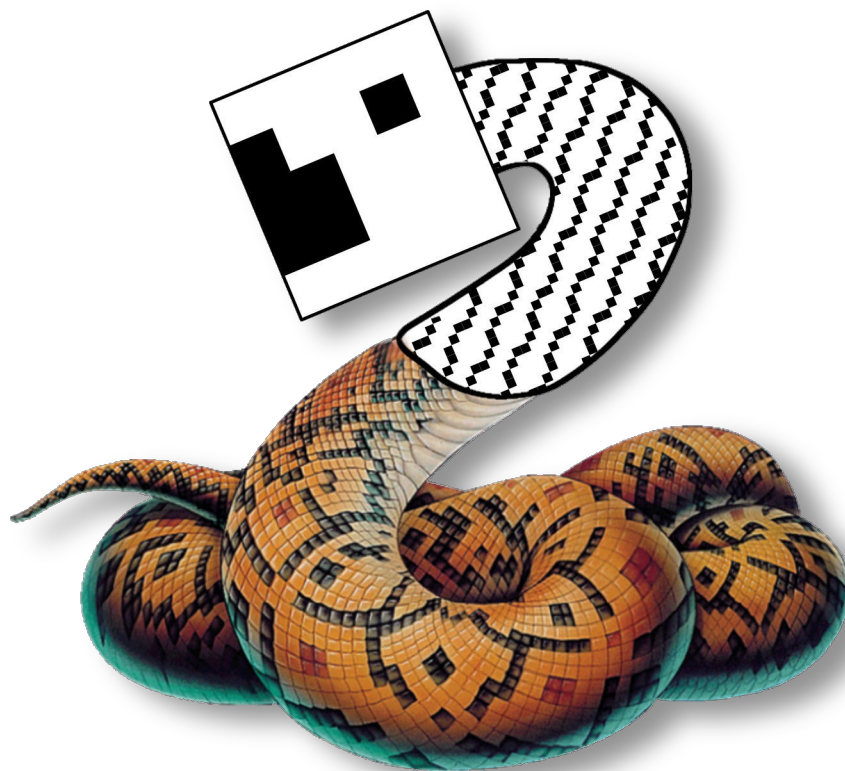


# Construcció d'una videoconsola i programació dels 25 "Snakes" que conté



2n Batxillerat C  
Curs acadèmic: 2017/2018

# ÍNDEX

1. INTRODUCCIÓ.....	pàg 1
2. EL JOC: EL PROCÉS CREATIU I SOFTWARE.....	pàg 5
2.1. Abans de començar: codi de plataforma i motor gràfic.....	pàg 5
2.2. Els modes de joc.....	pàg 6
2.2.1. El mètode creatiu “Hidra”.....	pàg 7
2.2.1.1. Passos a seguir.....	pàg 8
2.2.1.2. Usos.....	pàg 9
2.2.1.3. Pros i contres.....	pàg 10
2.2.2. L’ “Hidra” aplicat a l’ <i>Snake</i> : Els 25 modes resultants.....	pàg 12
2.2.2.1. Estudi de les regles i possibles anti-regles del joc.....	pàg 12
2.2.2.2. Llistat de modes finals i regles de les quals sorgeixen.....	pàg 16
2.2.2.3. Reptes més destacats a l’hora de programar-los.....	pàg 25
2.3. Menús i estructura general.....	pàg 26
2.4. Per acabar: Debugging i optimització.....	pàg 30
3. LA CONSOLA.....	pàg 33
3.1. Fases de prototipatge.....	pàg 33
3.1.1. Xip FT232H. Comunicació PC-pantalla.....	pàg 33
3.1.1.1. Components del muntatge.....	pàg 34
3.1.1.2. Esquema de connexions.....	pàg 38
3.1.2. Placa SAMD21 Mini. Consola autònoma.....	pàg 39
3.1.2.1. Components del muntatge.....	pàg 40
3.1.2.2. Esquema de connexions.....	pàg 43
4. CONCLUSIONS.....	pàg 45
5. BIBLIOGRAFIA I WEBGRAFIA.....	pàg 47

# 1. INTRODUCCIÓ

Aquest treball gira entorn del procés de fabricació d'una videoconsola que he muntat jo, a més de la programació dels jocs que conté, també de collita pròpia.

## JUSTIFICACIÓ

La intenció de centrar aquest treball al voltant d'un projecte d'aquestes característiques va sorgir primer de la intenció de fer un treball de recerca més aviat pràctic, del qual en pogués treure un resultat final físic i que em pogués resultar útil fins i tot després d'haver acabat el treball, i també de l'interès i coneixements que ja tenia sobre els videojocs, la programació i l'electrònica des d'abans d'haver de prendre cap decisió sobre el treball. Va ser gràcies al suport i guiatge puntual d'algú extern a l'Institut, del meu entorn familiar, qui em va fer interessar pel món de la programació per primera vegada ara fa uns anys, que em vaig veure capaç de realitzar un treball com aquest.

## OBJECTIUS

Així doncs, els objectius que em vaig proposar assolir durant aquest treball al començar es resumeixen en:

- Aprendre a dissenyar i muntar una videoconsola portable des de 0, component a component.
- Construir una carcassa per a la videoconsola que en contingui totes les parts.
- Programar els 3 o 4 jocs que haurien d'omplir la màquina.

## PROCÉS

Com acabo d'exposar, en un principi la meua primera idea va ser la de programar diversos jocs per a la consola, 3 o 4, per donar al jugador més d'una opció a triar. Així, vaig decidir arbitràriament que el primer que desenvoluparia seria un clon del clàssic joc *SNAKE*, ja que em va semblar un joc prou fàcil i distret de programar per començar.

En l'*Snake*, o "serp" en català, un joc que aparegué per primer cop el 1976, el jugador controla un ésser lineal llarg i prim, que a vegades s'anomena "serp" o "cuc", que s'arrossega dins d'un espai restringit, recollint aliments o algun altre element, intentant evitar xocar contra la seva pròpia cua o contra les parets que delimiten l'àrea de joc. Cada vegada que l'animal es menja un objectiu, en torna a aparèixer un altre a un altre punt aleatori del mapa, la cua de la serp s'allarga una mica, i això fa que esdevingui més difícil de maniobrar i, per tant, augmenta la dificultat del joc.

Captura de vídeo: <https://goo.gl/bY3EBC>



*Llegenda de la meua interpretació del joc Snake*

A mesura que vaig anar dedicant-hi temps, però, se'm començaren a acudir diverses versions alternatives del joc que em semblaren d'allò més interessants d'estudiar. Eren totes idees molt simples, que en canviaven ben poc les mecàniques (les regles segons les quals funciona un joc) però en canvi aconseguien un gran impacte en la seva jugabilitat. Així doncs, vaig abandonar la idea de desenvolupar diversos jocs diferents en pro de poder posar en pràctica totes aquelles idees amb més llibertat i provar de fer-hi tots aquests canvis que em venien al cap. Com que l'*Snake* va passar a ser l'únic joc que tindria la consola, això significava que l'hauria de convertir en un joc molt més complet i extens del que normalment és perquè fos capaç, per si sol, d'omplir la videoconsola i fer-la prou entretinguda de jugar-hi. Mode a mode, variant a variant, vaig anar ampliant-lo i, sense adonar-me'n llavors, vaig anant desenvolupant una sèrie d'estratègies per fer més àgil aquest procés de modificació del joc original, que es van acabar concretant en el mètode creatiu que més tard vaig anomenar "**Hidra**", que consisteix en:

- Fer-se ben conscient del conjunt sencer de regles i normes que regeixen un joc per, llavors, provar sistemàticament de trencar-les i pervertir-les una a una del major nombre de maneres possible. De l'agrupació d'aquestes noves *anti-regles*, en sorgeixen nous *anti-jocs*. Per això li vaig posar "**Hidra**", com el monstre mitològic, ja que de cada cap/regla que se li talla al joc, n'apareixen nous caps/anti-regles.

D'aquesta manera, quan em vaig adonar d'aquesta pauta que anava seguint sense haver-m'ho plantejat d'entrada, vaig fer la llista de totes les regles que se m'acudiren segons les quals el joc de l'*Snake* quedava definit, que foren unes 30, i de la contradicció de cadascuna d'aquestes en sortiren uns 50 modes alternatius a la versió original d'aquest. Les "anti-regles" dels nous modes poden sorgir o bé de la negació absoluta de l'enunciat primer o, en canvi, de modificacions d'aquest més treballades que acabin per fer-lo fals igualment. Que de la modificació de 30 normes inicials en sortissin fins a 50 variacions fou possible perquè sovint una mateixa regla es pot veure vulnerada de més d'una manera diferent. D'aquests 50 nous modes vaig escollir-ne els 25 que més em varen cridar l'atenció ja fos per la seva originalitat, complexitat, simplicitat, o repte a l'hora de programar-los, i d'aquí sorgí la llista final dels modes de joc que l'*Snake* de la meua videoconsola conté.

Així, alguns exemples de modes sorgits del procés creatiu que vaig idear durant aquest treball són:

- Un de més simple (negació absoluta de l'enunciat original) (mode 16: **Mòbils**):

L'objectiu/rata és estàtic, **no es mou** de lloc ➡ La rata **es mou** i prova d'escapar-se de la serp

- Un de més elaborat (alteració parcial de l'enunciat original) (mode 22: **Loops**):

La serp fa punt al **tocar amb el cap** la rata ➡ La serp ara fa punt a l'**envoltar-la amb el seu cos**

Aquests 25 modes els vaig acabar classificant en 7 tipus diferents segons el tipus de normes de les quals partien: regles relacionades amb la gestió de l'*input* i els **CONTROLS** del joc, les relatives a les restriccions espacials i **LÍMITS** del joc, al comportament i capacitats de la **SERP**, les corresponents a la naturalesa de l'objectiu de la serp o **RATA**, les que es refereixen a l'aspecte **GRÀFIC** i informació mostrada per pantalla del joc, les regles que afecten l'**OBJECTIU** del jugador i, en el setè tipus, "**ALTRES**".

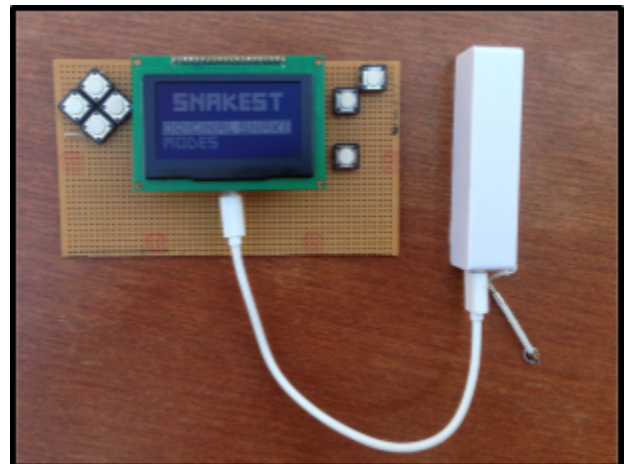
Un cop fets tots, va ser el moment de pensar en el disseny de la consola que hauria de contenir-los.

Per decidir quins materials comprar per cada versió i prototip em vaig guiar sempre per les necessitats tècniques que vaig anar tenint a cada moment, òbviament, però també vaig haver de donar una gran importància al preu econòmic dels diferents components per tal que no se'm disparés gaire el pressupost per al treball. Alguns dels components més destacats que vaig haver d'adquirir van ser la pantalla *OLED* de la consola, el xip *FT232H*, els botons o polsadors, i el microcontrolador final *SAMD21 Mini*, que tots plegats em van permetre construir els dos prototips diferents que he pogut fer de la meua consola.

La primera de les versions consistia en només la pantalla *OLED* connectada a l'ordinador a través del xip *FT232H*, que li permetia a la primera mostrar el joc que en realitat processava el segon (que també era el que rebia l'*input* del jugador a través del teclat), i en canvi el segon prototip, i el que presento junt amb aquest treball escrit, ja funciona del tot autònomament gràcies a la placa *SAMD21 Mini* i els botons que porta soldats. Aquesta segona versió només necessita estar endollada a un corrent de 5mA per funcionar, i aquest corrent el pot treure tant d'un endoll domèstic mitjançant un carregador adaptador de *miniUSB*, com d'una bateria de mòbil portable, per exemple.



*Pack ordinador+FT232H+pantalla*



*Consola engegada i funcionant*

Tot i que m'hauria agradat seguir treballant en un tercer prototip de la consola i millorar-la, no em va ser possible fer-ho a temps i em queda per més endavant, com a projecte personal fora d'aquest treball.

Seguidament, explicaré tot això amb més profunditat.

## 2. EL JOC: EL PROCÉS CREATIU I SOFTWARE

### 2.1. Abans de començar: codi de plataforma i motor gràfic

Fa anys, cap als 70, comunicar-se amb la pantalla d'un ordinador per tal d'ordenar-li explícitament pintar certs píxels d'una pantalla de cert color era una tasca molt més senzilla de dur a terme del que ara és. Llavors n'hi havia prou d'entrar al "mode gràfic" de la computadora i, mitjançant només dues ordres ben bàsiques, ja estaves llest per poder pintar la pantalla com més et plagués. Aquestes comandes eren:

- **PEEK**: en català "mirar", "espiar", o "donar una ullada", aquesta ordre et deia el color actual del píxel en concret que li haguessis especificat a l'hora de cridar-la.
- **POKE**: en català "tocar", "punxar", o "donar un copet", aquesta ordre et permetia pintar d'un color determinat el píxel en concret que li haguessis especificat a l'hora de cridar-la.



*Exemple de programa en Basic fent servir les comandes PEEK i POKE*

Avui en dia, aquest mateix procés de comunicar-te amb la pantalla del teu ordinador de forma tan directa per donar-li ordres del mateix estil, és molt més costós i difícil. Per aquest motiu, em vaig haver de valdre de l'ajuda d'algué extern que em pogués ajudar per aconseguir-ho. D'aquesta manera, la mateixa persona que em va fer interessar per tot aquest tema de la programació, ja fa uns anys, em va haver de donar un cop de mà una vegada més per tal de facilitar-me ser capaç de començar a programar els gràfics del meu joc i la meua consola. Així, gràcies a la seva contribució i única part del codi que no és meua, l'anomenat "codi de plataforma", que és el que em permet a mi com a programador comunicar-me amb el Sistema Operatiu de forma que sigui ell el que llavors doni les ordres a la pantalla, vaig poder per fi començar a treballar pel meu compte en aquest projecte.

Aquesta primera versió tan primitiva del codi amb què vaig començar a treballar sí que entenia els conceptes de *píxel* (punt al pla) i *bitmap* (pla on es dibuixen els punts), però no els de *sprite* (conjunt de píxels que junts formen una imatge més complexa). Per a cada *frame* de cada segon se li havia d'especificar a la màquina de quin color pintar cada píxel i, per tant, la primera tasca a dur a terme fou la de preparar un *motor gràfic* que fos capaç de saber moure tot un conjunt ordenat de píxels a la vegada. Això és: programar-me un seguit de recursos que em facilitessin a mi, el programador, el disseny i gestió de l'apartat gràfic del meu joc, que seria l'encarregat de transmetre al jugador tota la informació necessària del que està passant a cada moment al joc per fer-li possible jugar.

Així, la primera funció que vaig programar va ser ***ps()***, de “Pintar *Sprite*”, que em permetria poder pintar per pantalla un conjunt ordenat i compacte de píxels que es mourien com a bloc, per poder representar per exemple la serp protagonista i la rata que ha de perseguir. La segona en canvi fou ***ferforma()***, que em permetia mostrar per pantalla formes rectangulars sòlides d'un sol color, que vaig fer servir per pintar els menús i els marges del joc.

