



**Treball de Fi de Grau**

**GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques**

**Universitat de Barcelona**

---

**PROGRAMA DIDÀCTIC PER A NENS A L'IPAD**

---

**Rubén Fernández Nicolau**

Directora: Dra. Maite López Sánchez

Realitzat a: Departament de  
Matemàtica Aplicada i  
Anàlisi. UB

## Agraïments

La realització d'aquest projecte no hagués estat possible sense la col·laboració de diverses persones, a totes elles els hi dedico amb afecte.

A la doctora Maite López Sánchez per la seva confiança, supervisió, suport, correccions, ànims i crítiques. Gràcies per tot, aquest projecte no seria el mateix sense la teva ajuda i orientació.

Als nens i nenes que van regalar el seu temps i col·laboració provant l'aplicació. Ada Espinosa López, Margot Roperó Garcia, Violeta Escobar López i Valentín Escobar López. Gràcies per la vostra paciència, sense vosaltres LogiK no hagués estat possible.

A la meva família, sense el seu amor incondicional, ànims, suport i confiança no estaria escrivint aquestes línies. Gràcies per ajudar-me a ser qui sóc i a assolir els meus objectius.

A totes les persones que m'han il·luminat el camí amb la seva amistat, confiança, amor i suport. Sense vosaltres no seria qui soc avui.

## Índex

<b>1. Summary</b> .....	4
<b>2. Introducció i antecedents</b> .....	5
<b>3. Objectius i motivació del problema</b> .....	6
<b>4. Motors de creació d'aplicacions investigats</b> .....	8
4.1 Codea.....	10
4.1.1 Introducció .....	10
4.1.2 Descripció de funcionalitats .....	11
4.1.3 Lua .....	16
4.1.4 Codi d'exemple.....	17
4.2 Mobincube .....	19
4.2.1 Introducció .....	19
4.2.2 Descripció de funcionalitats .....	20
4.2.2.1 Creació.....	20
4.2.2.2 Edició .....	21
4.2.2.3 Disseny i visualització .....	22
4.2.2.4 Publicació .....	23
4.3 Construct 2 .....	24
4.3.1 Introducció .....	24
4.3.2 Explicació de funcionalitats.....	25
4.3.2.1 Layouts .....	25
4.3.2.2 Objectes.....	26
4.3.2.3 Behaviors.....	26
4.3.2.4 Sistema d'events .....	27
4.3.2.5 Preview i exportació.....	28
4.4 Gamesalad.....	29
4.4.1 Introducció .....	29
4.4.2 Explicació de funcionalitats.....	30
4.4.2.1 Scenes.....	30
4.4.2.2 Actors .....	31
4.4.2.3 Rules & Behaviors.....	32
4.4.2.4 Atributs o variables .....	34
4.4.2.5 Imatges i Sons.....	35
4.4.2.6 Taules .....	36
4.4.2.7 Exportació i publicació .....	37

<b>5. Motor escollit: Justificació</b> .....	39
<b>6. Explicació del projecte</b> .....	41
6.1 Anàlisi de requeriments i funcionament.....	41
6.2 Casos d'ús.....	43
6.3 Diagrames de seqüència .....	46
6.4 Captures de codi.....	52
6.5 Estructura de l'aplicació .....	54
6.5.1 Components principals: <i>Actors</i> i <i>Tables</i> .....	54
6.5.1.1 Actors .....	54
6.5.1.2 Tables .....	59
6.6 Captures de pantalla .....	61
6.6.1 Captures del projecte principal .....	61
6.6.2 Captures en fase de desenvolupament.....	64
<b>7. Justificacions, problemes i dificultats trobades</b> .....	68
<b>8. Test d'usabilitat</b> .....	71
8.1 Resultats test.....	71
<b>9. Ampliació</b> .....	72
9.1 Canvis realitzats.....	72
9.2 Captures de pantalla .....	73
9.3 Test d'usabilitat.....	81
9.3.1 Resultats test.....	81
9.4 Treball futur.....	82
<b>10. Conclusions</b> .....	83
<b>11. Hores invertides</b> .....	84
11.1 Investigació.....	84
11.2 Desenvolupament Projecte.....	84
11.3 Redacció memòria.....	85
<b>12. Referències</b> .....	86
<b>13. Annexos</b> .....	87

## 1. Summary

Nowadays, there are lots of app creation engines on the market with the ability to export software on different platforms, such as HTML5, Android or iOS among others.

Despite their limitations, allow the creation of simple software and can be used for prototype designs, which can be used as a reference to make an application with the same characteristics, but designed in a different programming language which allows a better use of resources and a better inner structure.

The main advantage of these engines is the short period of time in which you can make a simple app.

This project was created with the idea of using one of these engines to create a prototype application and, considering the low market in the educational field, the application was redirected to children and want to help them to enhance logical thinking.

The program must accomplish three essential needs, which are:

- Should include the concept of *Loop* somehow.
- Must be at least of one level.
- It has to consist of a two-part screen where commands are entered on one side and, on the other, show the game screen.

The goal of the project is to investigate different engines in order to find one that allows the implementation of a prototype application that fits our needs and, using it, create and apply a usability test in order to improve and adapt it to the needs of the end user.

## 2. Introducció i antecedents

Actualment han sortit al mercat nombrosos motors de creació d'aplicacions amb capacitat per a exportar el producte final a diverses plataformes, com ara HTML5, Android o iOS entre d'altres.

Tot i tenir les seves limitacions, permeten la creació de software senzill i poden ser utilitzats per a la realització de prototips, que es poden utilitzar de referència per a la creació d'una aplicació amb les mateixes característiques, però dissenyada amb un llenguatge de programació que permeti un millor ús de recursos i una millor estructura de la aplicació.

L'avantatge principal que presenten aquests motors a simple vista és la rapidesa en la permeten tenir feta una aplicació senzilla.

Amb la idea d'utilitzar un d'aquests motors per a crear un prototip d'aplicació neix aquest projecte i, pensant en l'escàs mercat en l'àmbit educatiu, es decideix que l'aplicació estigui destinada a nens i els ajudi a potenciar el pensament lògic.

Es decideix que el programa ha de complir tres requisits imprescindibles, que són:

- Ha d'incloure el concepte de *Loop* d'alguna manera.
- Ha de ser com a mínim d'una pantalla.
- Ha de constar d'una pantalla dividida on per una banda s'introdueixin comandes i, per l'altra, es visualitzi el joc.

Es pretén doncs, investigar diferents motors de creació per tal de trobar-ne un que permeti la implementació d'un prototip d'aplicació que compleixi aquestes característiques i, amb l'ús del mateix, crear-la i aplicar-li un test d'usabilitat per provar de millorar-la i adaptar-la a les necessitats de l'usuari final.

### 3. Objectius i motivació del problema

L'objectiu del projecte és desenvolupar amb èxit una versió Beta d'aplicació per a l'iPad i que satisfaci certes necessitats o objectius principals.

Aquests objectius es basen en el tipus d'aplicació i en el que ha d'aconseguir transmetre als usuaris, en aquest cas nens de 5 a 8 anys. En els tests d'usabilitat que es pretenen realitzar, se'ls donaran les pautes a seguir, però la interfície ha de ser molt simple, ja que els nens de 5 anys tot just comencen a llegir.

Ha de ser un software educatiu en el que es millori el pensament de tipus lògic i els ajudi de forma inconscient a estructurar la ment i optimitzar els seus recursos cognitius a l'hora de realitzar múltiples accions. Es pretén fer per una banda usant el moviment espacial, en les 4 direccions dels eixos de coordenades, en dues dimensions.

Per l'altra, es vol transmetre el concepte d'executar accions de forma seqüencial i ordenada per a assolir un objectiu i, junt amb aquest, reforçar el concepte de repetir accions simples un número concret de vegades. Això últim, correspondria al concepte de *Loop*.

Implementant aquests conceptes en el software d'alguna manera, es podria ajudar a millorar les àrees cognitives relacionades amb el pensament lògic

El software ha de ser un joc, les propostes principals no requerien d'un llenguatge específic de programació ni de plataforma, tot i que sí que havia de ser mòbil i a poder ser d'iPad.

Com que no hi havia cap requisit en quant a llenguatge, es va proposar com a objectiu pensar en una idea o prototip de joc que compleixi els requisits exposats, trobar un motor de creació d'aplicacions que s'adapti a aquestes necessitats i que permeti tenir-lo funcional en el temps estipulat.

Per a crear el software i assegurar la seva qualitat i compliment dels objectius, es proposa també realitzar tests d'usabilitat a nens amb el software un cop sigui funcional per tal d'obtenir un feedback.

Aquest ens ha de permetre crear o aproximar el que seria la següent versió del software, que contindrà les optimitzacions oportunes segons el feedback obtingut d'aquests tests.

En resum, els objectius a complir per el software han de contenir essencialment les següents característiques :

- Per a nens i nenes de 5 a 8 anys, per tant s'ha d'adequar la interfície i usabilitat a aquesta franja d'edat. No ha de fer cap distinció de gènere, ha de poder ser jugat indistintament per nens i nenes.

- Ha de ser de tipus joc educatiu.
- Compatible amb dispositius mòbils. Android i/o iOS.
- Ha de constar d'una pantalla dividida on per una banda s'introdueixin comandes i, per l'altra, es visualitzi el joc.
- Ha d'incloure el concepte de *Loop* d'alguna manera.
- Ha de ser com a mínim d'una pantalla.

Per a desenvolupar un software d'aquestes característiques, s'han de tenir en compte els recursos i personal disponibles en vers aquests requisits.

Tenint en compte que el temps de realització i documentació d'aquest és limitat i la idea és presentar un software funcional, es va proposar la idea d'investigar diferents motors de creació de jocs per a dispositius mòbils i HTML5 que poguessin estalviar encerta mesura la feina de programació per tal de poder tenir el que seria una maqueta o versió beta del software per a poder analitzar-la.

A partir d'aquest anàlisi caldria debatre si seria o no una aplicació exitosa en el mercat i, per tant, si seria viable desenvolupar-la amb algun tipus de software més potent per a tenir una aplicació totalment atractiva, funcional i sòlida que pugui tenir una mínima sortida.

També es vol comprovar si seria viable acabar l'aplicació amb el motor escollit i distribuir-la directament si la seva funcionalitat ho permet, sense la necessitat d'implementar-la en un altre llenguatge.

A l'hora d'escollir el motor de creació, apart dels recursos utilitzats, entre les seves funcionalitats és important tenir en compte la seva potència en quant a la exportació a diferents plataformes.

Tot i que l'interès principal és una aplicació per a iPad, seria útil disposar d'un sistema de conversió el més ràpid i còmode possible per tal de poder acabar distribuint el software en més plataformes en un futur.



## 4. Motors de creació d'aplicacions investigats

Durant el desenvolupament de la idea del que anava a ser el projecte, es van investigar tres motors de creació d'aplicacions per a dispositius mòbils.

L'objectiu era fer una valoració de les eines, potències i mancances de cadascun d'ells a fi d'escollir el més òptim per al tipus de projecte que volem implementar.

Els motors investigats són :

- **CodeA** : De pagament.
- **Mobincube** : Gratuït, amb possibilitat d'ampliació de tipus de subscripció segons necessitats.
- **Gamesalad** : Gratuït, amb possibilitat d'ampliació de tipus de subscripció segons necessitats.
- **Construct 2**: De pagament, amb subscripció.

Aquests motors de creació són un software amb un interface que permet crear software, facilitant la feina de programació, per a poder abordar projectes de petita i mitjana envergadura amb molt menys temps que si es fessin directament amb el llenguatge de programació corresponent a la plataforma per a la que es vulgui programar (XCode, Android, HTML5,..).

Òbviament aquests motors comporten una sèrie d'avantatges e inconvenients que cal tenir en compte a l'hora de decidir amb quines eines voldrem desenvolupar el nostre projecte.

Es mostra a continuació un breu resum de les limitacions i avantatges que han estat més rellevants a l'hora de decidir-se a utilitzar un motor de creació d'aplicacions.

### **Avantatges :**

- Disposen d'una interfície '*user-friendly*'
- Permeten tenir software funcional en menys temps
- La documentació és en general molt completa i hi ha força suport online al respecte.
- Hi ha Assets gratuïts i de pagament per a ajudar amb la creació de la interfície gràfica.
- En pocs dies es pot aprendre el suficient per començar a desenvolupar software.
- Fan la conversió a diverses plataformes (segons el motor) de forma ràpida i de forma més o menys efectiva.

- Són de ràpida instal·lació i configuració.
- Agilitzen molt el desenvolupament de software de petita envergadura i/o amb necessitats molt determinades i “senzilles”.
- Permeten la creació de maquetes de projectes més grans i permet tenir una altra mirada als desenvolupadors.

**Limitacions :**

- Limitacions en quant a exportació, cada motor té les seves.
- No són útils per a projectes grans o amb necessitats molt específiques degut a les limitacions de cada motor.
- Limitacions estructurals, no permeten l'ús d'alguns recursos de programació que facilitarien la feina, s'ha d'adaptar el projecte a les característiques del motor, que solen ser reduïdes per tal de facilitar la creació.
- Possibles problemes en la exportació/conversió a les diferents plataformes.
- Limitacions a l'hora d'utilitzar recursos externs (Bases de dades, suport de xarxa, etc..).
- Les pròpies característiques dels motors obliguen a haver de tenir una excessiva atenció al consum de recursos que suposa l'aplicació, ja que solen consumir-ne molts.
- Requereixen d'un SO en concret, no solen ser universals.

Cal tenir en compte aquests avantatges i limitacions abans de començar a desenvolupar el software, ja que és crucial escollir si serà viable fer servir un motor o no, i en cas afirmatiu el escollir-ne un o altre afectarà directament en la metodologia a adoptar en la creació i desenvolupament del projecte.

A continuació, es presenta una descripció de les característiques de cadascun dels motors i de les possibilitats que ofereixen, així com les limitacions e impediments de cadascun d'ells.

L'estudi de GameSalad ha estat més exhaustiu ja que ha estat el motor escollit per a la creació del software i, per tant, s'han dedicat moltes més hores a la seva investigació i aprenentatge.

## 4.1 Codea



### 4.1.1 Introducció

La distribució d'aquest projecte va ser aprovada per Apple sota el nom de Codify, a finals del 2011 i en format d'aplicació iOS per a dispositius iPad. Més tard es va canviar el nom del programa a Codea.

Consisteix en una plataforma pensada per a desenvolupadors de videojocs, educadors i estudiants per a la creació de videojocs. Dóna suport i ajuda amb eines a nivell visual per a la creació de codi.

Al ser una app, permet desenvolupar projectes directament des de l'iPad sense necessitat de més eines, gràcies a un *interface* molt treballat per a facilitar la edició de codi en un dispositiu tàctil.

Aquest *interface* permet interactuar amb Assets guardats a Dropbox o a Google Drive e importar-los directament al codi amb pocs clicks.

És una eina molt útil per al disseny de projectes de petita envergadura, està orientat al disseny de jocs.

Desenvolupat sota el llenguatge Lua, permet un ràpid aprenentatge. Tot el conjunt de llibreries incloses amb Codea permeten poder tenir codi funcional en molt poc temps, ja que amb l'ús de les funcions d'aquestes estalviem moltes línies de codi cridant a les funcionalitats pertinents per al nostre projecte.

Tot això fan de Codea un motor a tenir en compte a l'hora de desenvolupar software, i he decidit investigar-lo més a fons per a obtenir un referent sobre les seves capacitats i funcionalitats.

#### 4.1.2 Descripció de funcionalitats

L'interfície és molt còmode per ser un programa d'iPad, ja en la pròpia pantalla d'inici tenim visibles 2 zones :

- La zona superior amb un botó per a accedir a documentació, tutorials, API, manuals.. I la resta amb projectes d'exemple de les diferents funcionalitats de CodeA per a que l'usuari pugui visualitzar exemples senzills de codi.
- La zona inferior conté els projectes que anem creant per a accedir-hi tocant amb el dit.

A continuació, una captura de pantalla on es poden veure clarament aquestes zones :



Si premem el botó *Getting Started*, accedim a tota la documentació de Codea, tutorials, documentació i manuals del llenguatge LUA en el que està basat, API de les funcions, fòrums, una Wiki creada per als usuaris i alguns exemples.

Si fem un cop d'ull a l'apartat *Codea Reference* podem accedir a un menú amb les funcionalitats que ofereix, amb exemples de cada una en un format d'API.

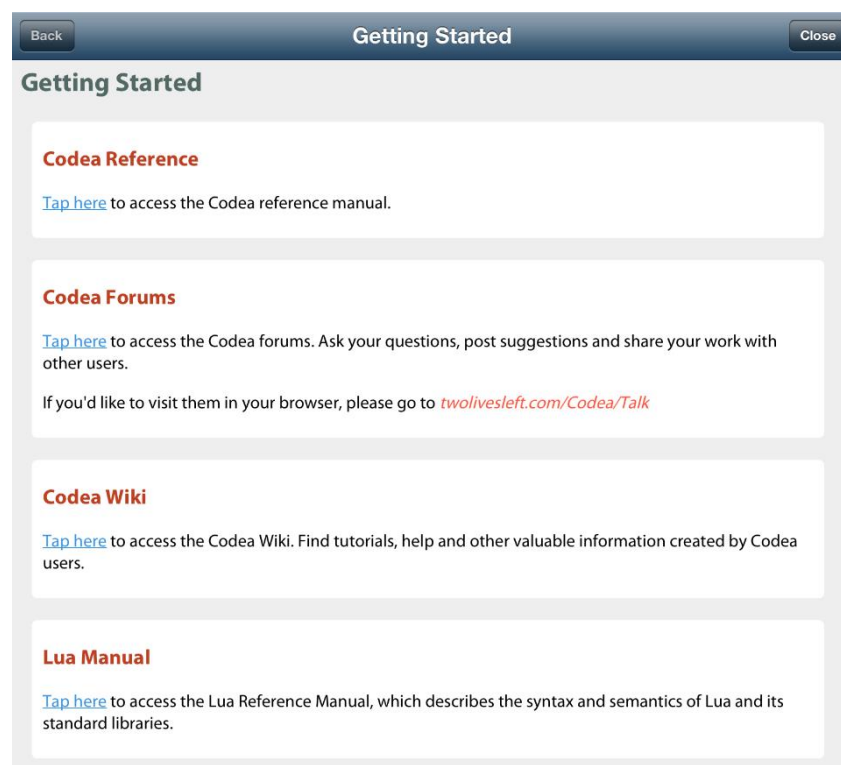
Mentre es realitza codi es pot accedir a tota la documentació de forma àgil des de la pròpia aplicació, cosa que facilita molt la programació.

També permet el canvi de variables, paràmetres e imatges mitjançant el Touch, simplificant bastant la feina, fent que tan sols s'hagin d'anar afegint les diferents funcions i configurant-ne els paràmetres.

És molt senzill d'aprendre i permet la creació de software amb necessitats força específiques, encara que això representi haver d'utilitzar les funcions amb que compta el sistema.

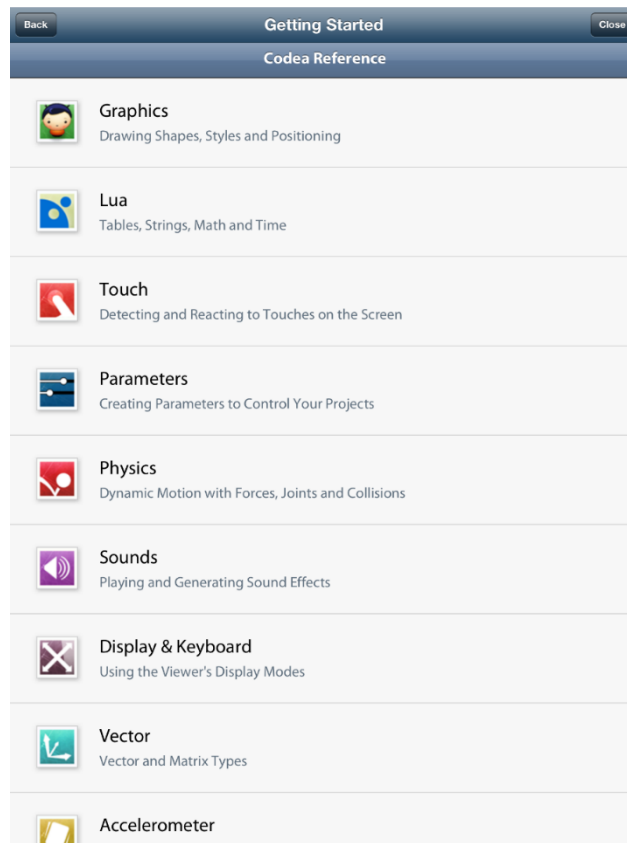
No permet importar cap tipus de llibreria externa, com la majoria de motors de creació permet a l'usuari fer servir només les funcions proveïdes pel propi motor. La feina de l'usuari doncs és mirar d'adaptar la seva idea a les capacitats d'aquest motor i comprovar si és viable o no.

A continuació algunes captures de pantalla de l' interface del motor :



Aquesta imatge correspon a l'apartat *Getting Started*, on està situada tota la documentació de l'aplicació.

Mentre es va desenvolupant l'aplicació, aquesta permet la consulta de la documentació de forma àgil, i facilita molt la programació, ja que no es perd temps en saber com fer les coses, les funcionalitats estan totes recollides i accessibles des de la pròpia aplicació i això facilita molt la recerca.



Aquesta segona imatge correspon a les referències, on estan explicades totes les funcions que permet Codea, amb exemples de codi de cadascuna d'elles.

Les funcions estan agrupades en diferents categories, cada categoria correspon a una funcionalitat diferent de Codea o, en termes de programació, les diferents llibreries. Les diferents llibreries ofereixen les següents funcionalitats :

- Gràfics
- Lua (Taules, Strings, operacions lògiques, Time)
- Touch
- Paràmetres
- Físiques
- Àudio
- Pantalla i Teclat
- Vectors i Matrius
- Acceleròmetre
- Xarxa
- Emmagatzematge de Dades


A continuació exemples d'algunes d'aquestes funcions en forma de captura de pantalla, per a ajudar a crear una apreciació visual de la comoditat d'accés a la informació que ofereix el software :



Aquestes referències permeten a l'usuari accedir de forma ràpida a les funcionalitats específiques que ofereix el llenguatge explicat més àmpliament a la documentació de Lua.

Amb tota la documentació s'inclou també un manual complet de Lua, al que s'accedeix des de l'apartat de Referències de Codea.


Back
Getting Started
Close



## Lua 5.1 Reference Manual

The reference manual is the official definition of the Lua language. For a complete introduction to Lua programming, see the book [Programming in Lua](#).

This manual is also available as a book:



**Lua 5.1 Reference Manual**  
 by R. Ierusalimsky, L. H. de Figueiredo, W. Celes  
 Lua.org, August 2006  
 ISBN 85-903798-3-3

Buy a copy of this book and [help to support](#) the Lua project.

[start](#) · [contents](#) · [index](#) · [errata](#) · [other versions](#) · [português](#) · [español](#) · [русский](#) · [magyar](#) · [deutsch](#)

Copyright © 2006–2012 Lua.org, PUC-Rio. Freely available under the terms of the [Lua license](#).

### Contents

- 1 – Introduction
- 2 – The Language
  - 2.1 – Lexical Conventions
  - 2.2 – Values and Types
    - 2.2.1 – Coercion
  - 2.3 – Variables
  - 2.4 – Statements
    - 2.4.1 – Chunks
    - 2.4.2 – Blocks
    - 2.4.3 – Assignment
    - 2.4.4 – Control Structures
    - 2.4.5 – For Statement
    - 2.4.6 – Function Calls as Statements
    - 2.4.7 – Local Declarations
  - 2.5 – Expressions
    - 2.5.1 – Arithmetic Operators
    - 2.5.2 – Relational Operators
    - 2.5.3 – Logical Operators

El manual conté tota la informació necessària per començar a programar amb Codea en poques hores.

La simplicitat que suposa l'aprenentatge de l'ús de Codea, apart de les facilitats d'importació d'imatges, multimèdia, canvi de paràmetres, la biblioteca de funcions i manual, és deguda a que Codea està fet en llenguatge Lua, que és un llenguatge d'scripting basat en Objective C i C++, que s'explica amb una mica més de detall a continuació.



### 4.1.3 Lua



Lua és un llenguatge de programació dissenyat per a una programació procedimental. També ofereix un bon suport per a la programació orientada a objectes, programació funcional i programació orientada a dades.

Es pretén que Lua sigui usat com un llenguatge de script per a qualsevol programa que ho necessiti. Està implementat com una biblioteca escrita en C, és a dir, en el subconjunt comú d'ANSI C i C++.

