

Creació del joc simon amb un llenguatge de programació



Nom: Naiara González Casanova
Grup: 4t ESO A
Tutor: Jaime Morcillo
Curs: 2020-2021
Data de lliurament: 9/4/2021

Índex

1. Introducció	1
2. Hipòtesi	2
3. Desenvolupament del tema	3
3.1 Llenguatge de programació	3
3.2 MIT App Inventor	4
3.2.1 Components	5
3.2.1.1 Components de la interfase d'usuari	6
3.2.1.2 Components de disseny	7
3.2.1.3 Components multimèdia	7
3.2.1.4 Components de dibuix i animació	9
3.2.1.5 Components del mapa	9
3.2.1.6 Components del sensor	10
3.2.1.7 Components socials	11
3.2.1.8 Components d'almacenament	12
3.2.1.9 Components de connectivitat	12
3.2.2 Blocs	12
3.3 Per què utilitzaré App inventor	12
3.4 Que és el Simon?	13
3.5 Programació del Simon	13
3.5.1. Disseny Simon	14
3.5.2. Blocs Simon	15
6. Webgrafia	27
7. Apèndix	28



1. Introducció

El meu treball consisteix a crear el lloc anomenat Simon amb un llenguatge de programació.

El Simon és un lloc que consta de quatre botons de diferents colors. Aquests colors s'il·luminen en un ordre aleatori formant una seqüència. Quan ja s'ha acabat la seqüència el jugador ha de recordar l'ordre en el qual s'han encès els botons i pulsar els botons repetint-lo. Si es fa malament els botons es tornaran a il·luminar en un ordre aleatori, en canvi si es fa bé anirà pujant el nivell de dificultat amb una seqüència més llarga.

En aquest món en el qual vivim s'estan deixant de costat els llocs de taula o que fomenten la memorística pels videojocs per això vaig decidir crear un lloc amb un llenguatge de programació perquè així la gent torni a jugar aquests jocs. A més últimament per la tecnologia la gent està perdent la seva capacitat de memòria, ja que tot es pot trobar a internet per això les persones en comptes de raonar una estona i trobar la solució que segurament coneixen la busquen en internet perquè és més fàcil i còmode. Per això vull utilitzar la tecnologia per a fomentar la memòria d'una forma entretinguda i que no resulti avorrida.

Primer el que faré serà buscar llenguatges de programació amb els quals puc realitzar el meu joc i miraré els pros i els contres per decidir amb quin em quedaré. Tot seguit investigaré més a fons sobre el llenguatge de programació que he escollit, els diferents components i el que he de saber per començar a programar. Tot seguit realitzaré el joc que he escollit i veuré si funciona bé o no, i en el cas que funcioni si puc fer alguna cosa per millorar-lo.



2. Hipòtesi

Per poder començar amb la recerca del treball, m'he plantejat la següent hipòtesi:

Serà possible realitzar el joc Simón amb un llenguatge de programació en un mes amb coneixements mínims de programació.

Per contrastar aquesta hipòtesi hauré de realitzar una recerca prèvia per tal de conèixer que he de saber per poder programar un joc, etc.

A partir d'aquí em sorgeixen unes qüestions, que són les següents:

- És possible realitzar el joc Simón amb un llenguatge de programació?
- Realitzar un joc sense coneixement de programació?
- És possible realitzar-la en menys d'un mes?



3. Desenvolupament del tema

3.1 Llenguatge de programació

Un llenguatge de programació és un llenguatge formal dissenyat perquè a través d'una sèrie d'instruccions permetre a un programador escriure un conjunt d'ordres, accions consecutives, dades i algorismes per crear programes que controlin el comportament físic i lògic d'una màquina. Bàsicament és un programa que té com a objectiu l'elaboració d'altres programes informàtics.

Els llenguatges de programació es poden classificar en:

- **Llenguatge de baix nivell:** Són llenguatges de programació que estan dissenyats per un hardware específic, per això per aplicar-ho correctament cal que el programador tingui coneixements sòlids del hardware. Treuen un molt bon profit al sistema per al qual van ser dissenyats però no es poden aplicar per a cap altre.
Exemples d'aquest llenguatge són el llenguatge màquina, és l'únic llenguatge que entén la màquina i és interpretat directament pel processador. Aquest llenguatge es forma per la successió de zeros i uns, anomenat codi binari. També està el llenguatge ensamblador que va ser creat perquè al programador no li costés tant programar com li costaria si ho fes en el llenguatge màquina.
- **Llenguatge d'alt nivell:** Són llenguatges de programació que a diferència dels anteriors es poden aplicar en diferents tipus de sistema. Aquest llenguatge és molt més fàcil que entendre i s'apropa més al nostre llenguatge que al llenguatge màquina. Aquest llenguatge després és transformat al llenguatge màquina mitjançant intèrprets i compiladors. Els compiladors tradueixen al llenguatge màquina d'un sol cop i els intèrprets ho fan instrucció per instrucció.



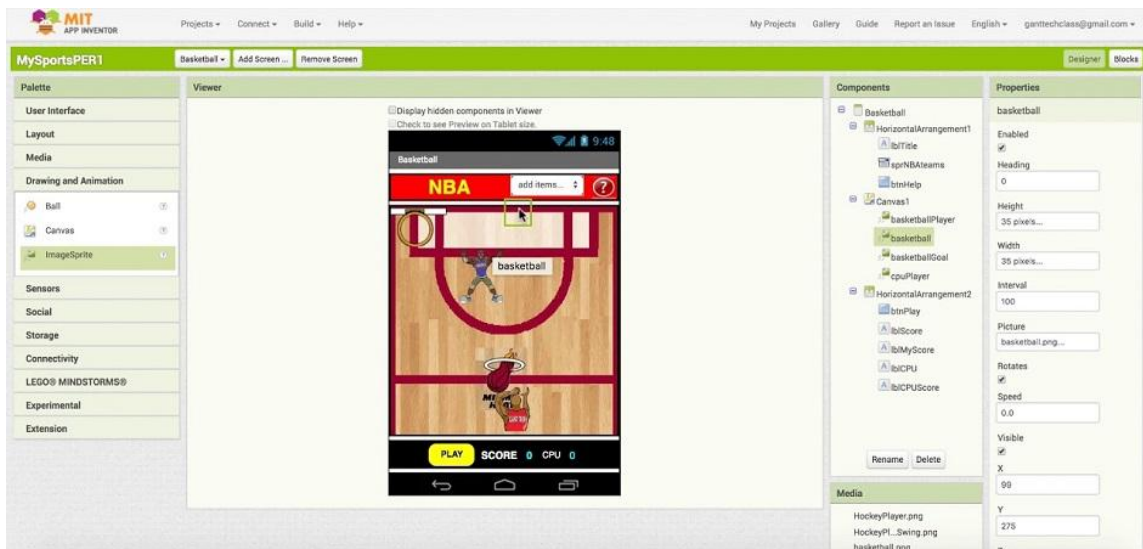
Imatge 1. Llenguatge d'alt nivell i llenguatge de baix nivell (llenguatge maquina). *Imatge estreta de:* <https://www.diarlu.com/lenguajes-de-programacion/>

