



# Happy Hour

CFGS Desenvolupament d'Aplicacions Multiplataforma

**Adrià González García**  
**Andrés Sebastián Quiceno Hernandez**  
**Victor Hermel Añezco Cadena**

**DAM2A**

31/5/2019

Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## **GNU Free Documentation License (GNU FDL)**

**Copyright © Adrià González García, Andrés Sebastián Quiceno Hernandez, Victor Hermel Añazco Cadena.**

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

## **Resum del projecte :**

*Happy Hour* és un joc interactiu per els nens petits que tinguin entre 2-5 anys d'edat, està dissenyat per Android i programat en Java. *Happy Hour* està pensat per a que els nens aprenguin mentre es diverteixen.

Té un total de sis jocs diferents per a que el nen tingui una experiència completa d'aprenentatge, cada joc serveix per desenvolupar diferents funcions cognitives del nen: matemàtica, lògica, lingüística, naturalista i visual-espacial.

## **Abstract:**

*Happy Hour* is an interactive game for young children between 2-5 years of age, designed for Android and in java code. *Happy Hour* is designed for children to learn while having fun.

It has a total of six different games so that the child has a complete learning experience, each game is used to develop the different cognitive functions of the child.

## **Paraules clau :**

Interactiu  
Joc  
Divertit  
Educatiu  
Scrum  
Github  
IDE

# Índex

## 1. Introducció

1.1 Context i justificació del Treball

1.2 Objectius del Treball

1.2.1 CARACTERÍSTIQUES DEL PROJECTE

1.2.2 OBJECTIUS DEL PROJECTE

1.2.2.1 OBJECTIUS PRINCIPALS DEL PROJECTE

1.2.2.2 OBJECTIUS SECUNDARIS DEL PROJECTE

1.3 Enfocament i mètode seguit

1.3.1 Enfocament i mètode implementat

1.3.2 Tecnologies Rebutjades

1.4 Planificació del projecte

1.4.1 Requirements Generals

1.5 Productes obtinguts

1.6 Descripció dels altres capítols de la memòria

## 2. Anàlisi i disseny

2.1 Requirements

2.1.1 Anàlisi i disseny

2.3 Mockup

## 3. Implementació

3.1 Estructura Del Projecte

3.2 Interfaz De Usuario

3.3 Interaccions

## 4. Conclusions

4.1 Futures Implementacions

## 5. Glossari

## 6. Bibliografia

## 7. Annexos

7.1. Guia rapida de l'aplicació

7.2. Codi font

# 1. Introducció

## 1.1 Context i justificació del Treball

És un joc fet per a que els nens aprenguin mentre es diverteixen.

Molts pares estan preocupats de que els seus fills juguin a jocs que no són adequats per ells, que es gastin diners en un entorn que no coneixen i moltes empreses intenten guanyar diners a costa d'això.

*Happy Hour* té la solució per aquells pares preocupats per aquests problemes.

El resultat que volem assolir és que els nens es diverteixin mentre aprenen i els pares estiguin tranquils.

## 1.2 Objectius del Treball

### 1.2.1 CARACTERÍSTIQUES DEL PROJECTE

Las característiques del projecte són les següents:

- Aplicació per mòbils amb Android
- Codi Java
- Orientada a l'aprenentatge i el desenvolupament creatiu.
- Diferents jocs cadascun amb una manera única de jugar.
- Múltiples interaccions.
- So i efectes animats.
- Desenvolupada amb Android Studio Project

## 1.2.2 OBJECTIUS DEL PROJECTE

L'objectiu principal del projecte és fer una aplicació per android orientada a nens i nenes petits amb la finalitat d'entretenir i que alhora puguin aprendre sense la necessitat de que els pares hagin d'estar pendents de que el seu infant estigui envoltat en un entorn competitiu i ple de publicitat, tot pare i mare vol poder deixar el mòbil sense que s'hagin de preocupar per els problemes anomenats anteriorment i que alhora els nens es puguin divertir mentre aprenen.

Els objectius educatius de l'aplicació son els següents catalogats per joc:

- **Figures geomètriques:** Es vol que els nens i nenes que hi juguin comencin a conèixer les figures geomètriques com el triangle, el cercle, el quadrat o el pentàgon en un divertit *DragNDrop*.
- **Números:** Es vol que amb sumes i restes les nenes i nens puguin aprendre els cinc primers números basics on n'hi han animacions i sons.
- **Abecedari:** Es vol que els nens i nenes aprenguin les vocals, per incentivar això, es fa amb animals on si es dóna clic a l'imatge, s'escolta el so que fa cada animal.
- **Pintar:** Es vol que les nenes i nens desenvolupin la seva creativitat amb unes plantilles predefinides on podran pintar entre altres, un gat, un mussol, etc. O si prefereixen un mode on n'hi ha una imatge en blanc per poder fer el que vulguin.
- **Puzzle:** Es vol que els nens i nenes puguin aprendre a obtenir un millor control d'una pantalla táctil.
- **Parelles:** Es vol que les nenes i nens millorin la seva capacitat de memoritzar amb imatges divertides, que tenen animacions i sons.

### 1.2.2.1 OBJECTIUS PRINCIPALS DEL PROJECTE

Els objectius principals del projecte, són els següents:

- Aprendre a fer servir Android Studio.
- Ús i implementació de llibreries externes.
- Saber més a fons que pot oferir les diferents eines que dóna Android Studio.
- Aprofundir més en el llenguatge Java.
- Aprendre a dissenyar i implementar una aplicació, que funcioni en qualsevol Android .
- Implementació de base de dades.
- Analitzar dades com el temps utilitzat en un joc.
- Aprendre a fer servir Github en un entorn de treball “real” i que sigui en un grup (Així es força l’ús de “merge”, “rebase”, etc...).
- Planificació en equip fent servir les metodologies vistes en classe.

### 1.2.2.2 OBJECTIUS SECUNDARIS DEL PROJECTE

- Investigar el funcionament de Google Play.
- Fer servir un controlador de so per Android.
- Implementar correctament l’ús de les regles de presentació.
- Correcta estructura de un projecte (Classes, Package....).
- Veure altres models de bases de dades.
- Comparar diferents tecnologies.

## 1.3 Enfocament i mètode seguit

### 1.3.1 Enfocament i mètode implementat

En aquest projecte s'ha decidit fer-ho amb *Android Studio* en llenguatge *Java* i per a mòbils amb android.



S'ha decidit fer servir l'*IDE Android Studio* per que és l'entorn de desenvolupament que més facilita el disseny del producte i també alhora de gestionar el treball que ha sigut molta ajuda amb la seva implementació de control de versions com *GitHub*.



[Java](#) és el llenguatge de programació que s'ha utilitzat per què és el que s'ha après en classe i *Android* funciona principalment en aquest llenguatge on també es tenen les alternatives com "*Kotlin*".





S'ha desenvolupat el nostre producte en *Android* per què voliem fer una aplicació per nens petits per mòbil i android es un sistema operatiu que utilitzen la majoria de mòbils del mercat i molt més accessible a un públic generic.



El control de versions per al projecte que s'ha agafat és fer servir els sistema de repositoris de [Github.com](https://github.com) per treballar en equip. D'aquesta manera s'aprèn a treballar en equip en un mateix projecte i es pot tenir un registre de els canvis que s'han anat fent i quan de temps es triga per cada commit.

### 1.3.2 Tecnologies Rebutjades

## LibGDX



LibGDX es un *framework* per al desenvolupament de videojocs multiplataforma, que soporta actualment Windows, Linux, Mac OS, Android, IOS i HTML5.

No s'ha utilitzat aquest framework perquè per fer el disseny gràfic de l'aplicació estava molt limitat i es precisa hores d'investigació prèvies necessàries en comparació a Android Studio.

L'avantatge que té es que alhora de fer animacions i fer la planificació de un joc és molt superior a Android Studio, però ja que el joc està orientat a nens petits el joc que es fa no té la complexitat necessària per fer servir aquest framework.

S'ha orientat el producte a un àmbit molt més "artístic".