Amb aquestes dues funcions per començar, ja em vaig veure capaç d'aventurar-me a programar el joc que fos, després, és clar, d'haver d'incloure al codi de plataforma inicial el nou concepte d'*input* (dades que introdueix el jugador a la màquina, que es processen, i produeixen un *output*) per tal de poder captar i processar les accions del jugador quan premés o deixés de prémer segons quin botó. Per això em vaig tornar a deixar en mans del mateix programador professional d'abans. Un cop ja vaig ser capaç, doncs, de rebre un *input* del jugador, en forma de controls i botons premuts, i d'emetre un *output* de tornada, en forma de formes i figures pintades per pantalla, va ser el moment de programar la lògica del joc que hauria de fer d'intermediària entre tots dos successos, i ho vaig fer en el llenguatge de programació C. Abans, però, em va tocar reflexionar sobre allò que realment voldria que el meu joc fos, o fossin, en el cas que decidís fer-ne més d'un.

## 2.2. Els modes de joc

Tal com ja he esmentat prèviament en la introducció, els jocs que abans se'm van acudir per tal d'incloure'ls a la meua consola van ser tots d'un mateix tipus i gènere, el *retro*, que és el nom que se'ls dona al conjunt de primers jocs que es van inventar durant els primers 20 o 30 anys de la indústria. Pertanyerien a aquest grup jocs com el *Pong* (1972), *Snake* (1976), *Space Invaders* (1978), *Pac-Man* (1980), *Donkey Kong* (1981), *Tetris* (1984), *The Legend of Zelda* (1986), *Mega Man* (1987)...



Vaig pensar en tots aquests jocs perquè no desentonarien amb el to i estil de la meua consola, de resolució limitada i paleta de colors mínima (en blanc i negre), i la meua primera intenció va ser programar-ne diversos per donar més d'una opció a triar al jugador, ja que em va semblar que tindria prou temps per desenvolupar-ne 3 o 4, si més no.

Tot i això, el primer amb què vaig començar va ser l'*Snake* (ja descrit a la introducció del treball), que vaig escollir arbitràriament com hagués pogut escollir qualsevol altre joc dels anteriors esmentats. A l'enllestir-ne una primera versió, i abans de posar-me amb el següent joc de la llista, que probablement hauria estat el *Tetris*, se'm començaren a ocórrer tot d'idees i canvis a fer al joc per la curiositat de veure com afectarien totes aquestes variacions a la jugabilitat del joc resultant. Disposat a provar de programar totes aquestes modificacions i incloure-les al joc, per posar-me a prova a mi a més de comprovar-ne l'efecte, vaig abandonar la idea de treballar en altres jocs en pro d'esplaiar-me al màxim en aquest, el de l'*Snake*. Per tal que se m'acudissin el major nombre possible de versions alternatives del joc per tal de fer-lo el màxim de complet que pogués, i després d'haver-ne pensat les primeres pel meu compte una a una, vaig idear un mètode creatiu que m'hi ajudés i pogués així agilitzar el procés d'obtenció d'idees. És el descrit a continuació.

### 2.2.1. El mètode creatiu "Hidra"

Com ja s'ha mencionat anteriorment en el treball, la troballa d'aquest mètode fou casual, col·lateral al projecte principal que llavors s'estava duent a terme, però quan en vaig prendre consciència i em vaig adonar del que de fet estava fent per trobar noves idees per a fer nous modes, em va semblar prou interessant d'explorar com per incloure'l en aquest treball i estudiar-lo en les seves pàgines per intentar esbrinar si realment es tractava de quelcom realment rellevant o no.

Mentre pensava en noves versions de l'*Snake* per afegir al llavors tan reduït llistat de modes que tenia per incloure a la consola, em vaig adonar del patró que realment s'estava seguint a l'hora de fer-ho: buscava les regles i normes segons les quals el joc original quedava definit i, una a una, a aquestes les intentava pervertir del major nombre de maneres que se'm poguessin acudir per crear, d'aquest procés de transgressió sistemàtica, nous jocs que d'alguna forma podríem anomenar *anti-Snakes*. Davant d'aquest descobriment imprevist, vaig provar de parar més atenció a com s'estava duent a terme aquell procés de cerca d'idees i, així, en va acabar sortint l' "*Hidra*".

L'Hidra, el monstre mitològic, era una enorme **serp** de set caps que quan se li tallava un d'aquests n'hi naixien dos o més. Igualment, en aquest mètode creatiu, de la negació o variació de cada norma o regla que regeix un joc, en sorgeixen més de noves, les *anti-regles*.

### 2.2.1.1. Passos a seguir

1. **Posar-se restriccions o ser ben conscient i estar agraït de les que ja es té.** Encara que pugui semblar contraproductiu posar-se limitacions innecessàries ja des del començament a l'hora de crear quelcom des de zero, aquest pas és beneficiós en gairebé qualsevol procés creatiu al donar al compositor certes barreres dins les quals es pugui sentir més còmode i menys desbordat de tantes possibles idees, algunes potser bones, d'altres de dolentes, del que es sentiria si se li "pogués" ocórrer absolutament qualsevol cosa. Aquestes limitacions poden referir-se tant al com vols que sigui el mitjà o material o entorn en el qual existirà el producte final, és a dir el "continent", o en com vols que el producte en si sigui o no (tema, estil o mètode a usar a l'hora d'idear-lo...), el "contingut".
2. **Enumerar totes les regles o pautes definitòries del producte** sobre el qual es vol innovar que se t'acudeixin en un full una darrere l'altra. Parant atenció als detalls més subtils així com també als més obvis, des dels més aparentment formals (com quina informació i elements són mostrats per pantalla) fins als més fonamentals a l'hora de descriure'l (les normes bàsiques del joc). Allargar aquest pas fins que et sembli que el joc que s'està explorant pot ser completament definit només mitjançant les normes que acabes de trobar.
3. Una a una, parar-se uns moments a **reflexionar com poden ser falsejades** o bé de forma absoluta, que és el més fàcil, contradient-la del tot, o bé modificant-la de forma més original de manera que l'enunciat primer també deixi de ser cert. Pot donar-se que una mateixa regla pugui ser alterada de diverses maneres i de fet això seria el més ideal, trobar el màxim d'alteracions possibles a fer de cada norma, i de vegades també que una sola alteració o canvi en les mecàniques del joc trenqui amb més d'una norma a la vegada.
4. **Agrupar les diverses normes**, ara alterades, en grupets que creus que podrien funcionar segons el teu gust i objectiu a l'hora d'idear el joc (ja sigui per exemple el de merament divertir, entretenir, o bé el de crear desafiaments que suposin un repte al jugador, ...), per tal de crear paquets de noves mecàniques que juntes conformin nous jocs que s'aguantin per si sols.

En el cas del meu *Snake* aquest pas va perdre pes per què em va interessar més el repte de programar el màxim nombre de modes possibles durant els mesos que em vaig donar de marge per fer-ho, que no pas procurar que tals modes fossin realment jugables i els més entretinguts i divertits de tots.

## 5. Escollir la/es millor/s combinació/ons de noves normes.

### 2.2.1.2. Usos del mètode

- Agafar idees per **innovar sobre quelcom** per a convertir-lo en un producte seminou o per a completar un joc amb altres modes alternatius a l'original.
- **Comprovar com de semblants són dos jocs.** Si te n'adones que els estudis de tots dos jocs són prou semblants amb l'única excepció d'un parell de regles que són modificacions unes de les altres, serà que un d'aquests jocs prové de l'altre, o que tots dos són versions d'un tercer joc encara més antic, el vertader original. També és possible que tot i que un major nombre de regles que els pautin siguin diferents, segueixis sent capaç d'entreveure el perquè en una sola norma que en variar comporta que tot un altre conjunt de regles també s'hagi d'adaptar per a mantenir el joc jugable d'acord a la nova norma modificada.
- **Comprovar com d'equilibrat és un joc, o fins a quin punt ja se li han explotat totes les variants possibles.** Si trobessis en realitzar l'estudi de les normes d'un joc que al variar-ne qualsevol aquest deixa de ser entretingut o suficientment divertit o complicat com per seguir sent considerat joc, o per altra banda, si trobessis que totes les vulneracions interessants d'aquestes ja han estat provades, podríem concloure que totes les possibles versions del joc han estat exhaurides, si és que mai n'hi ha hagut que poguessin valdre la pena en primer lloc.
- Tenir present aquesta manera de trobar noves idees per canviar un producte per provar de mantenir-lo viu (anar pervertint una a una les regles que el defineixen) pot ajudar-te a detectar usos d'aquesta forma d'innovar en marques del teu entorn, i buscar aquest tipus de patrons podria servir-te de **divertimento o entreteniment intel·lectual** si t'ho proposessis.
- Fer-se a un mateix tan **conscient dels límits i pautes que regeixen el teu joc** pot servir-te per provar de portar-los a les seves últimes conseqüències i veure com es comportarien davant de certes situacions hipotètiques extremes, que més tard podries decidir afegir a la jugabilitat del teu joc, si és que trobessis que resulten prou interessants d'explorar com per acabar fent-ho.

### 2.2.1.3. Pros i contres del mètode

#### a) Pros

- És dels pocs mètodes creatius **encarats principalment a l'acte d'innovar** sobre quelcom ja existent en comptes de crear quelcom nou. Això el diferencia de la resta i el fa la millor opció a escollir, o almenys una que faries bé a tenir en compte almenys, quan del que es tracta és precisament això; millorar, rejuvenir, complementar, ampliar o polir...
- **La seva manera de funcionar fa que perdi pes a l'hora de dur-lo a terme la imaginació i capacitat creativa de cadascú.** Al tractar-se d'un mètode tan pautat, "fred i calculador", en el qual l'esforç principal a realitzar és sols el d'observar i anotar les regles que fan de quelcom el que és, per després anar vulnerant-les una a una de forma sistemàtica de forma més o menys original, això pot, en efecte, ser considerat un benefici per totes aquelles persones que es sentiran més còmodes fent llistes que esperant que la inspiració un moment o altre els arribi.
- Saber d'aquest mètode i buscar-ne possibles exemples al teu entorn et pot servir de *divertimento* com ja s'ha mencionat més amunt, però el cert és que també et pot servir de molt més per **obrir la ment i ajudar-te** encara d'una altra forma **en el teu dia a dia.** Seguint-ne l'exemple i aplicant-te'l, o més aviat adaptant-te'l, a la vida quotidiana, pots aprendre a tractar amb certes situacions amb les quals et trobis d'una manera més racional i calmada, sent capaç d'identificar-ne tots els diferents detalls, motius i possibles solucions a cadascun d'aquests, de forma més senzilla i ordenada. Semblant al que proposa l'escola del *Computational Thinking* o "pensament computacional", que proposa que aprendre l'habilitat de programar pot ensenyar a l'individu a gestionar les problemàtiques amb les quals es trobi en la seva vida corrent d'una forma més pautada i organitzada i satisfactòria gràcies a aplicar-hi els coneixements que fer el primer li han atorgat.

## b) Contres

- **Coeficient d'innovació més aviat simbòlic.** El fet que s'hagi de partir sempre d'un quelcom ja existent per posar en marxa el mètode òbviament en condiciona el resultat de manera que faci molt difícil que el joc o producte que en resulti pugui ser considerat vertaderament original. Tot i això, això mateix també és una fortalesa del mateix al ser un dels pocs mètodes si no l'únic que existeix només encarat a precisament *completar* o crear *versions alternatives* de quelcom.
- **Mida del que s'està estudiant.** No és possible analitzar d'aquesta manera un conjunt de normes o mecàniques diferents entre elles que funcionin per separat en entorns apartats encara que formin part d'un mateix tot, ja sigui un joc o gamma de productes. Així doncs, seria possible analitzar un nivell o fase de qualsevol joc, però potser no el joc sencer, en el cas de jocs més complexos i moderns que no pas l'*Snake*, per exemple.
- Tot i que quan vaig ensopegar amb el mètode per primera vegada em va semblar molt clar que era quelcom que funcionaria només en el mitjà dels jocs i videojocs, una altra part de mi també opinava que potser serviria en absolutament tot, i no només en la resta d'arts, sinó fins i tot també a l'hora d'idear objectes físics com eines o mobiliari. Després de donar-hi un parell més de voltes, però, aquesta idea exagerada de què era quelcom aplicable a tot es veié ràpidament descartada a l'adonar-me que **per qualsevol cosa que hagi de respondre a quelcom més que no sigui un seguit de normes o pautes merament lògiques**, com precisament els jocs, ja sigui un sentit estètic o intenció expressiva o una certa utilitat requerida, **no és adequat executar-hi aquesta mena d'estudi**. Així, ens trobem que els jocs que es troben més a la línia de poder ser considerats experiències prou encarades a la narració o a la divulgació d'un missatge en concret i no tant a la diversió o a la intenció de ser entretinguts com per ser anomenats com a tals, jocs, són també els més difícils d'estudiar mitjançant aquest mètode, cosa que d'una manera o altra acaba per reforçar aquestes conclusions.
- **Els usos que se'n facin o pels quals es vulgui fer servir.** Si fas servir qualsevol eina per una feina per la qual no és l'adequada, el més probable és que no acabis gens satisfet del resultat que et procuri. Òbviament s'han de tenir en compte, doncs, les situacions per les quals és més útil, aquelles per les quals no et podrà servir de res, i per les quals potser seria millor optar per algun altre mètode diferent.

### 2.2.2. L'“Hidra” aplicat a l'Snake: Els 25 modes resultants

Així, quan després de pensar en 4 o 5 versions alternatives de l'Snake per mi sol, vaig adonar-me del mètode que en realitat estava fent servir, vaig voler parar-hi més atenció i, un cop identificats els seus passos, vaig provar de posar-lo en pràctica amb l'Snake per trobar la resta de modes:

#### 2.2.2.1. Estudi de les regles i possibles anti-regles del joc

##### a) Restriccions amb què treballar

Les limitacions que em vaig voler autoimposar per tal d'acotar la meua creativitat dins d'un espai més reduït en què m'hi pogués sentir còmode van ser de dos tipus i les següents:

Tècniques:

- El joc hauria de ser **jugable des de la consola** portàtil que jo em fabricaria.
- ... la pantalla de la qual seria de 128x64 píxels **(2:1)** i **monocroma**.
- El **pressupost** que em podria gastar en aquest projecte també en seria un de limitat.

Humanes:

- El joc **l'hauria de poder programar jo** i, per tant, no podria ser més complicat de dur a terme del que jo en fos capaç.
- L'hauria de desenvolupar durant el **temps** límit que tindria per fer aquest treball.

Totes aquestes limitacions van acabar traduint-se en el fet que el joc que triés hauria de ser-ne un de prou **simple** com perquè: pogués funcionar en un processador limitat com el del **microcontrolador que portaria inclosa la consola**, es pogués entendre en **blanc i negre**, el sabés **programar jo (2D segur)** i fos assequible de realitzar en un **període de temps reduït**, d'uns quants mesos.

##### b) Regles definitòries de l'Snake (i anti-regles)

De normes que regissin el joc original en vaig trobar fins a 30 i fruit de la seva alteració en sorgiren unes 50 variants. Hauria estat possible seguir duent a terme aquest procés d'observació i contradicció sistemàtica fins a aconseguir-ne encara més, però vaig acabar concloent llavors que ja hi havia destinat suficient temps en aquesta etapa del treball, i que ja seria hora de seguir amb la següent, així que vaig abandonar i començar a decidir quines de les diferents idees em feien més el pes o em semblaven més interessants de programar, d'entre totes les que havia trobat:

**1 - El joc es controla mitjançant l'input digital que emet el jugador (botons, no joysticks):**

- Mitjançant controls analògics permetre-li girar i desplaçar-se més lliurement per l'espai.

**2 - L'input que emet el jugador s'interpreta relatiu a la seva perspectiva (amunt és ↑, ...):**

- Només poder girar cap a l'esquerra i la dreta relativament a la direcció actual.
- La direcció en què es mou la serp és la invertida de la qual s'emet (amunt és ↓, ...) .
- La distribució de les direccions canvia de botó cada vegada que es premen.