3.2 MIT App Inventor

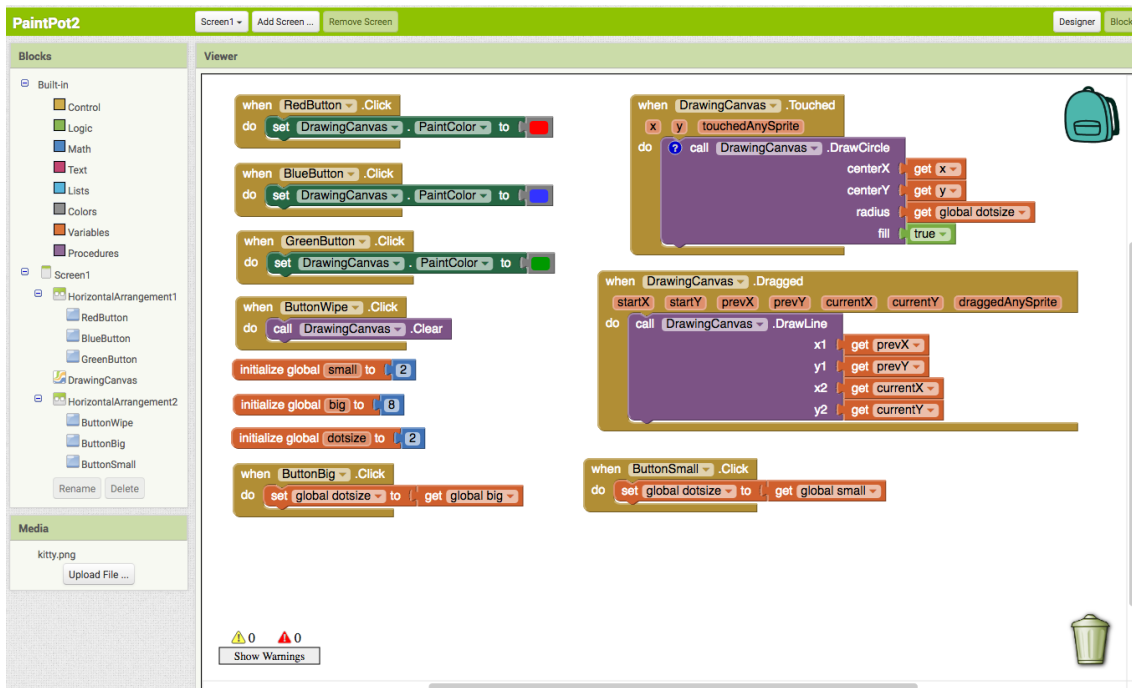
MIT App Inventor és un entorn de programació intuïtiu i visual que permet a les persones de totes les edats crear aplicacions completament funcionals per a telèfons mòbils i tauletes.

App Inventor està basat en blocs cosa que facilita la creació d'aplicacions molt més complexes. Amb el projecte d'App Inventor es busca que totes les persones, especialment els més joves puguin passar de consumir tecnologia a crear tecnologia i desenvolupar la seva creativitat, imaginació i la seva capacitat intel·lectual creant les seves pròpies aplicacions i fomentar l'ús i la interacció amb la tecnologia, un pilar molt important per al futur i el desenvolupament de la humanitat.

App Inventor consta de dues parts ben separades i diferenciades entre elles. La primera, el dissenyador, és tota la gran varietat de components que s'utilitzen per a la creació visual i el disseny de l'aplicació i després estan els blocs en l'editor de blocs que s'utilitzen per programar l'aplicació.



Imatge 2. Dissenyador d'App Inventor. Imatge estreta de <https://juegosrobotica.es/podcast-031/>



Imatge 3. Editor de blocs d'App Inventor. Imatge estreta de <https://www.formacionprofesional.info/guia-rapida-de-app-inventor/>

3.2.1 Components

Cada component pot tenir mètodes, esdeveniments i propietats. La gran part de les propietats es poden canviar per aplicacions, aquestes propietats tenen blocs que es poden utilitzar per obtenir els valors, no per establir-los. Cada



element s'anota amb text addicional per indicar el tipus d'element, si és de només lectura i si només existeix en el dissenyador o blocs.

3.2.1.1 Components de la interfase d'usuari

S'utilitzen perquè l'usuari pugui interactuar amb l'aplicació i moltes de les propietats de l'aspecte d'aquests components es poden modificar des de la part de disseny o en l'editor de blocs.

Dins d'aquests components podem trobar:

- Botó: Té la capacitat de detectar clics que es pot canviar.
- Casella de verificació: És un component que genera un esdeveniment quan l'usuari el pressiona.
- Selector de data: És un botó que quan es polsa inicia un quadre de diàleg emergent per a permetre a l'usuari seleccionar al calendari gregorià una data.
- Imatge: És un component per mostrar imatges i animacions bàsiques.
- Etiqueta: És un component utilitzat per mostrar text.
- Selector de llista: És un botó que quan es pressiona mostra una llista de textos perquè l'usuari els esculli.
- Visor de la llista: És un component visible que permet col·locar una llista d'elements de text a la pantalla per mostrar-los.
- Notificador: És un component que mostra missatges d'alerta i crea entrades de registre d'Android a través d'una varietat de mètodes.



- Camp de contrasenya: És un component que permet introduir contrasenyes, és pràcticament el mateix que un camp de text, però la seva diferència és que no es veu el que l'usuari està escrivint.
- Lliscador: És un component que mostra un control lliscant
- Desplegable: És un component que mostra un quadre de diàleg amb una llista d'elements.
- Switch: És un component que pot detectar els tocs que exerceix l'usuari i poden canviar el seu estat booleà en resposta.
- Camp de text: És un component que permet introduir text.
- Selector d'hora: És un botó que quan es pressiona obre un quadre de diàleg que permet introduir l'hora.
- VisorWeb: És un component que et permet veure pàgines web.

3.2.1.2 Components de disseny

Aquest component no són visibles per a l'usuari ni els utilitza. Aquests components li serveixen al creador de l'aplicació per distribuir i ubicar la resta de components visibles per a l'usuari. Hi ha diferents tipus, per col·locar els components en vertical, horitzontal, etc. Tots ells tenen la mateixa funció i diferents aspectes d'aquests components es poden modificar de la mateixa manera que els anteriors.

3.2.1.3 Components multimèdia

Aquests components tenen relació amb l'àudio o els vídeos i les imatges.

Dins d'aquests components podem trobar:

- Gravador: És un component que serveix per gravar un vídeo amb la càmera del dispositiu.



- Càmera: Aquest component serveix per fer una foto amb la càmera del dispositiu.
- Selector d'imatges: Quan l'usuari polsa aquest botó apareix la galeria d'imatges del dispositiu. Quan es selecciona una imatge es guarda. No es podran almacenar més de deu fotos, ja que si hi ha deu fotos i es selecciona una la més antiga s'esborrarà per poder guardar la més recent.
- Reproductor: És un component que reproduïx àudio i controla la vibració del dispositiu. Aquest components millor per àudios llargs com cançons.
- So: Aquest component reproduïx arxius de so. Per aquest component és millor els arxius de so curts com els efectes de so.
- Gravador de sons: És un component la funció del qual és gravar àudio.
- Reconeixement de veu: Aquest component escolta la veu de l'usuari que parla i converteix aquest so parlat en text mitjançant la funció de reconeixement de veu del dispositiu.
- Text a veu: Aquest component serveix per escriure automàticament el text que diu la veu de l'usuari. Es poden establir diferents idiomes.
- Reproductor de vídeo: És un component per reproduir vídeos. Inclou els controls per reproduir, aturar, saltar cap endavant i saltar cap enrere el vídeo.
- Traductor Yandex: Aquest component serveix per traduir paraules o frases. Aquest component necessita internet, ja que funciona amb el servei de Yandex.Translate.



3.2.1.4 Components de dibuix i animació

Moltes de les propietats de l'aspecte d'aquests components es poden modificar des de la part de disseny o en l'editor de blocs. Aquests components tenen relació amb les coordenades (X,Y) i el moviment.