Lua no té noció de programa principal (main), funciona incrustat en un client amfitrió, denominat programa contenidor o simplement amfitrió ( host ). Aquest pot invocar funcions per executar un tros de codi Lua, pot escriure i llegir variables de Lua i pot registrar funcions C perquè siguin cridades pel codi Lua.

Mitjançant l'ús de funcions C, Lua pot ser ampliat per tal d'abastar un ampli rang de dominis diferents, creant així llenguatges de programació personalitzats que comparteixin el mateix marc sintàctic.

La distribució de Lua inclou un programa amfitrió de mostra anomenat Lua, que utilitza una biblioteca per oferir un intèrpret de Lua complet i independent .

Lua és programari lliure i per tant no disposa de garanties, tal i com s'estableix en la seva llicència.

## 4.1.4 Codi d'exemple

A continuació captures de codi d'exemple inclòs amb el propi software :

```

Main
1  -- This example shows you how to use the parameter() function
2
3  function setup()
4      -- Here we set up a parameter called Radius
5      -- this creates a global variable called Radius
6      -- with a minimum value of 100 and max of 500
7      -- We do the same with StrokeWidth
8      parameter("Radius", 100, 500)
9      parameter("StrokeWidth", 0, 45, 20)
10     parameter("BlueAmount", 0, 255)
11
12     print("Use the slider above to adjust the ellipse parameters")
13 end
14
15 function draw()
16     background(10, 10, 20)
17     fill(255, 0, BlueAmount)
18     stroke(255)
19
20     -- Here we access the StrokeWidth parameter
21     -- exposed by the parameter() function in
22     -- setup
23     strokeWidth(StrokeWidth)
24
25     -- Move to the middle of the screen
26     translate(WIDTH*0.5, HEIGHT*0.5)
27
28     -- Here we access the Radius parameter which can
29     -- be adjusted by the user using a slider
30     ellipseMode(RADIUS)
31     ellipse(0, 0, Radius, Radius)
32
33 end
34

```

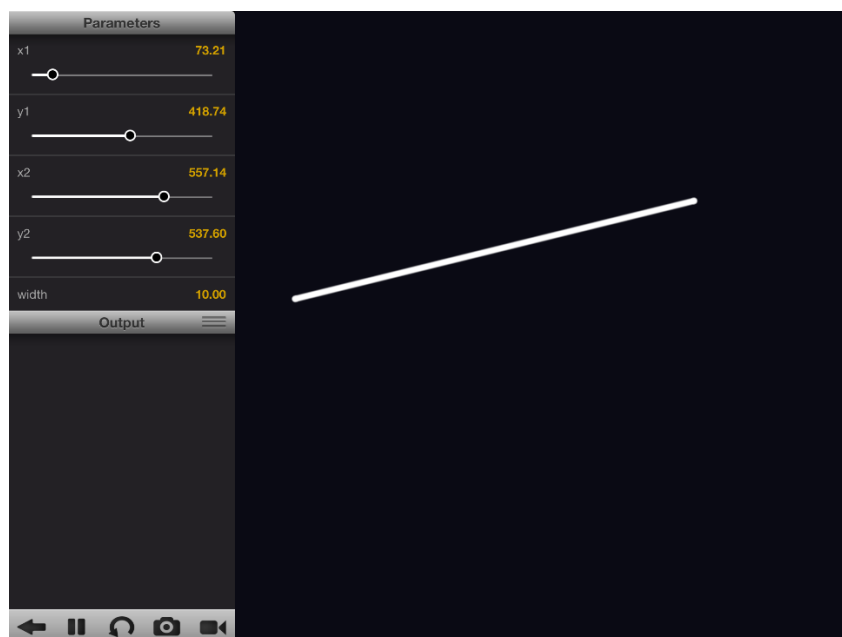
En aquest tros de codi veiem un exemple sobre com crear 3 paràmetres d'un cercle i poder interactuar amb ells mitjançant el *touchpad*.

Podem veure com amb molt poc codi ja podem mostrar molt per pantalla, això és gràcies a les llibreries incorporades amb el propi Codea.

```
1 supportedOrientations(LANDSCAPE_ANY)
2
3 t = 0
4
5 function setup()
6   parameter("x1", 0, WIDTH, 100)
7   parameter("y1", 0, HEIGHT, 100)
8   parameter("x2", 0, WIDTH, WIDTH-100)
9   parameter("y2", 0, HEIGHT, HEIGHT-100)
10  parameter("width", 1, 100, 10)
11  iparameter("lineCap", 0, 2, 0)
12 end
13
14 function draw()
15   background(10, 10, 20)
16   fill(255, 0, 0)
17   stroke(255)
18   strokeWidth(width)
19   if width < 3 then
20     noSmooth()
21   else
22     smooth()
23   end
24   lineCapMode(lineCap)
25   line(x1, y1, x2, y2)
26
27 end
28
29
```

En aquest exemple es pot veure un tros de codi d'exemple que mostra com dibuixar una línia configurant els paràmetres amb el touchpad, es pot veure com dins de la funció draw és on es dibuixa tot. De fet, en la funció draw és on Codea dibuixa cada segon tot el que hi ha sempre que es pugui.

A continuació una captura de pantalla del resultat d'aquest codi.



## 4.2 Mobincube



### 4.2.1 Introducció

Mobincube es un motor web de creació d'Apps per a dispositius mòbils de diferents plataformes.

Al estar dissenyat per a ser utilitzat per una plataforma web, no requereix cap tipus d'instal·lació prèvia de cap tipus de software i això agilitza molt el procés d'instal·lació i els requisits per a utilitzar-lo.

És un avantatge a tenir en compte ja que pot ser utilitzat des de qualsevol lloc, l'únic requisit és un navegador i connexió a internet.

Com tots els motors de creació d'aplicacions, Mobincube intenta facilitar i agilitzar al màxim la creació d'aplicacions per a que amb molt poc temps puguis tenir aplicacions funcionals.

També és gratuït e inclús permet publicar en un servidor propi de Mobincube de forma totalment gratuïta. També permet publicar de forma independent sota diverses condicions i preus.

Té els seus límits en quant a funcionalitats, però pot ser una eina molt potent si satisfà les necessitats del software que volem desenvolupar.

En el cas que ens ocupa, el vaig descartar ja que no era viable per al tipus de software que he realitzat, i per això l'estudi no serà tant exhaustiu com en els altres motors, tot i que és una eina a tenir en compte si el que es vol fer no és un joc, o un joc molt bàsic i les necessitats estan ben definides i no son molt complexes.

Amb aplicacions no complexes, referim a aplicacions orientades més aviat a la consulta, o amb un contingut molt específic d'informació, utilitzant inclús les funcionalitats bàsiques del dispositiu, però no és un motor per a crear funcionalitats que requereixin de molts càlculs ja que les funcionalitats estan molt definides i la creativitat queda reduïda a determinar la millor manera de fer servir aquests "blocs" que representen les diferents funcionalitats.

En els següents apartats s'explica breument les diferents funcionalitats i característiques que ofereix aquest motor.

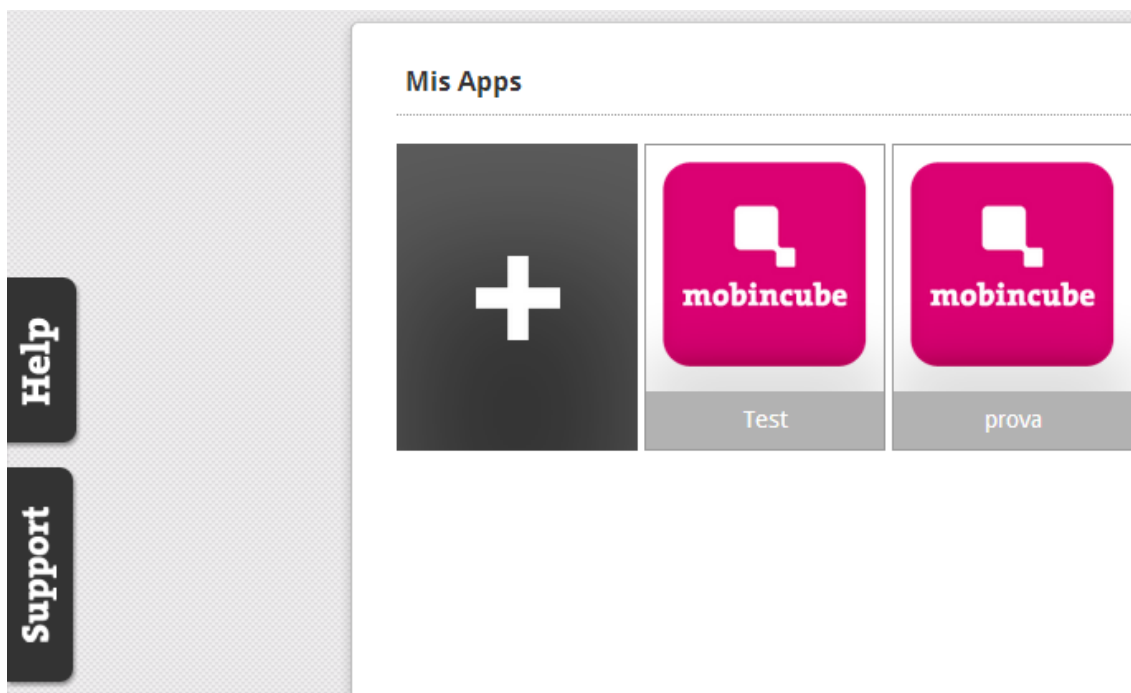
#### 4.2.2 Descripció de funcionalitats

Mobincube disposa d'una interfície molt fàcil d'utilitzar, de tipus *drag'n drop*, i això agilitza moltíssim la implementació de funcionalitats que, d'altra manera, es trigaria molt més en implementar.

Els passos a seguir per a fer servir l'eina són simplement registrar-se i, en menys de 3 minuts, ja es pot començar a desenvolupar la teva aplicació.

##### 4.2.2.1 Creació

Per a començar a crear una aplicació simplement se li posa un nom, s'escriu una categoria per a classificar-la segons el tipus de software, i es selecciona l'idioma principal en el que es desenvoluparà l'aplicació.



Dins de cada aplicació es pot editar tots aquests camps i configurar opcions de publicació, contractar diferents serveis Premium de Mobincube, utilitzar el servei CMS per a actualitzar el contingut de les aplicacions i gestionar els ingressos per publicitat.

## Datos

**Nombre:** Test **Fecha creación:** 18.06.2013

**Descripción:** Primera prova del TFG **Categoría:** Entretenimiento

**Idioma:** Castellano

**Versión 2** **Servicios contratados:** Ningún servicio contratado + info

**Última versión:** 25.06.2013

**Notas de la versión:**

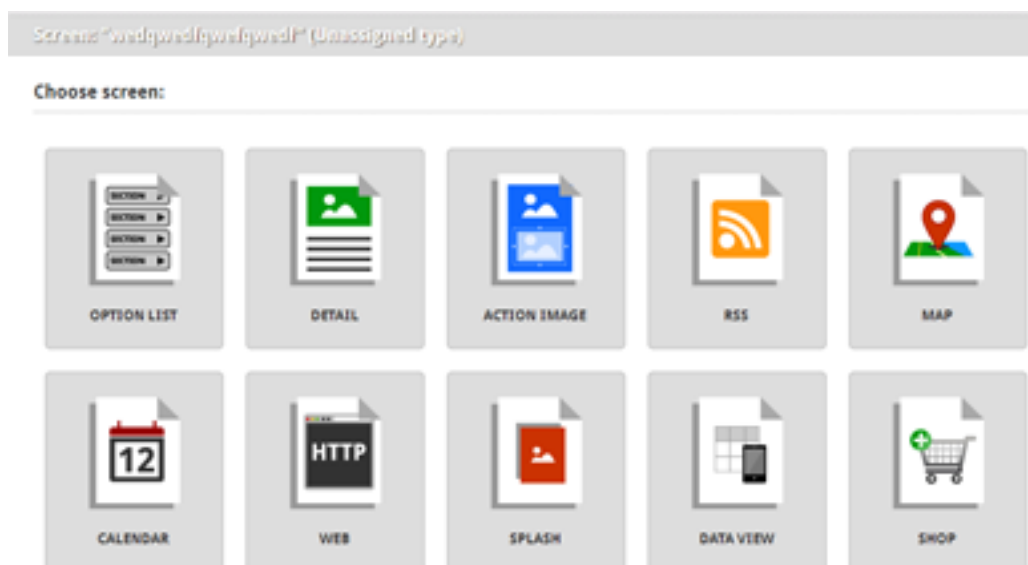
**Modificar datos**

## 4.2.2.2 Edició

En referència a les opcions d'edició, aquest ofereix funcionalitats molt limitades però no per això menys funcionals, ja que per contra, té un interface molt intuïtiu e inclús va acompanyat d'un sistema d'ajuda que explica com fer servir totes i cadascuna de les eines de que disposa.

Mobincube funciona per 'pàgines', cada pàgina és d'un tipus diferent i el dissenyador tan sols ha d'anar afegint les diferents pàgines dels tipus corresponents i editant-les de forma adequada per a complir les necessitats acordades.

Els diferents tipus de pàgines que es poden crear amb Mobincube es poden veure a la següent imatge :



Cadascuna d'aquestes pàgines ofereix un interface diferent per a facilitar la seva edició simplement fent copy-paste i drag'n drop.

No s'explicarà cadascuna de les interfícies a nivell intern ja que són molt intuïtives i al ser un motor descartat ràpidament, no és necessari, tot i això sí que s'enumeren a continuació per a oferir una breu idea dels mòduls de que disposa aquest motor.

Les funcionalitats que ofereix Mobincube per a totes les aplicacions són :

- Botiga Online
- Accepta pagaments amb PayPal
- Sistema de plantilles
- Sistema de Notificacions Push
- Barres de navegació
- Creació i gestió de formularis
- Google Maps
- Calendaris
- RSS
- Actualitzacions Online
- CMS
- Trucades telefòniques
- Enviament de SMS
- Enviament de correu electrònic
- Sistema de Publicitat
- Simulador Apple
- Simulador HTML5
- Enllaços amb el navegador d'Internet
- Bases de dades offline
- Base de dades actualitzables en línia
- Visors de dades configurables
- Publicació en App Store
- Publicació a Google Play
- Publicació a Windows Marketplace
- Versió Web App HTML5
- Integració amb Xarxes Socials
- Àudio i vídeo
- Splash Screens
- Textos
- Imatges
- Taules

#### 4.2.2.3 Disseny i visualització

Tot i que les funcionalitats d'edició són limitades, permet configurar-les per a proporcionar a cada App un aspecte diferent i no amb aspecte de "template app", que són aquelles aplicacions que una persona amb cert bagatge en creació d'aplicacions pot saber fàcilment que la aplicació ha estat creada amb un motor de creació.

Al permetre la creació de taules que es poden omplir amb contingut molt diferent, la creació de barres de navegació és totalment configurable, com ara amb separadors o textos web. Tot ho podem col·locar al nostre gust a la pantalla donant-li al software un aspecte molt més genuí.



Mentre es van editant els diferents estils i la aplicació en sí, Mobincube disposa d'un simulador online HTML5 que permet anar veient com queda l'aplicació en diferents Smartphones a mida que la vas dissenyant.

#### 4.2.2.4 Publicació

Mobincube permet la publicació d'aplicacions en el seu propi App Store de forma totalment gratuïta, i a altres plataformes, com ara Google Play, Windows Phone Market, la App Store d'Apple e inclús Blackberry mitjançant un sistema propi de subscripcions.

Per a la publicació per compte propi, simplement cal descarregar-se l'aplicació des de Mobincube i publicar-la on decideixi l'usuari.

Aquest pot escollir també si vol publicar sota el seu nom com a desenvolupador o si vol que Mobincube la publiqui en nom de l'usuari per a més renom.



## 4.3 Construct 2



### 4.3.1 Introducció

Construct2 és una plataforma de creació de jocs 2D amb capacitats per exportar a diferents plataformes a partir d'HTML5.

És una creació de Scirra, treballa sota l'entorn de Windows i té un gran renom en quant a creació de jocs ja que permet crear jocs funcionals en molt poc temps, de fet si la idea del joc és senzilla, en poques hores o dies es pot plasmar i provar a nivell funcional per a dedicar-se amb més èmfasi a la part del projecte corresponent al disseny i maquetació del software.

Disposa d'un interface també molt intuïtiu, de l'estil drag'n drop amb edició dels diferents paràmetres i comportaments.

Això permet que usuaris més orientats al disseny o estudiants de disseny d'aplicacions també puguin crear software al tenir totes les eines disponibles de forma tant intuïtiva i ràpida.

L'interface permet veure com quedarà l'aplicació al ser executada i permet dissenyar el joc a nivell visual i, per tant, de manera molt més ràpida e intuïtiva. Apart permet configurar tots i cadascun dels objectes importats canviant-los els paràmetres i comportaments.

També se'ls pot organitzar segons diferents capes per a millorar la seva organització durant la creació, e inclús editar els objectes des de el propi editor d'imatges incorporat amb Construct 2.

És un motor molt potent que permet tenir en molt poc temps jocs funcionals, per tant, encara que no s'utilitzi per a realitzar una versió final de software, es pot utilitzar per a testejar idees de forma ràpida, cosa que pot ser de gran valor a l'hora de crear software més complex del que no es puguin satisfer totes les necessitats i s'hagi de crear a més baix nivell.

La capacitat d'exportació a diverses plataformes que ofereix el converteix amb una eina a tenir en compte a l'hora de crear un joc en format HTML5, que és el format al que exporta Construct 2, i a partir d'aquest a les diferents plataformes, Android, iOS, etc.. Algunes d'aquestes exportacions porten problemes i s'han d'ajudar de software extern.

#### 4.3.2 Explicació de funcionalitats

Construct 2 té un interface molt potent i senzill de fer servir que permet crear i gestionar el projecte d'una forma còmode e intuïtiva mitjançant layouts, que serien les diferents pantalles de menú, de joc, d'opcions, etc.. del nostre software, el que serien els diferents *activities* en Android.

Això i el gestor de capes i objectes permet dissenyar amb facilitat l'interface gràfic del software i crear classes e instàncies d'aquestes en pocs clicks.

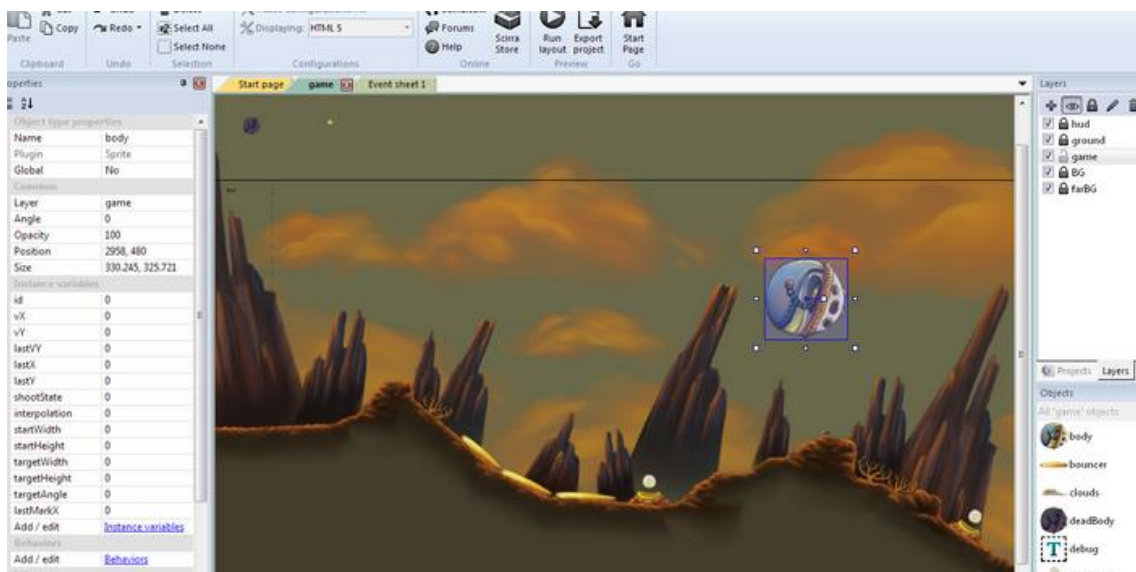
La forma visual en que es tracta tot, la rapidesa en crear software funcional, la capacitat d'exportació a diferents plataformes i la seva facilitat d'aprenentatge fan de Construct 2 una eina a tenir en compte.

A continuació s'expliquen les funcionalitats i forma de treballar de Construct 2:

##### 4.3.2.1 Layouts

En els layouts trobem nivells, pantalles de títol i demés objectes predefinits dels layouts. Aquests es consisteixen en una o varies capes que s'utilitzen per a col·locar els objectes a diferents nivells de profunditat.

És el que representa cada pantalla que es mostrarà en el software.



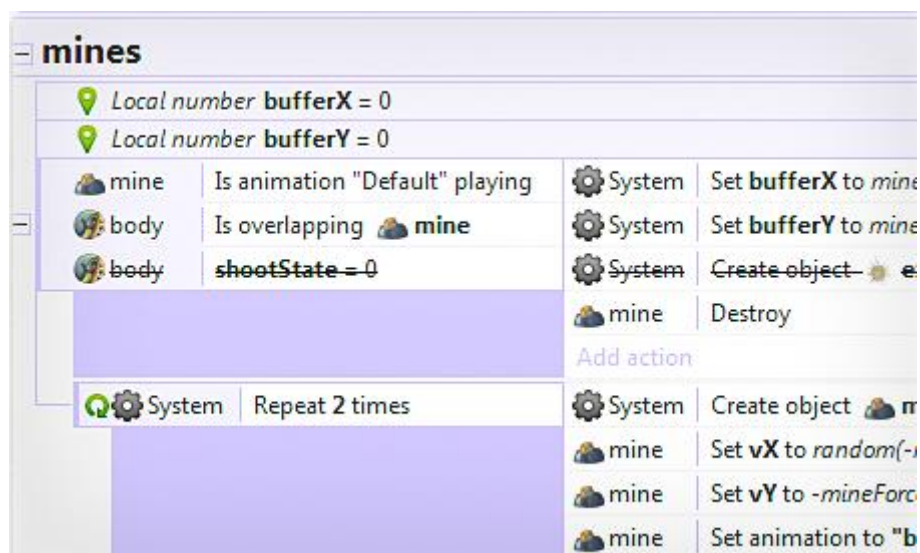
#### 4.3.2.2 Objectes

Un tipus d'objecte en Construct 2 és el que seria una classe en programació orientada a objectes. Un cop definim un objecte amb els paràmetres que vulguem, podem tenir diverses instàncies d'aquest.

També podem agrupar els objectes en famílies o grups.

#### 4.3.2.3 Behaviors

Són els diferents comportaments que poden adoptar els diferents objectes que anem creant. Es componen d'un conjunt de condicions, o funcions prefabricades que permeten assignar a un objecte les característiques que siguin necessàries.



Aquestes funcions poden ser per exemple, direccions, que es comporti com una plataforma o un cotxe, assignar-li físiques, o pathfinding. Apart també permet efectes com ara el fade o el drag & drop.

La majoria de behaviors poden ser replicats fàcilment amb events, però consumeix molt més temps.

Per exemple, si afegim el behavior *platform* a un objecte, automàticament ja podem córrer i saltar per damunt d'objectes marcats com a 'solid' i permet modificar variables com ara la velocitat, acceleració, distància de salt, gravetat, força, etc.. per a ajustar al màxim el comportament que vulguem que adoptin els nostres objectes.

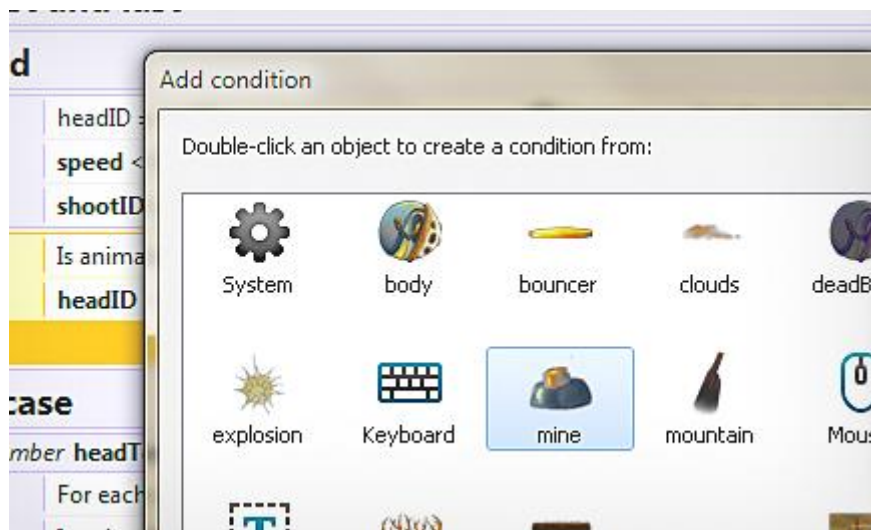
#### 4.3.2.4 Sistema d'events

Construct 2 té un Sistema de gestió d'events que permet encapsular el vèries funcions o mètodes diferents per tal d'utilitzar-los en cada layout si es vol.