# IntelliJ



IntelliJ és un IDE (Entorn integrat de Desenvolupament) per al desenvolupament de programes informàtics. És desenvolupat per JetBrains (anteriorment conegut com IntelliJ), i està disponible en dos edicions: edició per a la comunitat i edició comercial.

Es va prendre la decisió de no utilitzar aquest IDE perquè el que més s'adaptava a les necessitats del projecte era Android Studio, ja que és un IDE derivat de IntelliJ fet per fer aplicacions per Android.

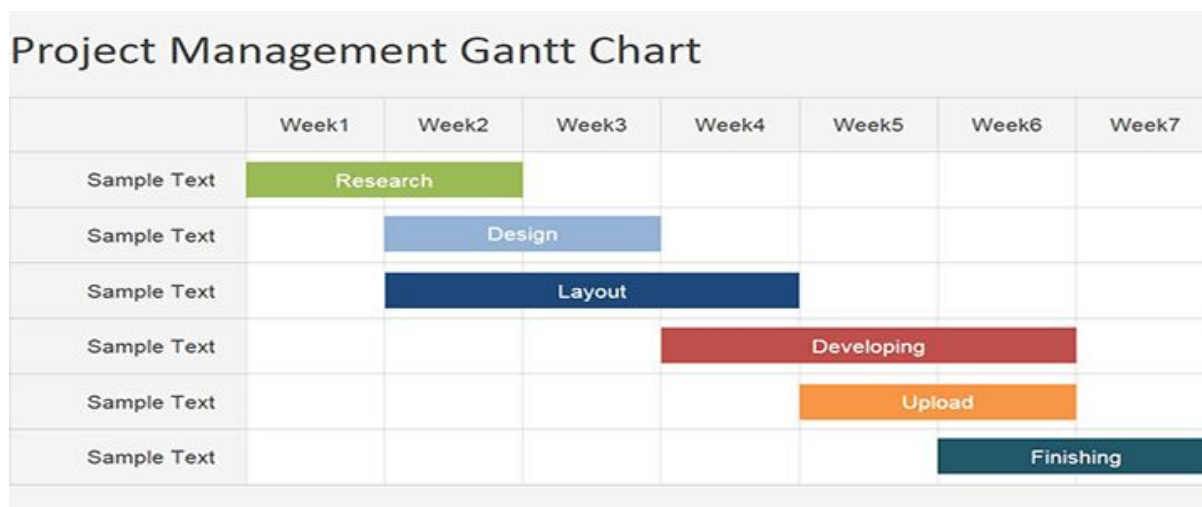
## 1.4 Planificació del projecte

A l'hora de gestionar un projecte existeixen diverses metodologies com les tradicionals Waterfall, o RUP, o més darrerament les metodologies àgils com SCRUM o Kanban.

### Metodologia tradicionals(Rebutjada):

Les metodologies com Waterfall organitzen les tasques de desenvolupament en diverses fases consecutives.

El diagrama de Gantt és una eina útil per poder planificar aquest tipus projectes. Proporciona una visió general de les tasques programades, totes les parts implicades han de saber quines tasques s'han de completar i en quina data.



Aquesta metodologia de treball es va descartar ja que avui en dia, les empreses estan més enfocades a metodologies àgils i que modular que permetin tenir un control en tots els àmbits del projecte ja que moltes vegades és necessari fer canvis d'última hora o alguna característica no es pot implementar.

Per això s'ha optat per la implementació de metodologia anomenada SCRUM ja que permet fer tot el que s'ha anomenat abans, i en aquest

projecte s'ha vist necessari aquesta metodologia ja que mai es pot sapiguer quan tens problemes amb alguna característica si pot arribar a la data prevista, amb *SCRUM* es pot retrasar i controlar millor aquests problemes.

### *SCRUM:*

Per a la planificació, s'ha utilitzat un mètode anomenat *SCRUM* com s'ha comentat anteriorment, per utilitzar aquest mètode, s'ha fet amb una eina online anomenada "*Trello*" que permet la creació de taules que es fa servir per dividir el projecte en diferents parts (*BackLog, current scrum, Ready, Tested, Implemented...*).

Cada *sprint* en el projecte dura una setmana, un *sprint* està compost de cinc etapes:

**1. Reunió:**

En aquesta reunió es defineixen els objectius de cada *sprint*

**2. Scrum diari:**

Petita reunió que es realitza durant el dia per veure com van va el desenvolupament.

**3. Treball durant el *sprint*:**

Desenvolupar el projecte amb el que es va pactar el dia de la reunió.

**4. Revisió del *sprint*:**

Al final de cada *sprint* es realitza una inspecció dels objectius, tant assolits com els que no.

**5. Retrospectiva del *sprint*:**

Reunió on el principal objectiu és millorar tant com a equip com cadascun dels membres, identificar els problemes que han sorgit, com s'han solucionat i com es pot fer que no tornin en un futur.

El projecte s'ha repartit de forma equitativa i tothom involucrat ha estat d'acord d'acceptar.

La part de disseny, s'ha fet entre tots els membres de l'equip. Al fer diferents jocs, es faran dos jocs per membre, cadascun dels jocs ha d'intentar de implementar una interacció diferent i que es puguin diferenciar correctament.

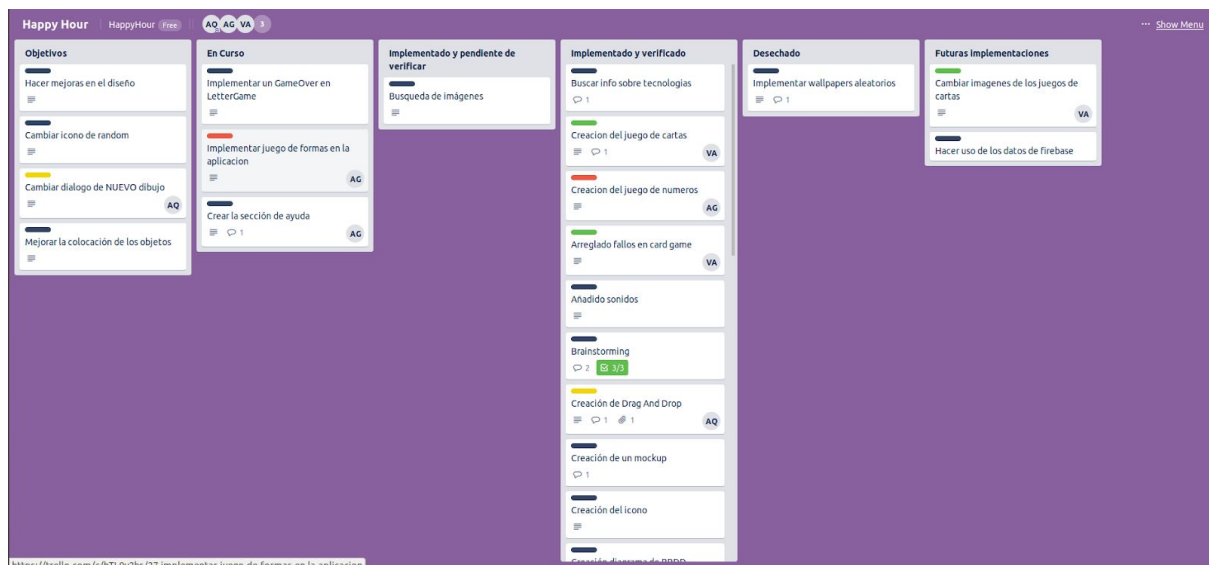
Un cop finalitzat els jocs, implementar una base de dades en *firebase* per poder guardar dades importants com sapiguer quin son els jocs on es gasta més temps, quin joc és el més jugat, etc.

## Trello

Aquí n'hi ha una petita imatge de l'eina *Trello* on es pot veure com s'ha anat estructurant el projecte i com s'ha repartit el treball.

Per no complicar molt a l'equip, es va acordar de que cada setmana, quan alguna característica no estigués preparada, es quedaria on està per continuar treballant en ella pel següent sprint.

Es proporciona un enllaç en anexos necessari per poder entrar a visualitzar el trello del equip.