**3 - L'input es rep i processa de forma simultània, la serp s'hi dirigeix immediatament:**

- Moure amb l'input un punter que indiqui on anar a la serp, i no a la serp en si.

**4 - Tu tries quan girar i en quin sentit fer-ho:**

- Girs aleatoris espontanis en contra de la voluntat del jugador.
- La distribució de les direccions canvia de botó cada vegada que es premen.

**5 - Restriccions espacials: 4 parets.**

- Poder passar d'una paret a una altra.
- Poder dissenyar parets i obstacles de més dins de la pantalla de joc.

**6 - Restriccions espacials: el teu cos.**

- Poder-se travessar a un mateix.

**7 - Continuïtat física de l'espai:**

- Poder deixar "caus" repartits pel mapa: entres per un i surts per l'altre.
- L'espai en el qual es desenvolupa el joc és major al que es pot veure de cop.
- Límits espacials (parets) mòbils, canviants.

**8 - Només hi ha una serp a la pantalla. Controles 1 serp:**

- Controlar simultàniament a més individus que es mouen tots a la vegada.

**9 - De la serp en dirigeixes el cap:**

- Poder canviar de dirigir el cap/dirigir la cua a voluntat.

**10 - El moviment de la serp és constant, mai para:**

- Desplaçar-se casella per casella.

**11 - La velocitat de la serp és constant, no varia:**

- La velocitat de la serp va augmentant progressivament a mesura que passa el temps.
- Fases de *turbo*, "acceleració puntual", depenent de la ratxa de punts que un porta.

**12 - La longitud de la serp es manté constant si no es menja:**

- Si no menges vas encongint-te fins a desaparèixer, i es conta el temps que triga a passar.
- Si no menges vas creixent fins a ofegar-te, i es conta el temps que triga a passar.
- Poder regular la longitud de la serp a mà.

**13 - La serp mai decreix, només mor:**

- Al xocar només perds cos o puntuació, però pots seguir jugant (el jugador té més “vides”).

**14 - Són 4 les direccions cap a les quals es pot moure la serp, les cardinals nord, sud, est i oest:**

- Girar 45° les 4 direccions per tal de jugar en diagonal.
- El jugador, a més, també pot moure's en diagonal i girar en angles múltiples de 45°.
- Mitjançant controls analògics permetre-li girar i desplaçar-se més lliurement per l'espai.

**15 - Continuïtat física de la serp:**

- Poder expulsar «parts del cos» de la serp per tal d'obtenir punts a distància.

**16 - La posició i direcció inicial des de la qual parteix la serp sempre és la mateixa:**

- Començar la partida sempre des d'un punt del mapa diferent i aleatori.

**17 - Menjar-se una rata o objectiu sempre té el mateix resultat, dóna puntuació i llargada:**

- Rates especials canvien les mecàniques de joc durant un temps limitat (*powerups*).
- Rates trampa que es confonen amb les normals, i maten la serp.
- Rates de diferents rangs donen diferent puntuació.

**18 - Seqüència de rates: una a la vegada, i quan te la menges apareix la següent.**

- Més d'una rata disponible per menjar-se a la vegada.
- Rates disponibles només durant un temps, que poden canviar de lloc abans d'agafar-les.

**19 - Les rates poden aparèixer en qualsevol punt del mapa i sempre són accessibles:**

- Les rates o objectius poden aparèixer només en certs punts concrets del mapa.
- Rates disponibles només durant un temps, que poden canviar de lloc abans d'agafar-les.
- Menjables només en un ordre concret (si n'hi hagués més d'una).
- Menjables només venint des d'una direcció específica.
- Rates trampa que es confonen amb les normals, i maten la serp.

**20 - Hi ha un sol tipus de rata o objectiu. És estàtic i no es mou ni canvia en cap moment:**

- La rata es mou, i prova d'escapar-se de la serp.
- Rates especials canvien les mecàniques de joc durant un període limitat de temps.
- Rates trampa que es confonen amb les normals, i maten la serp.
- Rates de diferents rangs donen diferent puntuació.

**21 - La serp sencera és visible en tot moment:**

- Només les caselles més pròximes al cap de la serp són visibles al llarg de la partida.

**22 - L'objectiu o rata és visible en tot moment:**

- La rata és invisible, i la serp reacciona a com de prop és d'ella per orientar al jugador.



**23 - Objectiu: obtenir el màxim de rates i puntuació abans de xocar:**

- Prestar més atenció a la divisió puntuació/[moviments o temps], premiar l'eficiència.
- Prestar més atenció al recorregut que s'ha de recórrer fins a l'objectiu:
  - Buscar omplir el màxim de caselles abans de xocar o arribar a l'objectiu.
  - Jugar en un circuit de reflexos o cursa.

**24 - Has de tocar la rata amb el cap per agafar-la i fer punt:**

- Haver d'empènyer-la fins a algun lloc per puntuar, per exemple al "cau".
- Haver d'envoltar-la amb el cos per puntuar, arraconar-la.

**25 - Controles la serp:**

- Controlar la rata, haver-se d'escapar d'una IA que et persegueix mentre fas punts.

**26 - Es tracta d'un joc en 2D:**

- Jugar-lo vist en primera persona des de la perspectiva de la serp, en 3D.
- En "1D": la serp es mou només amunt i avall, i els objectius són els que s'acosten a tu.

**27 - És un joc *singleplayer*, és a dir, d'un sol jugador:**

- Convertir-lo en multijugador competitiu:
  - Competeixen pels punts i es poden matar als contrincants barrant-els-hi el pas.
  - Un jugador controla la serp i els altres els objectius, que s'escapen.
- Convertir-lo en multijugador col·laboratiu:
  - Només un jugador veu la rata, un altre veu la serp, i un tercer la controla.

**28 - El moviment de la serp és pautat i quadriculat (divisió de l'espai en blocs de 5x5 píxels):**

- La serp es mou de forma més suau píxel a píxel i pot girar en qualsevol moment.
- Mitjançant controls analògics permetre-li girar i desplaçar-se més lliurement per l'espai.
- Canviar la forma en les que s'estructura l'espai a polígons amb més o menys cares.

**29 - La dificultat del joc va augmentant segons l'ambició i accions del jugador:**

- La velocitat va en augment progressivament.
- La dificultat va disminuint segons l'ambició i accions del jugador, recompensant-lo.
- Enemics mòbils encarregats d'entorpir el moviment de la serp cada tants segons.

**30 - Vista zenital de l'acció:**

- Vist des de baix, les direccions s'interpreten invertides.
- Jugar-lo vist en primera persona des de la perspectiva de la serp, en 3D.
- "De perfil": la serp es mou només amunt i avall, i els objectius són els que s'acosten a tu.

**31 - Les unitats de cos noves s'adhereixen a la cua de l'animal:**

- Augmentar en amplada en comptes de longitud.

### **c) Empaquetament i elecció de les millors variacions**

En el cas del meu *Snake* aquest pas va perdre importància perquè vaig prioritzar un nombre elevat de modes i reptes a programar, per tal de posar-me a prova, que no pas el fet que els escollits fossin els millors o més jugables. En aquesta decisió també hi jugà el factor temps, i és que tampoc volia allargar-me massa en el procés de selecció dels modes, i posar-me com abans millor a la seva programació. D'aquesta manera, tot i guiar-me pels que més m'agradessin i veiés més viables i entretinguts de fer, la seva elecció en última instància fou força arbitrària. Així doncs, el criteri principal en l'agrupament d'anti-regles per conformar les diferents noves versions de l'*Snake* quedà reduït només a ajuntar aquelles que semblessin dependre obligatòriament unes de les altres, en comptes de mirar de buscar quines es podien combinar de la manera més original i funcionar millor plegades.

#### **2.2.2.2. Llistat de modes finals i regles de les quals sorgeixen**

El llistat de modes finals, classificats en 7 tipus depenent de la classe de normes la negació de les quals parteixen, és el següent:

#### **a) CONTROLS**

##### **Invertit:**

Norma que es vulnera:

- L'*input* que emet el jugador s'interpreta relatiu a la seva perspectiva.

Norma reformada:

- La direcció en què es mou la serp és la invertida de la qual s'emet.

##### **Egocèntric:**

Norma que es vulnera:

- L'*input* que emet el jugador s'interpreta relatiu a la seva perspectiva.

Norma reformada:

- Només poder girar cap a l'esquerra i la dreta relativament a la direcció actual.

##### **Estratègic:**

Norma que es vulnera:

- L'*input* es rep i processa de forma simultània, la serp s'hi dirigeix immediatament.

Norma reformada:

- Moure amb l'*input* un punter que indiqui on anar a la serp, i no a la serp en si.

Captura de vídeo: <https://goo.gl/UgtL7m>

## b) LÍMITS:

### Vores:

Normes que es vulneren:

- Restriccions espacials: 4 parets. Si toques qualsevol de les parets que limiten el terreny de joc s'acaba la partida.
- Continuitat física de la serp. Cada "unitat de cos" que la forma està immediatament unida a l'anterior i a la següent.

Normes reformades:

- Al dirigir-te a qualsevol de les quatre vores el cap de la serp reapareix per la paret contrària, i eventualment la segueixen posteriorment tots el "cossos" que la formen.

Captura de vídeo: <https://goo.gl/To7Ts9>

### Fantasma:

Norma que es vulnera:

- Restriccions espacials: el teu cos. Si xoques contra el teu propi cos s'acaba la partida.

Norma reformada:

- Et pots creuar i travessar a tu mateix, però per seguir-li donant un cert sentit i dificultat al joc no pots trepitjar-te més d'una casella seguida.

Captura de vídeo: <https://goo.gl/qUaEL4>

### Editor:

Normes que es vulneren:

- Límits espacials: 4 parets (limiten un terreny de joc rectangular). Si toques qualsevol de les parets que limiten el terreny de joc s'acaba la partida.
- La posició i direcció inicial des de la qual parteix la serp sempre és la mateixa.
- Les rates poden aparèixer en qualsevol punt del mapa i sempre són accessibles.

Normes reformades:

- El jugador pot **modificar** el terreny de joc a voluntat abans de jugar-hi de manera que pot posar un major nombre de **parets i obstacles** dins del terreny de joc modelant així l'espai en formes diferents, així com determinar la **posició i direcció inicial** que prendrà la serp al començar la partida, i **designar caselles les quals no es tindran en compte a l'hora de buscar un lloc on fer aparèixer el pròxim objectiu** o rata si és que li interessés que en aquests punts no en poguessin sortir.

Captura de vídeo: <https://goo.gl/uKWMGE>

### **c) SERP:**

#### **Regulable:**

Normes que es vulneren:

- La longitud de la serp es manté constant si no es menja.
- La serp mai decreix, només mor.
- La dificultat del joc va augmentant segons l'ambició i accions del jugador. A mesura que fa més i més punts cada cop la serp es fa més llarga i es fa més difícil de maniobrar.

Normes reformades:

- Tot i que la puntuació del jugador segueix depenent del nombre d'objectius que va menjant-se, el control sobre la llargada de l'animal el té el jugador que pot decidir arbitràriament de fer-la créixer o decreixer a voluntat.

Captura de vídeo: <https://goo.gl/8L5w6G>

#### **Manual:**

Norma que es vulnera:

- El moviment de la serp és constant, mai para.
- La velocitat de la serp és constant, no varia.

Normes reformades:

- Desplaçar-se casella per casella. Moviment puntual només quan es rep l'input concret.

Captura de vídeo: <https://goo.gl/jMQeU6>

#### **Accelerant:**

Normes que es vulneren:

- La velocitat de la serp és constant, no varia.
- La dificultat del joc va augmentant segons l'ambició i accions del jugador. A mesura que fa més i més punts cada cop la serp es fa més llarga i es fa més difícil de maniobrar.

Normes reformades:

- La velocitat de la serp va augmentant progressivament a mesura que passa el temps. La dificultat a l'hora de conduir-la, per tant, també augmenta a mesura que es fa més ràpida, independentment dels punts que es facin.

Captura de vídeo: <https://goo.gl/sjM2za>

### **Capicua:**

Norma que es vulnera:

- De la serp en dirigeixes el cap.

Norma reformada:

- El jugador té la capacitat de commutar el cap i cua de la serp, alterant-ne així la direcció, a voluntat.

Captura de vídeo: <https://goo.gl/V4j1Yj>

### **Spitting:**

Norma que es vulnera:

- Continuïtat física de la serp. Cada “unitat de cos” que la forma està immediatament unida a l’anterior i a la següent.
- La serp mai decreix, només mor.

Normes reformades:

- El jugador té la capacitat d’expulsar parts del seu cos per la boca per obtenir puntuació a distància, fent-se més petit al desprendre’s d’elles, però recuperant el doble d’unitats de cos al fer punt.

Captura de vídeo: <https://goo.gl/P38zrz>

### **Diagonal:**

Norma que es vulnera:

- Són 4 les direccions cap a les quals es pot moure la serp, les cardinals N, S, E i O.

Norma reformada:

- El jugador, a més, també pot moure’s en diagonal i girar en angles múltiples de 45°.

Captura de vídeo: <https://goo.gl/DLf2Fp>

### **Smooth:**

Norma que es vulnera:

- El moviment de la serp és pautat i quadriculat. L’espai està dividit en seccions quadrades i el moviment de la serp a través seu és de casella en casella.

Norma reformada:

- La serp es mou de forma més suau píxel a píxel i pot girar en qualsevol moment.

Captura de vídeo: <https://goo.gl/MrD777>

## **d) RATA:**

### **Més rates:**

Normes que es vulneren:

- Menjar-se una rata o objectiu sempre té el mateix resultat, dóna puntuació i llargada.
- Seqüència de rates: una a la vegada, i quan te la menges apareix la següent.

Normes reformades:

- Existeixen “falsos objectius” que s’assemblen als veritables però que tocar-los significaria perdre la partida. El jugador ha de saber distingir-los per sumar punts i no morir. Fins a cinc objectius simultanis es troben disponibles arreu del terreny de joc, i els objectius es troben disponibles només durant un temps determinat i llavors reapareixen a un altre punt del mapa.

Captura de vídeo: <https://goo.gl/5crhMY>

### **Direccionals:**

Normes que es vulneren:

- Has de tocar la rata amb el cap per agafar-la i fer punt.
- Les rates poden aparèixer en qualsevol punt del mapa i sempre són accessibles.

Normes reformades:

- Els objectius només es poden aconseguir depenent de la direcció des de la qual s’hi arriba. Cada rata, cada vegada, es pot accedir des d’una sola direcció determinada. Menjar-se un objectiu des d’una direcció errònia significa perdre la partida.

Captura de vídeo: <https://goo.gl/RVNjyq>

### **Mòbils:**

Norma que es vulnera:

- L’objectiu o rata és estàtic i no es mou ni canvia en cap moment.

Norma reformada:

- La rata es mou a la mateixa velocitat que la serp, casella a casella, cap a qualsevol de les 4 direccions aleatòriament, de forma erràtica, escapant-se del jugador.

Captura de vídeo: <https://goo.gl/7zmXDb>

## **e) GRÀFICS:**

### **Survival:**

Norma que es vulnera:

- La serp sencera és visible en tot moment.

Norma reformada:

- Només les caselles més pròximes al cap de la serp són visibles al llarg de la partida.  
La resta de terreny de joc queda a les fosques quan s'hi està lluny.

Captura de vídeo: <https://goo.gl/KzyXTL>

### **Gallineta Cega:**

Normes que es vulneren:

- L'objectiu o rata és visible en tot moment.
- L'aspecte i naturalesa de la serp és tota l'estona igual.