Això facilita la organització i desenvolupament dels comportaments que han de tenir els diferents components del software.

Aquests queden recollits en la *event sheet* o full d'events, i està dissenyat per a ser el més fàcil d'entendre possible per a un dissenyador. L'avantatge dels events és que pots fer-los servir en altres layouts i, en el cas d'un joc, si la estructura es crea correctament fent servir les millors *best-practices*, es poden tenir varis nivells o gran part d'ells encapsulats de tal manera que permet tornar a fer servir grans estructures lògiques amb facilitat, cosa que encara agilitza més la part de disseny.

Amb l'ajuda de variables, sub-events i funcions recursives també es poden estalviar moltes línies de codi, tan sols cal escollir el tipus d'objecte correcte, la condició o acció que vulguem que succeeixi i agregar-ho a l'event actual que estiguem fent.



Podem tenir també grups d'events, que poden fer-se servir per a activar o desactivar varis events a l'hora i també, com és obvi, per a ajudar en la organització dels diversos comportaments.

Podem resumir els events en Construct 2 com al que gestiona la part lògica del joc. Tots els comportaments es gestionaran per separat en behaviors o en events, que podrem repetir o tornar a cridar quan vulguem, encapsulant així els conjunts de funcions que necessitem per a cobrir les necessitats del nostre software.

#### 4.3.2.5 Preview i exportació

Construct 2 permet visualitzar tal i com quedaria l'aplicació final amb el sistema de previsualització que incorpora.



Una característica a tenir en compte és la capacitat de Construct 2 de previsualitzar mitjançant tecnologia Wi-fi amb dispositius que suportin HTML5, com diverses tablets.

Aquesta característica és molt útil per a provar i testejar el software de manera rapidíssima i directament en diversos dispositius per a facilitar la observació dels possibles errors que poguessin anar sorgint durant el desenvolupament del software.

Construct 2 permet la exportació a múltiples plataformes. Aquest, converteix el projecte a HTML5 i a partir d'aquí l'exporta a la plataforma que calgui. Construct 2 crea apps HTML 5, i hi ha diverses maneres de publicar-les.

Les opcions que ofereix el software són :

- En el teu propi servidor
- Scirra Arcade
- DropBox/Google Drive
- Chrome Web Store
- Facebook
- iOS Web Apps
- Windows 8 apps
- Firefox Marketplace
- Android e iOS via CocoonJS

Com es pot veure, per a publicar per a Android e iOS, cal fer-ho des de CocoonJS, que funciona semblant a Phonegap (també compatible), tot i que poden haver-hi problemes a l'exportació segons el software.

## 4.4 Gamesalad



### 4.4.1 Introducció

Gamesalad és un motor de creació de jocs que funciona sota OSX, tot i que recentment ha desenvolupat una versió per treballar sota Windows.

Juntament amb Construct 2, és un dels motors de creació de videojocs que més ha donat a parlar per la xarxa amb unes 70 top 100 apps, una d'elles premiada com a n1 en la App Store d'Apple.

Té unes característiques semblants a Construct 2, tot i que aquest no depèn de software extern per a realitzar les exportacions als diferents dispositius.

Té un sistema d'escenes, events, actors i objectes que permeten tenir dissenyats videojocs senzills en qüestió d'hores.

El sistema drag & drop per a adjuntar els diferents components com ara imatges, sons o comportaments fan que la tasca d'importació sigui senzillíssima, tan sols cal respectar els formats requerits, que són diferents en la versió de Mac que en la de Windows, sent aquesta última molt més restrictiva. Aquest fet és degut a que la versió de Windows va ser desenvolupada molt més tard i sota petició del conjunt d'usuaris i persones que volien poder desenvolupar software amb un PC. Com que la versió és més nova, en el moment de la realització del projecte, aquesta presenta forces limitacions en alguns punts i dona certs problemes que s'expliquen a l'apartat d'explicació del projecte, ja que ha estat el motor escollit per a fer-lo.

La seva aparent simplicitat a l'hora de crear apps, la seva capacitat d'exportació en diferents plataformes i la seguretat de que és un motor que fan servir mils de persones en relativament poc temps que fa que l'aplicació estigui viva, la fan digne de ser tinguda en compte per a la creació de software.

#### 4.4.2 Explicació de funcionalitats

Gamesalad, al igual que Construct 2, té un interface que permet tenir totes les funcionalitats a la vista i accessibles, això li atorga simplicitat a la potència que ofereix.

Gamesalad té unes característiques estructurals semblants al seu competidor directe en Windows, Construct 2. Té un sistema d'escenes, actors i comportaments que són molt semblants als conceptes del motor prèviament explicat, la qual cosa no suposa un inconvenient en absolut, més aviat al contrari.

Els Assets (imatges i sons), es poden importar amb gran comoditat i els formats acceptats són els més utilitzats (.png, .mp3, .jpg..), així que és fàcil dissenyar-los i importar-los al motor. Els formats suportats varien segons la versió, essent la versió de Windows molt més limitada al respecte.

A continuació s'explica a mode funcional les característiques de Gamesalad.

##### 4.4.2.1 Scenes

Són l'equivalent al layout de Construct 2, són totes i cadascuna de les pantalles que contindrà el software (menús, pantalles del joc, pantalla d'opcions, etc..).



En podem fer tantes com vulguem i dins de cada escena tindrem els diferents actors (objectes en Construct 2) que duran a terme les accions per a les que estiguin dissenyats mitjançant els diferents comportaments o behaviors que tinguin adjuntats.

Canviar d'escena no és quelcom automàtic en Gamesalad, s'ha de controlar mitjançant un comportament anomenat *Change Scene*.

Les escenes poden tenir atributs d'escena propis com ara la gravetat o la grandària, que seran propis de cada escena individualment.

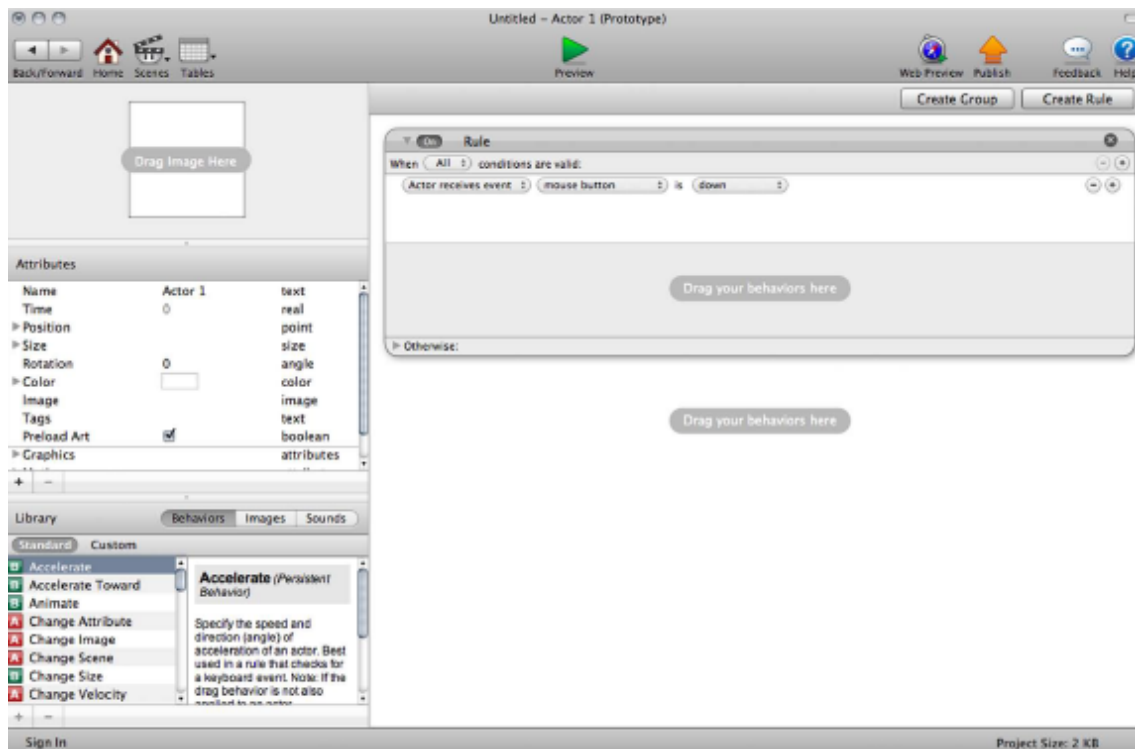
També hi ha variables globals a tot el projecte en si, que poden ser utilitzades en totes les escenes del projecte.

Apart dels paràmetres propis de cada escena i del projecte, l'usuari també pot crear noves variables de diversos tipus per a ús personal, fet molt útil que motors com ara Mobincube no ofereix i que és clau per a desenvolupar software.

#### 4.4.2.2 Actors

Els actors són classes encapsulades amb uns atributs per defecte als que afegirem els comportaments i normes necessàries per a que realitzin les funcions per a les que han estat creats.

El conjunt d'aquests actors determinarà com opera el projecte en si, és on es troba repartit el 'codi'.



Aquests es comporten com a classes, ja que representen qualsevol cosa, des d'un text mostrat per pantalla fins a personatges, objectes de l'entorn, objectes amb els que interacciona el protagonista ja sigui parlant, xocant-hi, etc.. durant l'execució del joc.

També s'utilitzen per a tot tipus de comportaments del projecte, gestió de taules, música i sons, control de càmera, moviment d'actors.. en definitiva, els actors són l'element clau de Gamesalad.

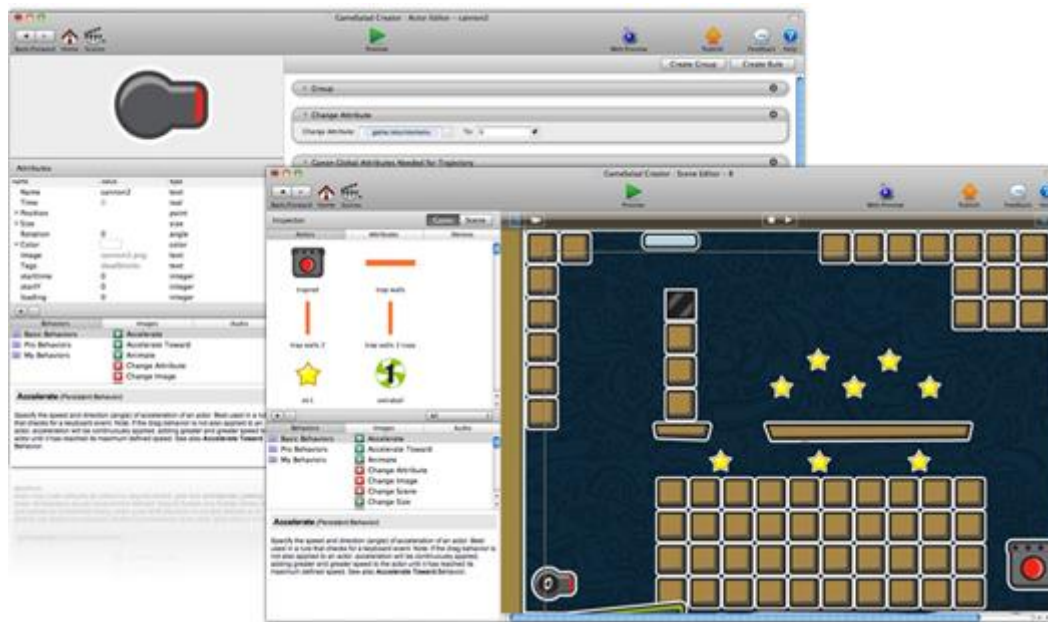
Hi ha dos tipus d'actors:

- **Actors d'instància** (instance actors)
- **Actors prototip** (prototype actors)

En programació orientada a objectes, els actors prototip serien l'equivalent al concepte de classe, i els d'instància tal i com diu el nom serien els equivalents a les instàncies o objectes d'aquesta classe.

Dins d'aquest actor prototip, hi hauran els atributs, normes i comportaments que tindran per defecte totes les instàncies que hi hagi en les escenes del projecte.



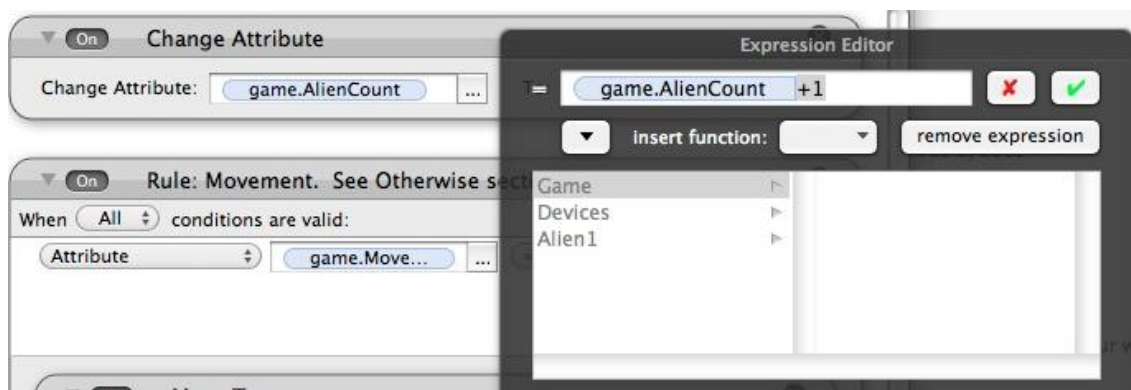


Cada actor instanciat d'aquest prototip, podrà ser modificat com es vulgui i conservarà les seves propietats de forma individual de la resta d'actors sense afectar en res a les propietats del prototip.

Amb les normes i comportaments és com es controla i es gestiona el projecte, són el cervell dels actors.

#### 4.4.2.3 Rules & Behaviors

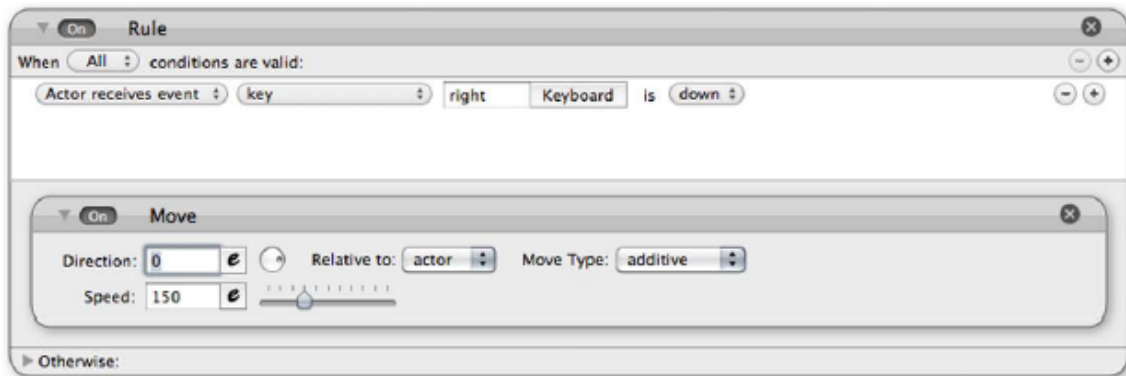
Les rules i behaviors són tot el conjunt d'accions que s'assignen als actors per gestionar les seves interaccions. Conformen la lògica del projecte, gestionen tots els events, com per exemple una col·lisió o un clic i l'acció que realitzarà, com ara moure's, canviar de velocitat o acceleració, desaparèixer, canviar d'imatge..



Gamesalad conté un total de 36 comportaments estàndard que es poden combinar per a obtenir els comportaments necessaris per a la nostra aplicació.

Hi ha 3 tipus de behaviors :

- **Containers:** Són els que s'utilitzen per a comportaments que volem que passin només sota condicions específiques. Un d'aquests comportaments es la *rule* o norma, que pot contenir una o varies condicions i dins les accions que volem realitzar. Un altre exemple seria el timer, en el que podem controlar cada quant de temps volem que passi un event.



- **Persistent Behaviors:** Són comportaments que passen contínuament a no ser que estiguin dins d'una *rule* que contingui alguna condició que ja no s'acompleixi.
- **Action Behaviors:** Passen una vegada i només es repeteixen si estan dins d'una condició que canvia el seu valor a *false* i torna a ser *true* un altre cop.

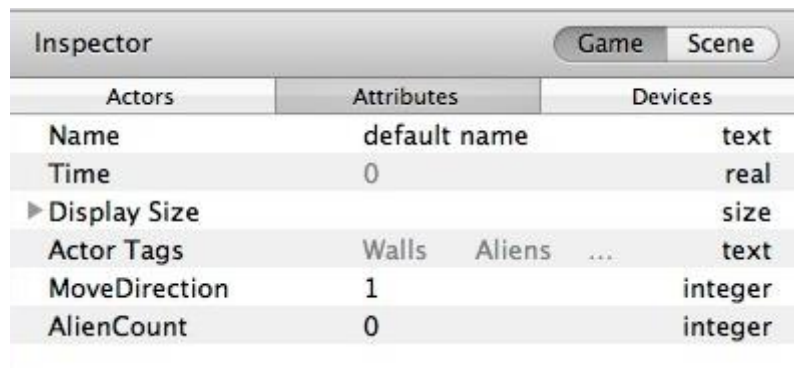
Alguns dels comportaments més usats són :

- Accelerate
- Accelerate Toward
- Animate
- Change Attribute
- Constrain Attribute
- Change Scene
- Change image
- Collide
- Destroy
- Move
- Play music
- Play sound
- Reset Scene
- Timer
- Save/Load Attribute

#### 4.4.2.4 Atributs o variables

Els atributs i variables són l'eina més potent de Gamesalad juntament amb el sistema de *behaviors*.

Hi ha atributs que ja tenen valors per defecte a les escenes i els actors, però també es poden crear de nou dins d'aquests a gust del desenvolupador. Això és útil per a guardar qualsevol tipus de valor, ja sigui text o un número com per exemple una puntuació, velocitat, etc.. en definitiva, qualsevol dada que necessitem guardar.



Inspector			Game	Scene
Actors	Attributes	Devices		
Name	default name	text		
Time	0	real		
▶ Display Size		size		
Actor Tags	Walls Aliens ...	text		
MoveDirection	1	integer		
AlienCount	0	integer		

Hi ha 3 nivells d'atributs o variables que, jeràrquicament, són:

- **De joc:** Un atribut de joc serà accessible des de totes les escenes i tots els actors dins d'elles.
- **D'escena:** Un atribut d'escena serà accessible tan sols des de dins de la pròpia escena per totes les instàncies d'actors que la componen.
- **D'actor:** Un atribut d'actor, si és accedit des d'una instància del propi actor afectarà a si mateix, no a la resta d'instàncies. Si l'atribut es modifica a l'actor prototip, afectarà a totes les instàncies d'aquest.

Per últim, tenim 6 tipus diferents d'atributs:

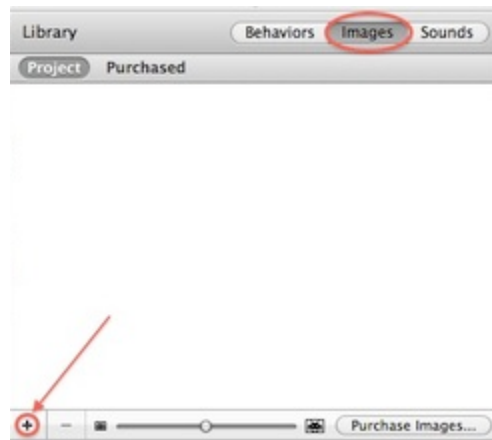
- **Boolean:** Valors *true/false*
- **Text:** Lletres i números, solen ser mostrats amb el *behavior* anomenat *display text*.
- **Integer:** Valors sencers.
- **Real:** Valors amb decimals, permet resultats fraccionaris, com ara el temps, o altres valors en que necessitem més exactitud de càlcul.
- **Angle:** Permet valors entre 0 i 359 amb decimals. Representen els angles d'un cercle.
- **Index:** Números sencers. No poden ser negatius. Aquest tipus d'atribut tindrà més sentit un cop Gamesalad implementi els *Arrays*. També es pot utilitzar per indexar taules, tot i que es pot usar de la mateixa manera que un *integer*.

#### 4.4.2.5 Imatges i Sons

Les imatges, música i sons són espectacularment fàcil d'importar a Gamesalad. La versió de Windows requereix passar a format \*.WAV o \*.WMA tots els sons i música per tal de ser reproduïda, i presenta alguns errors en la reproducció en algunes ocasions.

Això es deu a que la versió de Windows encara està en fase de desenvolupament tot i que ja incorpora gairebé totes les funcionalitats de Gamesalad.

En Gamesalad, quan creem el projecte disposem d'una biblioteca amb els comportaments i multimèdia(imatges i sons) buida.



En aquesta biblioteca podem accedir a tots els comportaments als que té accés Gamesalad i a les imatges i sons que anem afegint al projecte.

Per a afegir una imatge o so, tan sols cal clicar al botó i després a la icona '+' senyalada en la imatge. També podem accedir a la botiga d'assets de Gamesalad per a comprar-los e importar-los a la nostra biblioteca.



Un cop afegits els Assets a la biblioteca, per a afegir-los al projecte és tant fàcil com arrastar-les a un actor o bé a una escena directament. Si ho fem de la segona manera, es crearà automàticament un actor amb el nom de la imatge.

A partir d'aquí ja el podem animar amb una seqüència d'imatges o afegir-hi els *behaviors* necessaris.

No s'especifiquen les característiques de les imatges en Windows, de fet porten força problemes, s'han d'importar en el format exacte amb el que es faran servir.

En iOS, les resolucions a les que es farà reescalat a totes les imatges importades són:

- 16x16
- 32x32
- 64x64
- 128x128
- 256x256
- 512x512
- 1024x1024

Això significa que al importar una imatge, Gamesalad s'escalaran a les proporcions anteriors que més s'adeqüin al seu tamany. Per exemple, si una imatge és de 50x35, ocuparà el mateix que una de 64x64.

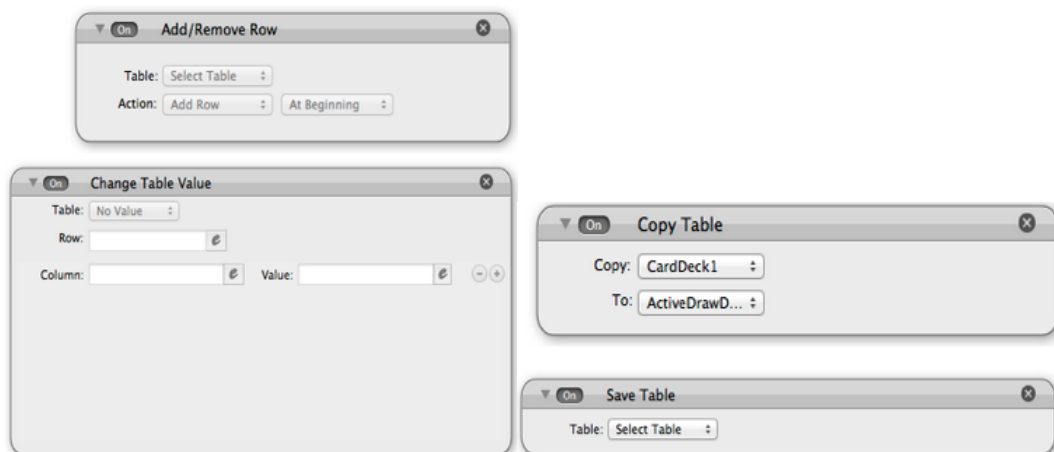
Per a afegir sons o música s'utilitza el mateix sistema que per a les imatges, i per a controlar-los ho fem amb els *behaviors* fets per a tal efecte com ara *Play Sound* o *Play Music*.

#### 4.4.2.6 Taules

Les taules suposen una manera molt més neta de guardar informació sobre el projecte.

El seu ús és limitat i no permet molta versatilitat, però si que ajuda en gran mesura poder guardar atributs de tots els tipus en tantes files i columnes com especifiquem.