- Pilota: És un component que es pot col·locar a un llenç, on pot reaccionar a tocs i arrossegades, interactuar amb altres sprites (tant pilotes com imatges sprite) i moure's segons els seus valor de propietat.
- Llenç: És un panel rectangular tàctil de dues dimensions en el qual es pot dibuixar i es poden moure sprites. Qualsevol ubicació del llenç s'especifica amb els valors (X, Y). X és el nombre de píxels lluny del costat esquerre del llenç i Y és el nombre de píxels lluny del costat dret del llenç.
- Imatge sprite: És un component que es pot col·locar a un llenç, on pot reaccionar a tocs i arrossegades, interactuar amb altres sprites (tant pilotes com imatges sprite) i moure's segons els seus valor de propietat. La seva diferència amb la pilota és que la imatge sprite pot obtenir la seva aparença d'un arxiu d'imatge i la pilota no.

3.2.1.5 Components del mapa

Dins d'aquests components podem trobar:

- Cercle: Aquest component crea un cercle d'un radi i centrar en una longitud i latitud.
- Col·lecció de característiques: És un component que agrupa diverses entitats de mapa.



- Cadena de línea: Aquest component serveix per dibuixar una seqüència de línies en un mapa
- Mapa: És un contenidor de dues dimensions que representa icones de mapa en un segon pla y permet que els elements marcador identifiquin punts en el mapa.
- Marcador: Aquest component serveix per indicar punts en el mapa. Aquests marcadors poden ser personalitzats.
- Navigació: És un component que entre dos ubicacions genera direccions.
- Polígon: Aquest component serveix per crear polígons que ocupen una àrea de dues dimensions al mapa.
- Rectangle: Aquest component crea rectangles amb latituds fixes i longituds per als límits dels punts cardinals.

3.2.1.6 Components del sensor

Aquests components són diferents sensors que mesuren diferents magnituds.

Dins d'aquests components podem trobar:

- Acceleròmetre: Pot detectar l'agitació i mesurar l'acceleració
- Lector de codi de barres: Aquest component serveix per escanejar un codi QR i recuperar el resultat
- Baròmetre: És un component que pot mesurar la pressió de l'aire del medi ambient.
- Rellotge: Component per mesurar i controlar el temps.



- Sensor Giroscopi: És un component que proporciona dades del sensor de giroscopi del dispositiu.
- Higròmetre: Aquests component mesura la humitat relativa del medi ambient
- Sensor de llum: Aquest component permet mesurar el nivell de llum
- Sensor d'Ubicació: És un component que dona informació sobre la posició, la ubicació, la velocitat i la direcció
- Sensor de camp magnètic: És un sensor que mesura el camp magnètic
- NearField: Aquest component només admet la lectura i escriptura d'etiquetes de text
- Podòmetre: Aquest component manté el recompte de passes utilitzat en l'acceleròmetre.
- Sensor de proximitat: És un component que pot mesurar la proximitat d'un objecte.
- Termòmetre: És un component que pot mesurar la temperatura de l'aire que és a l'ambient

3.2.1.7 Components socials

Aquests components permeten per exemple seleccionar contactes, seleccionar emails, trucar per telèfon, seleccionar un número de telèfon, compartir missatges o arxius, enviar textos, etc.



3.2.1.8 Components d'almacenament

Aquests components permeten almacenar dades a un servidor de base de dades, per almacenar i recuperar arxius, etc.

3.2.1.9 Components de connectivitat

Aquests components serveixen per connectar el dispositiu a altres dispositius, per iniciar una altra aplicació d'App Inventor, etc.

3.2.2 Blocs

Els blocs integrats es poden utilitzar sense tenir res a veure amb els components del projecte. A més a més dels blocs de llenguatge cada component del projecte té els seus propis conjunts de blocs específics.

Dintre d'aquests blocs hi ha:

- Blocs de control
- Blocs lògics
- Blocs matemàtics
- Blocs de text
- Llistes de blocs
- Blocs de diccionaris
- Blocs de colors
- Blocs de variables
- Blocs de procediments

3.3 Per què utilitzaré App inventor

Utilitzaré app Inventor per a crear el joc Simon perquè és un entorn de programació bastant fàcil gràcies al fet que la programació es fa mitjançant blocs i complet perquè té molts components diferents que faciliten la creació de diferents tipus d'aplicacions totalment funcionals.



App Inventor és fàcil i pràctic d'utilitzar i no et calen molts coneixements sobre programació. Gràcies a la programació en blocs no calen molts coneixements sobre la programació i són molt intuïtius i lògics d'utilitzar. Té una part per dissenyar gràficament l'aplicació molt simple d'utilitzar i modificar amb molta diversitat de components per poder crear una àmplia quantitat d'aplicacions diferents. Com no tenia molt temps per aprendre programació amb un altre entorn de programació i aquest és complet i fàcil de dominar ho vaig escollir.

3.4 Que és el Simon?

El Simon és un joc de memòria inventat per Ralph H. Baer i Howard J. Morrison. Va ser creat i comercialitzat per Milton Bradley Company (MB) fundada en 1860 per Milton Bradley i des de 1984 fins a l'actualitat és una divisió de la companyia més gran del món, Hasbro. Simon va ser comercialitzat per primera vegada l'any 1978. Va tenir molt èxit i va vendre milions d'unitats per tot el món.

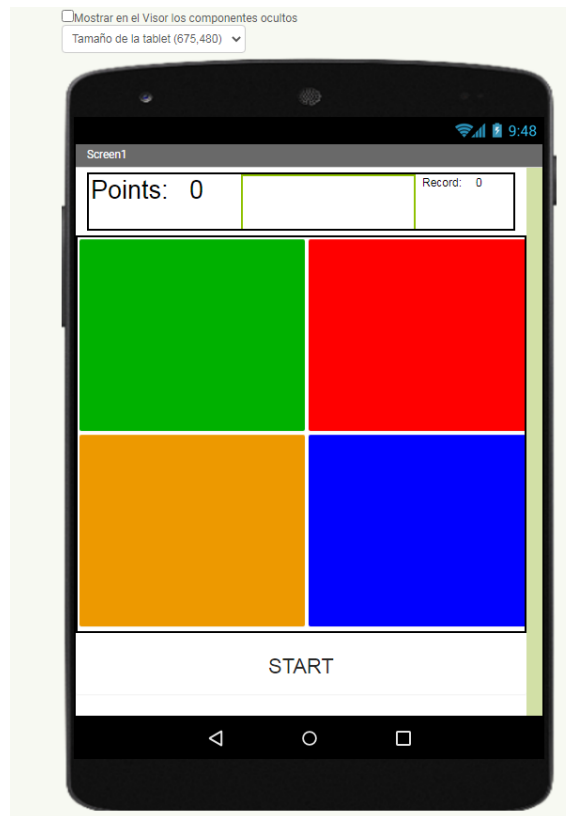
El joc Simon té una forma circular amb quatre botons de diferents colors (vermell, blau, verd i groc) i que emiten un so diferent i propi, a més els sons són harmònics entre si. Els botons s'il·luminen aleatòriament i el jugador ha de pulsar els botons en la mateixa seqüència de llum sense equivocar-se. La seqüència de botons que s'ha de repetir comença amb un únic botó i a mesura que el jugador aserti aquesta anirà incrementant amb un nou botó aleatori. El jugador perd al cometre un error o gana si aconsegueix repetir correctament la seqüència un nombre predeterminat de vegades.

3.5 Programació del Simon

A continuació explicaré com he realitzat el joc Simon, els components que he utilitzat i quins passos he seguit per programar-ho.