Normes reformades:

- La rata és invisible durant la partida (només se la veu breument a l'aparèixer, per ajudar el jugador a trobar-la i fer el mode més jugable) i l'aparença de cada "unitat de cos" que conforma la serp depèn de la seva proximitat a l'objectiu invisible en qüestió. Com més a prop s'hi està, el cos és d'un color més clar, mentre que com més se n'allunya més fosc es fa.

Captura de vídeo: <https://goo.gl/vkgSgd>

## **f) OBJECTIU:**

### **Màximes caselles:**

Normes que es vulneren:

- Objectiu: obtenir el màxim de rates i puntuació abans de xocar.
- La longitud de la serp es manté constant si no es menja.

Normes reformades:

- L'objectiu del jugador passa a ser el d'arribar a ocupar el màxim nombre de caselles possible abans de xocar i perdre la partida. En aquest mode de joc la serp creix contínuament a cada passa que fa i no hi ha rates a perseguir.

Captura de vídeo: <https://goo.gl/3Ug1nd>

### **Selecció Natural:**

Norma que es vulnera:

- La dificultat del joc va augmentant segons l'ambició i accions del jugador. A mesura que fa més i més punts cada cop la serp es fa més llarga i es fa més difícil de maniobrar.
- La serp mai decreix, només mor.

Normes reformades:

- A mesura que passa el temps la serp va disminuint de mida. Si no es fan punts prou sovint, acaba per desaparèixer completament, i es perd la partida.

Captura de vídeo: <https://goo.gl/9rfRHy>

### **Bàsquet:**

Normes que es vulnera:

- Has de tocar la rata amb el cap per agafar-la i fer punt.
- Les rates poden aparèixer en qualsevol punt del mapa i sempre són accessibles.
- La serp mai decreix, només mor.
- Restriccions espacials: 4 parets i el teu cos.

Normes reformades:

- El jugador ha d'empènyer les rates fins a un punt determinat de la pantalla, el seu "cau", per tal de fer punt. Fer fora una rata del terreny de joc farà que n'aparegui una de nova al mapa a canvi de decreïxer una unitat de cos. Si la mida de la serp ja era mínima abans de treure una altra rata fora del terreny de joc, el jugador perd la partida. Més restriccions espacials: xocar contra el forat que fa de cau o empènyer una rata contra el cos del jugador també significa perdre la partida.

Captura de vídeo: <https://goo.gl/WshdCF>



## Loops:

Normes que es vulneren:

- Has de tocar la rata amb el cap per agafar-la i fer punt.
- La dificultat del joc va augmentant segons l'ambició i accions del jugador

Normes reformades:

- Haver d'envoltar-la amb el cos per puntuar, arraconar-la. El jugador ha d'encerclar amb el seu cos, tancar, l'objectiu, per tal d'anotar punt i fer que en reaparegui un de nou. Com més llarga es faci la serp, menys li costarà anar encerclant la rata.

Captura de vídeo: <https://goo.gl/W6NPv3>

## 1D:

Normes que es vulneren:

- Es tracta d'un joc en 2D (la serp es pot moure verticalment i horitzontal)
- Vista zenital de l'acció.

Normes reformada:

- "Passa a ser un joc 1D": el moviment de la serp es duu a terme en una sola dimensió, en l'eix de les y, mentre que les rates s'acosten a ell i no a l'inrevés.

Captura de vídeo: <https://goo.gl/NdtnZA>

## IA:

Normes que es vulneren:

- Controles la serp.
- El moviment del jugador és constant, mai para.
- L'objectiu o rata és estàtic i no es mou ni canvia en cap moment.

Normes reformades:

El jugador controla la rata i per tant se'n deriva que:

- La rata deixa de ser estàtica i el jugador en pren el control. Es pot moure i ho fa a una velocitat major a la de la serp. El seu moviment no ha de ser constant, sinó que respon a l'*input* que s'està rebent a cada moment. La serp el persegueix mentre intenta fer punts. Les limitacions espacials que en limiten el moviment són semblants: les vores i el cos de la serp. L'objectiu del jugador són unes "anti-rates" que pot obtenir trepitjant-les com ho faria de normal. Si el cap de la serp les toca, també s'anota el punt, i la serp s'allarga.

Captura de vídeo: <https://goo.gl/B4zuk3>

## g) ALTRES:

### Retro:

“Norma que es vulnera:”

- L'*Snake* tal com l'he reproduït jo, sencer.

“Normes reformades:”

- ÉS UN HOMENATGE A LA VERSIÓ MÉS CLÀSSICA DEL JOC *SNAKE*:
  - Aspecte gràfic més simple, *retro*.
  - Divisió de l'espai en més seccions més petites.
  - Serp més menuda i veloç.

Captura de vídeo: <https://goo.gl/7CxP64>

### Multijugador:

Norma que es vulnera:

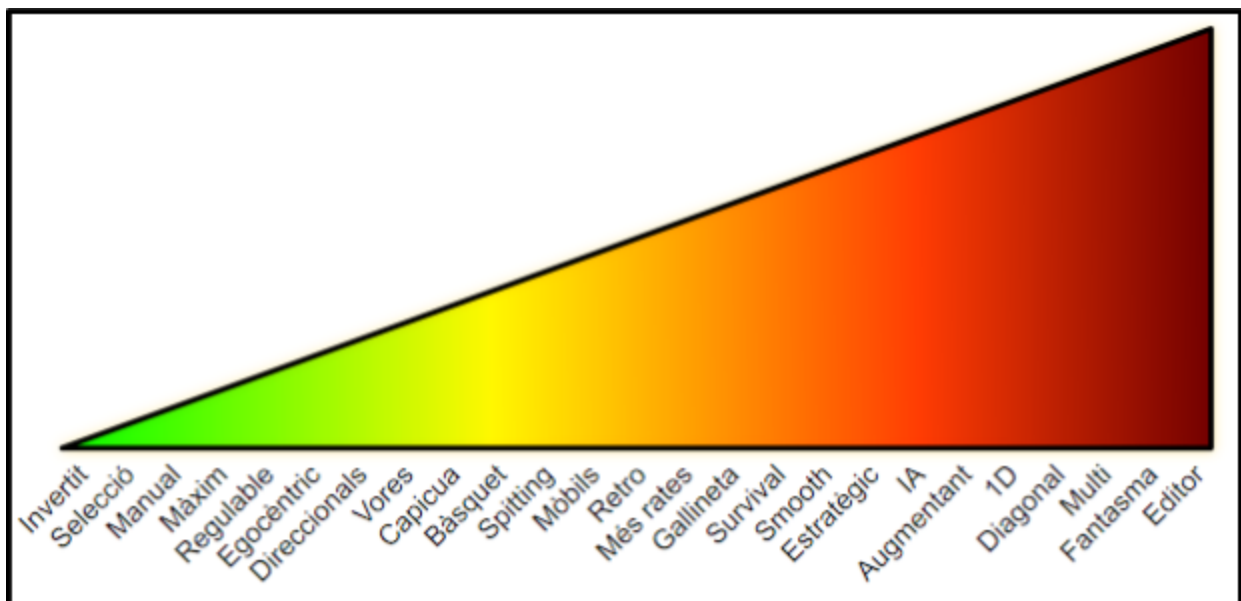
- És un joc *singleplayer*, és a dir, d'un sol jugador.
- Només hi ha una serp a la pantalla.

Norma reformada:

- Dos o més jugadors controlen cadascun una serp dins del mateix terreny de joc i competeixen per fer xocar els altres entre ells o contra les parets del mapa. Per aconseguir-ho podrien optar per intentar anotar el màxim nombre de punts possibles i així ser més gran que la resta, i influenciar així els seus moviments de manera més evident. Tot i haver-lo programat, aquest mode s'ha quedat fora de la consola al no tenir prou botons per fer-lo viable allí, i en canvi sí a l'ordinador.

### 2.2.2.3. Reptes més destacats a l'hora de programar-los

Tot i que la majoria de modes vaig ser capaç de programar-los amb relativa facilitat i rapidesa, n'hi va haver un grapat també prou nombrós que em va costar més de dur a terme del que m'hauria esperat i fins i tot en algun cas comptat vaig haver de demanar ajuda abans de poder-me'n sortir finalment. Així, mentre que la programació d'algun mode, dels primers de cada tipus a la llista anterior, ja que l'he ordenada de més simples a més complexos dins de cada classe de mode, va ser realment senzilla, els últims de cada apartat van ser els més difícils de dur a terme, sovint a causa de la necessitat d'idear una funció massa densa de programar o bé per alguna exigència especial en l'aspecte gràfic del mode. Tot i això, malgrat que alguns fossin tan fàcils de realitzar i alguns altres tan complicats, tota la resta van resultar el punt just de desafiants però a l'hora assolibles per mi, i treballant en aquest grup fou on més vaig gaudir en tot el treball sencer. Alguns exemples d'aquestes funcions especialment complicades de dur a terme varen ser: `pintarSERP_FANTASMA()`, `pintarSERP_DIAGONAL()`, `donarlordre()`, `casellesocupades()`, `recomptePOSICIONS_6()`, `toca_paret()`...



*Escalada en la complexitat dels diferents 25 modes de joc de SNAKEST*

## 2.3. Menús i estructura general

Un cop programats i provats tots els modes que em van semblar suficients, per separat, va tocar unificar-los tots 25 en un sol programa que els englobés i permetés al jugador navegar d'un a l'altre a través d'un menú general o interfície gràfica. Em va tocar, doncs, primer de tot, pensar com voldria classificar-los i ordenar-los dins d'aquest. Vaig decidir fer-ho segons el tipus de norma la perversió de la qual haguessin sorgit, com ja hem comentat prèviament, separats en 7 categories: **CONTROLS**, **LÍMITS**, **SERP**, **RATA**, **GRÀFIC**, **OBJECTIU** i **ALTRES**. Un cop vaig estar segur de com voldria que s'articulés el menú, vaig haver de pensar en l'aspecte gràfic que prendria tot plegat per pantalla: els marges i font de les lletres, el nom dels modes i del joc, **"SNAKEST"** (de l'anglès "el més *Snake* de tots", "*I Snake* absolut"); i, seguidament, em vaig poder posar a programar tot el seguit de funcions i rutines que em servien per convertir en realitat totes aquestes idees de com volia que fos.

Totes aquestes funcions tan concretes sobre el funcionament dels menús vaig separar-les en un fitxer propi, *funcions\_menu.h*, així com ja havia fet amb les funcions pròpies del motor gràfic a *funcions\_motor.h*, i amb funcions pròpies del joc (com per exemple `moureSERP()` o `ferCREIXERoDECREIXER()`) a *funcions\_joc.h*. Tots aquests fitxers es troben, a la vegada, dins d'una carpeta anomenada *funcions*, al costat d'altres carpetes del mateix estil que engloben tot de fitxers d'un mateix tipus com ara la carpeta *sprites*, que conté totes les imatges que el joc usa en els diferents modes de joc, també classificades segons si es tracten de *sprites* fets servir per representar la serp, la rata, o les lletres i números dels menús (122 *sprites* en total), i la carpeta *modes* on, separats en diferents documents segons la classe a la qual pertanyen, es troba el codi que controla les mecàniques i el funcionament dels 25 modes del meu joc.

Tots aquests fitxers són necessaris a l'hora de fer funcionar el joc i acaben sent tots cridats i fets servir en el fitxer "mare" *main.c*, que és el que controla com es relacionen entre si totes les funcions i recursos emmagatzemats en totes aquestes carpetes diferents per tal de donar forma a un sol joc que les fa servir a la vegada.

L'esquema del què conté el microcontrolador de la consola, del joc, al cap i a la fi, té un aspecte semblant al següent:

```
samd21_platform.c - Codi de plataforma que gestiona la relació main.c/microcontrolador
                    ([ordres programades] / [botons i pantalla, input i output])

main.c             - Arxiu que engloba tota la resta d'arxius i en controla la interacció

[CARPETA] funcions:

    funcions_joc.h  - pintarSERP(), moureSERP(), gestionarINPUT(), crearRATA()...
    funcions_menu.h - pintar_paraula(), pintar_puntuacio(), pintar_menus()...
    funcions_motor.h - ps(), ferforma()
    funcions_setup.h - inicialitzar_finestres(), iniciar_modes_i_serp()
    setup.h

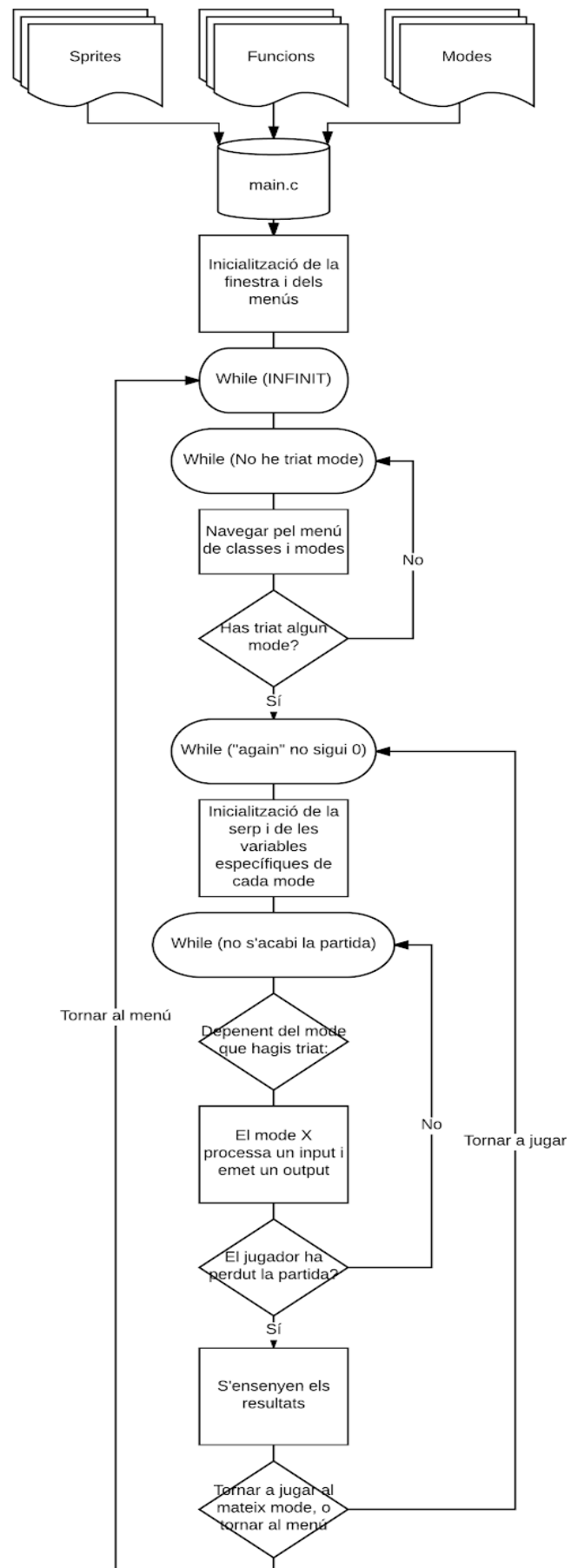
[CARPETA] modes:

    modes_control.h - mode_01_INVERTIT(), ... , mode_03 ESTRATEGIC()
    modes_limits.h  - mode_04_VORES(), ... , mode_06_PINTAR()
    modes_serp.h    - mode_07_REGULABLE(), ... , mode_13_SMOOTH()
    modes_rata.h    - mode_14_PLUS(), ... , mode_16_RATESMOBILS()
    modes_grafic.h  - mode_17_SURVIVAL(), mode_18_GALLINETA()
    modes_objectiu.h - mode_19_EASY(), ... , mode_24_IA()
    modes_altres.h  - mode_25_RETRO()

[CARPETA] sprites:

    HUDsEspecials.c - Variacions del HUD (interfície gràfica) en els diferents modes
    LletresNumeros.c - Lletres i números que es fan servir als menús i resultats
    Rates.c         - Variacions de l'aspecte de l'objectiu en els diferents modes
    Serps.c         - Variacions de l'aspecte del jugador en els diferents modes
    imatgeC.c       - Conté la definició de sprite: llista ordenada de píxels de color
```