Això ens permet accedir a informació des de qualsevol actor ja que les taules són visibles a tot el projecte. Així doncs, amb varis comportaments de Gamesalad es podrà crear i accedir a aquestes taules per a modificar i/o recuperar les dades que necessitem.

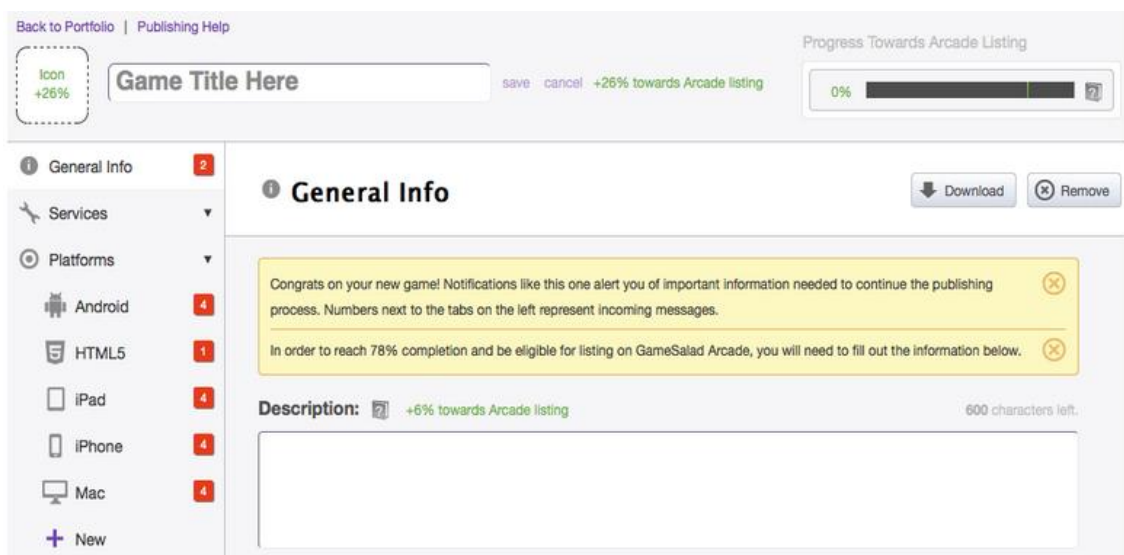


No obstant, les taules es defineixen inicialment des de el propi Gamesalad Creator amb els tipus d'atributs que necessitem per a cada columna, el nom de la taula i el número de files i columnes que tindrà.

Més endavant, podem modificar els valors de dins de la taula, afegir i/o esborrar files i/o columnes, guardar les dades de la taula o carregar les dades d'una taula a una altra.

#### 4.4.2.7 Exportació i publicació

Gamesalad permet la exportació directa a diverses plataformes. Un cop acabat el projecte o la part del projecte que volem testejar, publiquem l'aplicació al nostre perfil de Gamesalad per a convertir-la allà al format que vulguem o, si disposem d'un certificat d'Apple Developer i un iPad, podem compilar via Wi-fi directament en el dispositiu i veure com quedaria la nostra aplicació.



Un cop exportada l'aplicació, la podem convertir als següents formats, tal i com s'aprecia en la imatge anterior.

- Android
- HTML5
- iPad
- iPhone
- Mac

Tant en la versió de Windows com en la de Mac de Gamesalad, essent encara més apreciable en la primera, la conversió a HTML5 presenta molts errors de conversió i pot portar problemes a l'hora d'exportar creant errors inexistents en el codi que requereixen de mesures no gaire netes per a reparar-los, com ha estat el cas d'aquest projecte.

Si es disposa d'una llicència d'Apple Developer, tant el testeig com la posterior publicació es faciliten en gran mesura, de fet el testeig resulta comodíssim e instantani ja que prement un botó en el GameSalad Creator es compila via Wi-fi directament al nostre dispositiu Apple(iPhone o iPad), agilitzant moltíssim el procés.



El procés de testeig, juntament amb els logs, tot i que no són exhaustius i hi ha molts errors que no es veuen, són suficientment potents com per debugar l'aplicació amb prou precisió.

## 5. Motor escollit: Justificació

Després de valorar la informació exposada anteriorment i avaluar la millor manera de satisfer els objectius proposats, s'han tingut en compte principalment 3 factors :

- Preu vs prestacions del motor
- Capacitat de possible exportació a diverses plataformes
- Ràpida i fluïda a l'hora de desenvolupar software

Construct 2, Mobincube i Gamesalad tenen possibilitat d'exportació a diverses plataformes, permetent-ne així una futura implementació en cas necessari. No obstant, Codea tan sols permet la exportació en iOS format iPad i, per tant, requereix d'una llicència d'*Apple Developer* per a publicar i fer-lo servir ja des d'un inici.

Codea es va acabar descartant per dues raons:

- Pagament de l'aplicació i necessitat de disposar d'una llicència Apple des d'un inici.
- No permet la programació des d'un ordinador

Tot i que ara es disposa d'una llicència d'*Apple Developer*, quan es va començar a desenvolupar el projecte no era així, i tot i que la intenció era ja des de un inici fer una aplicació per iPad, es pretenia investigar i trobar alguna altra eina gratuïta o amb pagaments extres amb la que poder realitzar l'aplicació sense tenir una llicència Apple.

El fet que, tot i estar molt ben documentat i ser molt senzill de fer servir, només poder desenvolupar el codi des de l'iPad ha estat quelcom que, per un petit període de temps es va plantejar, però quan es van descobrir motors que permetien la programació des d'un ordinador i fins i tot la exportació a altres formats diferents del iOS, es va acabar descartant, pensant més en la comoditat de desenvolupar software amb un ordinador, fet que agilitza molt el procés de creació de software.

Ara que es disposa d'una llicència, no seria mala idea provar de fer algun altre petit projecte amb aquesta eina. Seria interessant provar-la amb profunditat, sota cap tipus de pressió. Per a projectes petits pot ser una eina molt curiosa d'utilitzar.

Mobincube sorprèn per la facilitat i rapidesa en que permet realitzar aplicacions de consulta. Va ser descartat ja que mentre s'anava investigant, es va observar que el tipus de jocs que es poden fer amb aquest motor no s'ajustaven gens a les necessitats d'aquest projecte.

Mobincube és un motor molt potent, però està més centrat en les aplicacions que en els jocs i les funcionalitats que ofereix el motor estan orientades a aquest fet.

El dubte més gran va ser entre Construct 2 i Gamesalad, ja que tots dos són motors potents que permeten l'exportació a diverses plataformes. Construct 2 és exclusiu de Windows mentre que Gamesalad era originalment per Mac, però recentment ha desenvolupat una versió Windows que encara està sent millorada.



Els 2 són gratuïts, amb possibilitat d'incrementar-ne les funcionalitats, reduir els anuncis, etc.. mitjançant pagaments extra.

Tant un com l'altre són coneguts per la seva qualitat i el nombre elevat d'usuaris que els utilitzen.

El sistema d'events i comportaments de tots dos és molt semblant i en quant a funcionalitats sembla que també estan més o menys a la par, tots dos fan servir un nombre elevat de recursos i s'ha d'anar amb compte quan en fem l'exportació, ja que depèn de com la dissenyem pot presentar problemes de ralentització.

En quant a exportació i actualitzacions del motor, Construct 2 és més potent i permet exportar a més plataformes, com ara un joc de Facebook, o publicar-ho en HTML5 en la teva pròpia web.

Aquestes característiques, ja que en aquest cas es necessita publicar per iPad i el temps per a realitzar el projecte ha estat limitat, no han estat rellevants per a decidir, tot i que ho serien si s'hagués de fer un projecte de més durada.

En quant a la rapidesa a l'hora de desenvolupar software, sembla ser que Gamesalad és una mica més ràpid i Construct 2 necessita de més paràmetres, però les contínues actualitzacions fan que creixi a un nivell més ràpid que Gamesalad.

Finalment es va optar per Gamesalad i es va començar a desenvolupar el projecte. De Construct 2, en vam sentir a parlar més tard i es va estar investigant ja que semblava molt interessant. El haver començat ja a desenvolupar el projecte i el temps del que es disposava, es va decidir continuar amb Gamesalad i ha estat una decisió acertada, ja que el software compleix els requisits mínims demanats i el projecte s'ha pogut acabar i inclús aplicar-hi millores.

Tot i això, si s'hagués descobert Construct 2 abans i s'hagués pogut investigar amb més temps, probablement hagués estat el motor escollit.

Primer es va utilitzar la versió de Windows de Gamesalad. Aquesta a la llarga portava molts problemes i es va migrar el projecte a la versió de Mac i, tot i no disposar-ne d'un, es va poder aconseguir-ne un de prestat amb el que poder-hi treballar, permetent acabar l'aplicació sense complicacions fins al moment de l'exportació, amb la que es varen experimentar força contratemps.

Aquests problemes en l'exportació sembla ser que no és cosa del motor, és problema de la conversió a HTML5, que encara no està del tot polida sigui quin sigui el motor que utilitzis. Tot i això, per segons quin tipus d'aplicació segueix sent un risc assumible si no es disposa d'un temps molt limitat per a realitzar el projecte.

## 6. Explicació del projecte

En aquest apartat s'explicarà en que ha consistit el projecte, en el que s'ha aconseguit, les seves necessitats, estructura, funcionament, casos d'ús, etc..

En definitiva, tots els passos i per totes les modificacions que ha anat passant el joc des de que es va pensar la idea fins al software que és ara.

### 6.1 Anàlisi de requeriments i funcionament

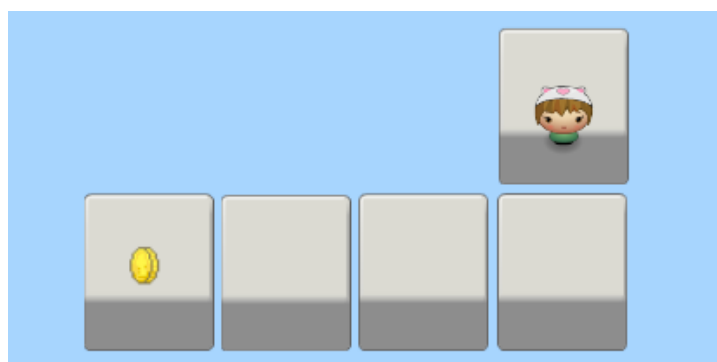
El software ha de permetre l'aprenentatge o exercici de les estructures mentals dedicades al processament lògic de la informació.

Tal i com s'exposa a l'apartat d'objectius, el software ha de satisfer el següent:

- Per a nens i nenes de 5 a 8 anys, per tant s'ha d'adequar la interfície i usabilitat a aquesta franja d'edat. No ha de fer cap distinció de gènere, ha de poder ser jugat indistintament per nens i nenes.
- Ha de ser de tipus joc educatiu.
- Compatible amb dispositius mòbils. Android i/o iOS.
- Ha d'incloure el concepte de *Loop* d'alguna manera.
- Ha de ser com a mínim d'una pantalla.

Així doncs, tenint en compte que es necessitava disposar d'una estructura de joc en la que s'introduïssin comandes per una banda i el joc respongués, es va decidir crear LogiK.

Com es pot apreciar en la següent imatge, s'ha tingut en compte especialment el públic a qui va dirigida la aplicació a l'hora de desenvolupar-la. No s'ha utilitzat cap tipus d'estètica que pogués ser catalogada a nivell de gènere, inclús s'ha fet servir un heroi de cada sexe alternant-se en cadascun dels nivells.



LogiK, es basa principalment en fer arribar a un heroi o protagonista fins a una moneda.

Per a fer-ho, aquest disposa d'unes comandes en la part inferior de la pantalla corresponents a les 4 direccions i una comanda 'repetir' o *Loop*.

Es pretén treballar el concepte de seqüencialitat fent que el jugador hagi d'anar arrastrant les accions en l'ordre correcte a la zona de comandes per a que l'heroi arribi a la moneda, un cop les hagi col·locat, haurà de prémer un botó per a indicar que ja ha acabat d'introduir la seqüència.

L'heroi doncs, es mourà en l'ordre en que el jugador hagi guardat prèviament. Si el jugador arriba a la moneda guanya, si no, perd. El joc ofereix un feedback sonor en ambdós casos evocant error o èxit segons convingui, no obstant també ho fa a nivell visual, amb un símbol de *check* de color verd al guanyar o una 'X' vermella al perdre.

El nombre de comandes a utilitzar en cada nivell són diferents, això afavoreix l'ús de la comanda repetir o *Loop*, que també tindrà un nombre limitat de moviments.

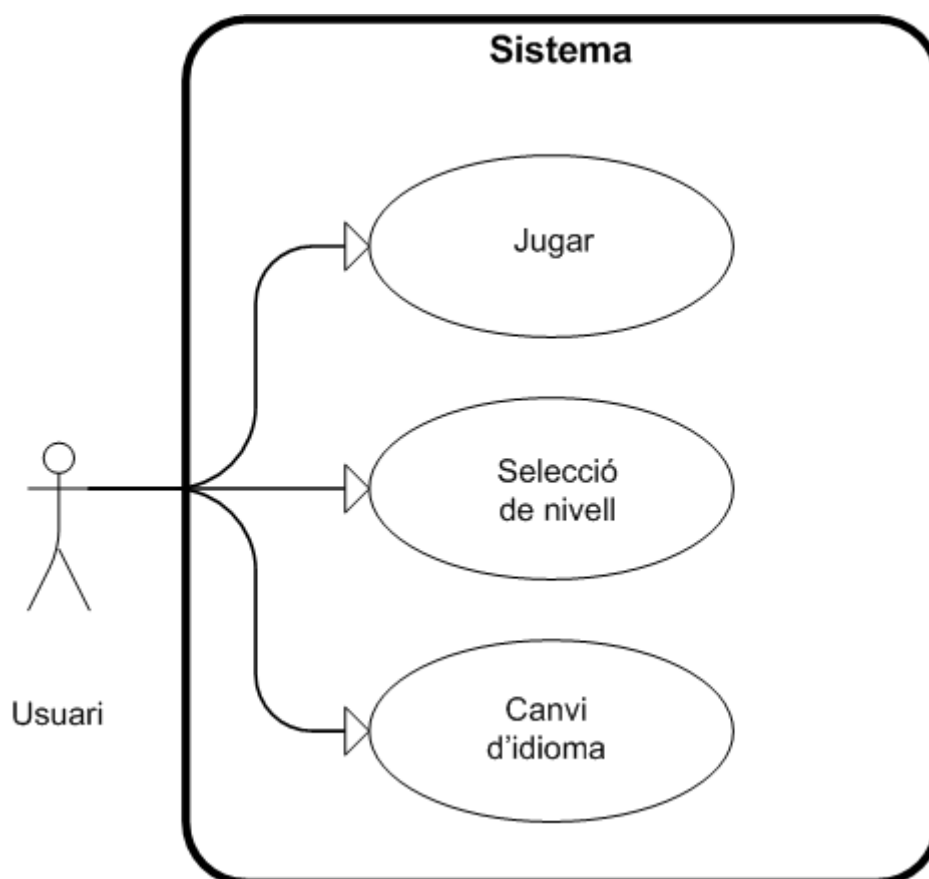
La limitació de comandes a realitzar per arribar a la moneda i el nombre i distribució del camí de caselles entre les que es podrà moure, determinaran la dificultat i creativitat necessària per a superar el nivell.

Al acabar la pantalla, tant si perd com si guanya pot provar de repetir-la o tornar al menú, si guanya també pot avançar de pantalla.

## 6.2 Casos d'ús

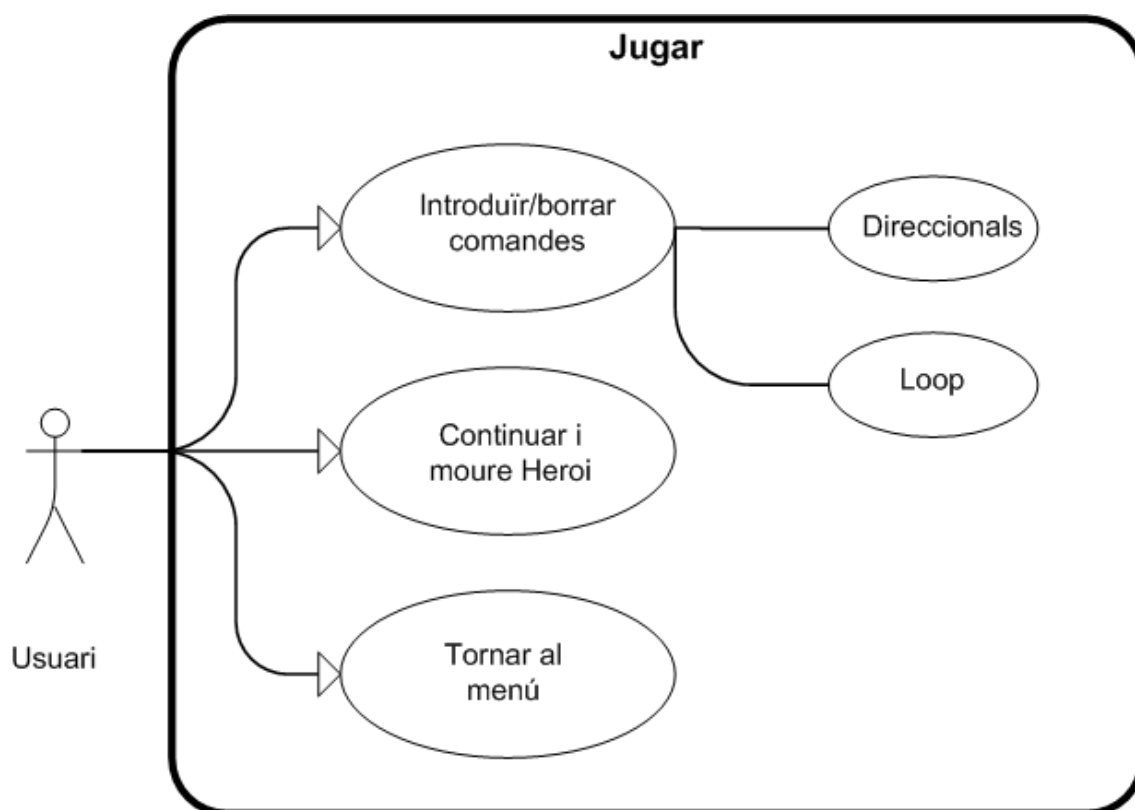
A continuació es mostren els diagrames dels casos d'ús del sistema per a explicar les interaccions que fa l'usuari amb el sistema d'una forma més visual.

S'utilitzaran varis diagrames per a explicar-ho:



Com es pot apreciar, hi ha 3 casos d'ús principals, als que s'accedeix mitjançant el menú principal:

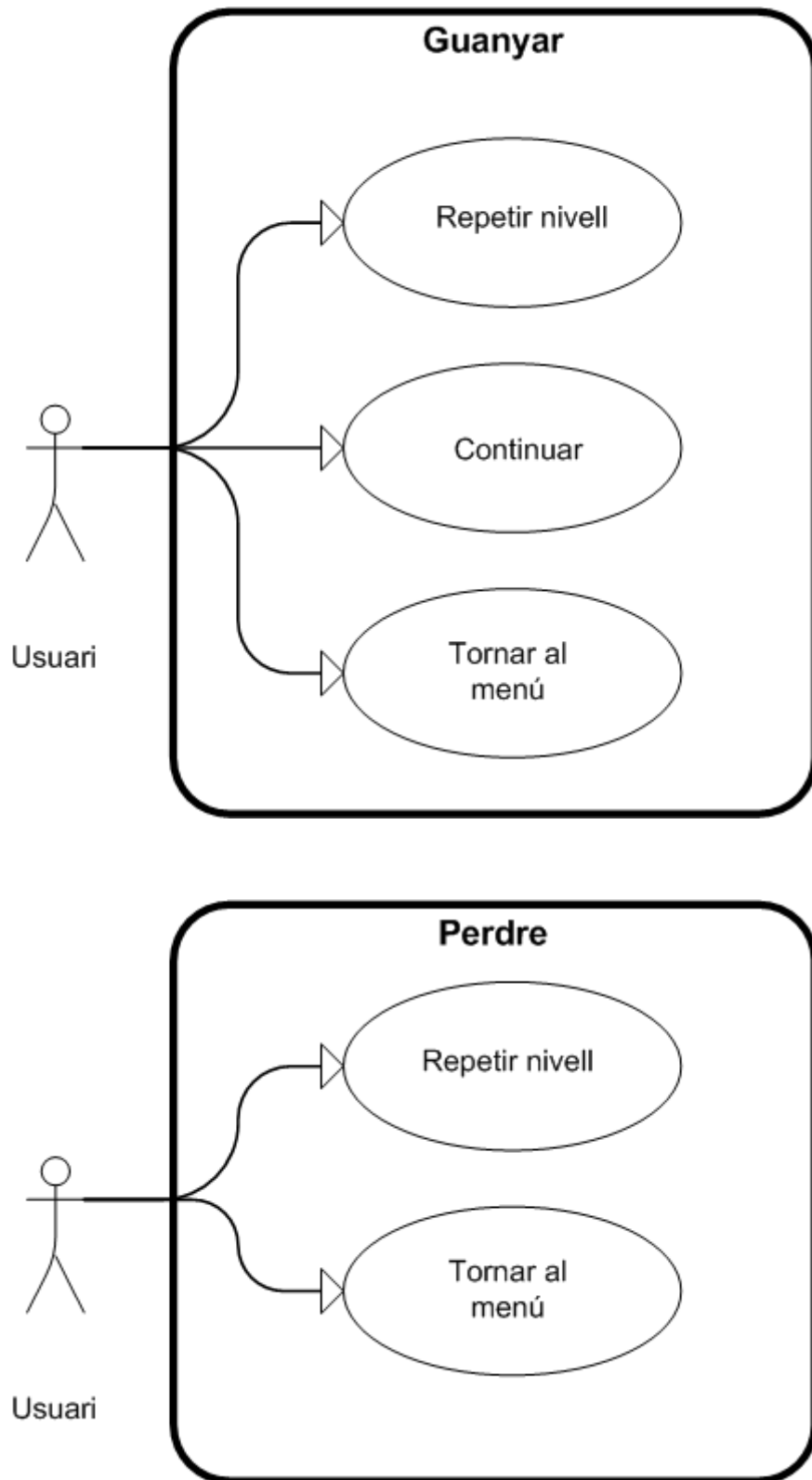
- **Jugar:** Començar la partida des del primer nivell. És el cas d'ús principal de l'aplicació ja que comporta el joc en si. L'explico amb més detall en el següent diagrama.
- **Selecció de nivell:** Aquest consisteix en un conjunt de requadres representant cadascun dels nivells que l'usuari pot prémer per a accedir directament al nivell seleccionat, començant la partida des d'allà.
- **Canvi d'idioma:** L'usuari pot prémer directament sobre l'idioma en que vol l'aplicació. Català, castellà o anglès. L'aplicació per defecte està en anglès.



Un cop dins d'un nivell, el flux de joc queda representat per aquest diagrama. El jugador introdueix comandes fins a tenir les que necessita per fer arribar al jugador a la moneda i un cop acaba pot continuar i fer moure l'heroi, o tornar al menú principal en qualsevol moment.

Els casos d'us són els següents:

- **Introduir/borrar comandes:** L'usuari fa drag & drop amb les comandes necessàries per a que l'heroi arribi a la moneda. Aquestes poden ser del tipus direccional: A dalt, a baix, esquerra, dreta, o poden ser un Loop format per un nombre limitat de comandes direccionals, que es repeteixen tantes vegades com defineixi l'usuari. Les vegades que es pot repetir un Loop és limitada.
- **Continuar i moure heroi:** Un cop introduïdes les comandes convenients, al prémer el botó de continuar simbolitzat amb un tick verd, l'heroi es mourà seguint en ordre les comandes que s'hagin introduït prèviament.
- **Tornar al menú:** En qualsevol moment de la partida l'usuari pot tornar al menú principal per tal de canviar l'idioma, tornar a començar des del nivell 1, o seleccionar un altre nivell per a jugar-lo directament.



En el cas de guanyar o perdre, l'única distinció és que al guanyar pots avançar de nivell, i al perdre no, però la resta d'opcions es mantenen. Aquestes opcions són:

- **Repetir nivell:** Torna a començar el nivell amb la llista de comandes buida si ha guanyat, o plena amb les accions que havia realitzat prèviament en cas d'haver perdut.
- **Continuar:** En cas d'haver arribat a la moneda, l'usuari pot decidir avançar al següent nivell. En cas d'haver arribat a l'última pantalla el sistema mostrarà el menú principal altre cop.
- **Tornar al menú:** Aquest és el mateix que en l'anterior diagrama, l'usuari pot tornar al menú principal en qualsevol moment de la partida.

