
            elif self.nom == 'acetile':  
                Global.ok_acetile = False  
            else:  
                Global.ok_maquina= False  
  
        elif self.estat_actual in [3,5]:  
            self.estat_actual = 2  
            Global.ok_maquina = False  
  
        print(self.nom, 'OFF', self.estat_actual)  
  
    def obrir (self):  
        if self.estat_actual == 0:  
            self.estat_actual = 2  
        elif self.estat_actual == 1:  
            self.estat_actual = 3  
        elif self.estat_actual == 4:  
            self.estat_actual =5  
  
        print (self.nom, 'Porta oberta', self.estat_actual)  
  
    def tancar (self):
```

```
if self.estat_actual == 2:
    self.estat_actual = 0
elif self.estat_actual == 3:
    self.estat_actual = 1
elif self.estat_actual == 5:
    self.estat_actual = 4

print (self.nom, 'Porta tancada', self.estat_actual)

def encendreFOC (self):
if Global.ok_aire==True and Global.ok_acetile==True:
    if self.estat_actual == 1:
        self.estat_actual = 4
    elif self.estat_actual == 3:
        self.estat_actual = 5
    print (self.nom, 'Flama encesa', self.estat_actual)
else:
    pass

def apagarFOC (self):
if self.estat_actual == 4:
    self.estat_actual = 1
elif self.estat_actual == 5:
    self.estat_actual = 3

print (self.nom, 'Flama apagada', self.estat_actual)

def doNothing (self):
    pass

def update (self):
if self.estat_actual == 0: #inicial
    self.estat_actual = 1 #moviment
elif self.estat_actual == 1:
    self.estat_actual = 2 #estatic
elif self.estat_actual == 2:
    self.estat_actual = 1

print (self.nom, 'updated', self.estat_actual)

def down(self):
if Global.ok_maquina == True:

    if self.estat_actual in [0,2,3,4,6,7,8,9,10,12,14,16]:
        self.estat_actual = self.estat_actual + 1
    elif self.estat_actual == 1 :
        self.estat_actual = 0
    elif self.estat_actual == 5:
        self.estat_actual = 2
    elif self.estat_actual == 11:
        self.estat_actual = 6
    elif self.estat_actual in [13,15,17]:
        self.estat_actual = self.estat_actual -1

    print (self.nom, 'tecla down', self.estat_actual)

def up (self):
if Global.ok_maquina == True:
```



```
    if self.estat_actual in [0,12,14,16]:
        self.estat_actual = self.estat_actual +1
    elif self.estat_actual in [1,3,4,5,7,8,9,10,11,13,15,17]:
        self.estat_actual = self.estat_actual -1
    elif self.estat_actual == 2:
        self.estat_actual = 5
    elif self.estat_actual == 6:
        self.estat_actual = 11

    print (self.nom, 'tecla up', self.estat_actual)

def intro (self):
    if Global.ok_maquina == True:
        if self.estat_actual == 0:
            self.estat_actual = 2
        elif self.estat_actual in [2,3,4,5]:
            self.estat_actual = 6
        elif self.estat_actual == 12:
            self.estat_actual = 14
        elif self.estat_actual == 13:
            self.estat_actual = 15

    print (self.nom, 'tecla intro', self.estat_actual)

def opt (self):
    if Global.ok_maquina == True:
        if self.estat_actual in list(range(6,11)):
            self.estat_actual = 12
        print (self.nom, 'tecla optimitzar', self.estat_actual)

def reading (self):
    if Global.ok_maquina == True:
        if self.estat_actual == 15:
            self.estat_actual = 17

        elif self.estat_actual ==18:
            if Global.m[4]== True:
                self.estat_actual = 19
        elif self.estat_actual ==19:
            if Global.m[0]== True:
                self.estat_actual = 20
        elif self.estat_actual ==20:
            if Global.m[1]== True:
                self.estat_actual = 21
        elif self.estat_actual ==21:
            if Global.m[2]== True:
                self.estat_actual = 22
        elif self.estat_actual ==22:
            if Global.m[3]== True:
                self.estat_actual = 23

    print (self.nom, 'tecla lectura', self.estat_actual)

def result (self):
    if Global.ok_maquina == True:
        if self.estat_actual in [16,17]:
            self.estat_actual = 18

    print (self.nom, 'tecla resultats', self.estat_actual)
```

```

def prem (self):
    Global.end = True
    if len (Global.M) != 0:
        if Global.M[-1][:3] != [0,0,0]:
            Global.vides = Global.vides -1
    else:
        Global.vides = Global.vides -2

```

D.9 textObjecte.py

```

import objecte
class TextObjecte(objecte.Objecte) :
    def __init__(self, text, font, color, pos) :
        super().__init__(self)
        self.font = font
        self.text = text
        self.color = color
        self.pos = pos

```

D.10 componentGrafic.py