### 1.4.1 Requirements Generals

Per fer aquest projecte son necessaris tenir coneixements de android , programació orientada a objectes i Java.

També fa falta tenir a disposició un telèfon mòbil que et permeteixi fer instalacion directes desde Android Studio.

El SDK que s'utilitza al projecte és el 28 que equivale al Android 9 què és la versió més nova de Android.

<https://developer.android.com/studio/releases/platforms>

## 1.5 Productes obtinguts

**HappyHour** és el resultat del producte. Es una aplicació per Android (pròximament en IOS) programada amb codi Java conté sis jocs que es diferencien entre ells per a nens i nenes petits amb un seguiment de el temps i quin joc han jugat per informació rellevant pel equip i pels pares. El temps es gestiona en una base de dades feta amb [firebase](#).

## 1.6 Descripció dels altres capítols de la memòria

En els següents capítols s'explica el procés de desenvolupament i funcionament de l'aplicació. L'estructura i els diferents diagramas de disseny de cada part de l'aplicació.

Ens les **conclusions** es raonan els resultats final i l'aprenentatge de cada tecnologia implementada o utilitzada i també les futures implementacions que es poder fer.

El **glossari** recull una sèrie de definicions de diferents expressions i paraules que són una mica difícils d'entendre a primera vista, junt amb el seu significat o algún comentari.

En els **annexos** es recullen una serie d'elements necessaris per l'explicació d'alguns temes relacionats amb el producte. Com per

exemple: el manual d'usuari,algún script necessari o alguna tecnologia utilitzada.

La bibliografia recull diferents enllaços d'interès implementats a el producte final o utilitzades en el procés.



## 2. Anàlisi i disseny

### 2.1 Requirements

Quan es dona click a la pantalla de login, ha de sortir un menú de les diferents comptes de google que te registrades el telèfon mòbil on s'executa l'aplicació. Així es fa login a firebase per poder fer servir la base de dades i registrar el temps de joc.

El menu principal ha de tenir uns botons que ens permetin escollir el joc al que volem jugar o un botó random que surti qualsevol joc aleatori

- **Informació (Futura implementació):**

Aquesta opció mostra informació general sobre el joc i serveix com una opció de (help) per els diferents jocs. No li hem donat prioritat ja que es una aplicació per nens i no li donen mai aquesta opció només el pares.

- **Jocs:**

Una pantalla que tindrà sis imatges representant cada joc.

#### Diferents jocs:

- **Joc de cartes (CardGame):** Es un joc per exercitar la memòria; es mostren parelles de cartes que es donen la volta i tens que trobar la save parella.
- **Joc de Lletres (LetterGame):** Es un joc que mostra un animal/objecte amb les diferents lletres que conté i es motra un espai en blanc, abaix mostra les vocals com opcions per ha que els nens omplin l'espai en blanc.
- **Puzzles (PuzzleGame):** Aquest joc mostra una serie de imatges per escollir, quan el nen tria una imatge es forma un puzzle amb la mateixa.

- **Pintar (PaintGame):** Es un joc que mostra una serie de imatges per escollir, quan el nen tria una imatge et deixa pintar-la.
  - **Figures (ShapeGame):** Aquest joc mostra una serie de figures amb figures més petites amb cares i es té que arrosegar les cares a les figures més grans
  - **Números (NumberGame):** Es un joc que mostra una suma, resta i falta un número per resoldre l'operació matemàtica, el nen tindrà que triar un número de les opcions que surten abaix i resoldre l'operació.
- 
- **Joc Aleatori:**

Aquesta opció generarà un joc aleatori dels sis que n'hi han.
  - **Consultar temps de joc (PlayTime):**

Aquesta opció permet consultar el temps de joc de cada joc que hi ha jugat el nen/a.

### 2.1.1 Anàlisi i disseny

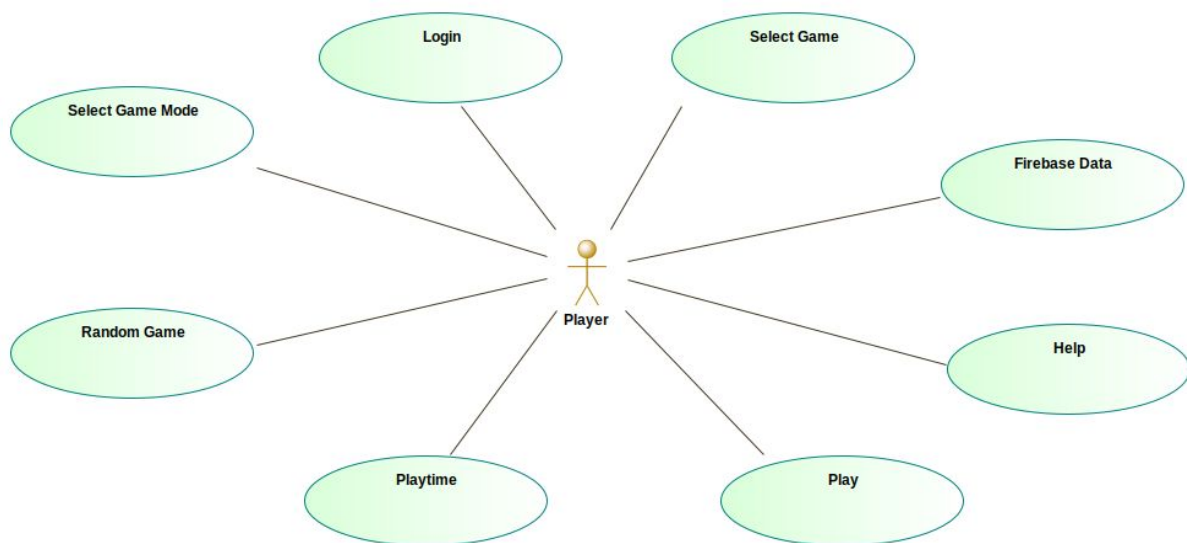
Per a la creació del projecte, no val només amb un mockup, és necessari la creació de diferents diagrames creats a partir de diferents anàlisis que han de permetre explicar amb un cop d'ull com pot el jugador interactuar amb l'aplicació i de com està formada l'aplicació.

Per resoldre això, es van crear un parell de diagrames amb la finalitat anomenada anteriorment.

El primer es un anàlisi de diferents casos d'ús:

- Ja que ens proporciona un o més escenaris que indiquen com hauria d'interaccionar l'usuari amb el sistema.

Fet l'anàlisi, és crea un diagrama a partir de l'informació obtenida.