### 6.3 Diagrames de seqüència

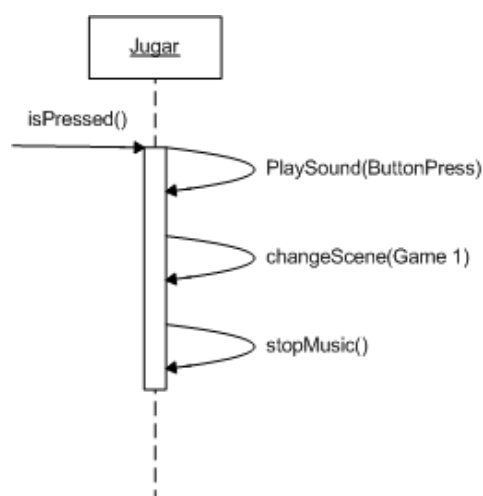
En aquest apartat s'exposaran els diagrames de seqüència dels casos d'ús del sistema:

En comptes de classes, es mostraran els diferents actors que realitzen els diferents comportaments. Es posaran en forma de mètode els diferents comportaments, tot i que en alguns casos es compactaran per tal de facilitar la comprensió del flux principal del cas d'ús, per tant els noms que es faran servir en els diagrames no han de correspondre exactament amb els noms dels comportaments en el codi del programa.

El que es pretén en aquesta secció és plasmar de manera entenedora el flux i comportaments del software en els diferents casos d'ús.

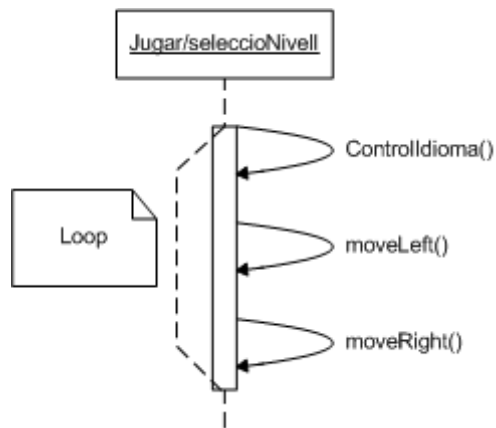
Algunes crides no es realitzen directament com s'indica en els diagrames, ja que un actor no es pot comunicar amb un altre directament, s'han de fer servir variables com a *triggers* dels diferents comportaments. Per facilitar la comprensió, es resumirà aquest fet representant-lo directament com una crida a l'actor corresponent.

#### Jugar/ Selecció de nivell



Quan el jugador prem el botó de jugar, s'executen els comportaments. En el cas del botó de selecció de nivell només canvia el canvi d'escena, que en comptes de 'Game 1', porta a 'Level Selection'.

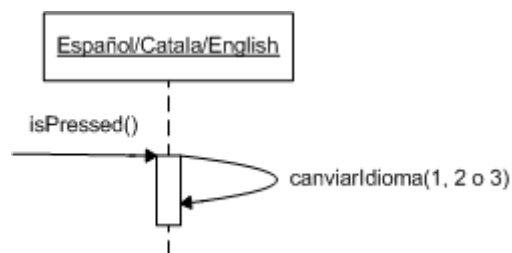
Mentre no es clica, els dos botons realitzen el següent Loop de comportaments:



El control d'idioma consisteix en comprovar la variable d'idioma i canviar la imatge del botó per a mostrar-lo en l'idioma corresponent.

moveLeft i moveRight són els comportaments que fan que els botons es moguin d'esquerra a dreta.

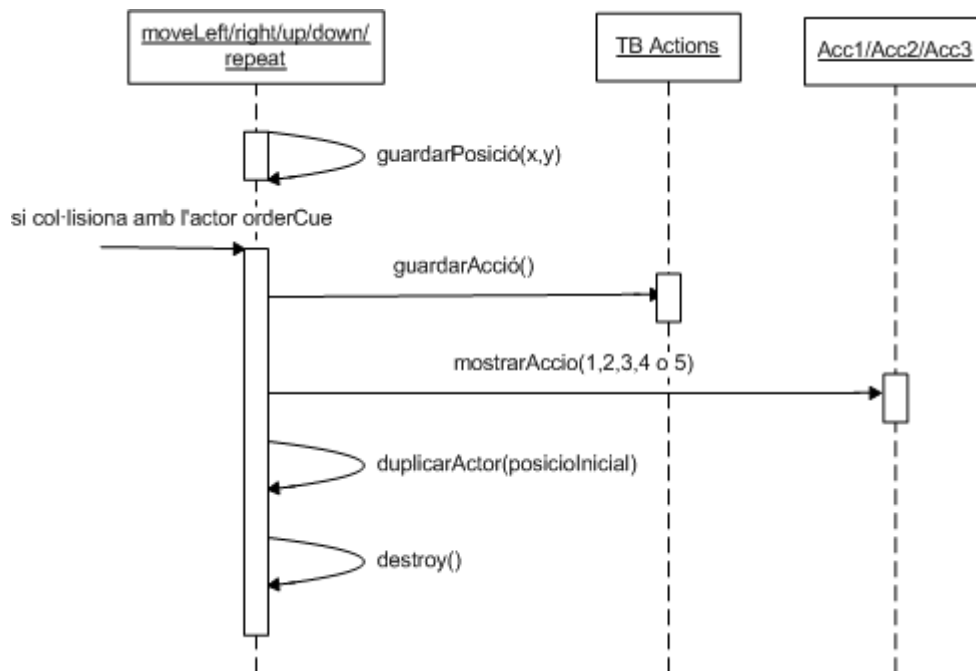
### Canvi d'idioma



Quan es prem qualsevol dels botons d'idioma, el que fa el software és canviar la variable de control d'idioma i doncs els actors que depenen de l'idioma, és a dir: les ajudes i els botons del menú.

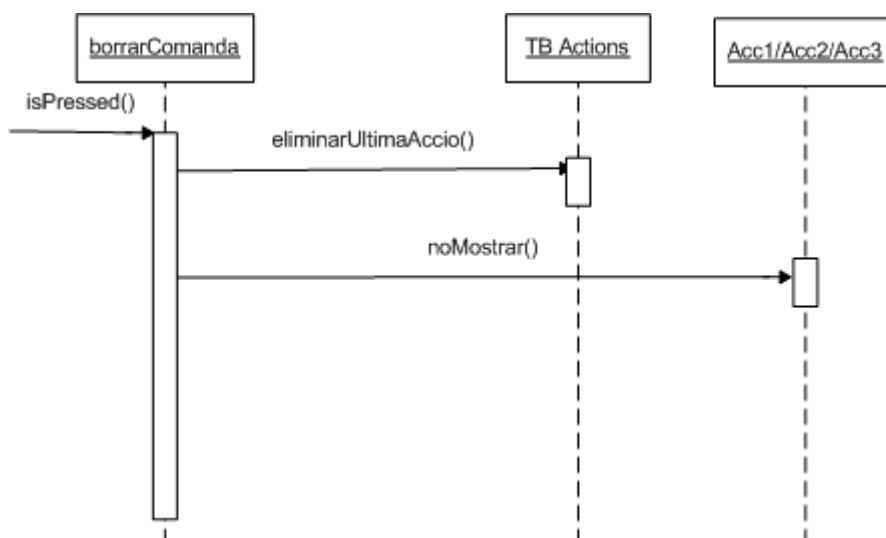


### Introduïr/borrar comandes



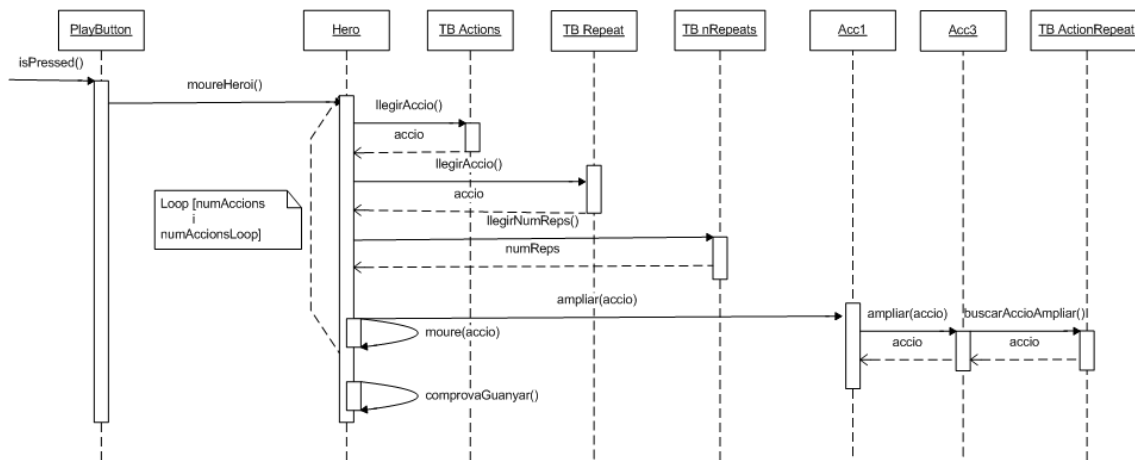
Quan arrastrem un dels botons al canvas que conté els espais buits per a les accions, guardem les accions en la taula TB Actions, en canvi si l'acció correspon a un Loop es guardaran a TB Repeat i TB nRepeats.

No he representat el control del número màxim d'accions que hi poden haver. En cas d'haver arribat al límit, el programa no deixarà a l'usuari afegir més comandes.



En el cas de l'actor borrarComandaLoop, el flux només canvia en que les accions s'esborren de diferents taules. Els actors Acc1, Acc2 i Acc3 els he col·locat aquí ja que segons el tipus de comanda es mostraran uns o altres.

## Continuar i moure heroi



Quan polsem el botó de continuar, el check verd, l'heroi comença a llegir una a una les accions a realitzar i les va realitzant una a una. Mentre s'executen, ampliarà els icones de dins del canvas.

Si l'acció a realitzar és un Loop, llegirà les accions corresponents d'aquest Loop a les taules corresponents per tal de llegir tant les accions com el número de repeticions del mateix. Per tal d'ampliar els botons, farà servir la taula TB ActionRepeat per anar llegint el botó actual a ampliar dels petits que hi ha a sota de les accions principals del canvas, ampliant el que toca en cada moment.

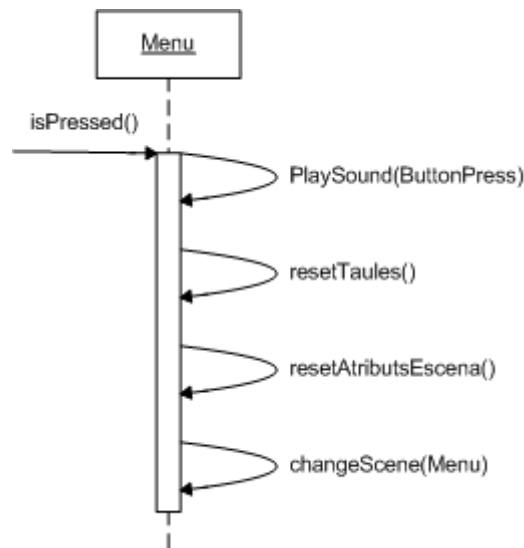
Els comportaments de comprovar guanyar els he hagut d'implementar diferent, en cada moviment detecta si s'ha arribat a la moneda o no, i si acaba la seqüència de moviments i no s'ha guanyat, detecta que s'ha perdut.

Cal indicar també que per a facilitar la comprensió, he inclòs a la nota els loops interns, un per buscar controlar les accions principals i un altre per a controlar les accions del Loop i les repeticions d'aquest.

En el diagrama he inclòs els comportaments de la comanda de repetir. Si la comanda és normal, només llegirà l'acció de la taula TB Actions corresponent i es mourà en conseqüència. Això ho farà per totes les accions. Si la comanda és repetir, es parará dins d'aquest Loop de TB Actions i realitzarà les comandes de dins tantes vegades com s'hagi indicat en el número de repeticions.

Per últim, indicar que la icona de la làmpada d'ajuda no està representat per no engrandir el diagrama, aquest seria un diagrama curt en el que al prémer el botó, l'ajuda es faria visible, i al tornar-lo a prémer, desapareixerà. Es pot visualitzar tants cops com calgui.

### Tornar al menú

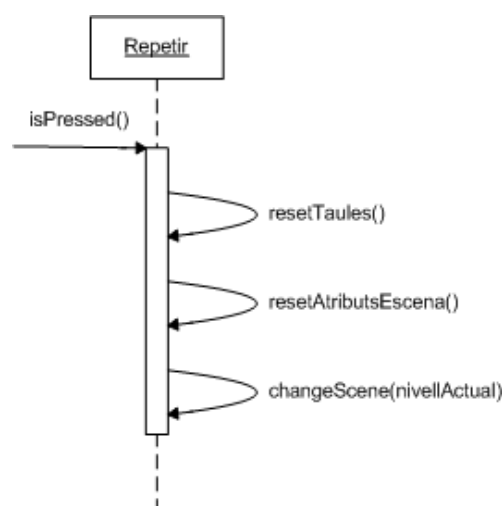


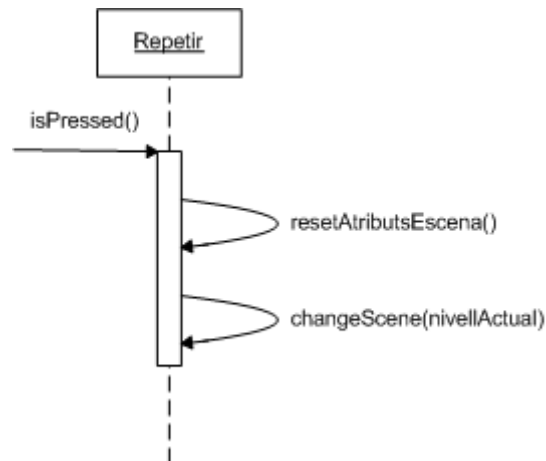
On posa resetTaules i resetAtributs escena, es refereix al conjunt de comportaments que posen els valors per defecte als atributs de control de l'escena i buiden el contingut de les taules.

### Repetir nivell

Quan premem el botó de repetir, varia el comportament segons si hem guanyat o hem perdut la partida. En el cas de guanyar, es reinicialitzen tots els valors i tornem al mateix nivell començant de zero, en canvi si hem perdut i seleccionem repetir, es guardaran les accions seleccionades prèviament i podrem continuar des d'allà.

A continuació s'exposen els 2 diagrames corresponents a l'actor Repetir, encarregat de gestionar aquest punt. El primer correspon a quan es guanya la partida, el segon a quan es perd.

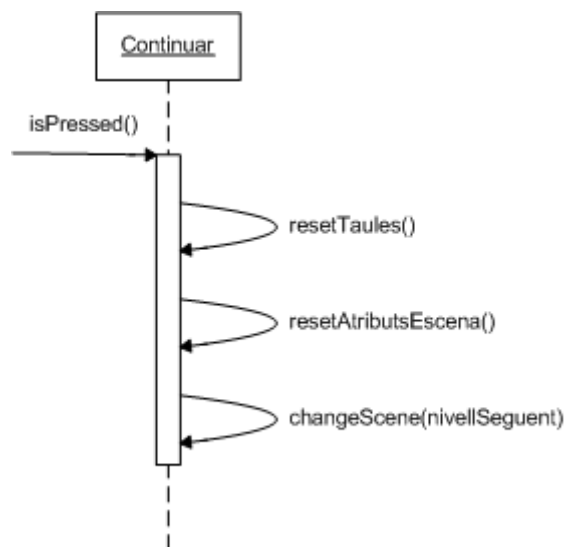




S'ha representat en dos diagrames el comportament, però es comproven alhora i s'executa un o altre segons si s'ha guanyat o no.

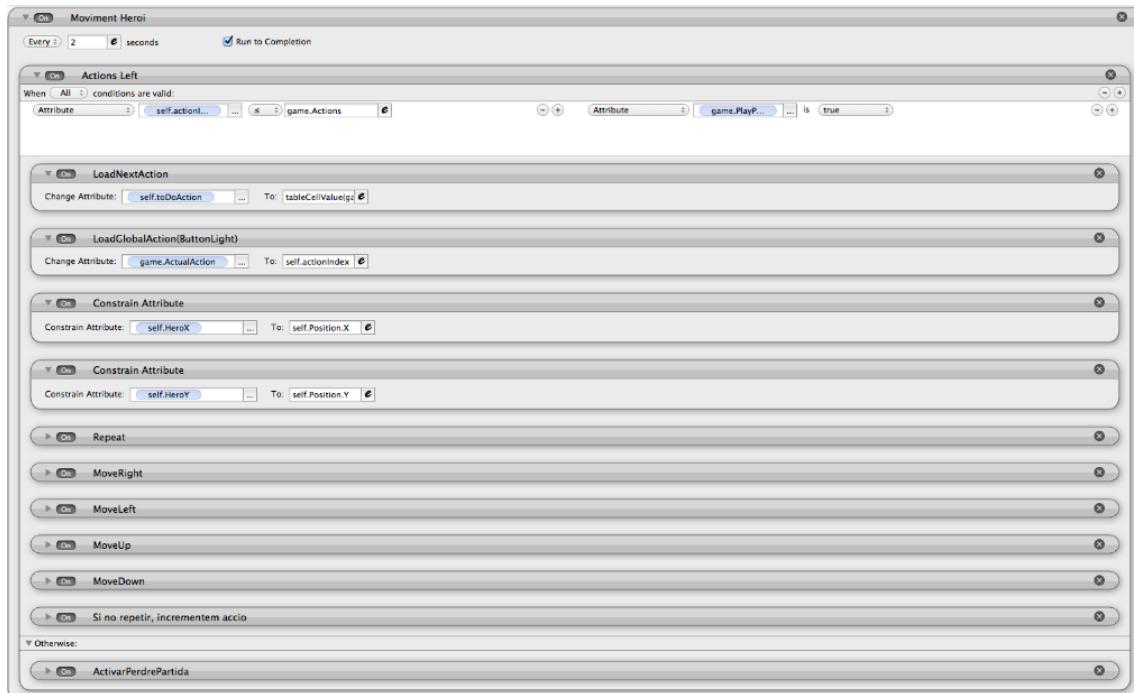
### Continuar

En cas d'haver guanyat la partida apareixerà, apart de la de repetir, la opció de continuar amb el següent nivell.

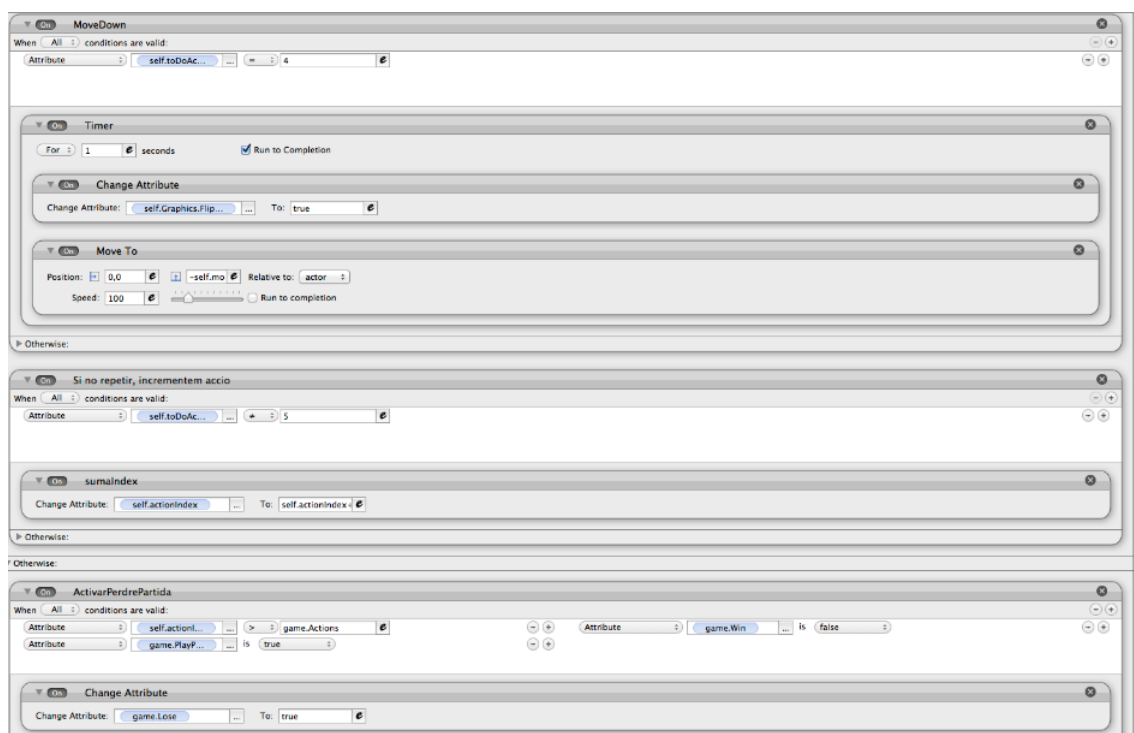


## 6.4 Captures de codi

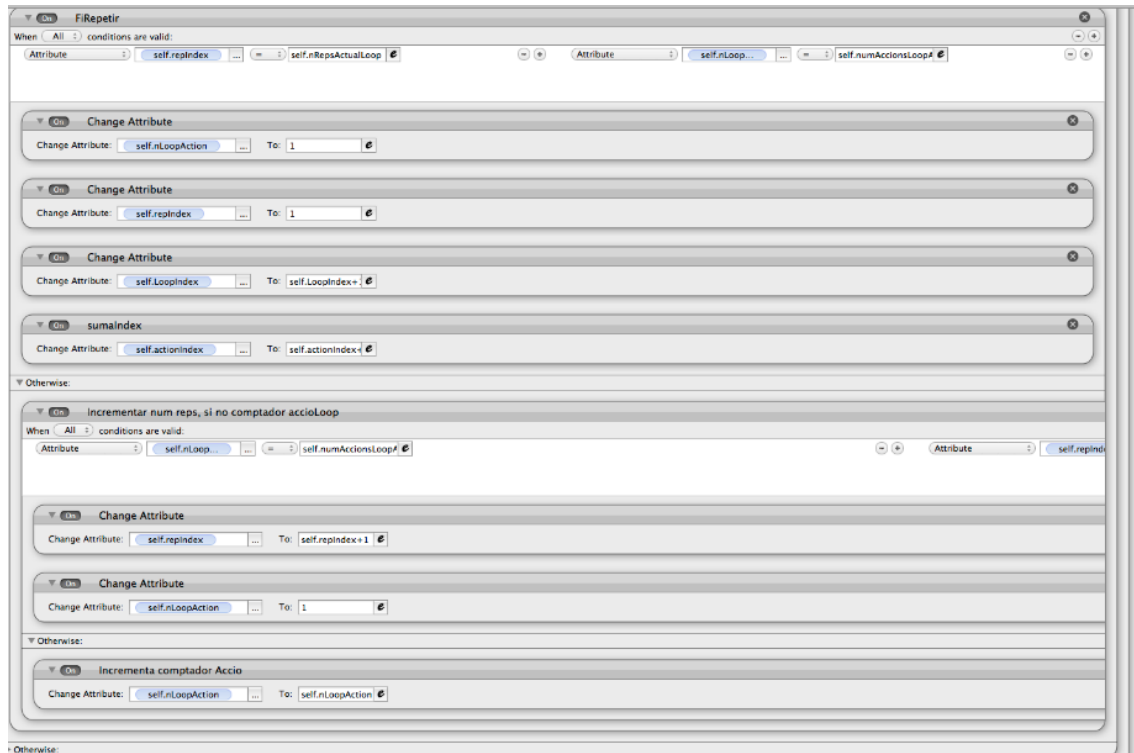
En les captures de pantalla que es mostren a continuació es pot veure com estan definits internament alguns dels comportaments de l'actor de l'heroi.



Aquesta imatge mostra l'estructura principal de l'heroi. Hi ha conjunts de comportaments que es troben minimitzats per tal de facilitar la visió general.



En aquesta captura es pot veure per dins la part del moviment cap a baix i la gestió del final del Loop, així com la detecció del final de la partida quan l'heroi no arriba a la moneda.



Per últim, es mostra la part del final de la execució de la part dels comportaments de l'heroi, les encarregades de gestionar les repeticions de les accions de dins del *Loop*. Va executant les accions fins que arriba a la última, doncs detecta si ha acabat amb les aquestes i mira si ha de repetir-les o no.