L'esquema de funcionament de *main.c*, el joc sencer:



L'aparença gràfica de totes aquestes fases és semblant a:



*Menú principal del joc SNAKEST*



*Primera pàgina del menú de classes*



*Segona pàgina del menú de classes*



*Menú específic de la classe CONTROL*



*Captura del mode de joc INVERTED*



*Captura de la PAUSA durant el joc*

Captura de vídeo de la fase de PAUSA: <https://goo.gl/7B181c>



*Pantalla de resultats al acabar-se la partida*

## 2.4. Per acabar: Debugging i optimització

Abans de poder donar per conclosa la fase del projecte relativa a la concepció creativa i programació del joc que hauria d'omplir la consola que em construïria, i de ficar-me a treballar en el muntatge d'aquesta, vaig haver de seguir treballant un temps més en el codi buscant-hi errades en el funcionament a corregir, i processos innecessàriament complexos o redundants a optimitzar per tal de fer-lo adequat per funcionar al microcontrolador que vaig triar per la consola, la placa *SAMD21 Mini*. Aquesta tasca de poliment va constar a la pràctica de tres fases prou diferenciades:

- **Debugging del programa:** paral·lel al procés de creació i programació en si cal també tenir molt present el procés de debugging mitjançant el qual es procura anar eliminant els errors i problemes i comportaments inesperats del codi que van sorgint durant el procés. Aquests errors o *bugs* es solen donar quan no s'han contemplat tots els casos possibles en la programació d'una funció o instrucció, i el programa es troba davant d'una situació per la qual no l'havien preparat, i actua erròniament. Per solucionar-los i depurar el programa cal anant jugant i rejugant múltiples hores als diferents modes, i provar de detectar-los i trobar-ne l'origen al codi, per arreglar-los.
- **Reducció de la mida** del programa: és a dir, de la memòria necessària per a poder-lo fer cabre i prou còmodament dins dels 32KB de *SRAM* i 256KB de *FlashMemory* de la placa. Al començar aquesta etapa el programa destinava fins a 34718 bytes a la gestió de totes les seves diferents variables i *structs* (264 en total), i 106288 bytes a la gestió de les funcions i rutines del joc (100 en total). En acabar, s'havien convertit en 6766 i 66405 respectivament.
- **Millora del rendiment** del programa: vaig haver de procurar que el cost de processament d'algunes funcions concretes es reduís per tal que el joc funcionés més fluidament en certs modes, cosa que el principi, abans què hi posés solució, provocava que alguns fossin injugables al funcionar tan feixugament. A l'haver-los programat tots primer a l'ordinador, on no tenia cap mena de problema ni escassetat de *CPU*, no em vaig adonar del fet que alguns n'usaven més del compte innecessàriament fins que no vaig provar de fer-los funcionar igual des del microcontrolador, amb una capacitat de processament molt inferior a la de qualsevol ordinador portàtil.

Després de tots tres ajustaments al codi, va estar finalment preparat per poder ser traslladar a la placa i funcionar des d'allà de manera prou satisfactòria.



## Mida de les 264 estructures del programa final

blanc	1 bytes	cuaEfant_data	14 bytes	lletra_i	16 bytes	rataO	16 bytes
espai_data	4 bytes	cuaNfant_data	14 bytes	lletra_j	16 bytes	rataS	16 bytes
posicions_op_inici	4 bytes	cuaOfant_data	14 bytes	lletra_k	16 bytes	record	16 bytes
dospunts_data	6 bytes	cuaSfant_data	14 bytes	lletra_l	16 bytes	record_data	16 bytes
lletra_i_data	6 bytes	capE	16 bytes	lletra_m	16 bytes	set	16 bytes
punt_data	6 bytes	capEfant	16 bytes	lletra_n	16 bytes	sis	16 bytes
capE_data	8 bytes	capN	16 bytes	lletra_o	16 bytes	SNAKEST	16 bytes
capN_data	8 bytes	capNE	16 bytes	lletra_p	16 bytes	Sud	16 bytes
capNE_data	8 bytes	capNfant	16 bytes	lletra_q	16 bytes	tres	16 bytes
capNO_data	8 bytes	capNO	16 bytes	lletra_r	16 bytes	un	16 bytes
capO_data	8 bytes	capO	16 bytes	lletra_s	16 bytes	vuit	16 bytes
capS_data	8 bytes	capOfant	16 bytes	lletra_t	16 bytes	zero	16 bytes
capSE_data	8 bytes	capS	16 bytes	lletra_u	16 bytes	cinc_data	21 bytes
capSO_data	8 bytes	capSE	16 bytes	lletra_v	16 bytes	dos_data	21 bytes
cercle_data	8 bytes	capSfant	16 bytes	lletra_w	16 bytes	lletra_a_data	21 bytes
cercleNEG_data	8 bytes	capSO	16 bytes	lletra_x	16 bytes	lletra_b_data	21 bytes
cosA13_data	8 bytes	cercle	16 bytes	lletra_y	16 bytes	lletra_d_data	21 bytes
cosB13_data	8 bytes	cercleNEG	16 bytes	lletra_z	16 bytes	lletra_g_data	21 bytes
cos_data	8 bytes	cinc	16 bytes	mira	16 bytes	lletra_h_data	21 bytes
creuC_data	8 bytes	cos_	16 bytes	Nord	16 bytes	lletra_k_data	21 bytes
cuaamunt_data	8 bytes	cosA13	16 bytes	nou	16 bytes	lletra_n_data	21 bytes
cuaavall_data	8 bytes	cosB13	16 bytes	Oest	16 bytes	lletra_o_data	21 bytes
cuaE_data	8 bytes	cosNEfant	16 bytes	pausa_A	16 bytes	lletra_p_data	21 bytes
cuaN_data	8 bytes	cosNOfant	16 bytes	pausa_E	16 bytes	lletra_q_data	21 bytes
cuaNE_data	8 bytes	cosSEfant	16 bytes	pausa_P	16 bytes	lletra_r_data	21 bytes
cuaNO_data	8 bytes	cosSOfant	16 bytes	pausa_S	16 bytes	lletra_s_data	21 bytes
cuaO_data	8 bytes	cosXfant	16 bytes	pausa_U	16 bytes	lletra_t_data	21 bytes
cuaS_data	8 bytes	cosYfant	16 bytes	p_cinc	16 bytes	lletra_u_data	21 bytes
cuaSE_data	8 bytes	creuC	16 bytes	p_cinc_data	16 bytes	lletra_z_data	21 bytes
cuaSO_data	8 bytes	cuaamunt	16 bytes	p_dos	16 bytes	nou_data	21 bytes
diagonalNE_data	8 bytes	cuaavall	16 bytes	p_dos_data	16 bytes	percent_data	21 bytes
diagonalNO_data	8 bytes	cuaE	16 bytes	percent	16 bytes	quatre_data	21 bytes
diagonalSE_data	8 bytes	cuaEfant	16 bytes	p_nou	16 bytes	set_data	21 bytes
diagonalSO_data	8 bytes	cuaN	16 bytes	p_nou_data	16 bytes	sis_data	21 bytes
Est_data	8 bytes	cuaNE	16 bytes	p_quatre	16 bytes	tres_data	21 bytes
mira_data	8 bytes	cuaNfant	16 bytes	p_quatre_data	16 bytes	vuit_data	21 bytes
Nord_data	8 bytes	cuaNO	16 bytes	p_set	16 bytes	zero_data	21 bytes
Oest_data	8 bytes	cuaO	16 bytes	p_set_data	16 bytes	lletra_m_data	24 bytes
posicions_op	8 bytes	cuaOfant	16 bytes	p_sis	16 bytes	lletra_v_data	24 bytes
puntaE_data	8 bytes	cuaS	16 bytes	p_sis_data	16 bytes	lletra_w_data	24 bytes
puntaN_data	8 bytes	cuaSE	16 bytes	p_tres	16 bytes	lletra_x_data	24 bytes
puntaO_data	8 bytes	cuaSfant	16 bytes	p_tres_data	16 bytes	lletra_y_data	24 bytes
puntaS_data	8 bytes	cuaSO	16 bytes	p_un	16 bytes	abc	41 bytes
rata_data	8 bytes	diagonalNE	16 bytes	p_un_data	16 bytes	pausa_A_data	69 bytes
rataE_data	8 bytes	diagonalNO	16 bytes	punt	16 bytes	pausa_E_data	69 bytes
rataFALSA_data	8 bytes	diagonalSE	16 bytes	puntaE	16 bytes	pausa_P_data	69 bytes
rataN_data	8 bytes	diagonalSO	16 bytes	puntaN	16 bytes	pausa_S_data	69 bytes
rataNEG_data	8 bytes	dos	16 bytes	puntaO	16 bytes	pausa_U_data	69 bytes
rataO_data	8 bytes	dospunts	16 bytes	puntaS	16 bytes	lletresPAUSA	80 bytes
rataS_data	8 bytes	espai	16 bytes	punts	16 bytes	menu_altres	88 bytes
Sud_data	8 bytes	Est	16 bytes	punts_data	16 bytes	menu_control	88 bytes
posicionsPAUSA	10 bytes	focus	16 bytes	punts_m	16 bytes	menu_editor	88 bytes
un_data	11 bytes	lletra_a	16 bytes	punts_m_data	16 bytes	menu_grafic	88 bytes
capEfant_data	14 bytes	lletra_b	16 bytes	p_vuit	16 bytes	menu_inici	88 bytes
capNfant_data	14 bytes	lletra_c	16 bytes	p_vuit_data	16 bytes	menu_limits	88 bytes
capOfant_data	14 bytes	lletra_c_data	16 bytes	p_zero	16 bytes	menu_objectiu	88 bytes
capSfant_data	14 bytes	lletra_d	16 bytes	p_zero_data	16 bytes	menu_principal	88 bytes
cosNEfant_data	14 bytes	lletra_e	16 bytes	quatre	16 bytes	menu_rata	88 bytes
cosNOfant_data	14 bytes	lletra_e_data	16 bytes	rata	16 bytes	menu_serp	88 bytes
cosSEfant_data	14 bytes	lletra_f	16 bytes	rataE	16 bytes	focus_data	158 bytes
cosSOfant_data	14 bytes	lletra_f_data	16 bytes	rataFALSA	16 bytes	SNAKEST_data	401 bytes
cosXfant_data	14 bytes	lletra_g	16 bytes	rataN	16 bytes	lletres	656 bytes
cosYfant_data	14 bytes	lletra_h	16 bytes	rataNEG	16 bytes	p	664 bytes

## Mida de les 100 funcions del programa final

crearRATA	16 bytes	inicialitzar_finestres	386 bytes
ferCREIXERoDECREIXER	16 bytes	moureORIGEN_EDITOR	398 bytes
gestionarRATES	16 bytes	mode_22_LOOPS	399 bytes
moureSERP	21 bytes	moureSERP_FINAL	404 bytes
moureSERP_FANTASMA	21 bytes	moureRATA_IA	417 bytes
inicia_generador_aleatoris	32 bytes	mode_07_REGULABLE	419 bytes
moureSERP_dir	34 bytes	mode_13_SMOOTH	423 bytes
pintarHUD	36 bytes	mode_66_BARRERES	424 bytes
ps	51 bytes	mode_12_DIAGONAL	433 bytes
aleat	53 bytes	moureRATA_BASQUET	433 bytes
ferforma	81 bytes	mode_03 ESTRATEGIC	476 bytes
gestionarINPUT_1D	86 bytes	pintarSERP_SMOOTH	497 bytes
gestionarINPUT_IA	111 bytes	mode_23_1D	511 bytes
pintarSERP_RETRO	115 bytes	mode_21_BASQUET	519 bytes
ferCREIXERoDECREIXER_mode	151 bytes	pintarSERP_1D	523 bytes
gestionarINPUT_DIAGONAL	157 bytes	crearRATA_mode	532 bytes
inicialitzar_menu	175 bytes	mourePIT	542 bytes
gestionarINPUT	187 bytes	pintarSERP	570 bytes
gestionarINPUT_INVERS	187 bytes	crearRATA_BARRERES	575 bytes
posicionsXY	190 bytes	crearRATA_DIAGONAL	578 bytes
comptar_llargada_pix	194 bytes	mode_24_IA	583 bytes
ferCREIXERoDECREIXER_1D	198 bytes	donarlordre	585 bytes
gestionarINPUT_STRA	217 bytes	toca_paret	603 bytes
pintar_puntuacio	225 bytes	moureRATA_MOBILS	610 bytes
recomptePOSICIONS_6	251 bytes	gestionarINPUT_CAPICUA	621 bytes
mode_19_EASY	258 bytes	mode_15 DIRECCIONALS	670 bytes
gestionarINPUT_EGO	261 bytes	moureSERP_1D	676 bytes
gestionarINPUT_MANUAL	263 bytes	pintarSERP_GALLINETA	735 bytes
recompteCOSSOS_5	272 bytes	crearRATA_5i13	813 bytes
mode_17_SURVIVAL	277 bytes	gestionarINPUT_REMOT	823 bytes
pintar_paraula	277 bytes	moureSERP_RETRO	848 bytes
capicua	282 bytes	pintar_menus	848 bytes
moureMIRA	283 bytes	casellesocupades	859 bytes
mode_14_PLUS	285 bytes	ferforma_mode	886 bytes
mode_08_MANUAL	288 bytes	moureSERP_SMOOTH	924 bytes
mode_18_GALLINETA	296 bytes	pintar_menu_XIFRES	925 bytes
mode_25_RETRO	296 bytes	moureSERP_dir_mode	937 bytes
mode_00_CLASSIC	297 bytes	gestionarRATES_mode	987 bytes
mode_01_INVERTIT	297 bytes	moureSERP_FANTASMA_dir_mode	1037 bytes
mode_04_VORES	297 bytes	pintarSERP_CAPICUA	1069 bytes
mode_10_CAPICUA	297 bytes	inicialitzar_TOTS_menus	1130 bytes
mode_05_FANTASMA	300 bytes	moureSERP_VORES	1212 bytes
mode_16_RATESMOBILS	300 bytes	ps_mode	1498 bytes
mode_02_EGO	303 bytes	moureSERP_DIAGONAL	1514 bytes
mode_09_INCRE	319 bytes	iniciar_modes_i_serp	1690 bytes
gestionarINPUT_PINTAR	324 bytes	pintarHUD_mode	1693 bytes
recompteCASELLES_13	341 bytes	pintarSERP_FANTASMA	2636 bytes
mode_06_PINTAR	355 bytes	pintarSERP_DIAGONAL	4270 bytes
mode_11_REMOT	360 bytes	pintarhoTOT	6990 bytes
mode_20_SelNat	379 bytes	main	7843 bytes

## 3. LA CONSOLA

### 3.1. Fases de prototipatge

#### 3.1.1. Xip FT232H. Comunicació PC-pantalla

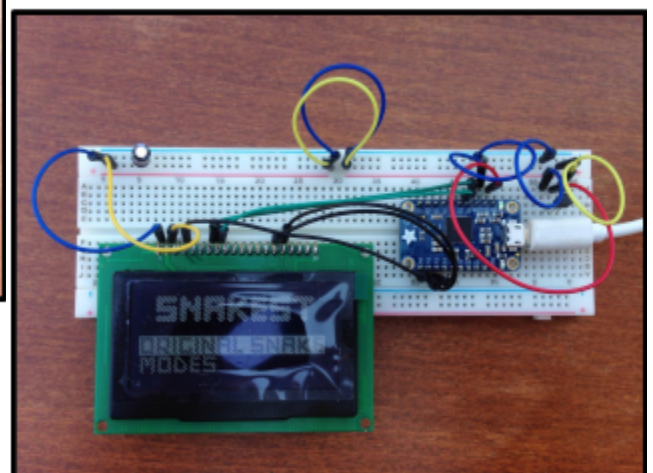
De fet el primer prototip de la meua consola que vaig construir no ho va ser pas, un prototip de consola, en absolut. De fet, funcionava tan sols com a “monitor” des d’on visualitzar el joc, que realment funcionava i es jugava des d’un ordinador totalment independent i extern. Així, quan vaig tenir el joc enllestit fins a un punt amb el qual m’hi vaig sentir còmode com per començar a temptejar fases més avançades del procés sense patir perquè al final el resultat pogués semblar inacabat o no prou polit, vaig voler provar de veure com es veuria tot allò que havia estat programant i jugant fins llavors des del meu ordinador portàtil des d’una pantalla molt més petita i de menys resolució com la que hauria de tenir la consola final.