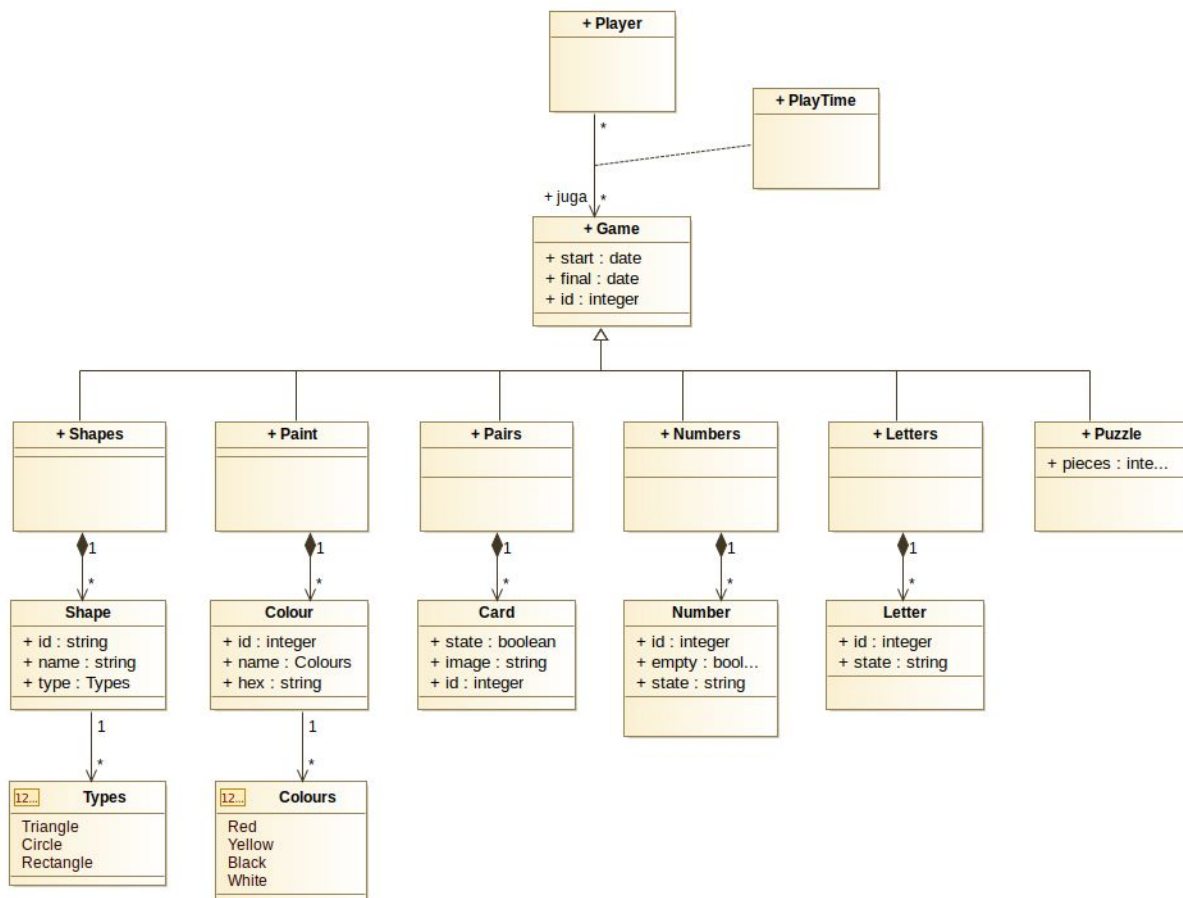


El segon és un anàlisi de classes:

- Els diagrames de classes són el pilar fonamental del modelatge amb UML, sent àmpliament utilitzats tant per anàlisi com per a disseny de sistemes i programari en general.

Ens permetrà representar gràficament i de manera estàtica l'estructura general de l'aplicació, mostrant cadascuna de les classes i les seves interaccions (com herències, associacions, etc.), representades en forma de blocs, els quals són unitats mitjançant línies i arcs.

Fet l'anàlisi, és crea un diagrama a partir de l'informació obtenida.



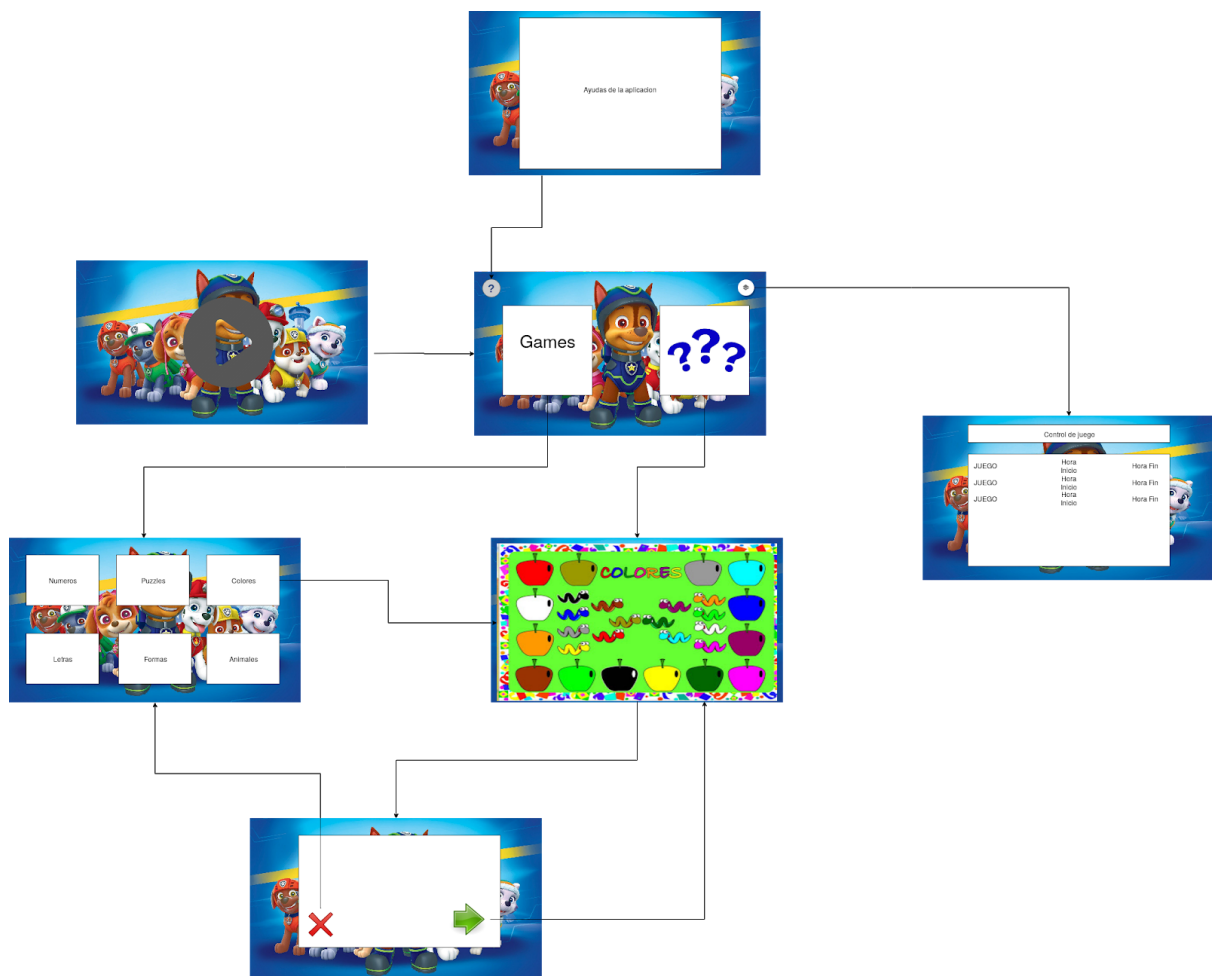
## 2.3 Mockup

Per començar a realitzar un projecte, la primera part d'un nou desenvolupament es la realització de un *brainstorming* (pluja d'idees), amb aquestes idees, es recopilen i s'utilitzen per fer un *sketch* principal de com hauria de ser l'aplicació.

Per això es crea un anomenat *mockup* que és molt útil ja que permet desenvolupar un disseny inicial molt fàcilment i si s'han de fer retocs, es poden fer en l'acte sense malgastar temps de desenvolupament.