```

import pygame
import Global
import principal

class ComponentGrafic(pygame.sprite.Sprite) :
    def __init__(self, c, scale) :
        pygame.sprite.Sprite.__init__(self)
        self.component = c
        self.nom = c.nom
        self.scale = scale
        self.setGraphics()

    def setGraphics(self) :
        self.estat = self.component.getEstat()
        capsa = self.component.getCapsaActual()
        size = int(capsa[2]*self.scale[0]),
int(capsa[3]*self.scale[1])
        nom_image = self.component.getImatgeActual()
        ima = pygame.image.load(nom_image).convert_alpha()
        self.image = pygame.transform.smoothscale(ima, size)
        self.image.set_colorkey(Global.WHITE)
        pos = capsa[0]*self.scale[0], capsa[1]*self.scale[1]
        self.rect = pygame.Rect(pos, size)
        if self.component in Global.l and self.estat == 2:
            pos = pygame.mouse.get_pos()
            self.rect = pygame.Rect (pos, size)

    def isOnTopOf(self, pos):
        return self.rect.collidepoint (pos)

    def isOnTopOfWhat(self, pos):
        capsa = self.component.getCapsaActual()
        capsaboto = (0.12,1,0.104,0.104)
        capsaportalamp = (1.28, 1.2, 1.12, 1.04)
        capsaflamaOn = (0.12,0.84,0.08,0.112)

```

```

capsaflamaOf= (0.12, 0.696, 0.08, 0.096)
capsa_up = (1.37, 1.18, 0.113, 0.07)
capsa_down = (1.37, 1.46, 0.113, 0.07)
capsa_intro = (1.37, 1.33, 0.113, 0.07)
capsa_opt = (2.14, 1.32, 0.193, 0.058)
capsa_read = (2.48, 1.477, 0.209, 0.2)
capsa_result = (2.14, 1.617, 0.193, 0.058)

if self.nom == 'maquina':
    capses = [capsaboto, capsaportalamp, capsaflamaOn,
capsaflamaOf]
else:
    capses = [capsa_up, capsa_down, capsa_intro, capsa_opt,
capsa_read, capsa_result]
    i = 0
    l = []
    while i< len(capses):
        capsafin = (capsa[0]+capses[i][0], capsa[1]+capses[i][1],
capses[i][2], capses[i][3])
        size = int(capsafin[2]*self.scale[0]),
int(capsafin[3]*self.scale[1])
        posi = capsafin[0]*self.scale[0],
capsafin[1]*self.scale[1]
        rect = pygame.Rect (posi, size)
        if rect.collidepoint(pos):
            l.append(i)
            self.component.addPart(i)

        i= i+1
    if len (l) != 0:
        return True
    else:
        return False

def collide (self, l):
    return self.rect.collidelistall (l)

def processClick(self) :
    self.component.doAccioActual ()
    self.setGraphics ()

```

D.11 componentGraficText.py

```

import pygame

class ComponentGraficText(pygame.sprite.Sprite) :
    def __init__(self, c, scale) :
        pygame.sprite.Sprite.__init__(self)
        self.component = c
        self.nom = c.nom
        self.scale = scale
        self.setGraphics ()

    def setGraphics (self):
        text = self.component.getText ()
        font = self.component.getFont ()
        color = self.component.getColor ()
        self.titol = font.render (text, True, color)
        posicio = self.component.getPos ()
        dim = font.size (text) # dimensió en pixels, no cal escalar

```

```

centre = posicio[0]*self.scale[0] - (dim[0]/2)
size = int (dim[0]*self.scale[0]), int (dim[1]*self.scale[1])
pos = centre, posicio[1]*self.scale[1]
self.rect = pygame.Rect (pos,size)

```

D.12 escenaGrafica.py

```

import pygame
import textObjecte
from pygame.locals import *
import component
import componentGrafic
import componentGraficText
import Global
import principal

class EscenaGrafica:
    def __init__(self,e, mida, color, nom_fons) :
        self.mida = mida
        self.area = pygame.display.set_mode(mida, HWSURFACE |
DOUBLEBUF | RESIZABLE)
        self.end = False
        self.color= color
        self.fons = pygame.image.load(nom_fons).convert()
        self.fons = pygame.transform.smoothscale(self.fons, self.mida)
        self.components = pygame.sprite.OrderedUpdates()
        scale = mida[0]/e.mides[0], mida[1]/e.mides[1]
        self.e = e
        for c in e:
            if isinstance(c, textObjecte.TextObjecte) :
                s = componentGraficText.ComponentGraficText(c, scale)
            else :
                s = componentGrafic.ComponentGrafic(c, scale)
            if c in Global.l:
                Global.t.append (s)
            self.components.add(s)

    def resize(self, newsize) :
        self.newsize = newsize
        self.area = pygame.display.set_mode( newsize , HWSURFACE |
DOUBLEBUF | RESIZABLE)
        nscale = newsize[0]/self.e.mides[0],
newsize[1]/self.e.mides[1]
        self.components = pygame.sprite.RenderPlain()
        self.fons = pygame.transform.smoothscale(self.fons, newsize)
        pygame.display.flip()
        del Global.t[:]
        for c in self.e:
            if isinstance(c, textObjecte.TextObjecte) :
                s = componentGraficText.ComponentGraficText(c, nscale)
            else:
                s = componentGrafic.ComponentGrafic (c, nscale)
            if c in Global.l:
                Global.t.append (s)
            self.components.add(s)

```