Per fer-ho, per aconseguir aquesta primera sensació de que ja m’anava acostant cada cop més a la construcció talment dita de la màquina, vaig haver de comptar amb l’ajuda del xip **FT232H**. El que aquest em va permetre va ser connectar l’ordinador, des del qual es rebia l’*input* del jugador i feia anar el joc, amb la pantalla que m’havia comprat, de tipus *OLED* i de 128x64 píxels de resolució, de manera que el què processava el primer es podria veure també en aquesta segona pantalla “auxiliar”. Això, [processar el joc al PC però veient-lo per la pantalla de la consola], em va servir de pas intermedi entre [processar el joc i veure’l al PC] i [processar el joc i veure’l a la consola].



*Pack ordinador+FT232H+pantalla*

*Detall del conjunt FT232H-pantalla*



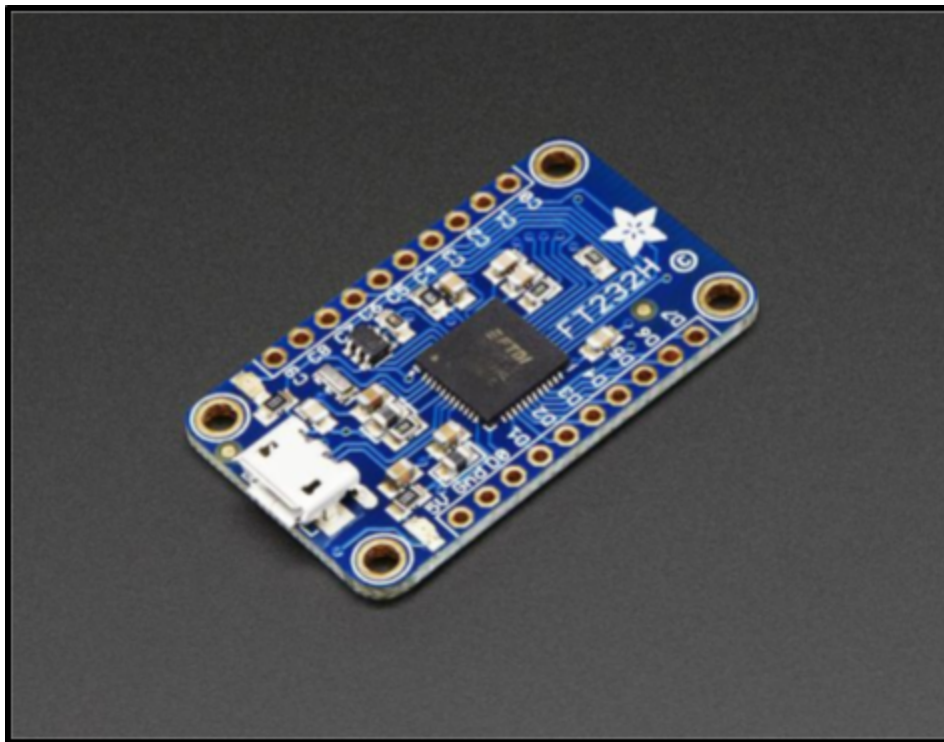
### 3.1.1.1. Components del muntatge

#### Placa FT232H

És la peça clau i la gràcia de tot el prototip. Parlant de forma una mica més tècnica, el que fa aquesta placa és atorgar a l'ordinador al qual es connecta la capacitat de comunicar-se de forma directa amb tot un seguit de dispositius i sensors externs amb els que de normal necessitaria algun microcontrolador o ARDUINO per fer d'intermediari entre tots dos. Permetent a l'ordinador "aprendre a parlar" alguns protocols de control nous com *SPI*, *I2C*, *serial UART* o *JTAG*, fa que aquest de cop sigui capaç per si sol de fer funcionar pantalles, encendre LEDs o llegir sensors, sense l'ajuda de cap més component addicional. Al ser aquesta placa, de la marca FTDI, una de les poques amb una funcionalitat semblant, i al costar relativament poc, \$14.95 en la botiga online Adafruit, la van fer un candidat perfecte del tot adient a les meves necessitats i butxaca.

Enllaç:

<https://learn.adafruit.com/adafruit-ft232h-breakout/overview>



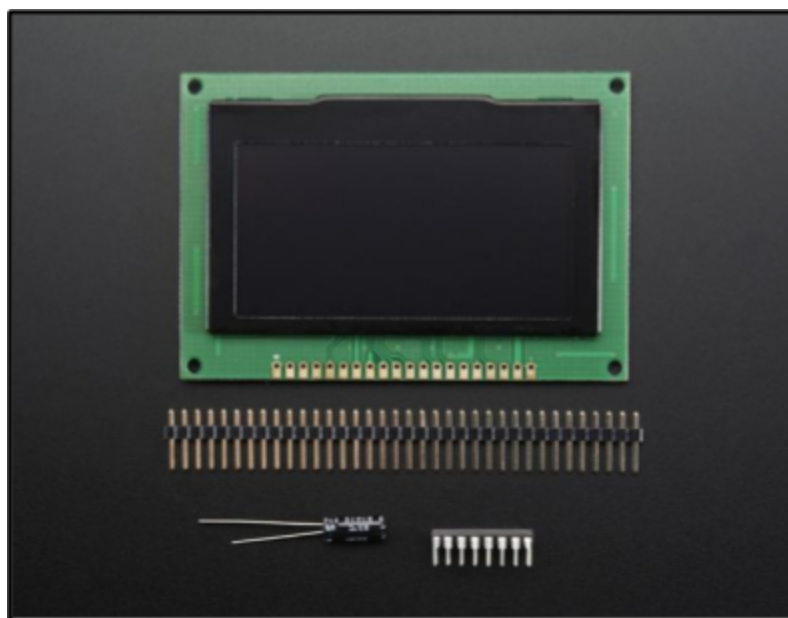
*Imatge del xip FT232H*

### Pantalla “ 2.7” Monochrome 128x64 OLED Display Module ”

Les característiques d'aquesta pantalla les vaig haver de tenir en compte molt abans d'arribar a aquest punt del procés de producció, i és que ja hi vaig haver de pensar tot just quan m'estava plantejant quin tipus de jocs haurien d'omplir la meva consola, ben bé al principi de tot plegat. Vaig autoimposar-me les limitacions de què es tractés d'una pantalla monocroma i de ràtio 2:1 per tal que el ventall de possibles jocs a programar-hi se'm reduís i no m'aclaparés tant el nombre de possibles idees que se'm poguessin acudir al complicar-me una mica la vida al obligar-me a pensar en un joc què es pogués jugar en blanc i negre i en una pantalla el doble d'ample que d'alta, i d'una resolució prou justa. Dins de les diferents pantalles al mercat que compleixen aquestes dues característiques em vaig acabar decantant per la més gran possible, de 2'7 polzades, que fabrica la marca Adafruit i posa a la venda en la seva web oficial a \$49.95. El protocol que fan servir aquest tipus de pantalla és el SPI (de l'anglès *Serial Peripheral Interfície*, en català *Interfície per a perifèrics sèrie*), i precisament és per això necessitava la placa *FT232H* per comunicar-m'hi i fer-hi arribar el joc perquè me'l mostrés.

Enllaç:

<https://learn.adafruit.com/2-7-monochrome-128x64-oled-display-module/overview?view=all>



*Imatge de la pantalla 2.7" Monochrome 128x64 OLED Display Module*

### Regulador de tensió LDO a 3.3V

El necessitem perquè la pantalla anteriorment esmentada demana una alimentació de 3.3V per funcionar correctament i la sortida de corrent de la placa *FT232H*, que serà la que alimentarà la pantalla, és de 5V, així l'hem de fer servir per convertir-los en els 3.3V desitjats que ens interessin. El regulador és el component de la imatge que té 3 pins o "potes". Una pota es connecta al corrent que vols canviar, una altra al *GROUND* o al pol negatiu del circuit, i per l'última n'emanarà el corrent modificat que busques.

Enllaç:

<http://es.rs-online.com/web/p/reguladores-de-tension-de-baja-caida/0403888/?sra=pstk>



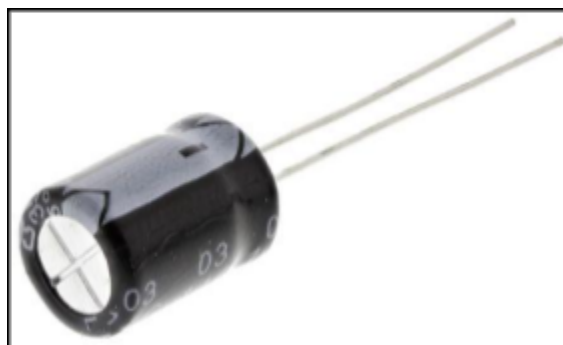
*Imatge del regulador de tensió LDO a 3.3V*

### Condensador de 220µF

En el circuit ens serveix per suavitzar les fortes fluctuacions de corrent que es poden donar en els 3.3V que alimenten la pantalla. Els condensadors són components que poden "retenir" o "emmagatzemar" energia durant un cert període de temps, i llavors anar-la alliberant de forma progressiva. És per això que funcionen tan bé per aquest tipus de tasques de mitigació de variacions en la quantitat de corrent que passa per un circuit i els travessa. En aquest cas, ens serveix per alleugerir els pics de demanda de corrent de la pantalla.

Enllaç:

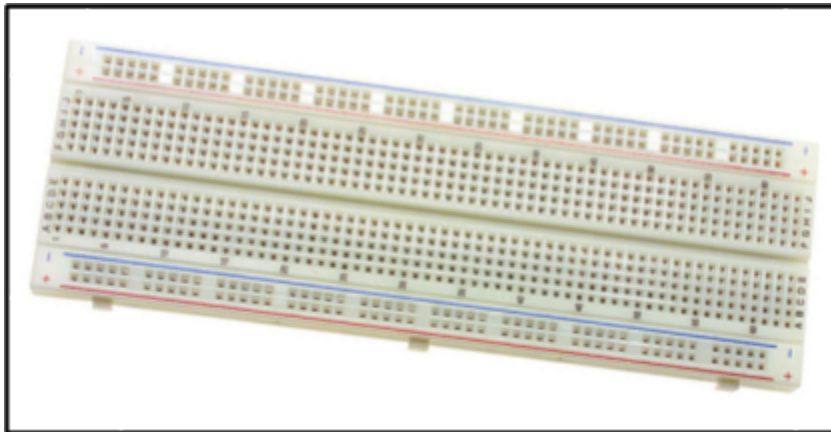
<http://es.rs-online.com/web/p/condensadores-de-aluminio/7111264/>



*Imatge del condensador de 220µF*

### Breadboard

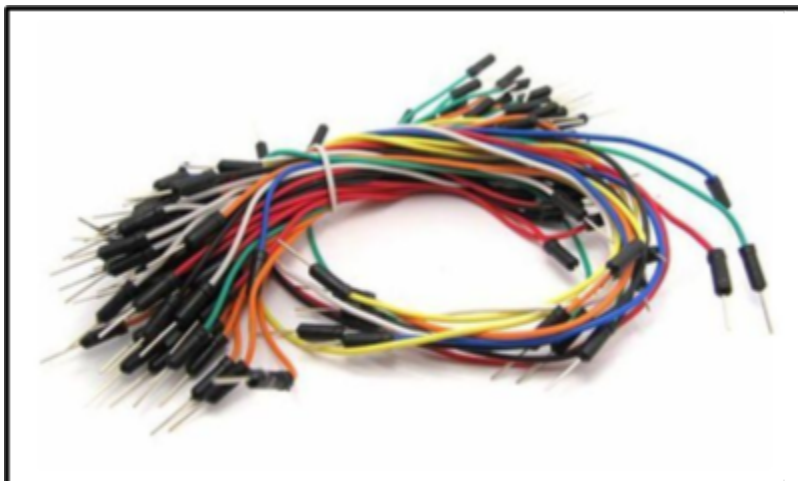
La què he fet servir és d'un model prou estàndard, de 830 contactes, que ja tenia per casa al ja haver-ne necessitat una per dur a terme altres projectes en anys anteriors. Sobre la *breadboard* és on es connecten tots els components que formen un circuit de manera que sempre se'ls pot tornar a treure i fer servir per a altres feines, sense haver-los de soldar i dessoldar cada vegada.



*Imatge d'una breadboard de 830 contactes*

### 13 cables "jumper"

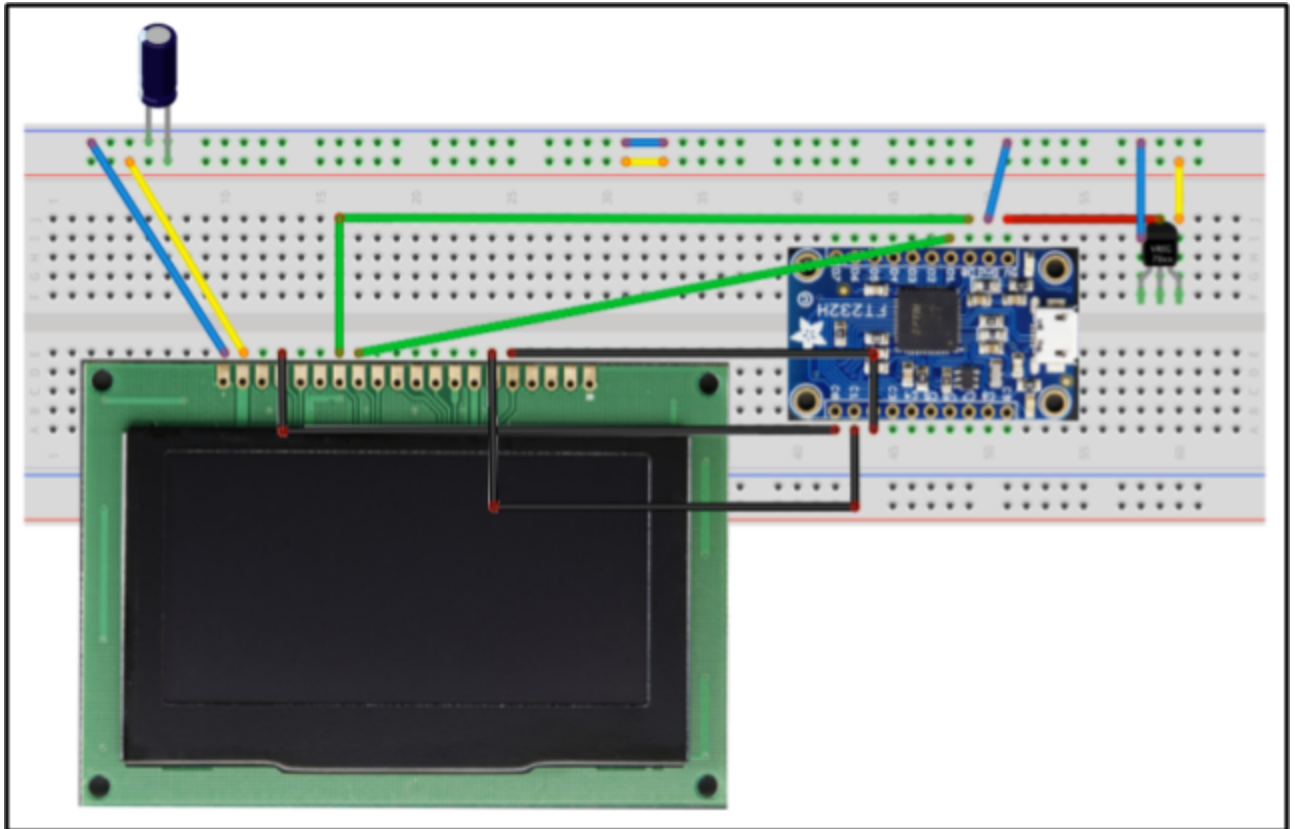
Són els que em van caldre per unir totes les parts del circuit, connectats com s'explicarà seguidament.