3.5.1. Disseny Simon

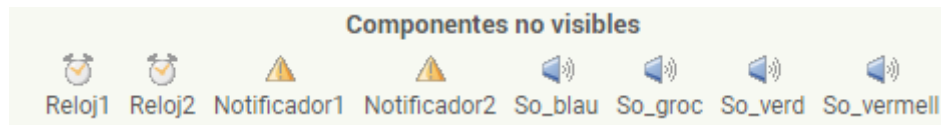
El primer que vaig fer per programar el joc Simon va ser realitzar el disseny que tindria la meua aplicació i escollir els components necessaris.



Imatge 4. Disseny Simon. *Imatge pròpia.*

A la imatge de dalt es veu el disseny de la meua aplicació. Consta d'un marcador, situat a la part superior i esquerra de la pantalla, que cada vegada que es polsi un botó correctament sumarà un punt. Al costat d'aquest cap a la dreta hi ha un marcador on es veu la puntuació més gran que has realitzat fins al moment. Aquests dos marcadors estan dins d'una disposició horitzontal la funció de la qual és posar els components en vertical. Situat a baix d'aquests components es troben quatre botons de diferents colors (verd, vermell, groc i blau) que s'encendran aleatòriament. Cada botó té unes dimensions d'amplada del 50% de la pantalla i d'altura del 35% de la pantalla. Estan col·locats en una disposició tabular. Sota els botons de colors està situat el botó de començament per començar la seqüència aleatòria.

A més dels components visibles que es veuen també hi ha d'altres que no es veuen.



Imatge 5. Components no visibles. Imatge pròpia.

La meua aplicació utilitza uns quants components no visibles. He posat dos rellotges, dos notificadors que serviran un pel missatge de què has fallat i l'altre pel missatge de superació de l'antic record. També he posat quatre sons, un per a cada botó perquè quan s'il·luminin soni el seu so característic.

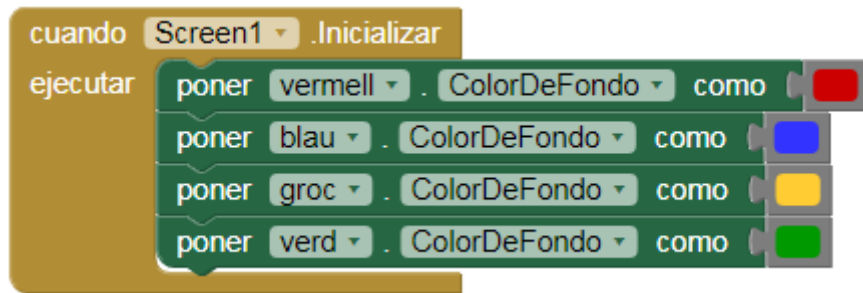
3.5.2. Blocs Simon

La meua aplicació està formada d'un seguit de blocs encaixats i relacionats entre si que fan que l'aplicació funcioni.



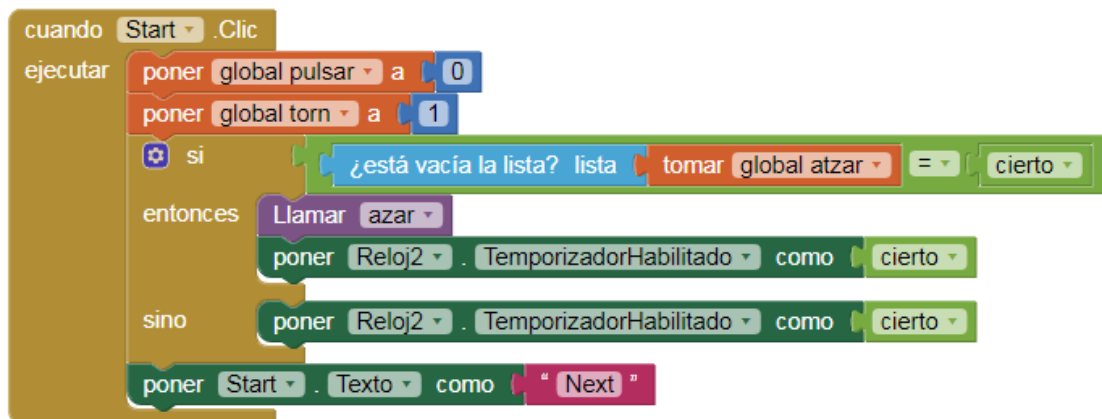
Imatge 6 . Variables. Imatge pròpia.

Primer vaig anar declarant variables que després en el procés de programació utilitzaré i explicaré la seva funció. Totes parteixen de 0 menys la variable “atzar” que correspon a una llista buida que he creat i utilitzaré durant el procés.



Imatge 7. Bloc pantalla. Imatge pròpia.

Aquest bloc és el primer que es realitzarà en la meva aplicació. Quan la meva aplicació s'inicia posaré el color de fons del botó vermell del color que es veu a la imatge i així amb els altres tres. Això ho faig perquè tot i que a la pantalla de disseny ja havia posat el color de fons dels botons als colors que volia, en la pantalla de disseny es poden seleccionar uns colors i als blocs uns altres per això quan després que els meus botons s'encenien i havien de tornar al color inicial aquest no tenia el mateix to que el primer.

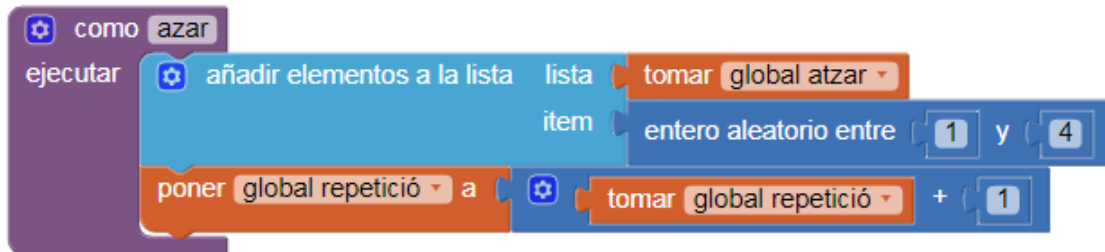


Imatge 8. Bloc començament. Imatge pròpia.

Aquest bloc es posarà en marxa quan el jugador pressioni el botó de començament. El que farà serà posar la variable "pulsar" a 0, això vol dir que la variable "pulsar" en aquest moment té el valor de 0. A més també canviaré el valor de la variable "torn" a 1.