## 6.5 Estructura de l'aplicació

A mida que s'ha anat desenvolupant el software, es va observar que es podia encapsular de tal manera que permetés tenir un motor de creació de pantalles amb el que pogués fer-ne més d'una i tenir un joc més complet.

La idea ha agafat forma i així ha estat finalment. S'ha creat una escena prototip a partir de la qual es poden anar generant diferents pantalles en molt poc temps. De fet el joc actualment té 9 pantalles en comptes d'una, que era el que es proposava en un principi.

El software doncs, consta de 4 tipus d'escenes diferents:

- **Introducció:** Logotip UB i noms de tutora i alumne.
- **Menú principal:** Canvi d'idioma, començar partida o selecció de nivell.
- **Selecció de nivell:** Es mostren tots els nivells disponibles i s'hi pot anar directament fent clic en la foto que representi el nivell escollit.
- **Nivell:** És el tipus d'escena que representa cada nivell per separat.

### 6.5.1 Components principals: *Actors* i *Tables*

Gamesalad utilitza *actors* per a representar el concepte de classe en programació orientada a objectes, tot i que no es comporten ben bé com a tal i, per tant, optimitza la facilitat de creació, però limita la optimització del codi o comportaments, afectant directament als recursos utilitzats pel software resultant.

Els actors instanciats no poden canviar paràmetres o interactuar amb els altres actors a no ser que sigui en un joc.

A continuació s'explica amb detall els principals actors o classes del projecte i els seus rols en el projecte, així com les taules utilitzades i la seva funcionalitat.

#### 6.5.1.1 *Actors*

Tal i com s'ha explicat en l'apartat dels motors investigats, Gamesalad funciona mitjançant events efectuats pels diferents actors que componen el projecte.

En aquest apartat mencionaré els actors utilitzats en el projecte i la seva funcionalitat. En l'apartat de propostes d'ampliació, explicaré les modificacions fetes en els actors que han estat necessàries per a fer possibles els canvis.

Els actors exposats representen els actors instanciats, que són els que proporcionen la funcionalitat al projecte. Tot i això, la majoria d'actors instanciats tenen els mateixos

comportaments que els prototip. La diferència són els valors d'algunes variables internes al crear-se, que han d'inicialitzar-se per separat, actor a actor.

Els actors utilitzats són:

- **Play/Nivel:** Els actors són iguals però amb imatges i paràmetres diferents. Surten a la pantalla de menú, són els que representen el botó de  *jugar i seleccionar nivell*. Al clicar-los, salten a les escenes del primer nivell i a la pantalla de selecció respectivament.



- **Coin:** És l'actor que representa la moneda a la que l'heroi ha d'arribar per superar cadascun dels nivells. Es fa servir també en la pantalla de menú a nivell decoratiu tan sols per a que roti i hi doni dinamisme.



- **Move(right/left/up/down):** Són els botons de fletxa corresponents a les direccions. Apareixen en tots els nivells i serveixen per seleccionar les accions a emmagatzemar. Els comportaments interns els permet ser arrastrats fins a tocar amb el canvas que conté els cercles blaus que representen les diferents accions disponibles a realitzar.
- **Repeat:** És la comanda de *Loop* pròpiament. Si s'arrastra fins al canvas, ocuparà un espai d'acció i farà visible el canvas que gestiona el *Loop*.



- **Move(right/left/up/down) Loop:** Són els botons situats a dins del canvas del *Loop*, i activen els cercles de l'interior d'aquest, corresponents a la seqüència d'accions que volem repetir. Aquests no s'arrastren ja que estan dins del propi canvas del *loop*, s'activen al clicar-los. Cal dir que després de passar el primer test d'usabilitat, s'han eliminat ja que els nens presentaven confusió amb els botons del canvas principal.





- **Hero:** És l'actor que representa l'heroi. És el que accedeix a les comandes enregistrades per l'usuari i les va executant una a una i en ordre. Reacciona als obstacles i als murs de col·lisió invisibles.



- **OrderCue:** És el canvas que conté els espais de les accions, la goma d'esborrar i el *check* per a executar les ordres. És on s'ha de fer xocar les accions.



- **PlayButton:** Aquest actor correspon al botó amb el *check* que, un cop clicat, activa a l'heroi per a que comenci a executar les accions enregistrades, que es mostren en ordre en els espais blaus.



- **Acc1, Acc2, Acc3:** Són els actors corresponents als espais en blanc que corresponen a les accions introduïdes. El nom és l'abreviatura d'acció. Acc1 correspon a les accions principals del canvas de la dreta, Acc2 a les del *Loop* i Acc3 a les que hi ha a sota de les primeres, que equivalen a les mateixes accions de dins del *Loop* representades pels actors del tipus Acc2.



- **nReps:** Botó que mostra el número de repeticions de les accions del *Loop*. N'hi ha un a sota de cada acció, i es fa visible si s'hi introdueix la comanda *Loop*.



- **BorrarAccio/BorrarAccioLoop:** Corresponen a la goma d'esborrar. El primer esborra la última acció de la taula que guarda les accions del canvas principal. El segon fa el mateix per a les accions de dins de cada Loop. Per a fer-ho esborra l'acció de la taula corresponent.



- **Colision/ColisionRock/Colision Gem:** Objectes amb els que pot xocar l'heroi. Corresponen a les pedres i a les parets o murs invisibles que delimiten el camí pel que passa el protagonista, que és qui controla el comportament quan això passa. En un principi eren gemes i una marieta. Tenir en compte que aquests actors no tenen una imatge pròpia, són actors sense imatge, cada instància té la seva. Abans de la primera ampliació eren representats per gemes i la marieta, més endavant per imatges de pedres.



- **nActions:** Controla els paràmetres necessaris a ajustar en cada nivell, com ara el número d'accions, el número de pantalla, o si s'ha d'activar l'ajuda de la pantalla o no.
- **LoopWindow:** És el canvas que conté tots els components del Loop. Apareix quan arrastres una orde de Loop al canvas principal.



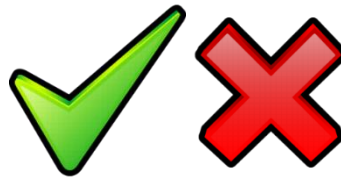
- **Ok/Cancel Loop:** Botons rodons situats dins de l'actor *LoopWindow*, un en forma de *check* i l'altre de creu vermella. Són els que activen o cancel·len l'acció de Loop un cop introduïdes les accions que desitgi l'usuari. Si es clica l'actor *OkLoop*, es guardaran les accions assignades a la taula corresponent i es mostraran en el canvas principal. Si es clica *CancelLoop*, s'esborraran les accions introduïdes i també es cancel·larà la última acció del canvas principal corresponent a aquest Loop.



- **ButtnLeft/Right:** Actores que representen les fletxes amb que, dins el Loop, seleccionem el número de repeticions que surt representat a l'actor *nReps*.



- **Win/Lose:** Representa el *tick* verd o la creu vermella que surt per pantalla quan guanyes o perds la partida, a sota apareixen al mateix temps un actor del tipus *Repeat* i un del tipus *Continue*, explicats a continuació.



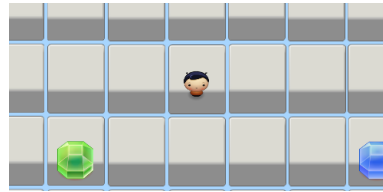
- **Repeat:** S'encarrega de reiniciar l'escena. Si la partida s'ha perdut, guardarà abans les dades de la partida actual per tal de repetir la partida amb les comandes actuals introduïdes. Això permet de reutilitzar les últimes comandes introduïdes per tal d'arribar a la solució correcta. Per altra banda, si s'ha guanyat, esborrarà les accions introduïdes i començarà de 0 el nivell. En la primera ampliació ha estat canviat per un altre icona de color verd.



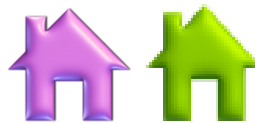
- **Continue:** Apareix quan guanyes a sota de l'actor *Repeat*, aquest té forma de fletxa. Al clicar-lo accedeixes al següent nivell. En la primera ampliació ha estat substituït per una fletxa més senzilla de color verd.



- **Level:** Actor que al clicar-lo fa un canvi d'escena a un nivell concret introduït per paràmetre. Són els nivells que apareixen a la pantalla de selecció de nivell. N'hi ha tants com nivells al joc.



- **Menu:** Actor en forma de casa que apareix a la cantonada superior esquerra de cada nivell. Es pot clicar en qualsevol moment de la partida i es realitzarà un canvi d'escena al menú.



- **Anglès/Català/Español:** Són els botons de selecció d'idioma del menú principal. Només apareixen en la pantalla de menú i canvien l'idioma dels icones i les ajudes al llarg del joc al clicar-los.



- **showHelpBttn:** És l'actor que representa el botó de lampareta màgica que apareix a la cantonada inferior dreta dels nivells que tenen algun text d'ajuda. Serveix per a mostrar/ocultar l'ajuda. Clicant al damunt del propi text també desapareix. A la versió de Mac has de clicar a la icona, no funciona clicant al damunt del text.



#### 6.5.1.2 Tables

Per a guardar totes les accions i dades necessàries per a gestionar la partida he utilitzat diverses taules. Internament, en les taules, cada tipus d'acció té un número del 1 al 5, els quatre primers corresponen a les accions de moviment i el 5 correspon al *Loop*.

L'objectiu d'aquestes taules és poder emmagatzemar en memòria, les accions que l'usuari va introduït al canvas principal, s'encarreguen d'enregistrar-les en ordre i de

gestionar les comandes introduïdes dins dels loops que hi hagi, així com el número de repeticions dels mateixos.

També gestionen la il·luminació de les comandes internes del *Loop*, el fet que es facin més grans els icones de l'acció actual mentre l'heroi es mou.

Les taules utilitzades són les següents:

- **TB Actions:** Taula que guarda les accions que es van arrastrant al canvas. Faig servir un número diferent per a cada acció, i la fila *n* indica el número en la llista. En el cas de l'acció repetir, es deixa indicada també i apart guardo les accions de dins i el número de repeticions de cadascuna.
- **TB Repeat:** Guarda les accions dins de cada *Loop*. La columna correspon a la posició del *Loop* en el canvas d'accions, la fila al número d'acció dins del *Loop*.
- **TB nRepeats:** Taula que guarda el número de repeticions de cada *Loop*. Les columnes equivalen al número de *Loop*. El número de *Loop* serà equivalent a la posició que aquest tingui entre les accions del canvas principal, per exemple, si es tracta de la primera acció, serà el *Loop* 1, i es guardarà el seu número de repeticions en la columna 1.
- **TB RepeatOn:** Es tracta d'una taula de booleans inicialitzats a *False*, que guardarà *True* en la posició de la fila equivalent a la acció principal si aquesta és un *Loop*, i la deixarà com estava en cas contrari. Serveix per detectar si la acció actual és un *Loop* o no. Actualment la taula no es fa servir ja que la informació es pot extreure de la taula *TB Actions*.
- **TB ActionRepeat:** Actua com a indicador per als botons de les accions internes del *Loop* que es mostren a sota d'aquest un cop creat, si estan a *true* es mostren. Això fa que quan creem un *Loop* es mostrin a sota les comandes.

Així doncs, imaginem que introduïm una comanda de *Loop* com a tercera acció en el canvas principal i l'omplim amb una acció d'anar a la dreta, seleccionant 4 com a número de repeticions per a que l'heroi es desplaci quatre espais.

La informació es guardarà de la següent manera:

- **TB Actions:** Guardarà el valor 5, que és el número d'acció que representa el *Loop*, en la tercera fila de la taula.
- **TB Repeat:** Guardarà el valor 1, que representa l'acció d'anar a la dreta, en la primera fila de la tercera columna de la taula.
- **TB nRepeats:** Es guardarà el valor 4, el número de repeticions, en la tercera columna.
- **TB ActionRepeat:** Mentre l'heroi executi les ordres del *loop*, s'aniran posant a *true* els valors de la taula. Els actors que representen les comandes de *Loop* s'il·luminaran, o es faran més grans, quan el valor de la seva posició sigui *true*.

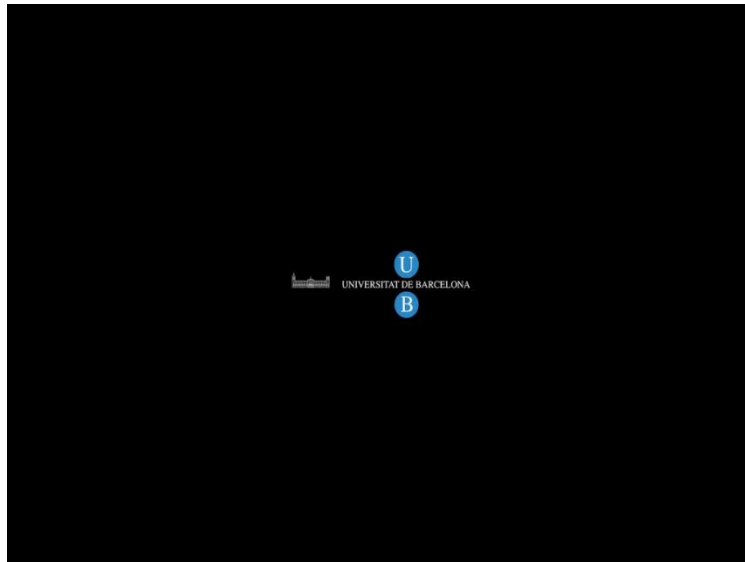
## 6.6 Captures de pantalla

En aquest apartat s'exposaran captures de pantalla del software quan estava en la seva versió funcional i algunes en les versions prèvies per tal de que es puguin apreciar visualment els canvis i, en conseqüència, l'evolució de la feina realitzada. Posteriorment a l'apartat d'ampliacions es mostren captures del joc després d'aplicar-les, que seran les captures corresponents al joc tal i com es troba actualment.

### 6.6.1 Captures del projecte principal

No es mostren captures de tots els nivells, tan sols les captures corresponents als diferents tipus d'escenes del projecte.

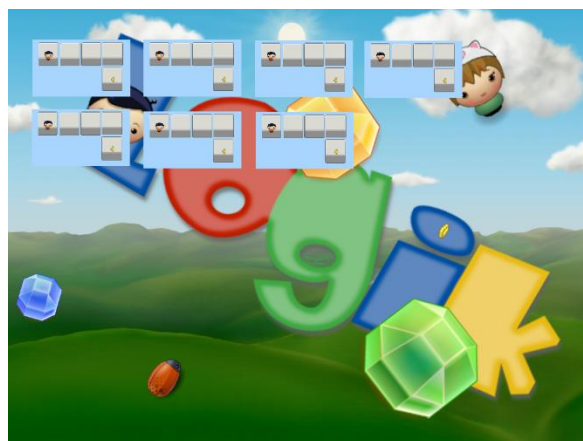
#### **Introducció o splash d'inici:**



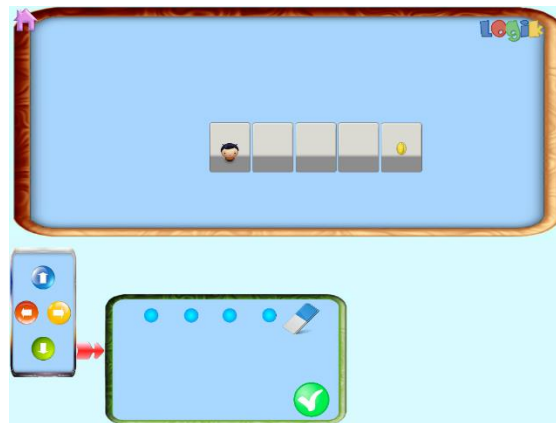
En aquesta pantalla, es mostra el logotip de la UB 3 segons abans de passar a l'escena del menú principal. Si fem clic en el logotip també passem al menú sense haver d'esperar els 3 segons.

**Menú principal:**

És la pantalla on s'escull començar a jugar des del primer nivell o si per contra escollim anar a la selecció de nivell. En aquesta pantalla també podem canviar l'idioma de l'aplicació si cliquem als icones de bandera corresponent. Ha estat una pantalla que ha patit molts canvis estètics de de que es va crear.

**Selecció de nivell:**

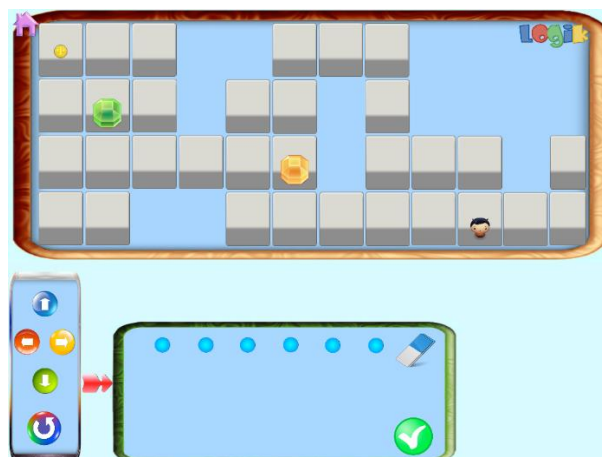
En aquesta pantalla tenim miniatures dels diferents nivells als que accedim clicant-hi a sobre. Com que no era la versió definitiva, es va utilitzar la mateixa imatge per a tots els nivells per tal de tenir la funcionalitat encara que no es tinguessin definitivament dissenyats els nivells.

**Nivells:**

Aquesta captura correspon al primer nivell d'aquesta versió del software. No s'exposaran captures de cada nivell per no ocupar més espai del necessari ja que es poden veure els nivells executant el joc i consultar-ho des d'allà.

El nivell, tal i com s'ha explicat en seccions anteriors, consisteix en anar arrastrant les comandes de l'esquerra al canvas de la dreta en l'ordre adequat per a arribar a la moneda. Un cop introduïdes cal clicar a la icona de *check* per a fer que l'heroi es mogui seguint l'ordre proposat.

En els nivells més avançats s'han inclòs obstacles en forma de gemes i una marieta que estan distribuïts per nivell.

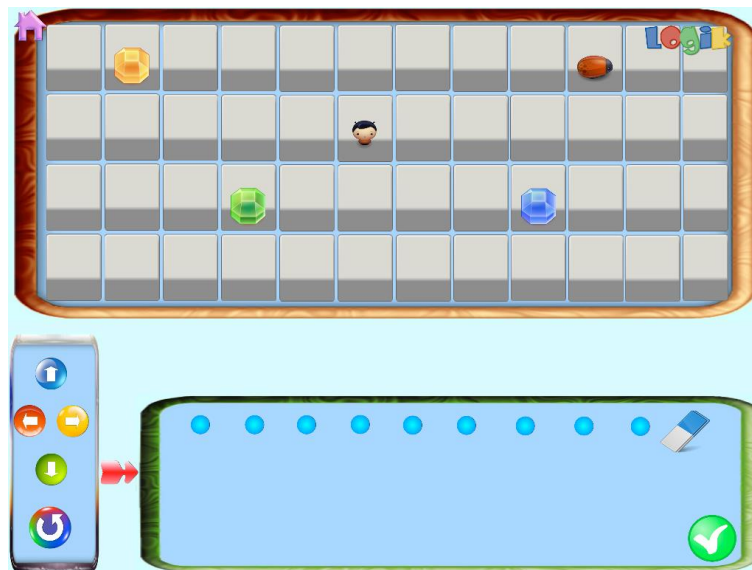
**Nivell prototip:**

Com s'ha mencionat en apartats anteriors, s'ha decidit fer una escena que representi un nivell prototip, amb les taules i atributs necessaris per tal de poder fer-ne més a partir d'aquest i poder unir-les.

Aquesta consisteix en una mena de plantilla amb tots els *slots* possibles per a les accions i tot el canvas del joc ple de caselles per tal d'esborrar les que calgui per a



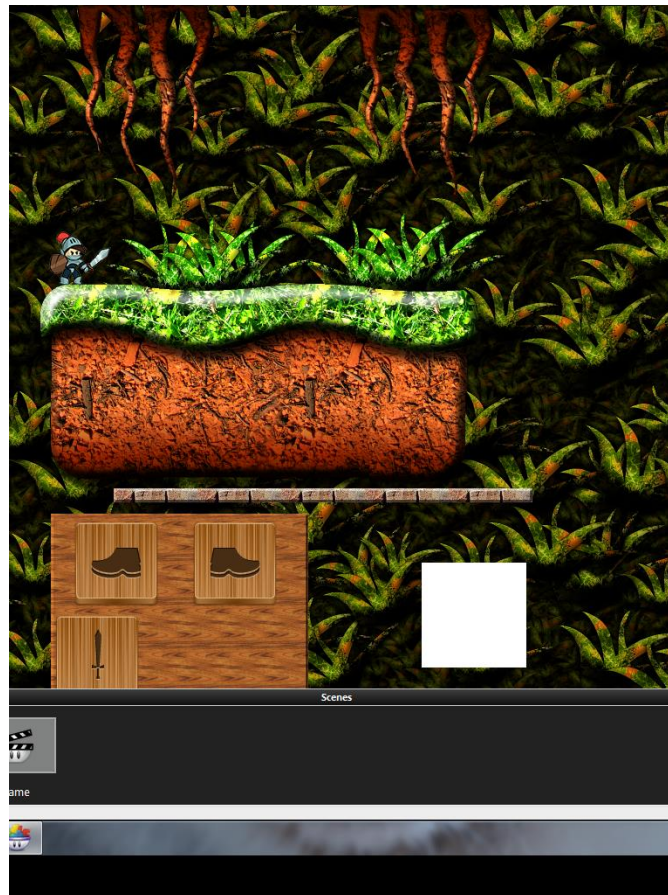
dissenyar el camí desitjat i llavors col·locar a l'heroi i la moneda on vulguem per a crear el nou nivell.



#### 6.6.2 Captures en fase de desenvolupament.

Abans de començar a desenvolupar el projecte des de Mac, vaig començar a desenvolupar sota la versió de Windows de Gamesalad. Al poc de començar, vaig experimentar molts problemes i vaig decidir canviar i començar de zero amb la versió de Mac. Vaig arribar a fer la pantalla de menú i un petit screenshot del que podria haver estat el primer nivell de LogiK en un principi.



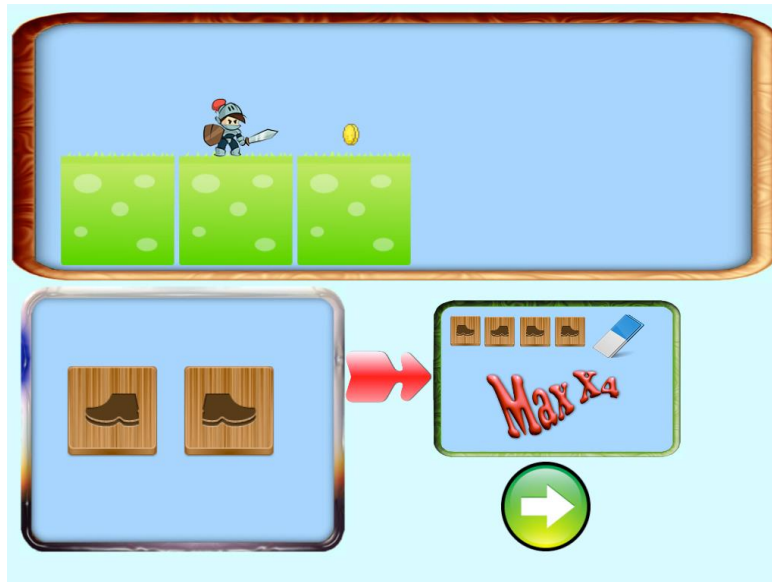


Com es pot veure, l'aparença no té res a veure amb el que és ara el joc. Va ser a partir d'aquí quan es va decidir canviar el look del joc per un de més infantil i senzill.