```

def draw(self):
    self.area.fill(self.color)
    self.area.blit (self.fons, (0,0))
    for c in self.components :
        if isinstance (c,
componentGraficText.ComponentGraficText):
            self.area.blit (c.titol, c.rect.topleft)
        elif c in Global.t and c.estat == 1:
            c.rect.center = pygame.mouse.get_pos()
            self.area.blit (c.image, c.rect)
        else:
            self.area.blit (c.image, c.rect)

def actualitza (self):
    comp = []
    estats= []
    for c in Global.actius: #objectes
        comp.append (c)
        estats.append (c.estat_actual)

    for c in self.components: #componentsGrafics (colisions)
        if c.nom == 'maquina':
            mostra = c.collide (Global.t)

    if len(mostra) != 0:
        for i in mostra:
            m = i+4 #relacio entre ordre estats i ordre Global.t
            estats[m] = 3
            Global.m[i] = True
    else:
        Global.m= [False]*5

    Global.estat_actual = estats
    print (Global.estat_actual)

```

D.13 estats.py

```

import Global
import escenaGrafica

def process_estats():
    v=vides()
    p=puntua()
    return p,v

def vides() :
    eactual = Global.estat_actual
    if eactual[0] == 1 and eactual[1] == 0: #acetile aigua
        Global.vides = Global.vides -1
    if 3 in eactual[4:8] and eactual[8]== 0:
        Global.vides = Global.vides -1

    return Global.vides

def compara(eactual, esperat) :
    r = True
    for i in range(4) :
        if eactual[i] != esperat[i]:
            r=False

```

```

for i in range(4,9) :
    if eactual[i] != esperat[i] :
        if eactual[i] != 2 or esperat[i] != 0:
            r=False
return r

def puntua() :
    eactual = Global.estat_actual
    if compara(eactual, Global.tasca[Global.estat_esperat] ):
        Global.punts = Global.punts + 100
        if Global.estat_esperat < 40:
            Global.estat_esperat = Global.estat_esperat +1
    else :
        Global.punts = Global.punts -10

return Global.punts

```

D.14 input.py

```

import pygame, pygame.font, pygame.event, pygame.draw
import string
from pygame.locals import *
import Global
import estats

def display_box(screen, message):
    "Print a message in a box in the middle of the screen"
    fontobject = pygame.font.Font(None,30)
    x = screen.get_width()
    y = screen.get_height()
    x1 = x/4
    y1 = y/12
    pygame.draw.rect(screen, Global.WHITE ,((x / 2) - (x1/2), (y / 2)
- (y1/2),x1,y1), 0)
    pygame.draw.rect(screen, Global.AQUA,((x/ 2) - ((x1+6)/2),(y / 2)
- ((y1+6)/2),x1+6,y1+6), 3)

    if len(message) != 0:
        screen.blit(fontobject.render(message, 1, Global.BLACK),((x/2)
- (x1/3), (y/2)))
    pygame.display.flip()

def evalua (list):
    if len(list)!=0:
        if ',' in list:
            n = list.index(',')
            list[n] = '.'
            w = "".join (list[:n+2])
        elif '.' in list:
            n = list.index ('.')
            w = "".join (list[:n+2])
        else:
            w = "".join (list)

    num = float (w)

```

```

if num in [1.4,1.5,1.6]:
    Global.punts= Global.punts +1000

elif num in [0.5,0.6,0.7]:
    Global.punts = Global.punts +500
    Global.vides = Global.vides - 0.5

else:
    Global.vides = Global.vides -1
    Global.punts = Global.punts +100

return Global.punts, Global.vides

```

E. QÜESTIONARI PROVES D'USUARI

* 1. Valora la impressió general que t'ha causat el joc

Molt dolenta	Dolenta	Ni bona ni dolenta	Bona	Molt bona
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

* 2. Valora els següents aspectes del joc

	1	2	3	4	5
Facilitat per entendre el joc (1= molt poca)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aspecte visual del joc (1= molt pobre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oportunitat per aprendre els conceptes associats (1=cap)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Entreteniment (1=molt pobre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Motivació per tornar a jugar (1=cap)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

* 3. T'ha servit com a formació prèvia a la pràctica?

- Sí
- No

* 4. Què consideres que es podria millorar del joc?

- Aspecte visual
- Complexitat
- Claredat dels objectius i/o instruccions
- Llibertat de moviments i/o opcions
- Altres (especifiqueu)