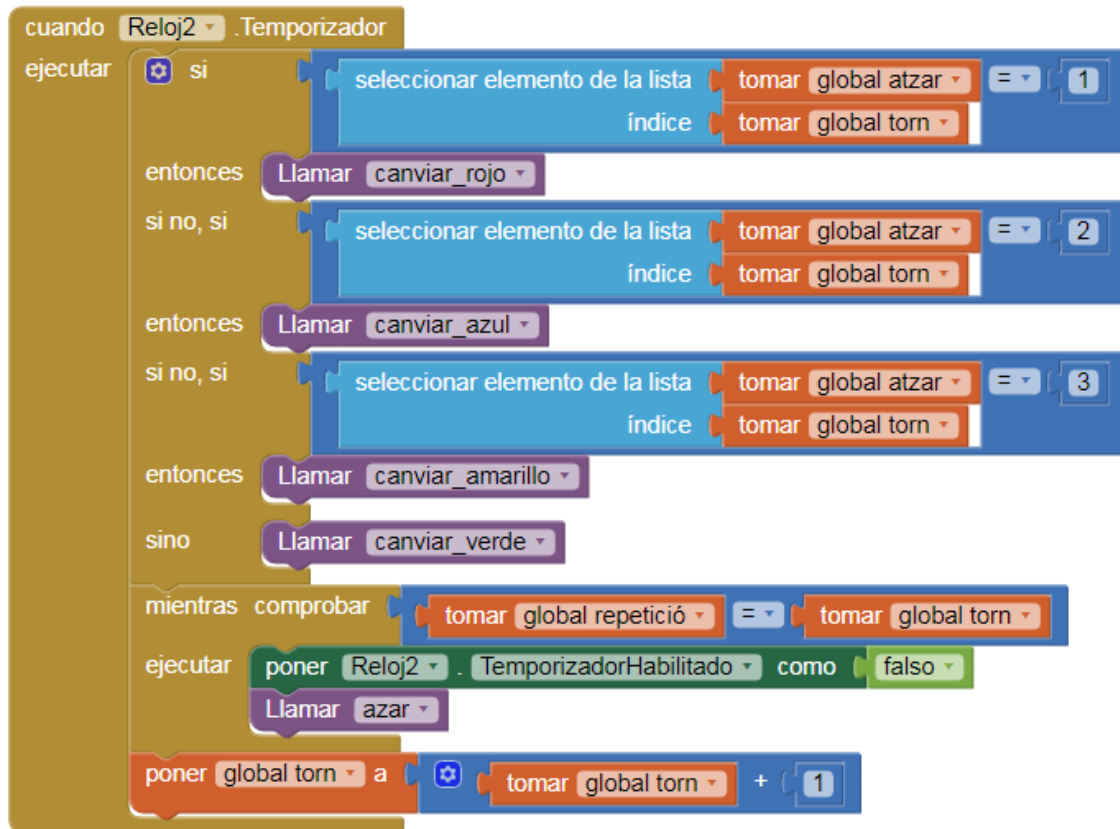
Ara preguntarà si és cert que la llista "atzar" que al principi havia creat que està buida. Si està buida llavors es trucarà al procediment "azar" i després

s'habilitarà el temporitzador perquè al principi no estava habilitat. Si la llista en canvi no està buida només s'habilitarà el rellotge dos. Tot seguit el text del botó de començament que al principi de l'aplicació posava "Start" ara canviarà i posarà "Next". La llista "atzar" només estarà buida al principi de cada partida.



Imatge 9. Procediment "azar". Imatge pròpia.

Al bloc d'abans quan la llista estava buida, que només passarà això al principi de cada partida, es trucava al procediment atzar que és aquest. Cada cop que es truqui a aquest procediment afegirem un nou ítem a la llista atzar que vaig crear. Aquest ítem que s'afegirà serà un valor aleatori enter entre l'1 i el 4 amb dos inclosos. A més la variable repetició es posarà a la suma de la variable "repetició" més 1, això vol dir que si en aquell moment la variable "repetició" té el valor 2 doncs ara tindrà el valor de $2 + 1$.



imatge 10 . Bloc rellotge 2. *imatge pròpia.*

El bloc del rellotge 2 s'activarà quan s'activi el temporitzador al bloc de començament després d'executar si cal el procediment “azar”.

El primer que pregunta aquest bloc és si l'element de la llista “azar” amb l'índex corresponent a la variable “torn” que és la que sempre que es polsa el botó de començament agafa un valor d'1, és igual a 1. La llista al principi estava buida i es va omplint amb nombres aleatoris, si la variable “torn” té un valor d'1, s'agafa el primer element de la llista. Aquest valor de la llista si és igual a 1 trucarà al procediment “canviar vermell” en canvi si és igual a 2 es trucarà al procediment “canviar blau”, si és igual a 3 trucarà al procediment “canviar groc” i si no es cap de les tres, es trucarà al procediment “canviar verd”. Aquesta part servirà perquè els botons s'encenguin de manera aleatòria.

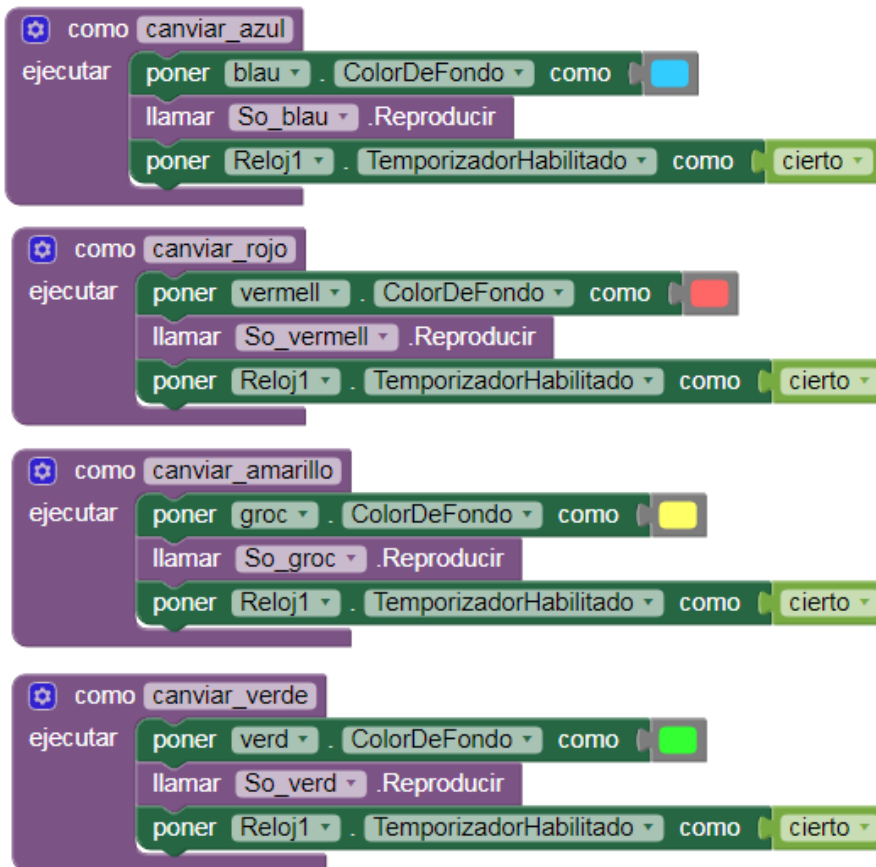
Tot seguit tenim un bloc que diu que mentre que el valor de la variable “repetició” sigui igual que la “variable torn” es posarà el rellotge 2 deshabilitat i es trucarà un altre cop al procediment “azar”. Si això no es compleix, passarà al



següent bloc. El següent bloc consisteix a posar la variable “torn” com el seu valor en aquell moment més 1.

La variable repetició sempre tindrà el valor del nombre de rondes que és fan perquè va augmentant cada ronda com els elements de la llista. En canvi la variable “torn” al principi de cada ronda sempre serà 1. Com que en el bloc rellotge es necessita que la variable “torn” i la variable “repetició” siguin iguals per desactivar-ho es repeteix.

Un exemple seria que estem a la ronda 2 per la qual cosa hi hauran dos elements en la llista i la variable “repetició” serà 2 mentre que la variable “torn” serà 1. Per ordre primer es compararà el primer objecte de la llista per saber quin botó encendre, com que la variable “torn” no és igual a la variable “repetició” es sumarà un 1 a la variable “torn” i es tornarà a fer el mateix. Es compararà el segon objecte de la llista, perquè ara la variable “torn” equival a 2, i es tornarà a encendre el botó corresponent i com ara la variable “torn” i la variable “repetició” són iguals sortim d’aquest bloc perquè aquest només s’executa quan el temporitzador del rellotge 2 està habilitat.



Imatge 11 .Procediments “canviar”. Imatge pròpia.