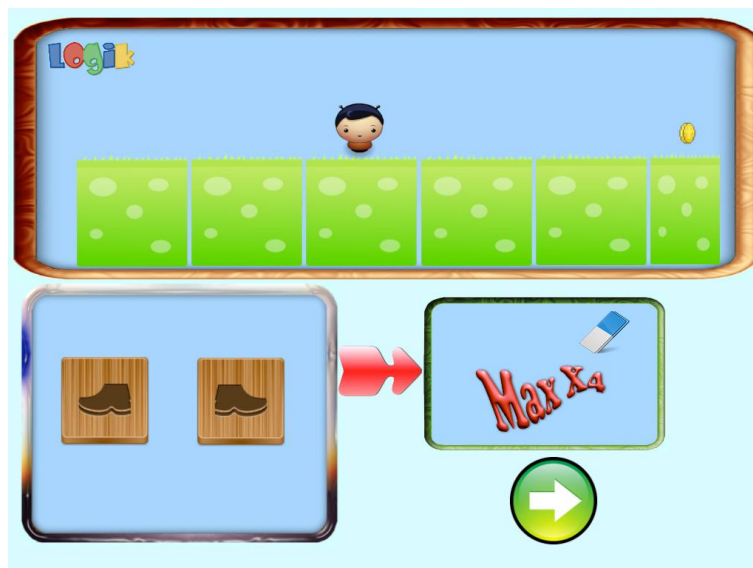
En vista d'aquest canvi necessari i els problemes a l'hora de testear l'aplicació, es va decidir canviar a la versió de Mac i tornar a començar el projecte sense tants contratemps. El primer canvi important en la visualització va ser el canvi a la visualització horitzontal per tal de poder mostrar nivells i el canvas d'introducció de comandes més grans.



Es van poder aconseguir i descarregar *free-Assets* més acord amb la imatge de joc que es volia donar en un principi, com ara l'escenari, els canvas o els nous herois. A partir d'aquí i paral·lelament es comença a desenvolupar la funcionalitat de les comandes de moviment i a preparar i pensar en com incloure la comanda de *loop*.



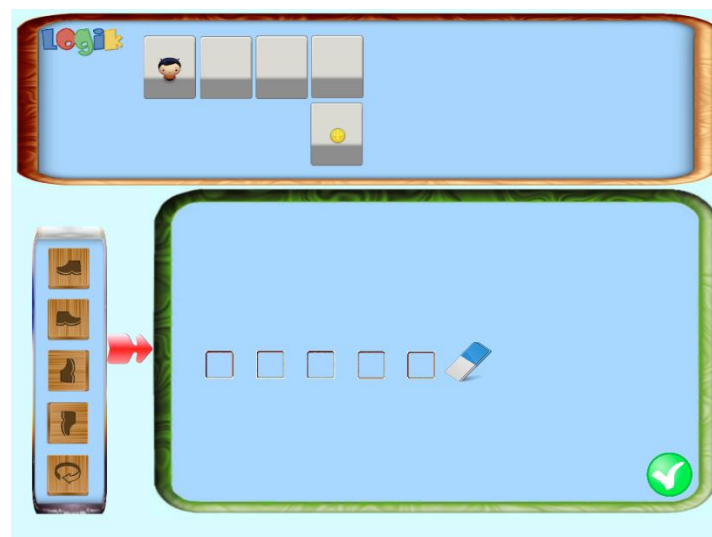
Com es pot veure en la imatge anterior i en la següent, en un principi el número màxim de comandes apareixia en forma de text a sota del canvas de les accions a executar. També en un principi els moviments eren horitzontals.



Per tal d'augmentar la jugabilitat, es va decidir incloure els moviments verticals i canviar els *Assets* del canvas de joc per tal d'habilitar-los. Amb això es guanya moltes més possibilitats de camins per a arribar a la moneda aprofitant el propi espai que permet el canvas de joc.



Com a últims canvis, es va fer que es veiessin les accions buides en el canvas de la dreta i les comandes de repetir també en cas de que la acció fos un Loop, es va reajustar el tamany de les comandes i també el tamany dels canvas per a que estigués més acord amb la usabilitat, reduint el de les comandes i engrandint el que conté les accions a realitzar.



Després d'aquests canvis ja la resta va consistir en testeig, retocar el sistema de col·lisions, canviar i dibuixar alguns icones com ara els de les comandes, o el de repetir i continuar.

Després d'això es va crear el nivell prototip i es va iniciar la creació dels nous nivells per a preparar l'aplicació per el primer test d'usabilitat.

## 7. Justificacions, problemes i dificultats trobades

En aquest apartat s'expliquen tots els contratemps i justificacions de certs comportaments d'alguns actors que s'haguessin pogut fer d'una altra manera però que es van haver de modificar per tal d'evitar problemes de conversió.

També s'exposaran tots els problemes experimentats a l'hora de desenvolupar i les decisions preses al respecte.

Per començar, com ja s'ha esmentat en altres apartats, es va començar el projecte amb la versió de Windows de Gamesalad.

Després d'haver llegit la documentació, manuals i haver començat a dissenyar l'esquelet funcional del joc, es va començar a experimentar problemes a l'hora d'executar i provar el software. Els Assets no apareixien on els havia col·locat i havia de col·locar-los manualment, apart no podia moure'ls de lloc un cop s'havien creat, s'havien de crear ja directament en la seva posició final.

Més endavant, al començar a introduir més actors, es bloquejava el programa al compilar.

També hi ha hagut problemes amb els comportaments ja que el software no es comportava de la manera indicada en aquests.

Arrel d'això i en vista del temps per a realitzar el projecte es va decidir no seguir desenvolupant sota aquestes condicions i es va poder fer ús d'un Mac, instal·lar-hi la versió d'aquest i desenvolupar des d'allà.

La versió de Windows no està encara 100% funcional i presenta alguns problemes que pel tipus de projecte no ha resultat ser gens viable. Tot i això ha permès apreciar les diferències entre les versions de Gamesalad, ja que la versió de Mac es diferencia en molt poques coses, apart de ser infinitament més còmode de fer servir, en relació a la interfície, un dels punts molt a favor ha estat el fons transparent de les imatges i el poder ampliar, reduir, rotar i moure els actors sense problemes a l'hora de compilar.

Per a publicar per a l'iPad de forma legal des de qualsevol motor, es necessita disposar d'una llicència d'Apple Developer, i es va optar per exportar l'aplicació a HTML5 ja que és un format que permet ser executat des de gairebé totes les plataformes.

No obstant, en la versió de Mac també s'han experimentat problemes a l'hora de convertir l'aplicació a HTML5, obligant a canviar i retocar els comportaments per a que l'aplicació es mostrés correctament.

Canvis que per sort, ja que el projecte és petit, no afecten al rendiment d'aquest, però que afecten molt a l'ús de recursos del dispositiu en que s'executa pel tipus de joc que és.

Entre ells, per exemple, està el fet de que els actors en si mateixos no poden actuar entre si, a no ser que sigui en un xoc.

Això obliga a l'hora de desenvolupar el software, a haver de fer servir variables de control i taules per tal d'organitzar el comportament dels actors, ja que en el cas que ens ocupa, els moviments no són directes, és a dir, que l'heroi no s'ha de moure al prémer cap botó sinó que ha d'anar guardant les comandes que anem arrossegant per a després executar-les en el mateix ordre. Això implica haver d'activar una variable global quan l'actor es comporti de la manera que vulguem gestionar.

Aquest fet ha afectat directament a la encapsulació, ja que si s'hagués fet servir algun altre sistema en que es programés directament en codi com ara XCode, probablement s'hagués pogut encapsular d'una altra manera els actors, així com degut als problemes de conversió a HTML5 també es va haver de canviar la disposició i funcionalitat d'alguns actors i taules per tal d'ajustar-ho. Finalment es va optar per obtenir una llicència d'Apple Developer per a poder realitzar de forma més òptima i ràpida la tasca d'exportació ja que aquesta presentava molts problemes.

Curiosament, al deixar d'exportar el projecte a HTML5 i exportar-lo directament en format \*.ipa va deixar de donar la majoria de problemes i s'ha pogut finalitzar el software i obtenir-ne una versió funcional.

Tot i això, degut a la relativament petita envergadura del projecte, no ha afectat en absolut al rendiment d'aquest i el software funciona correctament segons les funcionalitats aplicades, però sí en el seu disseny.

El haver optat per a fer una "pantalla prototip" i crear a partir d'aquí un conjunt de nivells, ha suposat haver de compartir i controlar certes variables, actors i taules per tal de no causar conflictes a l'hora de tornar a carregar dades.

Entre els canvis més substancials i rellevants, es troba el fet d'obligar a utilitzar *timers* per a realitzar comportaments de Loop. En el propi fòrum de Gamesalad es parla de l'abús de recursos que suposa aquest comportament, i de solucions al respecte, com ara la variable interna '*Game.Time*'. No obstant, al exportar el joc a HTML5 el joc no es comportava de la manera desitjada i es van haver de tornar a fer-los servir, afectant a molts dels comportaments i estructura del joc.

També per aquest fet i problemes de comportaments a l'hora de la exportació, es va haver de trencar algunes encapsulacions que milloraven el rendiment i estructura del projecte per altres comportaments més simples i que utilitzen més recursos.

Per altra banda ha limitat el fet de no poder interactuar d'un actor amb un altre, o poder fer inicialitzacions de forma global en un conjunt d'actors. Això s'ha hagut d'implementar amb variables i taules.

En quant a les variables i taules, al realitzar l'exportació es va experimentar molts problemes de funcionalitat i es va haver de reestructurar la aplicació una i altra vegada, experimentant cada vegada problemes diferents.

En vista de que el temps començava a passar i la aplicació no acabava de ser funcional, es va decidir investigar al respecte, el per què al compilar des de Gamesalad l'aplicació era 100% funcional i no obstant, al exportar-la a HTML5 sempre presentava un problema o altre.

No va ser difícil, navegant poca estona per la xarxa es varen poder consultar bastantes planes web que parlaven dels problemes que tenen tots els motors que exporten a HTML5 actualment, moltes parlaven també de Gamesalad i en el propi fòrum es poden llegir comentaris d'usuaris que també experimenten problemes al respecte.

Finalment, i tenint en compte aquests factors, es va optar per tornar a canviar la plataforma d'exportació, ja que el projecte inicialment es va pensar per a ser desenvolupat per a l'iPad, es va poder adquirir una llicència d'Apple Developer i experimentar amb la exportació en plataforma Mac i iPad.

Com no era d'estranyar, l'exportació a l'iPad mitjançant Wi-fi és comodíssima i ràpida, i els problemes d'aquesta van disminuir gairebé al 100%.

Va ser a partir de llavors quan es va poder acabar de dissenyar el primer prototip jugable de LogiK.

Es va decidir no refer de zero l'aplicació degut al temps del que es disposava, però probablement si s'hagués tingut la llicència des d'un principi no s'haguessin experimentat tants problemes i de ben segur s'hagués pogut conservar l'estructura interna que es tenia en un principi, aprofitant molt més els recursos i més encapsulada dintre dels límits de Gamesalad.

## 8. Test d'usabilitat

Un cop es va tenir el software 100% funcional i amb nivells de sobra per a jugar, es va realitzar un test d'usabilitat amb dues nenes de 5 anys.

Es van gravar en vídeo per tal de poder extreure'n un feedback que poder analitzar en busca de canvis i noves necessitats que poguéssim trobar.

L'objectiu d'aquest test ha estat poder actualitzar el joc de manera que sigui més intuïtiu i més acord amb l'estil de joc i necessitats del públic al que va dirigit.

Es van dedicar uns 10 minuts a cadascuna i es va prestar especial atenció amb el que tenien dificultat o problemes a l'hora de jugar per tal d'actualitzar el joc de la forma més útil possible per a l'usuari.

Els vídeos es troben en els annexos en format digital.

### 8.1 Resultats test

Els resultats del test van ser satisfactoris i les nenes van aconseguir jugar diversos nivells amb les explicacions pertinents sobre el funcionament del joc, no obstant també hi van haver punts que podíem millorar.

Al observar a les dues nenes jugant, es va descobrir que tenien problemes en diversos punts del joc que s'esmenten a continuació:

- Problemes al fer desaparèixer l'ajuda del joc a causa de tenir un fons transparent, apart de que no s'acabava de llegir del tot bé al barrejar-se les lletres amb el fons del joc.
- Arrastraven les comandes de direcció del canvas de l'esquerra en comptes de fer servir els botons de dins del canvas del Loop. Se'ls havia de recordar que s'havien d'introduir des d'allà.
- Els costava veure les fletxes de repetir i continuar al acabar de partida.
- Tenien dificultat en veure i clicar en els botons per escollir el número de repeticions, no el trobaven.
- La corba de dificultat del joc era massa elevada i es cansaven ràpid quan l'escenari tenia camins més llargs.
- El concepte de Loop, tot i que l'acabaven utilitzant correctament, semblava que no l'acabaven d'assimilar del tot.
- Els obstacles els hi semblaven més aviat objectes a aconseguir que no pas el que eren i els confonia. Se'ls hi havia d'explicitar que eren obstacles i havien de provar d'evadir-los per a arribar a la moneda.
- La lentitud de l'heroi i la falta de dinamisme feia avorrida l'espera mentre l'heroi executava les comandes.



## 9. Ampliació

Arrel d'aquestes observacions, s'havia de trobar la manera de pal·liar o reduir al màxim totes les dificultats que van tenir en comú les dues nenes.

Es va decidir doncs canviar una mica l'aspecte del joc, alguns aspectes funcionals i afegir-hi algunes millores que millorin l'experiència tenint en compte el temps disponible per a l'entrega.

Al següent apartat s'explica en quins punts ha consistit aquesta ampliació i més endavant es mostren captures de com ha quedat finalment el software.

Un cop implementades aquestes millores, s'ha pogut tornar a testejar l'aplicació amb una nena i un nen de 5 i 7 anys respectivament, dels que s'ha extret un nou feedback i es proposen futurs canvis a realitzar e implementar en LogiK.

### 9.1 Canvis realitzats

Els canvis que s'han acabat fent al software per a implementar els punts a millorar obtinguts del feedback del test són els següents :

- S'han modificat els textos d'ajuda per a que es visualitzin correctament. Els he afegit un requadre per tal de que no interfereixi el text amb el fons de pantalla i el nen pugui clicar tant en el mateix requadre com en la lampareta per a fer que desaparegui.
- S'ha fet que els icones de les comandes s'ampliïn mentre l'usuari els selecciona i els arrastra.
- S'han eliminat els icones de dins del Loop, ara s'utilitzen els icones del canvas de l'esquerra per a totes les accions.
- Els obstacles s'han substituït per pedres, que són molt més representatives d'obstacles que no pas les gemes i la marieta que hi havia.
- Els nivells de la pantalla de selecció, apart de tenir cadascun la imatge del nivell en qüestió, les he numerat per més simplicitat.
- S'ha canviat la fletxa indicadora entre els dos canvas per una de més senzilla.
- S'han substituït les fletxes de Repetir, Continuar per unes de més senzilles. Conseqüentment he canviat la del menú per una del mateix color que les altres per mantenir la coherència visual.
- S'han reescrit les ajudes en tots els idiomes per a fer-les més entenedores per a l'usuari.
- S'han reestructurat els nivells per tal d'ajustar la corba de dificultat i fer el joc més fàcil.

- S'han canviat els sons de quan arrastres una comanda al canvas i de quan perds. També s'ha canviat el so quan no hi ha més comandes per a esborrar.
- En la segona pantalla, s'ha limitat el Loop a una acció per a que l'usuari reforci més el concepte de les repeticions. A la sisena es limita el número de repeticions a 1 per a reforçar el concepte de funció, el encapsular varies accions en una sola comanda.
- El canvas del Loop s'ha reestructurat per a ser més còmode visualment. Les fletxes per seleccionar el número de repeticions són ara més grans i visibles.

Tots aquests canvis han suposat un rentat de cara per a LogiK, donant-li un aspecte més agradable i una interfície molt més atractiva i fàcil d'utilitzar per a l'usuari.

## 9.2 Captures de pantalla

A continuació, es mostra el que han suposat tots aquests canvis comentats a nivell visual amb captures de pantalla de les diferents escenes.

### Introducció o splash d'inici:



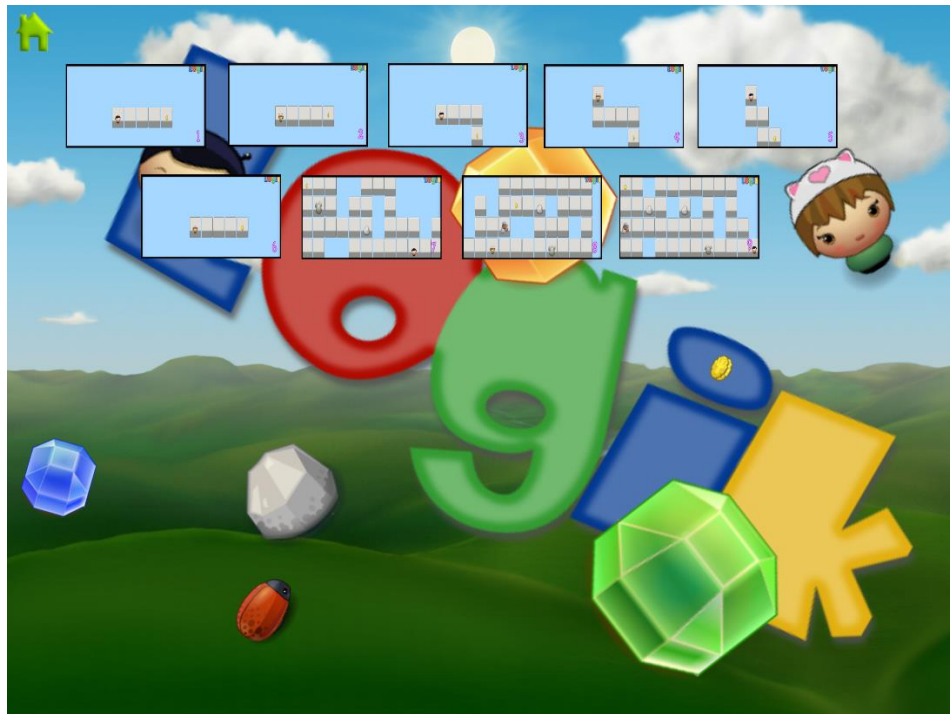
En aquesta escena s'ha inclòs el nom de l'autor i de la supervisora. No s'ha comentat en l'apartat d'ampliació ja que no ha estat quelcom per a millorar el software.

**Menú principal:**



En aquesta pantalla s'ha canviat el fons per un amb un dels obstacles a mode decoratiu. La resta està igual que abans.

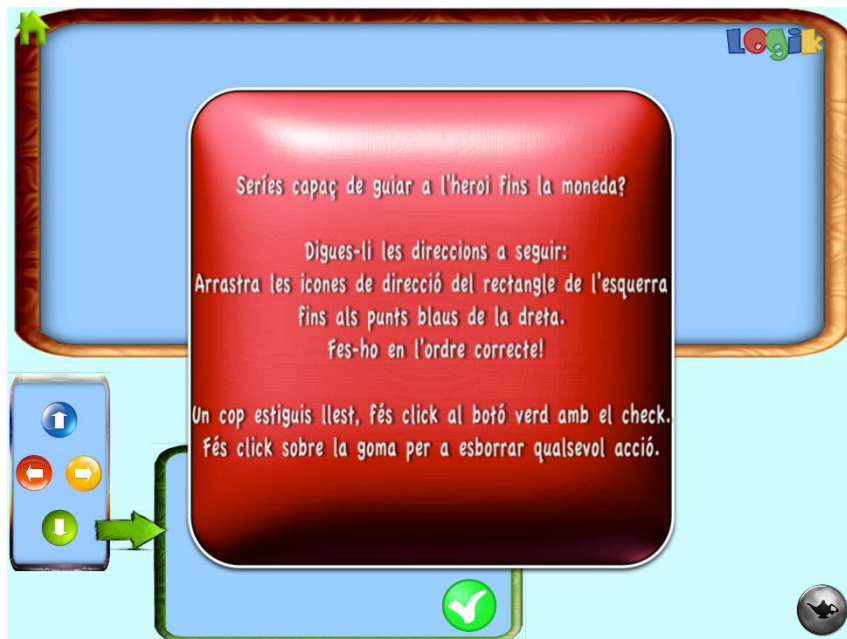
**Selecció de nivell:**



En aquesta pantalla, s'ha afegit un requadre negre, el número de nivell i una captura del mateix per a identificar cada un d'una forma més còmode visualment. També s'ha canviat el fons afegint-hi una pedra dels obstacles per a que es noti que s'ha afegit.

**Ajuda:**

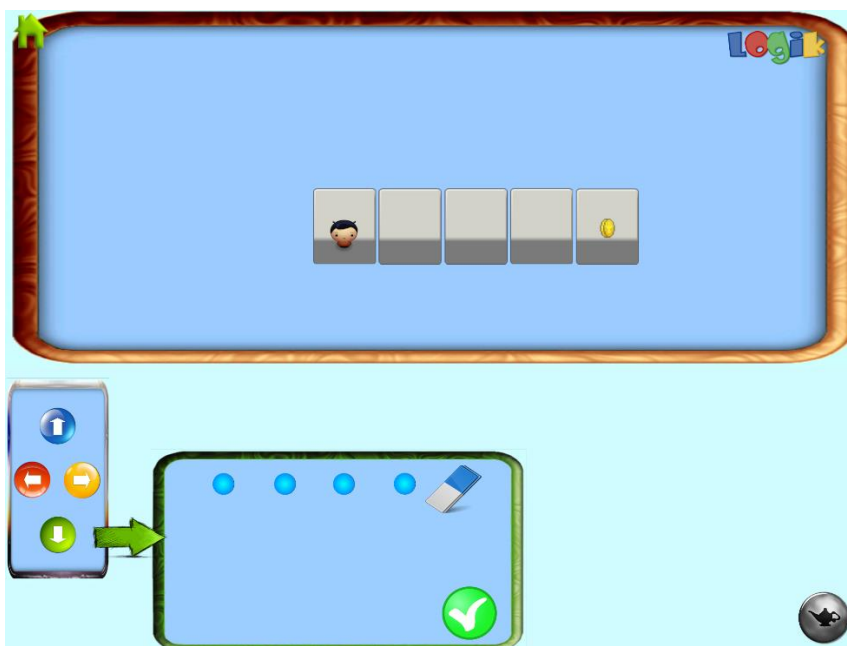
Abans de mostrar els nivells pròpiament, s'exposa una captura de la ajuda del primer nivell per tal de donar una referència visual del canvi realitzat.



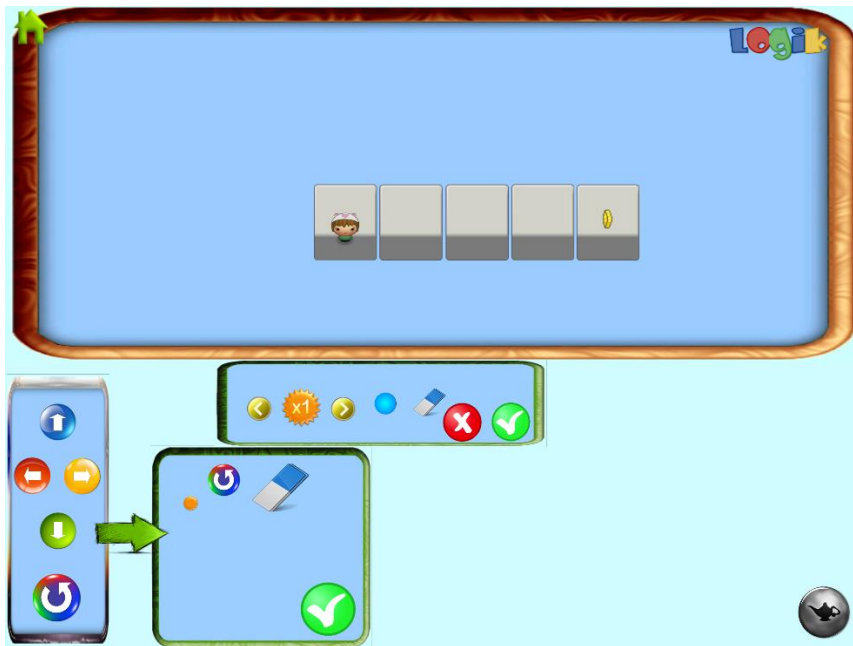
Per treure l'ajuda es pot fer fent clic sobre el text o bé prement sobre la icona de la làmpada de baix a la dreta de la pantalla.