N'hi ha una frase molt bona que enfoca el que representa fer un mockup:

*Vostè pot fer canvis ara en el dibuix amb una goma d'esborrar o més tard en l'obra amb un martell.*





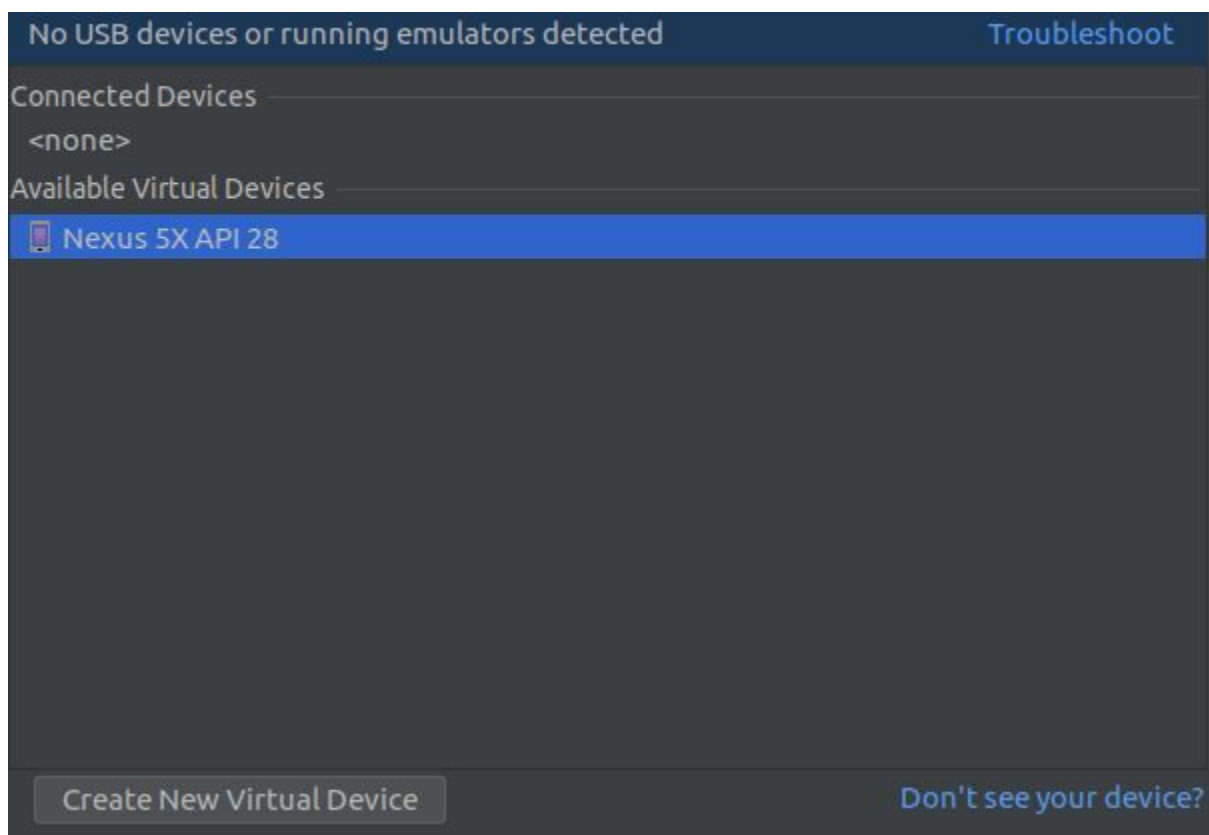
## 3. Implementació

### 3.1 Estructura Del Projecte

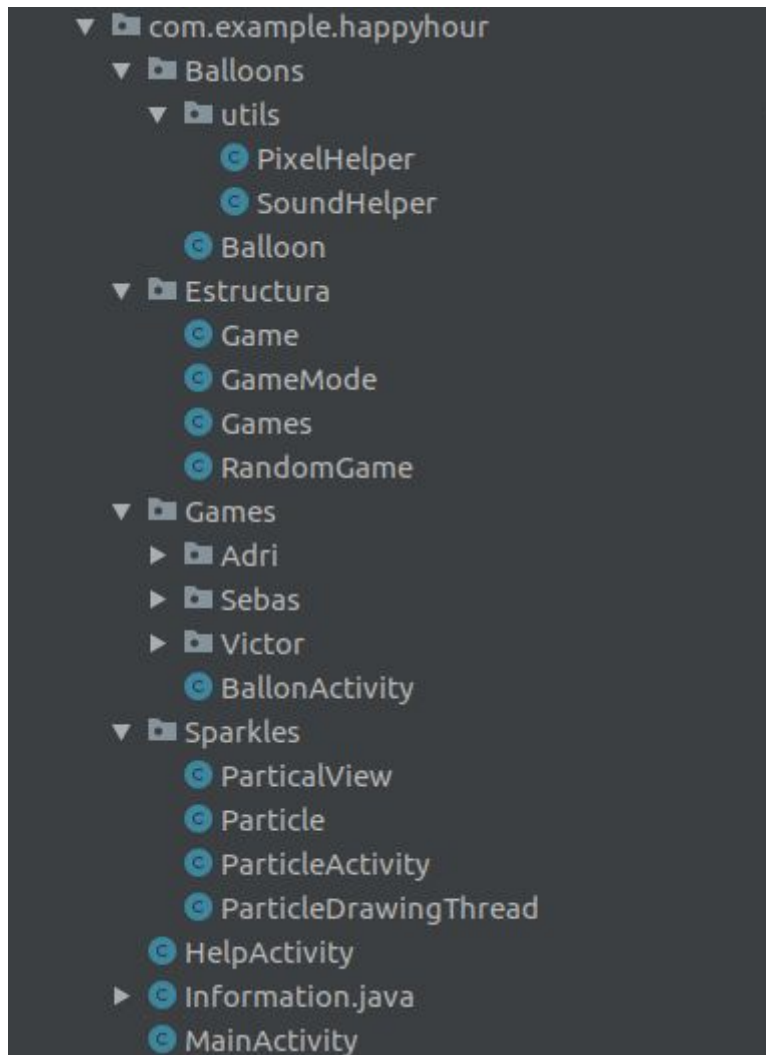
És necessari un IDE per al desenvolupament de l'aplicació, un exemple de IDE molt recomanat es *Android Studio*.

També és necessari que la versió SDK d'android sigui compatible amb la versió 28 que és equivalent a la versió 9 d'Android.

Un telèfon mòbil per provar l'aplicació o un emulador de telèfon mòbil (Android Studio per defecte et dona una opció per instal·lar un emulador).



L'estructura del projecte és la següent:



**Package Balloons:** Conté les classes necessàries per el funcionament de l'activitat final de cada joc que es mostra els globus que es poden clicar i fan soroll.

**Package Estructura:** Conté la estructura principal de l'aplicació les principals activitats. Com per exemple el menú principal i els diferents modes de joc.

**Package Games:** Conté el codi dels diferents jocs implementats, també es separen per els diferents programadors que han fet el joc.



En aquest cas els tres programadors són: Adrià, Sebastià i Victor.

- Els jocs d'en Adrià es troben els jocs de les figures (**shapes**) i el joc de números (**numbers**).
- Sebas ha fet els jocs de pintar (**paint**) i el de puzzles.
- Victor ha fet els jocs de parelles de cartes (**card**) i el joc de les lletres(**letter**).

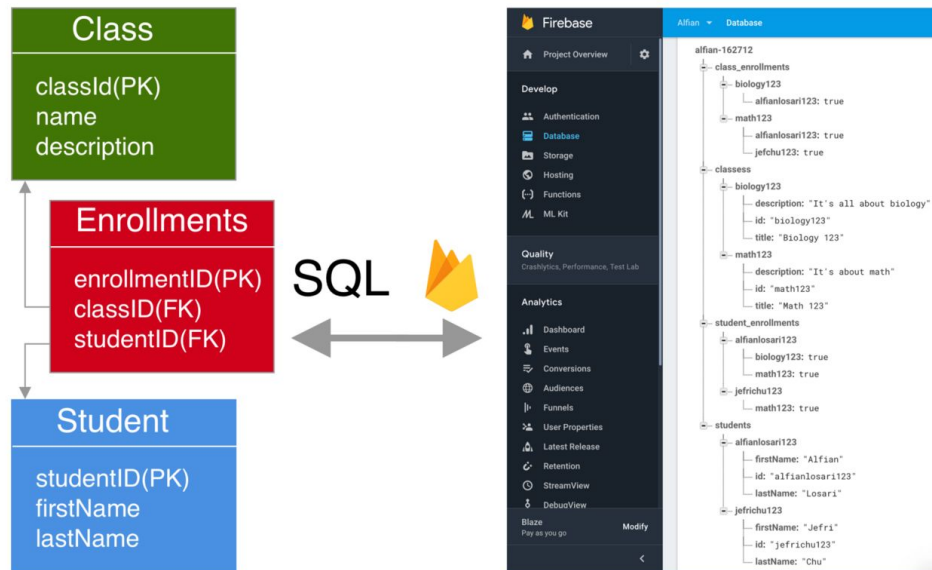
A més, n'hi ha un "joc" final que es l'activity Ballon que es fa servir com a premi en la majoria de jocs implementants.