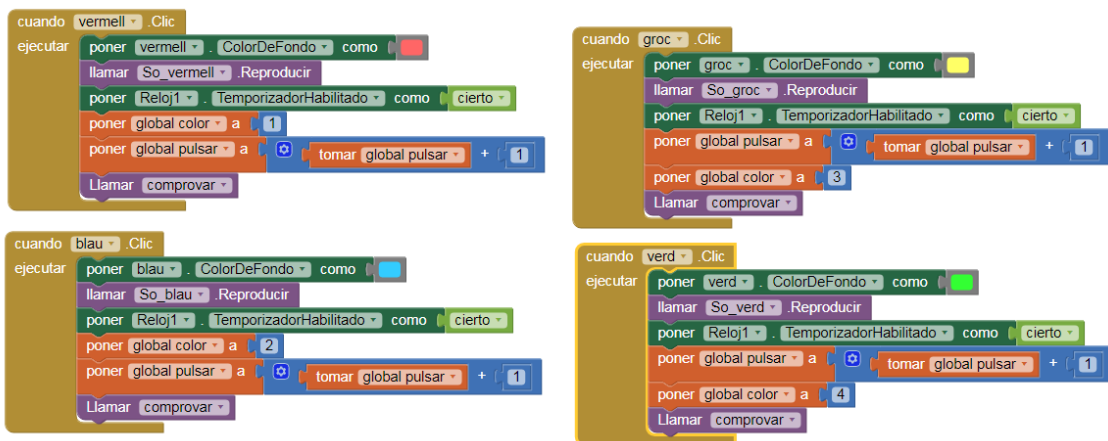
Els procediments per encendre els botons de colors són molt simples. Quan es truca a aquest procediment al bloc del rellotge 2 el primer que s’executa és que el color dels botons canvia a un més clar per fer l’efecte de què s’està il·luminant el botó. També es reproduïx un so diferent per a cada botó i finalment s’habilita el rellotge 1.



Imatge 12 . Bloc rellotge 1. Imatge pròpia.

El bloc del rellotge 1 s'activa després de canviar els botons a un color més clar. El que fa és posar tots els botons al color original per fer l'efecte de que s'apaga el botó i desactivem el rellotge 1.

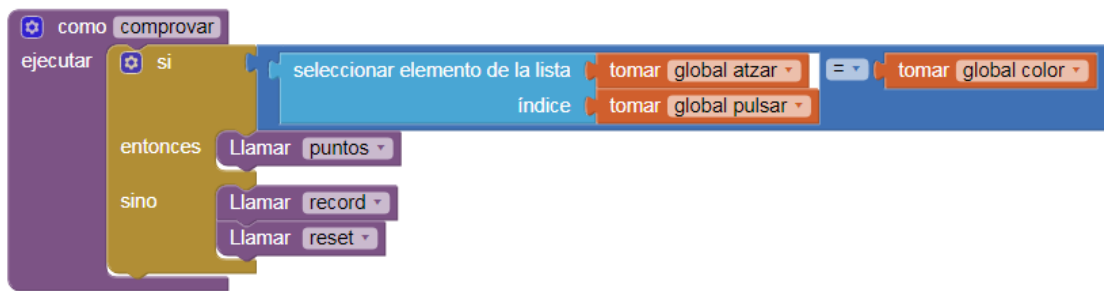
Fins aquí seria el procés que es fa des de que es polsa el botó per començar fins que la seqüència de colors s'acaba. Ara explicaré els blocs que intervenen en la repetició de la seqüència per part del jugador fins que es torna a començar una altra ronda.



Imatge 13 . Blocs botons. *Imatge pròpia.*

A continuació venen els blocs que intervenen quan el jugador toca els botons. Tots els botons tenen els mateixos blocs, l'únic que canvia és la variable "color" i el color del botó i so respectiu de cada botó.

Quan es polsa un botó el que passa és que es canvia el color a un més clar, sona el so i s'habilita el rellotge 1 per tornar a posar el botó del color original. Després d'això es posa la variable "color" al seu respectiu nombre per diferenciar quin color és, com en el bloc del rellotge 2 i a més la variable "pulsar" pren el valor del seu valor en aquell moment més 1. Tot seguit es truca al procediment "comprovar" que servirà per comprovar si el jugador ha pulsat el botó quan es devia.



Imatge 14. Procediment “comprovar”. Imatge pròpia.

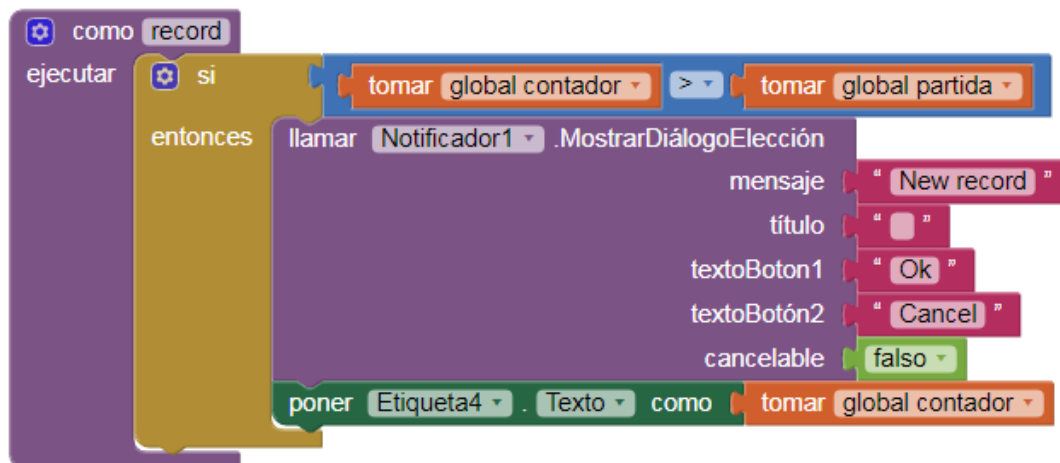
El procediment “comprovar” serveix per saber si has polsat els botons en l’ordre correcte. El primer que diu és que si l’element de la llista atzar que té com ordre la variable “pulsar” és igual a la variable “color” es trucarà al procediment punts, sinó es trucarà al procediment “record” i al procediment “reset”.

Un exemple seria que el jugador ha polsat el botó blau després d’haver polsat correctament l’anterior botó. En aquest cas la variable “pulsar” serà dos perquè cada cop que polses un botó es suma 1 i aquest és el segon botó que polsa. Per aquest motiu es veurà si el segon element de la llista és igual a la variable “color”. Si és igual es trucarà al procediment punt per sumar un punt per cada botó pressionat correctament, siné es trucarà a “record” i “reset” que faran que la partida es torni a començar.