*Imatge d'un grapat de cables "jumper"*



### 3.1.1.2. Esquema de connexions



*Esquema de muntatge del primer prototip (xip FT232H)*

#### Alimentació

- Dels 5V de la placa a la pota del mig del regulador de tensió (Vermell).
- De la pota de la dreta del regulador de tensió a la fila horitzontal positiva (Groc).
- Unint la fila horitzontal positiva de la secció de la dreta amb la de l'esquerra (Groc).
- De la fila horitzontal positiva al pin 2, el de l'alimentació, de la pantalla (Groc).
- Condensador entre les files positiva i negativa a la secció de l'esquerra de la *board*.
- Del terra o GND de la placa a la fila més superior que recorre la *board* lateralment (Blau).
- Del terra de la fila més superior a la pota esquerra del regulador de tensió (Blau).
- Unint la fila horitzontal negativa de la secció de la dreta amb la de la esquerra (Blau).
- Del terra de la fila horitzontal negativa al pin 1, el del terra, de la pantalla (Blau).

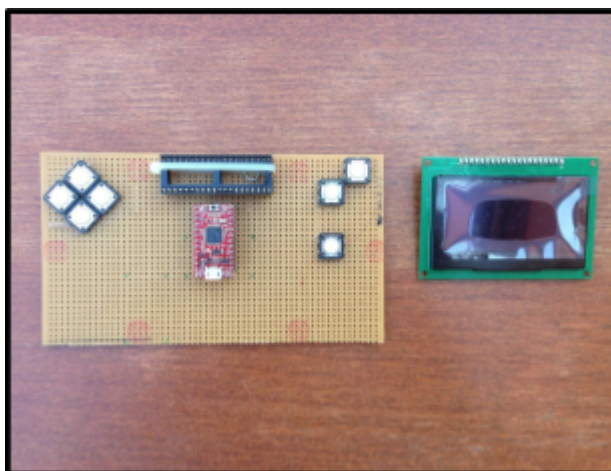
#### Comunicació

- Del pin D0 de la placa al pin 7, que controla l'*SPI clock*, de la pantalla (Verd).
- Del pin D1 de la placa al pin 8, que controla l'*SPI data*, de la pantalla (Verd).
- Del pin C0 de la placa al pin 4, el selector entre dades i ordres, de la pantalla (Negre).
- Del pin C1 de la placa al pin 15, que controla l'*SPI ChipSelector*, de la pantalla (Negre).
- Del pin C2 de la placa al pin 16, que controla el *RESET*, de la pantalla (Negre).

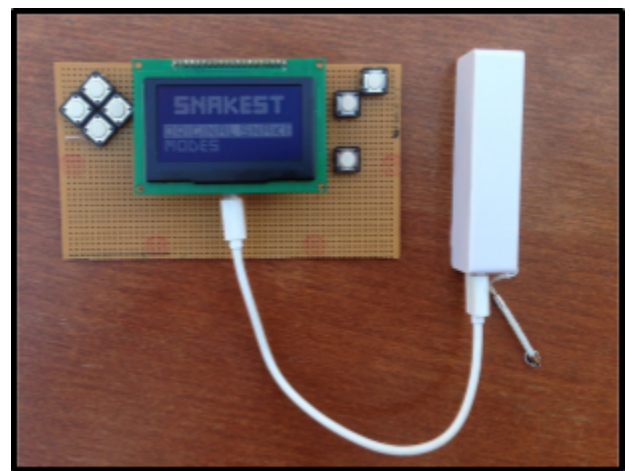


### 3.1.2. Placa SAMD21 Mini. Consola autònoma

Aquest segon i, de moment, últim prototip que vaig fer, ja s'assembla molt més al que podríem anomenar "videoconsola portable", tot i tampoc ser ben bé tan polit i estar tan ben acabat com m'agradaria que m'hagués quedat. Si hagués disposat de més temps m'hauria agradat poder treballar en una tercera i una quarta versió de la màquina però, finalment, això no fou possible. Aquest, a diferència del prototip que el precedia, ja sí que rep l'*input* del jugador i el processa tot per ell mateix gràcies al microcontrolador **SAMD21 Mini** que porta incorporat. Les principals diferències entre tots dos són precisament aquesta placa, que fa les funcions de la *CPU* de l'ordinador i se sap comunicar amb la pantalla tot a la vegada, i els 7 botons que té soldats, que reben l'*input* del jugador en substitució del teclat del portàtil, com era fins ara. També cal fer notar que aquesta vegada, com que l'aparell ja no depèn ni té perquè estar connectat de cap manera a cap portàtil ni ordinador per funcionar, ja no pot refiar-se tampoc del seu corrent com a font d'energia. Això significa que tot i que "autònoma" en tant que processament de l'*input* i *output* del joc, no ho és en termes energètics, i ha d'estar connectada mitjançant el port *miniUSB* de la placa a alguna font que l'alimenti per poder-se encendre i jugar-hi. Aquesta font pot ser o bé una presa de corrent domèstica de les més corrents, o una bateria portàtil com les que s'usen per carregar els mòbils quan ets lluny de qualsevol presa disponible, o bé el port *USB* d'un altre dispositiu amb font d'energia pròpia, com podria ser un ordinador, precisament. L'addició d'una bateria de liti i d'un interruptor d'encès i apagat a la consola per fer-la més independent encara són algunes de les innovacions amb les quals comptarien posteriors prototips respecte dels anteriors si els hagués pogut dur a terme finalment, a més d'una millor distribució de l'espai i components físics i potser també la incorporació d'algun sistema d'àudio a la consola que acompanyés al jugador en jugar.



*Prototip final desmuntat*



*Consola engegada i funcionant*

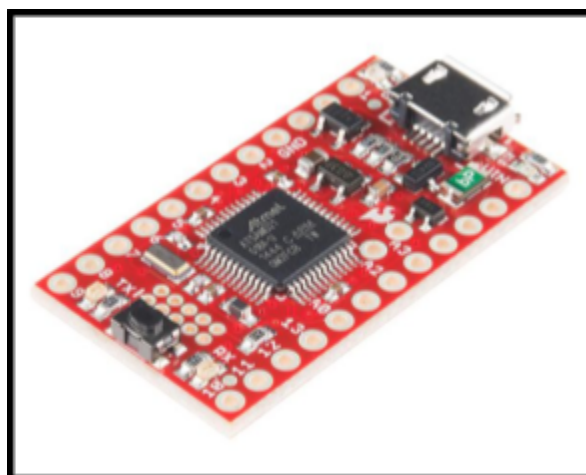
### 3.1.2.1. Components del muntatge

#### Placa SAMD21 Mini

Prou petita i discreta com per poder-se integrar en un circuit o aparell de forma que no en compliqui massa l'ergonomia o en trastoqui la forma, just prou potent i capaç com per poder fer funcionar el meu joc de manera fluida i correctament, i a la venda a la botiga online oficial del seu fabricant, *SparkFun*, a \$19.95. La idea de fer servir una placa d'aquestes característiques la vaig treure del blog d'un noi que s'havia construït una consola pel seu compte com hauria de fer jo i que l'havia fet servir per la seva reduïda mida i en canvi sorprenent potència en comparació, però altres alternatives que també van estar sobre la taula durant un temps van ser altres plaques com ara els microcontroladors *Arduino*, més coneguts i convencionals, o un *Raspberry*. Mentre que els primers tenien a favor que ja hi estava més familiaritzat i ja en tenia a casa i potser no n'hauria de comprar de nous, en canvi tenien en contra la seva mida i limitada capacitat processadora, que hauria estat incapaç de fer funcionar el meu joc a causa de la seva relativament elevada complexitat i magnitud. Un ordinador "monoplaca" com seria per exemple un *Raspberry* hauria pogut fer-se'n càrrec sense problema, però en canvi va ser descartat perquè precisament seria massa fàcil de dur a terme el port d'ordinador a consola (bàsicament perquè no hauria estat necessari, perquè una *Raspberry* de fet també és un ordinador), a més de pel seu pes i mida i forma encara més irregular que la d'un *Arduino*. Finalment em vaig decantar per la *SAMD21 Mini* perquè tant com per la mida com el preu em va semblar ideal, i les seves prestacions, 256KB de memòria Flash, 32KB de SRAM i 32KB of EEPROM, permetrien que amb un cert esforç fos capaç de fer-hi encabir el meu programa sense que ni per una banda em resultés massa difícil, ni tampoc massa senzill.

Enllaç:

<https://www.sparkfun.com/products/13664>



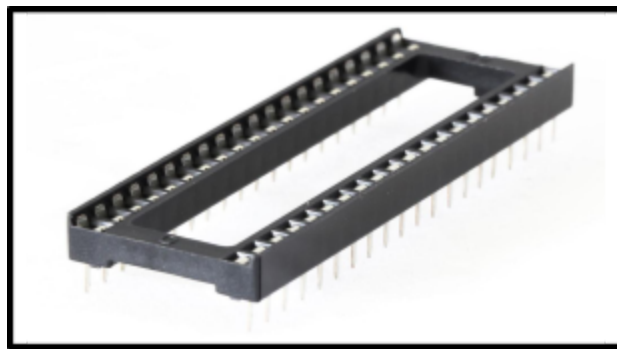
*Imatge del microcontrolador SAMD21 Mini*

### Pantalla “ 2.7" Monochrome 128x64 OLED Display Module ”

És la mateixa pantalla que vaig estar fent servir per fer les primeres proves. Aquesta vegada es comunica directament amb el microcontrolador sense haver de comptar amb l'FT232H per fer-ho, i ara és l'única pantalla des de la qual es veu el joc, que en la fase de prototipatge anterior, en canvi, funcionava simultàniament amb la del portàtil, per comprovar que totes dues mostressin el mateix en cada instant.

### 2 sòcols de 24 i 40 pins

Soldats a la *board* com si es tractessin dels components reals, permeten que siguin més fàcils de connectar i desconnectar del circuit la placa i la pantalla, respectivament. Consten d'una part inferior plena dels pins mascle que es claven i solden a la *board*, que queda fixa, i d'una superior femella en la qual es claven els pins dels components en qüestió, que és de més fàcil col·locació i extracció. No em va caler comprar-ne enlloc, els vaig poder aconseguir gratuïtament.



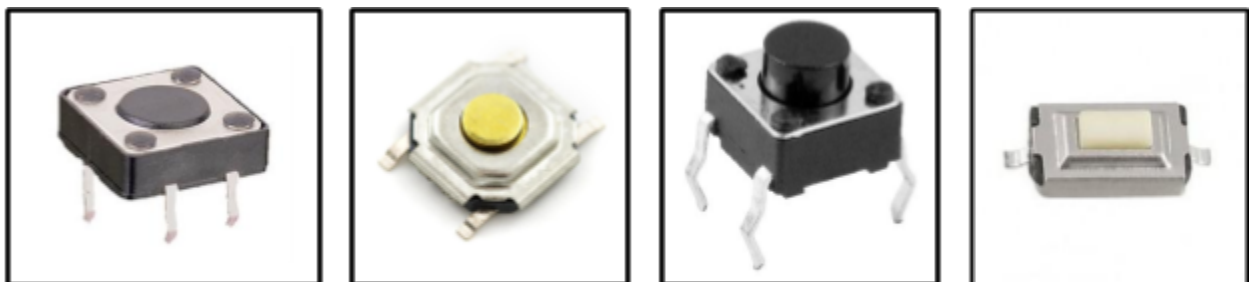
*Imatge d'un sòcol de 40 pins*

### 7 botons pulsadors de 12x12mm

4 botons són usats per a les fletxes que apunten amunt, avall, esquerra i dreta, 2 més fan de A i B, i el 7è és el de “PAUSA”. El cert és que en vaig haver d'adquirir de diverses mides i tactes, tots a un preu molt barat, a cèntims/botó, i provar-los en primera persona abans de decidir-me per quin tipus fer servir en la meva consola. Al final vaig decidir-me per uns de rodons de 12x12mm.

Enllaç:

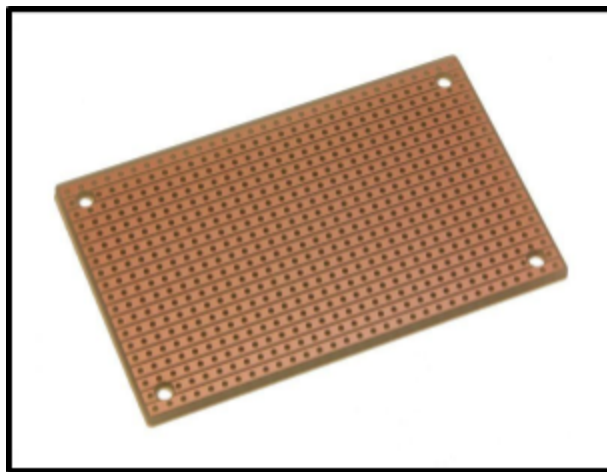
[http://www.diotronic.com/pulsador-12x12-mini-p11-521-2\\_1882/](http://www.diotronic.com/pulsador-12x12-mini-p11-521-2_1882/)



*Imatges dels diferents pulsadors que vaig provar abans de decidir-me pel primer, de 12x12mm*

### Protoboard

De 60x20 forats, la que he usat és de tipus *stripboard*, és a dir, té les files connectades entre elles: això redueix la quantitat de soldadura que s'ha de fer servir per unir 2 o més punts que es trobin a una mateixa línia horitzontal perquè com ja he dit les files comparteixen un mateix corrent de per sí, però dificulta molt més la lectura del circuit a l'acabar. En posteriors prototips m'agradaria treballar amb una que no estés connectada així, una *perfboard*. És doncs, un altre aspecte que m'hauria agradat canviar del prototip que finalment entrego. Vaig haver de fer-ho amb aquesta perquè vaig poder fer-me amb ella gratuïtament i en canvi una d'una altra classe me l'hauria d'haver comprada expressament.