**Package sparkles (futura implementació/tecnologia rebutjada):** Es una llibreria externa serveix per fer una animació cada vegada que es premi la pantalla o es llisqui.



- **Help Activity (futura implementació / tecnología rebutjada):** Una activity que mostra una ajuda sobre l'aplicació i el seu funcionament. Ha sigut rebutjada o té una baixa prioritat per que al ser una aplicació orientada a nens petits no li donen us algú.

- **Information (Firebase):** Conté tot lo relacionat amb la connexió de la base de dades feta amb firebase i la seva implementació la consulta i com es mostra.



Aquest diagrama mostra el funcionament general de Firebase relacionat amb database. Ja que firebase permet fer ús de moltes altres utilitats.

El model de classes objecte que tenim es **game** l'utilitzem per l'enviament i la consulta de les dades que farem servir per la funcionalitat de **Playtime**.

Després tenim la classe **information** que té un "**adapter**" que junt amb un **recyclerview** fem la consulta i la mostrem.



## 3.2 Interfaz De Usuario

A la primera activity trobem un botó que fa un login amb [firebase](#) (que fent el login permet desar les dades de entrada i sortida de quan entrem als diferents jocs i les guarda a firebase) i ens passa a la següent activity que es la de game\_mode(a totes les activity hi ha un include amb el fons de pantalla ):



A l'activity de game\_mode que es el menú principal de l'aplicació es troben diferents botons que ens permeten navegar per l'aplicació, el botó de la dreta amb la rosca ens envia a l'activity que es connecta amb el firebase (activity\_information) ,el botó del dau ens envia a un joc aleatori de la llista (és una classe amb un random en el que a cada joc se li assigna un número i amb el random depenent del numero que surti ens envia a un joc o un altre ), i el botó de l'osset a la capsa es el menu de seleccio de joc manualment, que al fer clic anem a activity\_games.



activity\_information es la activity que es connecta amb el firebase i treu informació sobre quins son els jocs als que s'accedeixen mes i a quines hores s'ha jugat.

Juego	Dia y Hora	Finalizado
Pintar	29 19:38	29 19:38
Puzzle	29 19:38	29 19:38
Letras	31 16:34	31 16:34
Letras	31 16:34	31 16:34
Letras	29 19:16	29 19:16

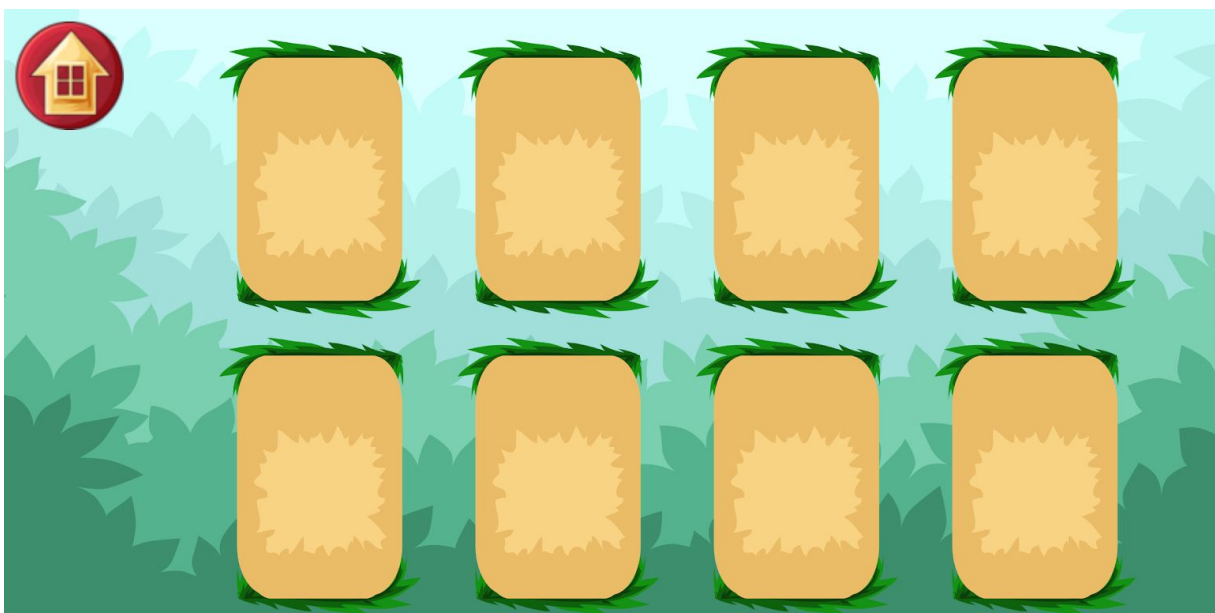
activity\_games aquí hi trobem el menú de selecció de jocs on podem escollir el joc al que volem jugar on hi ha un OnClick en tots el botons i cadascun redirigeix al seu joc corresponent



**Joc de Cartes (CardGame):** En aquest joc son dos classes diferents i una té més cartes que l'altre.



Després de un contador es giren les cartes junt amb una animació i es queda així:



## Joc de lletres (LetterGame) i joc de números(NumberGame):

Aquest jocs tenen una cosa en comú i es que gràcies a un numero **random** i un **include** es poden carregar diferents layouts amb la mateixa classe.

Numero random:

```
random = (int) (Math.random() * 3 + 1);
```

Include:

```
<include layout="@layout/vocals"/>
```

Aquest layout es un layout en general que només mostra les vocals( en el cas de el joc de lletres). Pero al fer el include es mostra l'altre layout junt amb les vocals. D'aquesta manera es poden fer diferents nivells amb una sola classe.

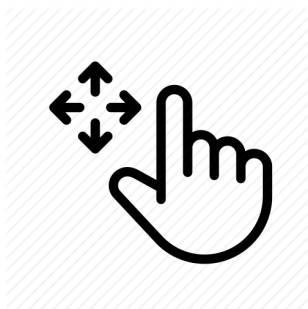
Layout numeros:



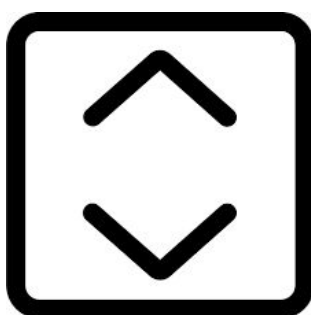
### 3.3 Interaccions



**Click:** Es pot fer click en qualsevol part de de les diferents activitats que es fara un soroll o una animació diferent.



**Drag and Drop:** Es pot arrossegar diferents elements de l'aplicació. Com per exemple en el joc de les figures(**shapes**) s'arrosseguen les figures per posar les cares o en el joc de **puzzles** s'arrosseguen les peces per posarles al lloc que toquin.



**Scrollview:** Es pot arrossegar per amunt i per avall per mirar una llista. En el cas de **HappyHour** es pot arrossegar la llista que mostra el temps de joc (**playtime**).



## 4. Conclusions

La conclusió del projecte ha sigut positiva ja que compleix amb els requisits més importants i bàsics proposats al començament del projecte:

- Creació de sis jocs diferents
- Implementació del firebase
- Saber estructurar un projecte
- Implementació de llibreries externes i el seu funcionament.
- Funcionament Android Studio i les eines que té.
- Implementació d'un disseny gràfic homogenic i agradable a l'ús.
- Creació d'un projecte en grup amb l'eina Github.

La planificació plantejada s'ha assolit de manera positiva en la majoria de casos.

Alguns jocs han sigut modificats o canviats completament per assolir una major varietat i homogeneïtat del producte.

### 4.1 Futures Implementacions

En aquest apartat parlarem sobre futures implementacions que hem mencionat anteriorment i que parlarem aquí, que no s'han pogut fer per falta de temps, falta d'interès o poca prioritat sobre altres.

- Menú d'ajuda (No implementat perquè no ho veiem necessari)
- Menú de configuració, com ara el volum, nivell de dificultat, etc...
- Més dades emmagatzemades en firebase sobre l'ús de l'aplicació
- Implementar més animacions, efectes i sons als diferents jocs.