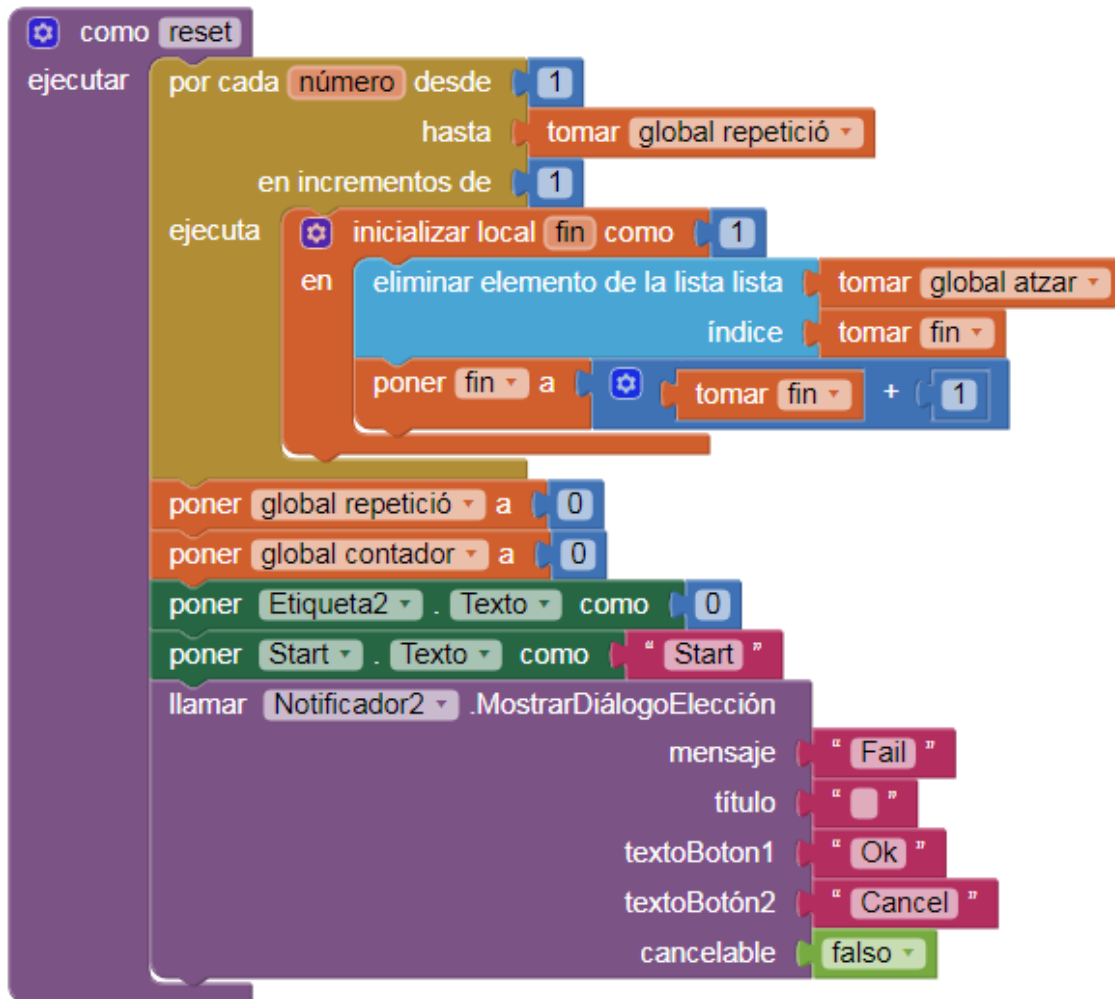
Imatge 15 .Procediment “puntos”. Imatge pròpia.

El procediment “punt” fa que en el marcador se sumi un punt cada cop que s’encerta un botó. Compta amb una variable que anirà incrementant el seu valor en 1 i una etiqueta que és el marcador on es veurà reflectit el valor de la variable “contador”.



Imatge 16.Procediment “record”. Imatge pròpia.

El procediment “record” consisteix a mirar si la puntuació de la partida del jugador que correspondria a la variable “contador” és més petita que la puntuació més alta fins ara que es representa en la variable “partida”. Si és així es mostrarà un missatge dient que ha aconseguit un nou record i l’etiqueta quatre que és la que correspon al marcador del record es posarà el valor de la variable “contador”. Si la variable “contador” no és més petita que la de “partida” no es farà res.



Imatge 17.Procediment "reset". Imatge pròpia.

El procediment "reset" elimina tots els elements de la llista. També posa la variable "repetició" a 0, la variable "contador" també a 0 i el text de l'etiqueta del marcador de la partida també a 0. Es torna a posar el text del botó de començament de "Next" a "Start". També es truca al notificador 2 per mostrar un missatge que digui que el jugador ha perdut.



4. Conclusions

Al final he aconseguit crear el joc Simon amb un llenguatge de programació, concretament App Inventor.

L'aplicació funciona correctament, té un marcador per anar sumant punt cada botó encertat i es registra la puntuació més alta del jugador. Tot això és a part de la funció principal del joc que és treballar la memòria repetint la seqüència de botons que es van repetint sumant un nou botó a cada ronda. També l'he afegit un so diferent per a cada un dels quatre botons quan s'il·luminen. A més els sons dels botons, com al joc Simon original, són harmònics entre ells per això dóna igual en què ordre sonin els botons perquè la combinació sempre li sonarà bé a l'orella.

Vaig escollir un llenguatge de programació que fos ràpid i fàcil d'aprendre pel poc temps amb el qual disposava que era amb un mes. Amb els meus coneixements mínims de programació i amb la investigació i la recerca d'informació per saber com utilitzar App Inventor va ser suficient per poder crear el Simon amb el seu funcionament bàsic i alguns components més.

A més per la forma en què està creat aquest Simon no té límit en la seva seqüència, si no es falla, es podria estar jugant infinitament. També vaig afegir els punts i el marcador amb la puntuació més alta per crear competitivitat. Així el jugador pot aspirar a anar superant el seu record i entretenir-se superant-se contínuament.

El temps del qual disposava tot i ser força limitat per a tot el procés que havia de fer per crear el joc Simon amb un llenguatge de programació a estat suficient per poder entendre, dissenyar i programar la meva aplicació.



5. Agraïments

Agraeixo al meu professor encarregat del treball de recerca per l'ajuda a l'hora de decidir el tema a tractar en aquest treball i els consells i l'ajuda a l'hora de la resolució d'alguns dubtes.



6. Webgrafia

CLARÍN. *Procesos mentales El impacto de las tecnologías en la memoria*
<https://www.clarin.com/sociedad/impacto-tecnologias-memoria_0_BkhnLDq4G.html> [Consulta: 3 de març 2021]

MIT APP INVENTOR. *About us.*
<<https://appinventor.mit.edu/about-us>> [Consulta: 15 de març 2021]

MIT APP INVENTOR. *Component reference*
<<http://ai2.appinventor.mit.edu/reference/components/>> [Consulta: 15 de març 2021]

Formación profesional. *Guía rápida de App Inventor*
<<https://www.formacionprofesional.info/guia-rapida-de-app-inventor/>> [Consulta: 15 de març 2021]

DIARLU. *Los tipos de lenguajes de programación: bajo y alto nivel*
<<https://www.diarlu.com/lenguajes-de-programacion/#:~:text=Los%20lenguajes%20de%20bajo%20nivel%20son%20aquellos%20que.procesador%20y%20dependen%20de%20la%20arquitectura%20del%20hardware.>>> [Consulta: 16 de març 2021]

CCM. *¿Qué es un lenguaje de programación y qué tipos existen?*
<<https://es.ccm.net/contents/304-lenguajes-de-programacion>> [Consulta: 16 de març 2021]

Concepto.de. *Lenguaje de programación*
<<https://concepto.de/lenguaje-de-programacion/>> [Consulta: 18 de març 2021]