*Imatge d'una stripboard de 30x20 forats*

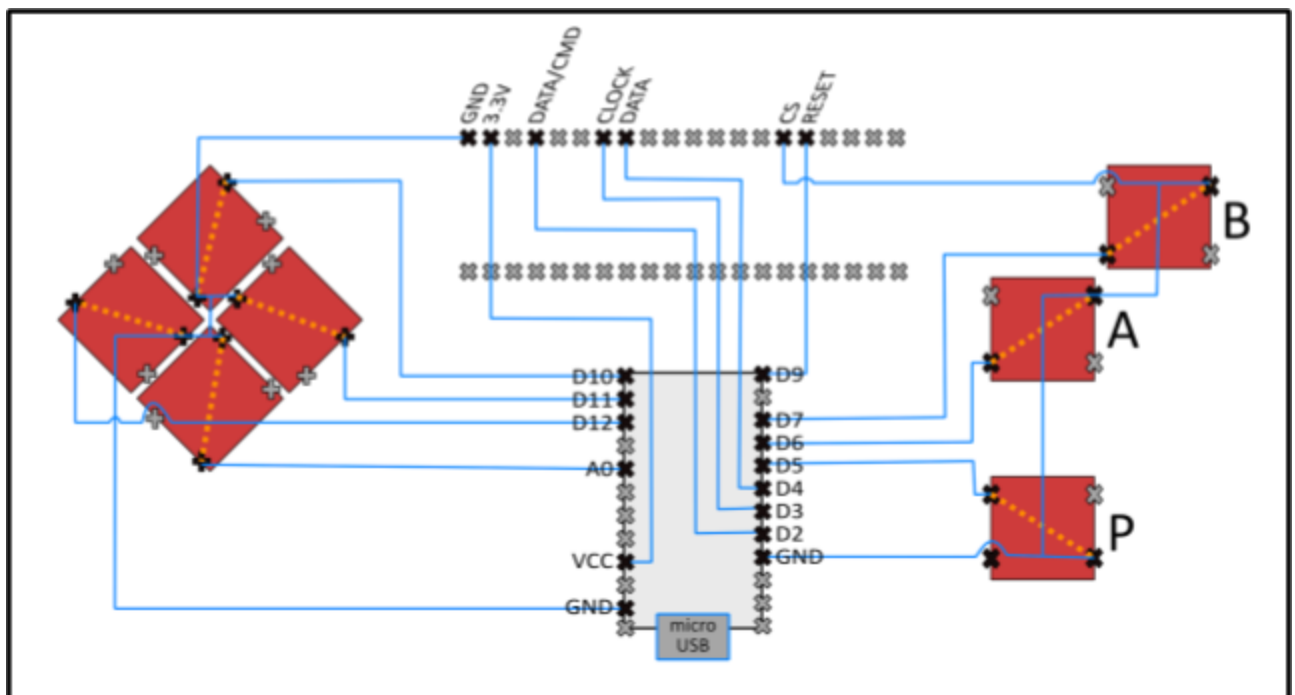
### Estany i soldador

Els vaig poder aconseguir sense haver-me'n de comprar enlloc. Em van caler per unir les columnes de forats verticalment i els 2 sòcols i 7 botons a la *board*, seguint l'esquema exposat a continuació.

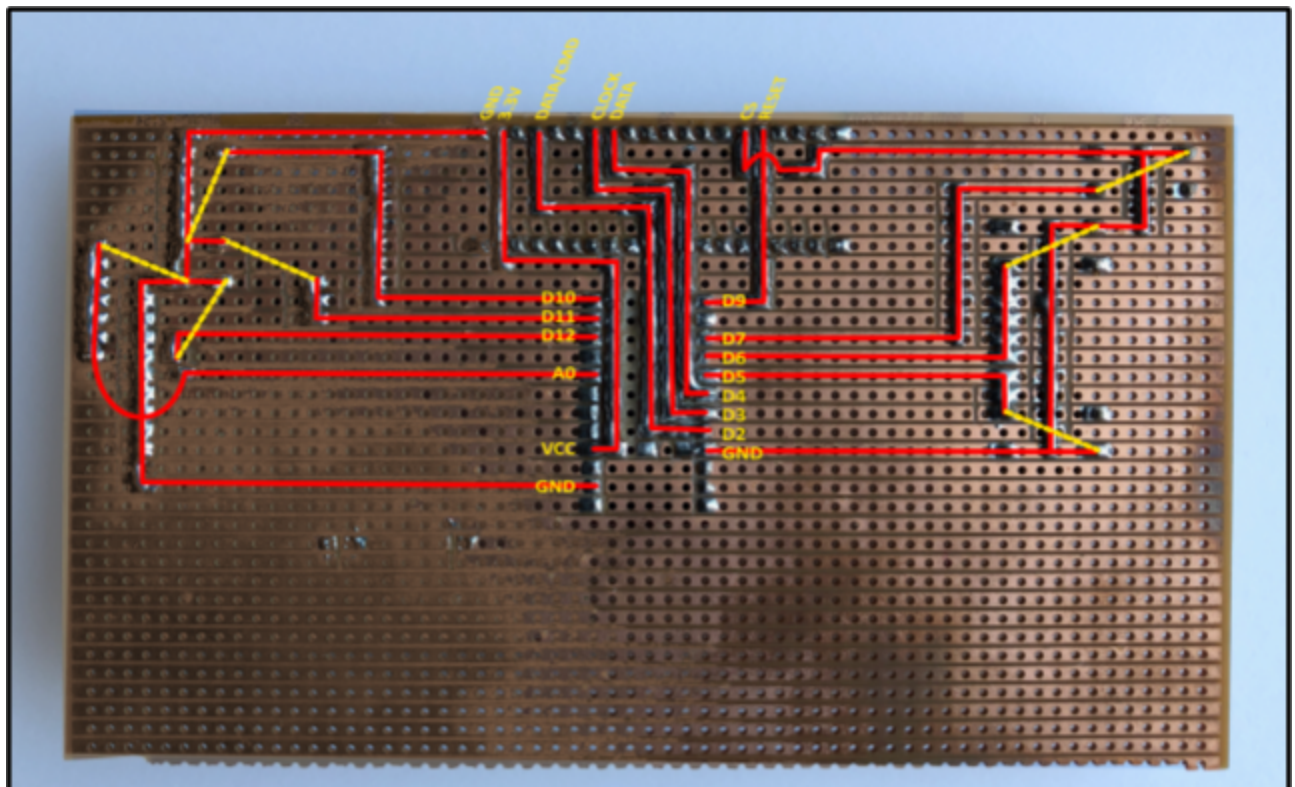


*Imatge d'un soldador elèctric d'estany*

### 3.1.2.2. Esquema de connexions



*Esquema orientatiu del funcionament i muntatge del segon prototip (placa SAMD21 Mini)*



*Saldatge de la placa seguint les indicacions de l'esquema superior*



## Connexions

- Del pin 1 de la pantalla al botó “amunt” (GND del botó a la pantalla).
- Del botó “amunt” al pin D10 de la placa (*input* del jugador a la placa).
- Del pin 1 de la pantalla al botó “dreta” (GND del botó a la pantalla).
- Del botó “dreta” al pin D11 de la placa (*input* del jugador a la placa).
- Del pin 1 de la pantalla al botó “esquerra” (GND del botó a la pantalla).
- Del botó “esquerra” al pin D12 de la placa (*input* del jugador a la placa).
- Del pin 1 de la pantalla al botó “avall” (GND del botó a la pantalla).
- Del botó “avall” al pin A0 de la placa (*input* del jugador a la placa).
- Del pin 1 de la pantalla al GND de la placa (GND de la placa al GND de la pantalla).
- Del pin 2 de la pantalla al pin VCC de la placa (3.3V de la placa a la pantalla).
- Del pin 4 de la pantalla al pin D2 de la placa (selector dada/comanda).
- Del pin 7 de la pantalla al pin D3 de la placa (SPI Clock).
- Del pin 8 de la pantalla al pin D4 de la placa (SPI Data).
- Del pin 15 de la pantalla al botó “B” (GND del botó a la pantalla).
- Del botó “B” al pin D7 de la placa (*input* del jugador a la placa).
- Del pin 15 de la pantalla al botó “A” (GND del botó a la pantalla).
- Del botó “A” al pin D6 de la placa (*input* del jugador a la placa).
- Del pin 15 de la pantalla al botó “PAUSA” (GND del botó a la pantalla).
- Del botó “PAUSA” al pin D5 de la placa (*input* del jugador a la placa).
- Del pin 15 de la pantalla al GND de la placa (GND de la placa al ChipSelector de la pantalla).
- Del pin 16 de la pantalla al pin D9 de la placa (RESET de la pantalla a la placa).

## 4. CONCLUSIONS

Després de realitzar tot aquest treball, només queda comentar-ne els resultats i la impressió que me n'ha quedat al finalitzar-lo.

Es podria concloure que s'han complert almenys dos dels tres objectius principals que em vaig plantejar a l'inici. També és cert que, durant el procés, aquests s'han anat veient matisats (de fer 3 o 4 jocs a un sol "súper" *Snake*) i les meves prioritats a l'hora de perseguir-los i explicar-los en aquest treball també han anat variant al llarg dels mesos. Així, tot i que la idea que en un principi havia tingut de com seria tot aquest projecte i el que ha acabat sent són prou diferents, la intenció de construir-me una consola i programar-ne tots els continguts s'ha vist realitzada igualment.

Tal com s'ha anat desenvolupant el treball, també he acabat entrant en temes i camps diferents amb els quals en un principi no havia comptat, com per exemple el disseny gràfic/artístic dels elements mostrats per pantalla del joc, o el procés creatiu que es va haver de donar abans per decidir què programar. Aquest és un clar exemple d'etapa del projecte a la qual en un principi no tenia previst donar gaire importància i, en canvi, més endavant vaig decidir donar-li un paper prou destacat respecte a la resta en el treball i també estudiar en les seves pàgines l'"Hidra", el mètode creatiu que vaig idear mentre es donava.

Per dur a terme el projecte, ha estat necessari seguir tot de passos que tampoc disten massa dels que habitualment hi ha darrere del procés de creació de qualsevol producte o joc que es pugui trobar al mercat. Això és: detecció d'unes necessitats a cobrir, selecció d'un objectiu a assolir, anàlisi dels recursos (físics, coneixements i capacitats) amb què es compta, recerca de les eines necessàries que no es tenen, disseny del producte a crear, aprenentatge constant en anar superant els diferents obstacles amb els que un es pot trobar durant un procés d'aquestes característiques, contínua revisió, adaptació i redefinició del projecte segons convé, acoblament i muntatge del conjunt de parts que formaran el resultat final i avaluació de la feina feta a l'acabar. En el món i mercat real, però, aquests passos es podrien complicar i ampliar més per tenir en compte altres factors com els costos de producció de l'objecte i la distribució o creació de marxandatge d'aquest, per exemple.

En cadascun d'aquests àmbits enumerats ha calgut invertir-hi una quantitat rellevant d'hores més o menys intenses de reflexió o de treball físic per arribar al producte final (en el disseny de més de 100 imatges o *sprites* i la programació de més de 2500 línies de codi en les quals fins a 100 funcions diferents gestionen les més de 260 variables i tipus de dades necessàries per a fer funcionar el joc).

Quant a la programació, no deixa de sorprendre el treball que requereix qualsevol petit assoliment, ja sigui definir què vols que es vegi per pantalla en un moment donat, com pautar un cert moviment, o dissenyar com es relacionen els diferents elements i fases del joc.

Així doncs, la valoració del treball només pot ser positiva tant pel que fa als resultats que ha donat com pels coneixements adquirits durant el seu transcurs. A més dels coneixements externs que he anat aprenent, també estic especialment satisfet de la descoberta personal de l'“Hidra”, que ha resultat un sistema prou distret i interessant d'estudiar i analitzar per anar-ne trobant les diferents implicacions i possibles usos i exemples.

Com en tot projecte, també hi ha hagut alguns moments d'esgotament i, en especial, fou d'allò més complicat i pesat repassar un cop més tot el procés dut a terme per redactar-ne la memòria posteriorment al seu desenvolupament. I és que m'he adonat com de difícil se m'ha fet provar de resumir i redactar prou clarament un projecte com aquest en què m'hi he bolcat tant i que vaig començar fa més d'un any i ha durat fins avui. D'aquí l'espessor general del document que teniu entre les mans, en el qual s'ha intentat que totes aquestes idees i matisos que s'han anat elucubrant durant tot aquest període hi quedessin prou ben recollits, sense caure en resultar massa opac per lectors menys entesos en el tema i procurant també que tampoc es passessin massa per alt alguns aspectes més o menys tècnics, igualment necessaris per entendre el treball.

Només em sap greu no haver tingut més temps a destinar en la part pràctica del treball, que ha estat sens dubte la que més he gaudit. Em queda pendent per fer, fora d'aquestes pàgines, seguir treballant en més prototips i versions més ben acabades de la consola que entrego juntament amb aquest treball. Alguns dels canvis que encara m'agradaria fer-hi són l'addició d'una bateria de liti pròpia i d'un interruptor d'encès i apagat per fer-la més independent encara, una millor distribució de l'espai i components físics, i també la incorporació d'algun sistema d'àudio a la consola que acompanyés al jugador en jugar. La construcció d'una carcassa que la contingui també era un dels principals objectius que m'havia proposat assolir en un començament però és quelcom que no he arribat a poder construir, i és que tenint en compte que voldria seguir treballant-hi més enllà d'aquest treball, vaig pensar que valdria més la pena esperar a tenir-la més acabada per ficar-me en el seu disseny.

Tot i això, estic prou satisfet i orgullós de la immensa feina que he fet i dels fruits que ha donat.

No vull acabar aquest treball sense donar les gràcies i el reconeixement que es mereix a [REDACTED], que m'ha fet de mentor durant tot el procés guiant-me i animant-me en les seves diferents etapes.



## 5. BIBLIOGRAFIA I WEBGRAFIA

### Manuels de programació

- SANDE, Carter; SANDE, Warren. *Hello World!: Computer Programming for Kids and Other Beginners*. Nova York: Manning, 2009.
- KERNIGHAN, Brian W; RITCHIE, Dennis M. *The C Programming Language*. Nova York: Pearson Education; Series: Always learning, 1988.

### Llenguatge de programació BASIC

- [https://en.wikipedia.org/wiki/PEEK\\_and\\_POKE](https://en.wikipedia.org/wiki/PEEK_and_POKE)
- <https://www.youtube.com/watch?v=seM9SqTsRG4>

### Jocs retro

- <https://en.wikipedia.org/wiki/Pong>
- [https://en.wikipedia.org/wiki/Snake\\_%28video\\_game%29](https://en.wikipedia.org/wiki/Snake_%28video_game%29)
- [https://en.wikipedia.org/wiki/Space\\_Invaders](https://en.wikipedia.org/wiki/Space_Invaders)
- <https://en.wikipedia.org/wiki/Pac-Man>
- [https://en.wikipedia.org/wiki/Donkey\\_Kong\\_%28video\\_game%29](https://en.wikipedia.org/wiki/Donkey_Kong_%28video_game%29)
- <https://en.wikipedia.org/wiki/Tetris>
- [https://en.wikipedia.org/wiki/The\\_Legend\\_of\\_Zelda](https://en.wikipedia.org/wiki/The_Legend_of_Zelda)
- [https://en.wikipedia.org/wiki/Mega\\_Man](https://en.wikipedia.org/wiki/Mega_Man)

### Mètodes creatius (per comprovar l'originalitat d'"Hidra")

- [https://en.wikipedia.org/wiki/Creativity\\_techniques](https://en.wikipedia.org/wiki/Creativity_techniques)
- [https://ca.wikipedia.org/wiki/Hidra\\_de\\_Lerna](https://ca.wikipedia.org/wiki/Hidra_de_Lerna)

### Xip FT232H

- <https://learn.adafruit.com/adafruit-ft232h-breakout/overview>

### Pantalles LCD i " 2.7" Monochrome 128x64 OLED Display Module "

- <https://learn.adafruit.com/2-7-monochrome-128x64-oled-display-module/overview>
- <https://www.youtube.com/watch?v=hZRL8luuPb8>
- <https://www.youtube.com/watch?v=vV8FbwobrKY>

### Condensadors

- <https://nergiza.com/condensadores-que-son-y-para-que-sirven/>

### Placa SAMD21 Mini

- <http://www.instructables.com/id/Arduino-MINTIA-Game-Console-in-a-Candy-Box/>
- [https://cdn.sparkfun.com/assets/learn\\_tutorials/4/5/4/graphicalDatasheet-Mini.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/4/5/4/graphicalDatasheet-Mini.pdf)
- <https://learn.sparkfun.com/tutorials/samd21-minidev-breakout-hookup-guide/samd21-mini-breakout-overview>