- Implementar més jocs i nivells en els diferents jocs ja implementats.

## 5. Glossari

**Java:** El Java és un llenguatge de programació dissenyat el 1990 per James Gosling amb altres companys de Sun Microsystems a partir del llenguatge C.

**Android:** Android és un conjunt de programari per a telèfons mòbils que inclou un sistema operatiu, programari intermediari i aplicacions.

**Android Studio (IDE):**Android Studio és l'entorn de desenvolupament integrat per a la plataforma Android.

**Gimp:** GIMP és una aplicació per al tractament d'imatges, creat per voluntaris i distribuït sota la llicència GPL.

**GitHub:** GitHub és un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font així com afegint les seves característiques pròpies.

**Framework:** és Infraestructura de programari que, en la programació orientada a objectes, facilita la concepció de les aplicacions mitjançant la utilització de biblioteques de classes o generadors de programes.

**SDK:**SDK es el acrónimo de “Software Development Kit” (Kit de desarrollo de software). El SDK reúne un grupo de herramientas que permiten la programación de aplicaciones móviles.

**StackOverflow:** Stack Overflow es un sitio web utilizado por una comunidad de desarrolladores informáticos, en la cual otros desarrolladores pueden encontrar soluciones a problemas de programación en diferentes lenguajes.

**Firestore:** és una plataforma per al desenvolupament d'aplicacions web i aplicacions mòbils desenvolupada per James Tamplin i Andrew Lee el 2012 i adquirida per Google el 2014.

És una plataforma situada en el núvol, integrada amb Google Cloud Platform , que utilitza un conjunt d'eines per a la creació i sincronització de projectes que seran donats d'alta qualitat, fent possible el creixement del nombre d'usuaris i donant resultat també a l'obtenció de una major monetització.

**Mockup:** Un mockup es una maqueta digital o fotomuntatge que vol donar credibilitat a un disseny. Es tracta de dissenys a escala real que aporten una característica que fa que les idees es tornin tangibles.

**Adapter:** Un adaptador és un objecte d'una classe que implementa la interfície *Adapter*.

Aquest actua com un enllaç entre un conjunt de dades i un adaptador vista, un objecte d'una classe que estiguin a la classe abstracta *AdapterView*.

El conjunt de dades pot ser qualsevol cosa que presenti dades en una manera estructurada. Arranjaments, objectes *List* i objectes *Cursor* amb usats, generalment, amb conjunts de dades.

**RecyclerView:** El widget de RecyclerView és una versió més avançada i flexible de ListView.

Al model RecyclerView, diversos components diferents treballen junts per mostrar les vostres dades. El contenidor global de la vostra interfície d'usuari és un objecte RecyclerView que afegiu al vostre disseny. El RecyclerView s'omple de visualitzacions proporcionades per un gestor de disseny. Podeu utilitzar un dels gestors de disseny estàndard (com LinearLayoutManager o GridLayoutManager), o bé implementeu els vostres propis.



## 6. Bibliografia

<https://developer.android.com/guide>

<https://stackoverflow.com>

<https://stackoverflow.com/questions/3364010/android-animation-flip>

<https://github.com/>

<https://developer.android.com/reference/android/animation/ObjectAnimator>

<https://developer.android.com/reference/android/os/Handler>

<https://developer.android.com/studio/releases/platforms>

<https://blog.trello.com/how-to-scrum-and-trello-for-teams-at-work>

<https://www.vogella.com/tutorials/AndroidDragAndDrop/article.html>

[https://www.youtube.com/watch?v=fqU4zc\\_XeX0](https://www.youtube.com/watch?v=fqU4zc_XeX0)

<https://code.tutsplus.com/tutorials/android-sdk-create-a-drawing-app-touch-interaction--mobile-19202>

<https://dragosholban.com/2018/03/09/how-to-build-a-jigsaw-puzzle-android-game/>

<https://github.com/rohan20/BalloonGame>

<http://www.iconarchive.com/show/cute-icons-by-yohproject/camera-2-icon.html>

# 7. Annexos

## 7.1. Guia rapida de l'aplicació

En aquesta guia de l'aplicació es veu el seu funcionament general

Quan entrem al joc veurem la pantalla inicial on trobem el botó de play



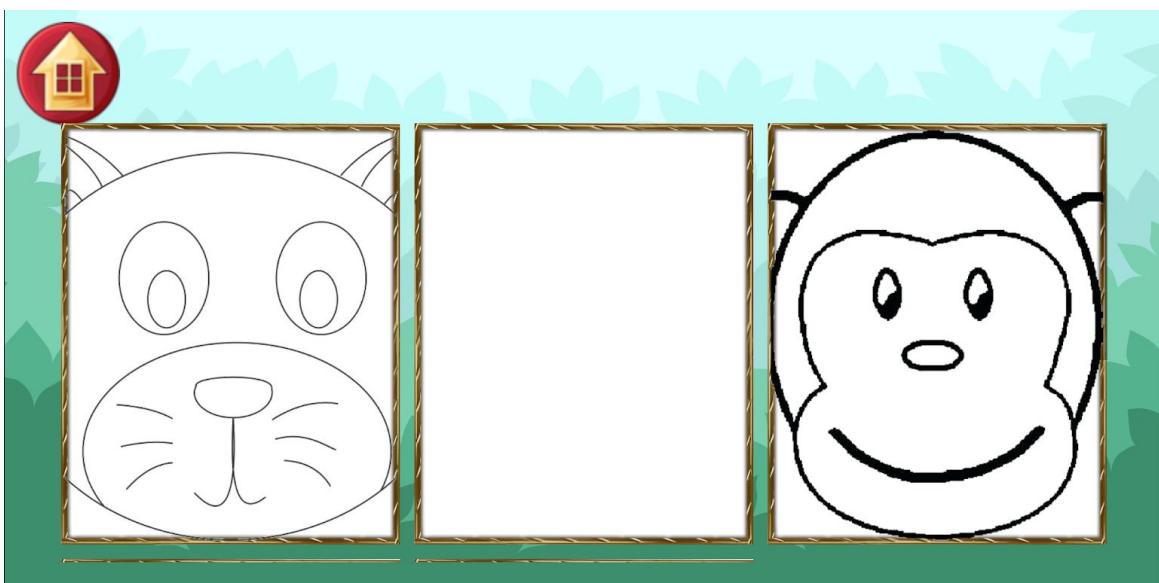
Una vegada li hem donat al play ens portarà a la pantalla principal de l'aplicació on veurem que hi ha diferents opcions: Informació de l'aplicació, pantalla de selecció de jocs, tria un joc aleatori i consultar la informació de entrada als jocs.



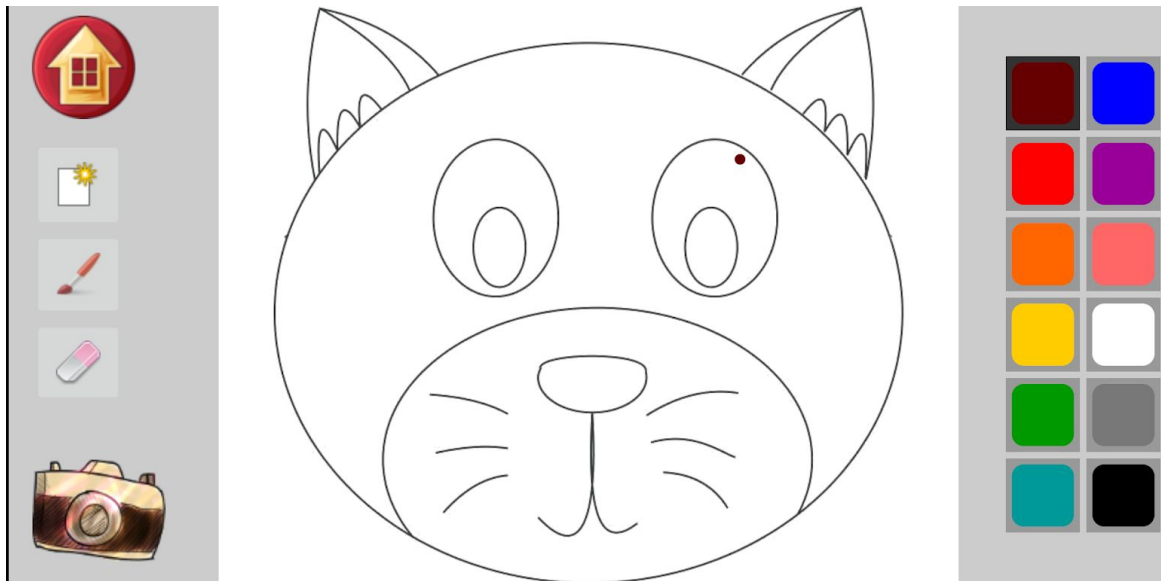
Quan anem a la pantalla de selecció de jocs ens trobem amb els diferents jocs que hi ha:



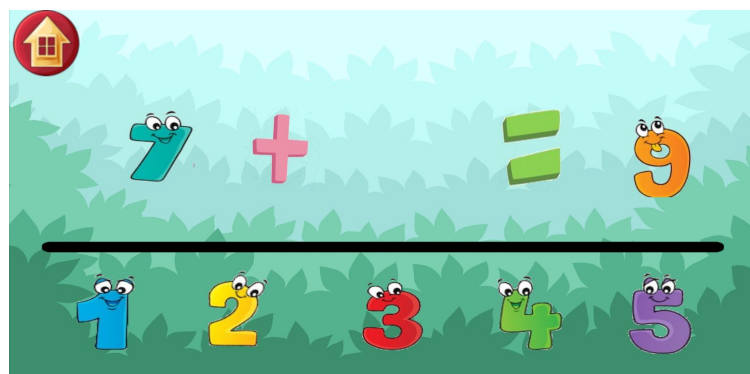
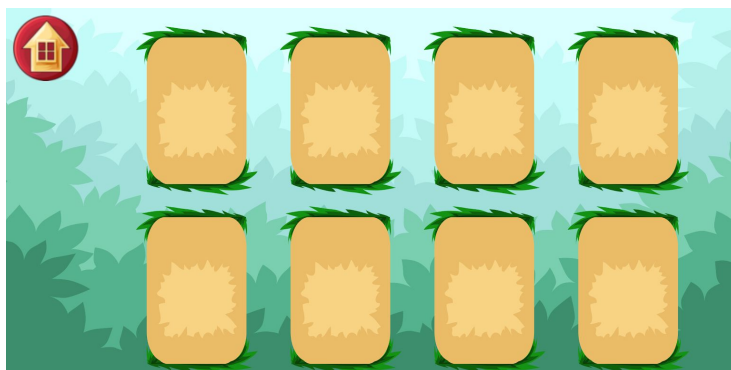
Al joc de pintar podem escollir una plantilla o fer-ne un dibuix desde zero.



Al escollir un dibuix ens apareixerà una paleta amb els diferents colors a la dreta que el podem seleccionar fent click en el color que desitgem , podem canviar el tamany del pinzell de dibuix i en cas de tornar a començar li donem a pagina nova i s'esborraran tots els canvis fets al dibuix.

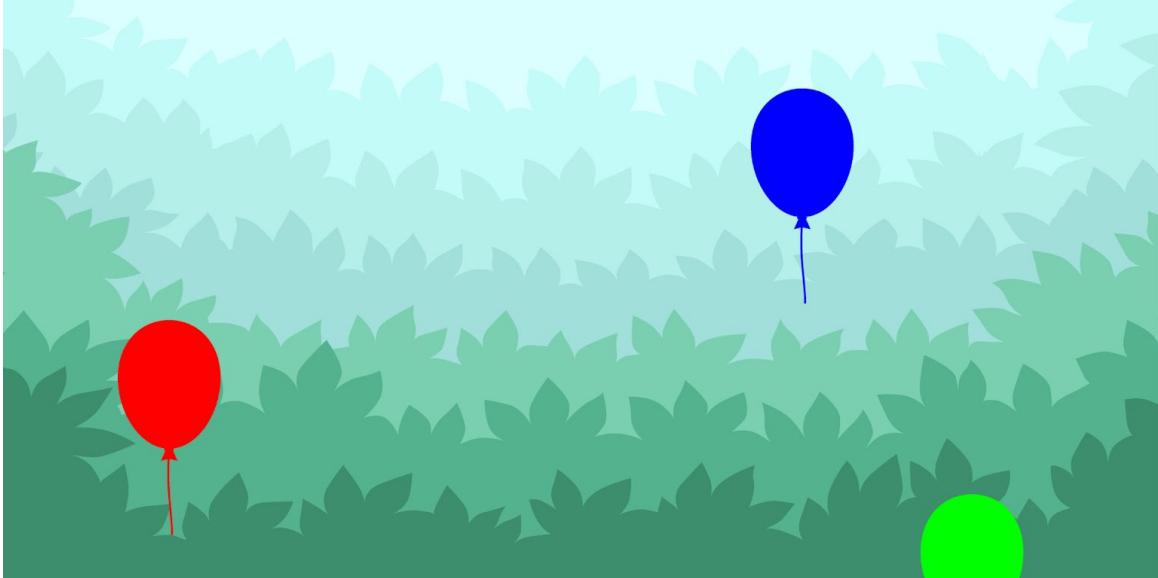


En cas d'escollir un joc en el que no hi ha plantilles per triar en portarà directament al joc escollit.

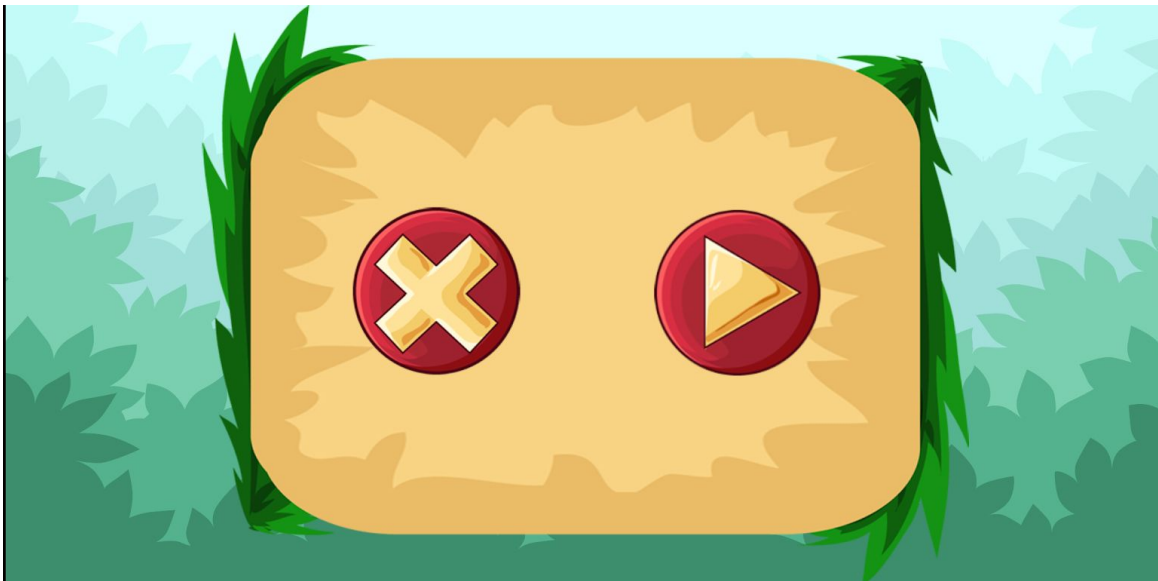




Una vegada ens hem passat un joc ens posarà una pantalla de celebració on es pot explotar uns globus:



Quan acabem la celebració amb els globus ens sortirà una pantalla en la que podrem escollir si continuar jugant o sortir del joc.



## 7.2.Codi font i Trello

<https://github.com/vanazco/HappyHour>

<https://trello.com/b/J3mES0e6/happy-hour>