A continuació s'exposaran en ordre les captures dels 9 nivells de que consta Logik:

**Nivell 1:**

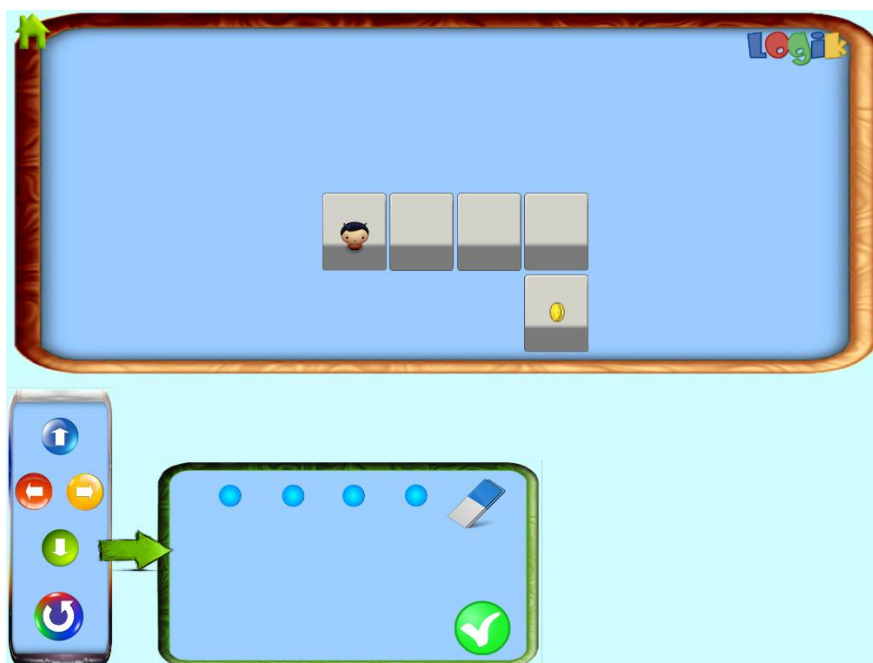


**Nivell 2:**

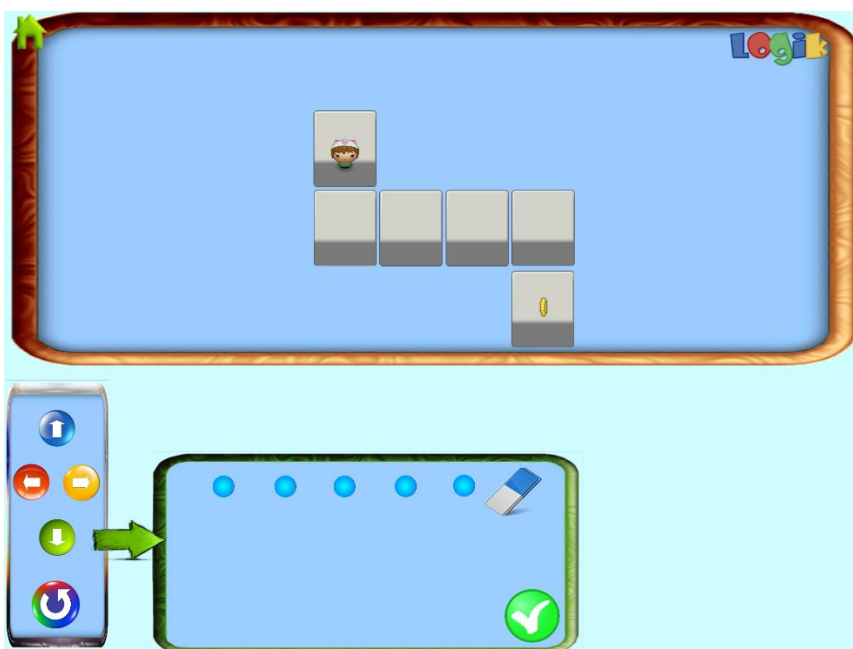


En aquest segon nivell és en el que s'ha reduït a una acció el Loop. També es pot apreciar el canvi d'aspecte sense botons dins i amb la selecció del número de repeticions més a la vista i gran.

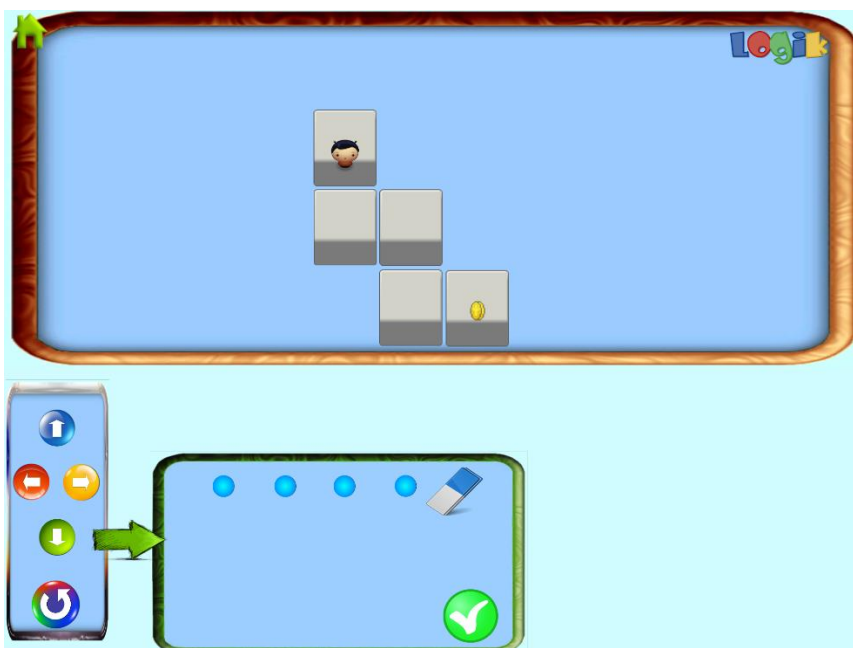
**Nivell 3:**



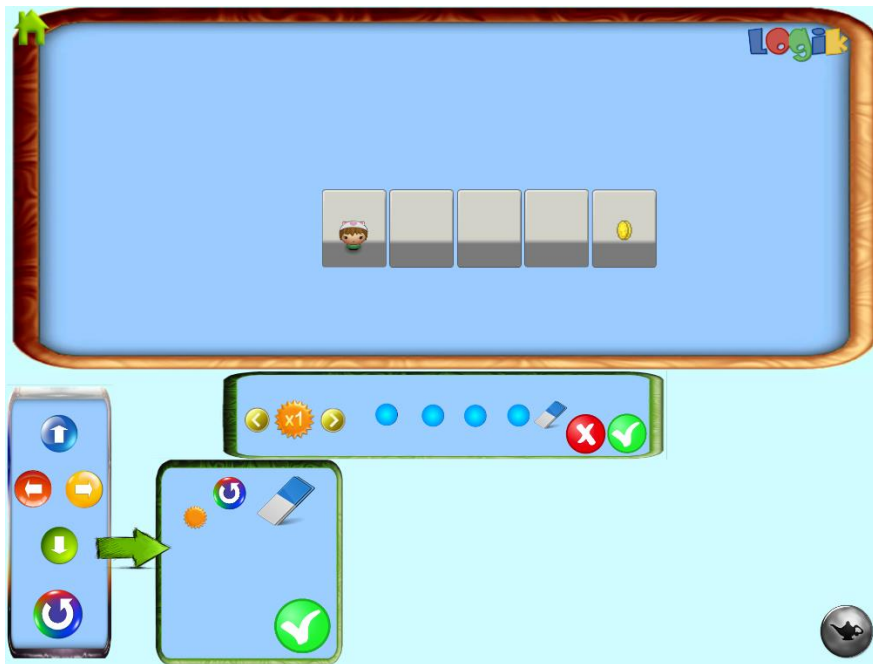
**Nivell 4:**



**Nivell 5:**

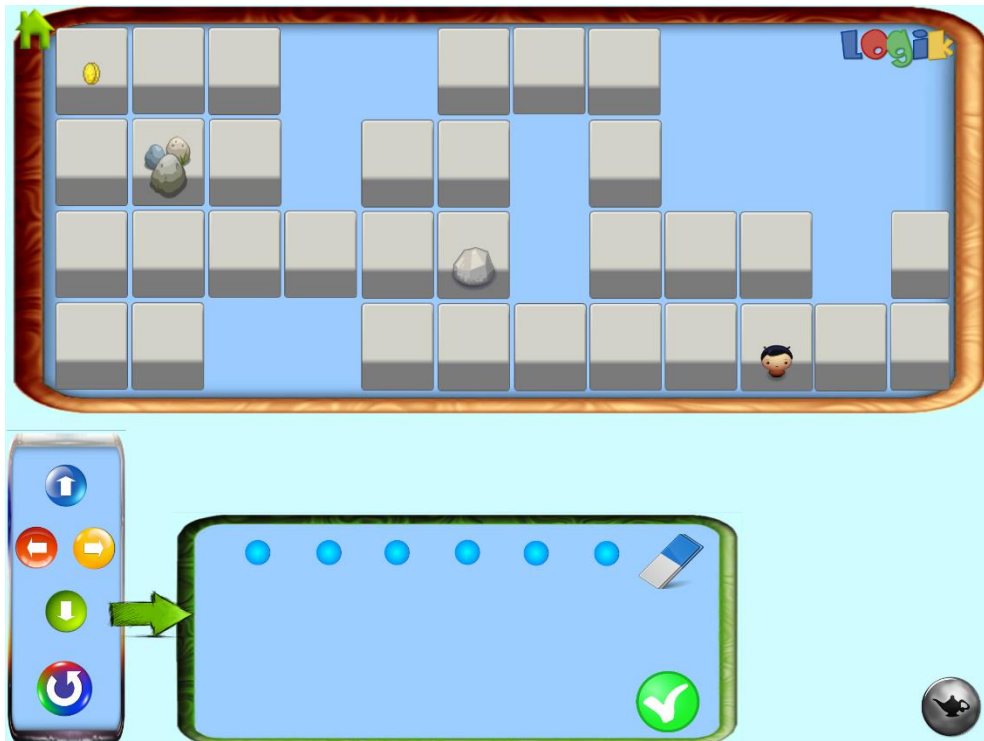


**Nivell 6:**

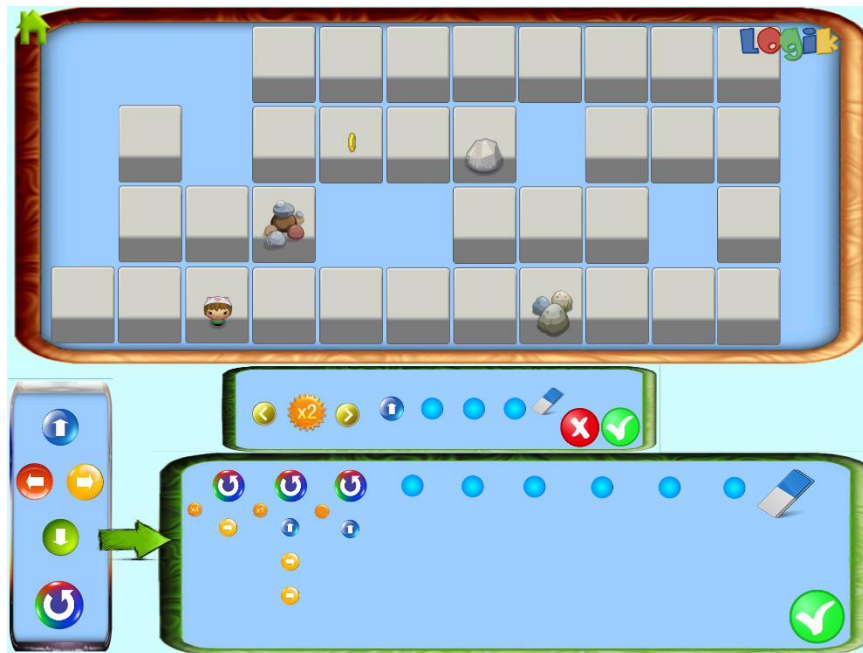


En aquest nivell es va limitar el número de repeticions a 1.

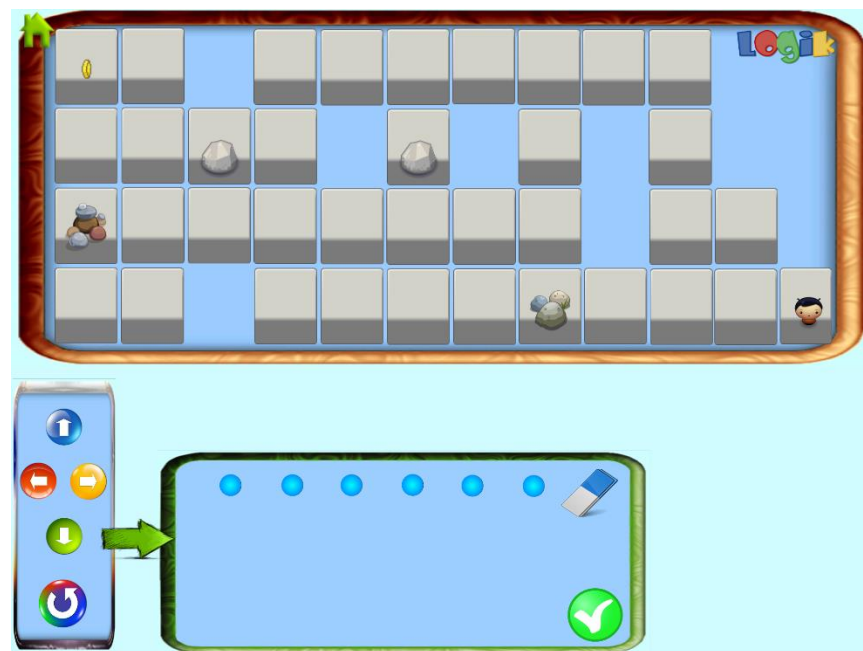
**Nivell 7:**



**Nivell 8:**



**Nivell 9:**



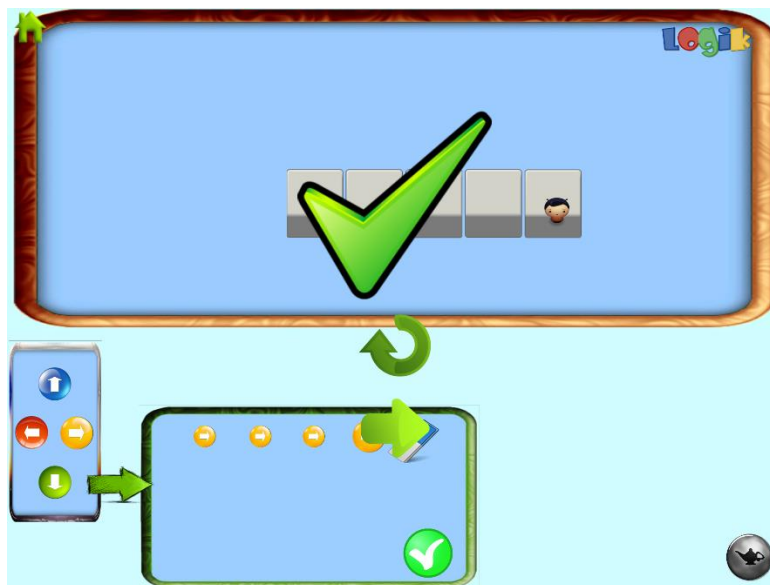
En els últims nivells, concretament a partir del 7, on hi comencen a haver obstacles, s'ha fet que hi hagi més *slots* d'acció de les accions necessàries si es fan servir comandes de Loop, això s'ha fet per a augmentar la dificultat i mirar de forçar als nens a fer servir la comanda de Loop per a encapsular vàries accions i la combinin amb el número de repeticions adequat per a poder encapsular més eficientment les accions.



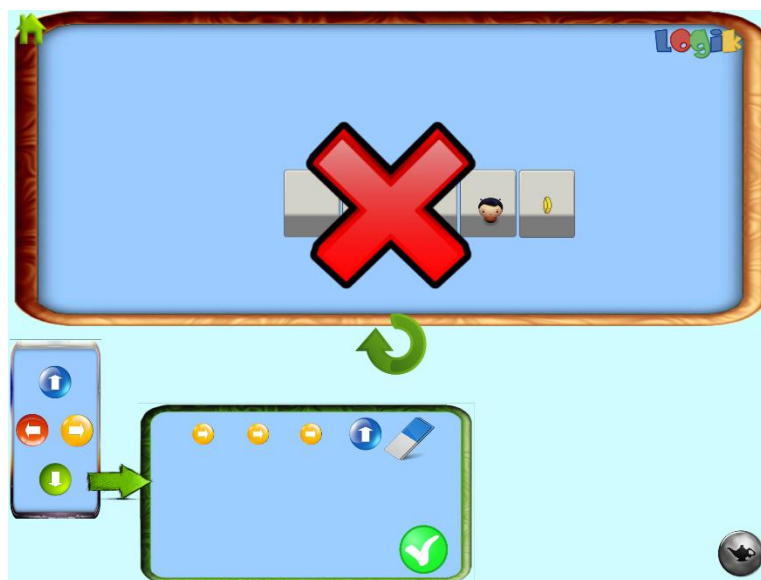
En el segon test s'ha vist que els costa molt combinar els dos conceptes de Loop i que la dificultat és massa elevada per als nens. És molt millor limitar el número d'accions a una solució possible i més senzilla.

Per acabar amb les captures, es mostrarà, per últim, el que surt per pantalla quan es perd o es guanya la partida.

**Guanyar:**



**Perdre:**



En aquestes captures es pot apreciar com, al guanyar o perdre, canvien els icones de Continuar i Repetir.

### 9.3 Test d'usabilitat

Després d'implementar totes les millores mostrades en l'apartat anterior s'ha pogut realitzar un últim test amb dos nens diferents del primer test, una altra nena de 5 anys i un nen de 7.

L'objectiu d'aquest test era comprovar si les millores han estat efectives o no.

També, es volia comprovar si se'n podia extreure més conclusions per proposar una possible ampliació futura.

#### 9.3.1 Resultats test

El feedback obtingut en aquest test ha estat molt positiu, als nens sembla que els agrada força el joc. Es noten molt els canvis a nivell visual i d'interfície a l'hora de jugar.

Tot apunta a que ara per ara LogiK pot ser un bon prototip de joc a desenvolupar a futur.

Tot i això, s'ha volgut tornar a fer un estudi de les parts en que s'ha vist que els nens tenien dificultat per tal de simplificar, si es pot, encara més la usabilitat del joc.

S'ha observat que els nens no acaben d'assimilar el concepte de Loop com a repetició i el Loop com a funció per a encabir comandes extra tots junts. Sembla ser que el primer que proven es repetir la ultima idea provada, en aquest cas el encabir comandes normals en comptes d'utilitzar el concepte de Loop en els nivells amb els camins més llargs fins la moneda, de manera que normalment els falten espais d'acció per omplir i no poden arribar-hi.

A les pantalles grans, a partir de la setena, pels nens és tot un repte el tenir tantes restriccions d'espai i haver d'utilitzar forçosament el Loop. La nena ha estat vora vint minuts per a resoldre-la. El nen no tant, però al acabar el primer nivell llarg ha preferit deixar el joc.

En aquest test es va poder apreciar una característica interessant, ja que va ser útil el fet que s'emmagatzemin les accions introduïdes al perdre com a sistema de *debugging* i fer proves abans no s'arriba a la solució correcte. Això ajuda als nens a veure on s'han equivocat en les pantalles de més dificultat per a ells.

A vegades els nens proven de ficar les comandes de dins del Loop a dins del canvas principal, cosa que ens fa pensar que podríem provar d'intentar reduir aquests casos canviant una mica més la disposició dels canvas.

## 9.4 Treball futur

Amb aquest últim feedback, es mostra una última proposta per a una possible futura ampliació amb idees que podrien minimitzar les dificultats que s'han pogut observar durant el test:

- Fer més nivells utilitzant el nivell prototip en el que s'utilitzi només un dels conceptes de Loop, i després realitzar nivells amb l'altre, per separat. Distingir també els dos conceptes de Loop amb un botó diferent, potser així ajuda als nens a interioritzar més els dos conceptes, en comptes d'anar-los barrejant en un sol tipus de comanda.
- Reestructurar els nivells a partir del 7 i fer-los més curts. Potser podríem utilitzar algun d'aquests nivells com a últim repte del joc, i la resta de nivells fer-los amb una dificultat que segueixi la linealitat dels nivells del 1 al 5.
- Es podria mirar de redimensionar el canvas principal. L'objectiu seria d'encabir-hi el del Loop a dins i fer que les comandes normals i de Loop s'activin totes amb el canvas principal fent ús d'una variable de control. Això potser eliminaria el dubte a l'hora d'arrastrar les comandes a un canvas o altre.
- Per últim, es podria afegir més tipus de comandes, com ara variables, o una comanda que executi un comportament de tipus *if-then*.

## 10. Conclusions

Al acabar la implementació del software, s'han pogut complir els objectius exposats. El feedback obtingut pels nens ha acabat sent força positiu, fent d'aquest prototip un possible projecte a tenir en compte com a futura aplicació.

Tot i les dificultats trobades s'ha pogut desenvolupar una aplicació 100% funcional amb totes les necessitats cobertes, així doncs, penso que el projecte ha estat un èxit.

Crec que aquesta aplicació amb un disseny gràfic genuí, podria lluir molt diferent i podria ser publicat amb més possibilitats per a qualsevol plataforma mòbil.

En quant a l'estudi dels motors, cal dir que per al tipus d'aplicació que volem desenvolupar, aquest ha estat viable. Tot i l'excés d'ús de recursos, per la grandària que té ara l'aplicació no ha suposat un problema.

En cas de futura implementació amb Assets i més ampli, sí que recomanaria fer-ho amb un altre llenguatge com ara XCode, per tal d'assegurar una estructura interna del software estable, que permeti aplicar futures ampliacions i millores d'una forma més estructurada, amb més possibilitats i aprofitant al màxim els recursos de la plataforma per a la que es vulgui desenvolupar.

La realització d'aquest projecte m'ha servit per comprovar la utilitat que tenen els motors de creació d'aplicacions, tant per a la creació d'aplicacions senzilles com de prototips funcionals d'aplicacions més complexes.

Aquests prototips, per a un equip de desenvolupament, suposen poder començar a desenvolupar una aplicació sobre unes bases testejades prèviament, amb la seguretat de saber que serà un software útil i amb característiques fiables.

## 11. Hores invertides

A continuació es comptabilitzen les hores invertides en el projecte :

### 11.1 Investigació

Hores dedicades a la investigació sobre el projecte : 48.

Aquest procés d'investigació inclou:

- Anàlisi de les necessitats.
- Elaboració de la idea sobre què anirà el joc i com serà.
- Elaboració teòrica dels objectius i característiques principals del software.
- Investigació sobre eines a utilitzar per al desenvolupament.
- Cerca de diferents motors gràfics que puguin cobrir les necessitats del projecte.
- Estudi i comparació dels diferents motors per facilitar la futura decisió sobre quin fer servir.
- Formació en el motor escollit

### 11.2 Desenvolupament Projecte

Hores dedicades al disseny i desenvolupament del software: 290.

Aquestes inclouen :

- Cerca d'assets que satisfacin les necessitats del software.
- Disseny dels diferents blocs funcionals del software.
- Implementació de la part lògica/programació de tot el projecte.
- Modificació/adequació del codi per a la exportació a HTML5.
- Exportació del software a les diferents plataformes (HTML5 i iOS).
- Testeig i correcció d'errors.
- Implementació de les millores proposades a partir del feedback obtingut dels tests d'usabilitat.

### 11.3 Redacció memòria

Hores dedicades a la part de redacció de la memòria del projecte: 69.

Inclou :

- Documentació sobre els diferents motors de creació de videojocs.
- Documentació dels diagrames de flux del projecte.
- Redacció del desenvolupament del projecte i de la resta de la memòria.

## 12. Referències

### **Codea :**

<http://twolivesleft.com/Codea/>

<http://www.lua.org>

### **Mobincube :**

<http://www.mobincube.com/>

### **Gamesalad :**

<http://gamesalad.com/>

### **Construct 2 :**

<https://www.scirra.com/construct2>

[http://www.xmoggo.com/construct2/manual/manual\\_2.html#anchor-32](http://www.xmoggo.com/construct2/manual/manual_2.html#anchor-32)

### **Webs Descarrega Free Assets :**

<http://gsanimator.com>

<http://opengameart.org>

[www.freesounds.com](http://www.freesounds.com)

<http://www.soundbible.com>

### **Fonts :**

<http://www.fontspace.com/category/kids>

### **Heurístiques :**

<http://www.nngroup.com/articles/ten-usability-heuristics/>

### **Ortografia :**

<http://www.uoc.edu/serveilinguistic/criteris/ortografia/acentuacio.html>

<http://www.uoc.edu/serveilinguistic/criteris/ortografia/apostrof.html>

## 13. Annexos

Els annexos es troben en format digital juntament amb aquest document i consten de:

- Manual d'usuari de LogiK en format pdf
- Les diferents versions de Logic en Format Gamesalad. La última versió és la v0.9, i d'aquesta última s'adjunta també una versió en format Mac per a poder testejar-la en cas de no disposar d'una llicència d'Apple Developer.
- Vídeos dels tests d'usabilitat. N'hi ha sis.