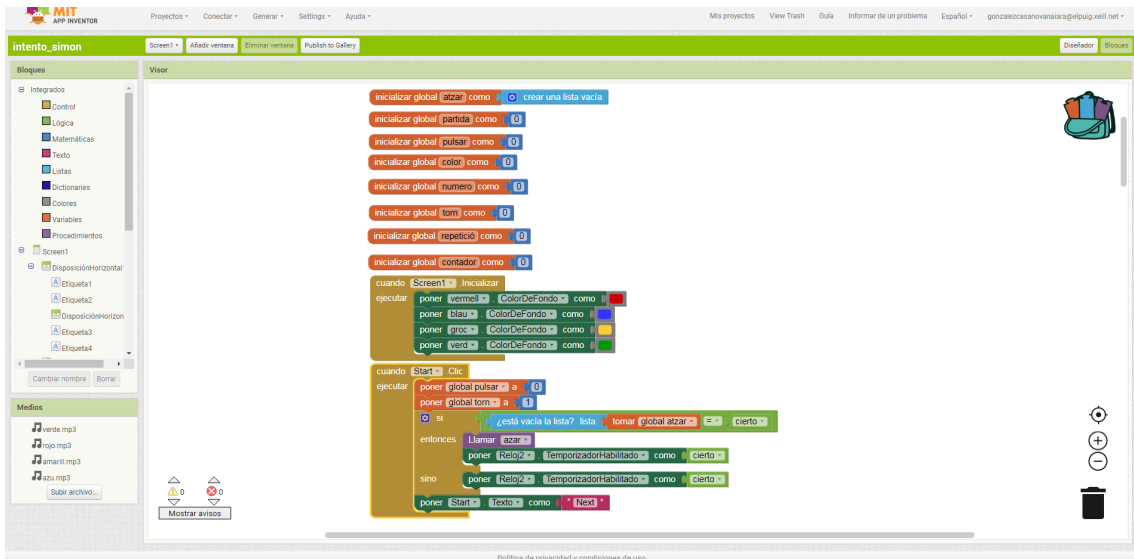
Pixfans. *El juego ochentero de MB*
<<https://www.pixfans.com/simon-el-juego-ochentero-de-mb/>> [Consulta: 19 de març 2021]



Encyclopedia. *Milton Bradley Company*

<<https://www.encyclopedia.com/books/politics-and-business-magazines/milton-bradley-company>> [Consulta: 20 de març 2021]

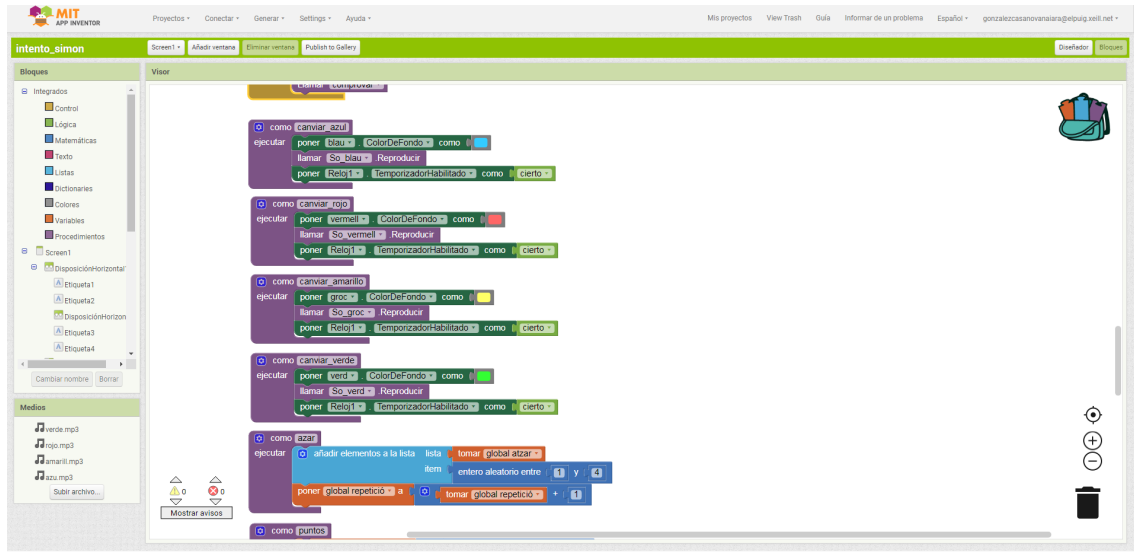
7. Apèndix



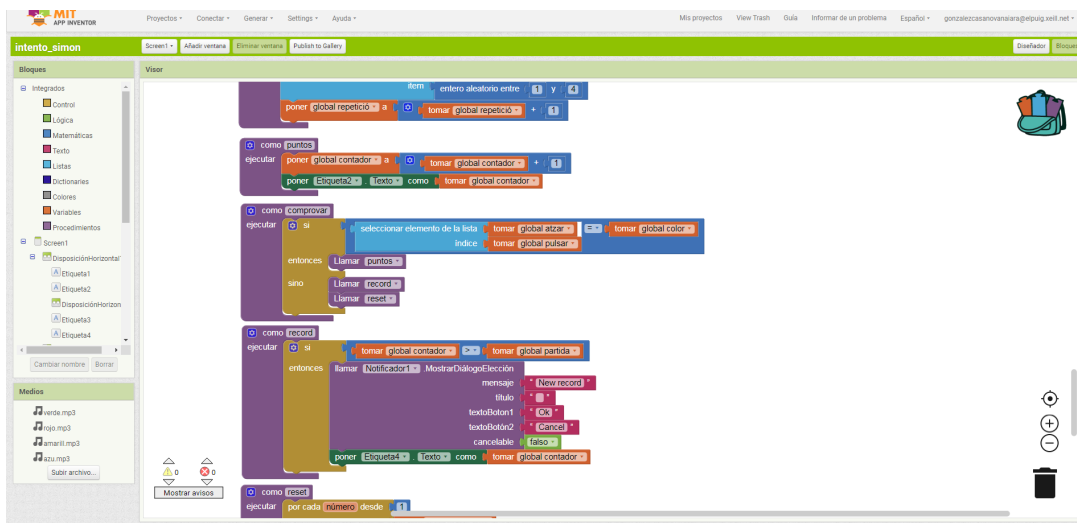
Imatge 1 Annex: Programació App inventor. *Imatge pròpia.*



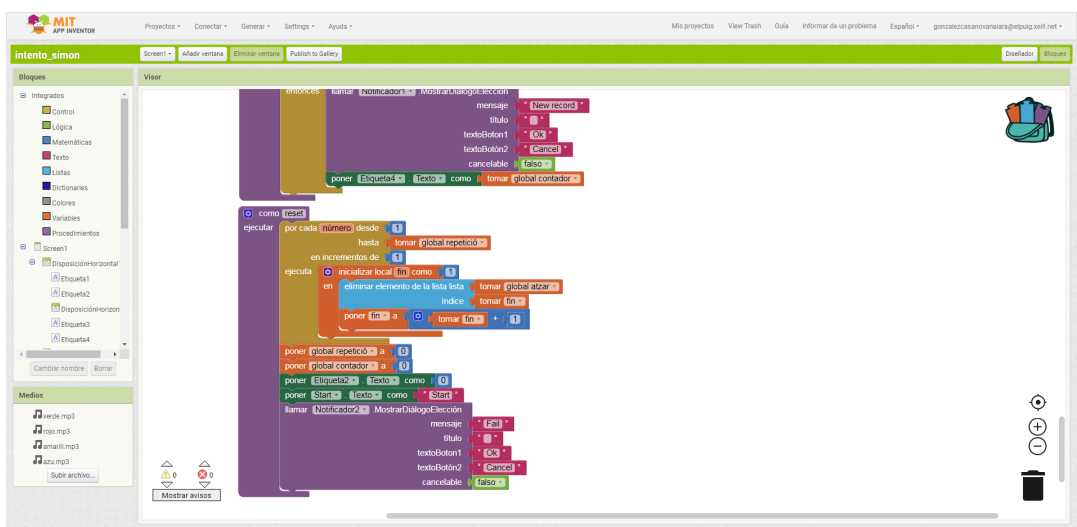
Imatge 2 Annex: Programació App inventor. *Imatge pròpia.*



Imatge 3 Annex: Programació App inventor. *Imatge pròpia.*



Imatge 4 Annex: Programació App inventor. *Imatge pròpia.*



Imatge 5 Annex: Programació App inventor. *Imatge pròpia.*