

# **MEMORIA DEL PROJECTE**

## **JOC CARCASSONNE**

  
**Universitat  
de Girona**  


**JAVIER MUÑOZ  
MARC GRAU  
TENG LIU**

## INDEX

---

El projecte.....	Pag 2.
Gestio treball en grup.....	Pag 3.
Explicació algoritmes més importants.....	Pag 4.
Intel·ligencia del jugador CPU.....	Pag 6.
Millores del programa.....	Pag 5.
Dificultats obtingudes.....	Pag 7.

## El projecte

En aquest que hem dissenyat del joc de Carcassonne hem après moltes coses sobretot del disseny amb java, fabricació de interfície amb javaFX, estructuració inicial d'un projecte i sobre tot el treball en grup.

Inicialment el final del projecte el veiem una mica lluny i amb moltes dificultats, però a mida que anàvem avançant amb el projecte, no era tant complicat i avançàvem sense gaires complicacions, a excepció d'un parell de cops, que explicarem en un apartat més endavant.

El disseny inicial que vam fer de les classes amb el seu diagrama al principi no ho veiem gaire clar però realment aquesta documentació inicial ens va servir força d'ajuda per el repartiment de treball en grup i anar a la idea de lo que cadascú havia de fer, sempre amb alguns possibles canvis.

Per part de el disseny de la Gui ens ha agradat fer en un projecte com es aquet el disseny d'una interfície gràfica, i ens ha motivat veure el funcionament el funcionament de la aplicació a través de la interfície.

Per a la prova d'aquest i els jocs de proves hem utilitzat els fitxers p1.txt i p2.txt. Mitjançant la modificació dels dos hem aconseguit provar totes les funcionalitats desenvolupades i trobar bugs els quals la majoria hem pogut arreglar.



## Gestio treball en grup

El repartiment del treball creiem que ha sigut correcte, inicialment el treball que cadascú feia era força individual però a mida que anàvem avançant, començàvem a necessitar classes que implementava un altre i ens anàvem juntant per posar-nos d'acord.

El repartiment de les classes han sigut:

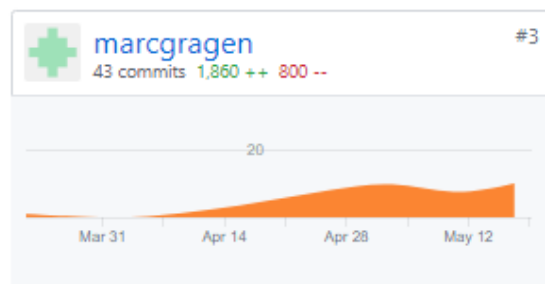
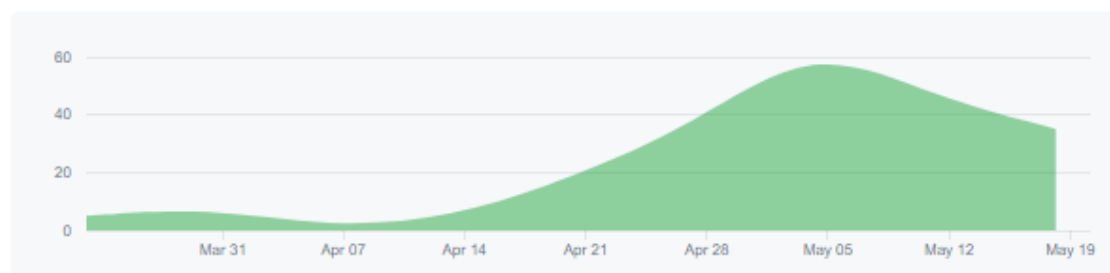
- Teng Liu: REGIO FITXA POSSESSIO CAMP ESTRUCTURA CAMI CREUAMENT CIUTAT ESGLESIA
- Javi Muñoz: POSICIO GUI JOC TAULER
- Marc Grau: BARALLA TIRADA JUGADOR LLEGIR FITXER CONTROLABLE MAQUINA

Hem utilitzat el Git amb Github creant un repositori per aquest projecte i cadascú per cada tasca diferent anava creant PullRequest i un altre intentava sempre de revisar i acceptar aquestes peticions.

Mar 24, 2019 – May 21, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



## Explicació algoritmes més importants

Realment no tenim algoritmes massa complicats en el projecte, però un dels algoritmes més complicats, que tampoc ens va costar gaire de dissenyar es:

```
///@pre Hi ha almenys una fitxa al tauler en la posicio inicial (5,5)
///@post Retorna llista de les posicions on es pot ficar la fitxa f
public ArrayList<Posicio> getPosDisponibles(Fitxa f)
```

Que fa cridar la funció recursiva:

```
///@pre Funcio recursiva de getPosDisponibles(), 9<=x<=9 && 9<=y<=9
///@post Retorna la llista de posicions on es pot colocar la fitxa
private ArrayList<Posicio> buscaColocacioFitxes(int x, int y, Fitxa f, ArrayList<Posicio> posicionsVisitades)
```

Aquesta funció recursiva parteix des de el centre del tauler i busca a traves de les posicions de les fitxes al voltant on pot encaixa la fitxa passada per referencia i es va expandint posició a posició.

## Intel·ligència del jugador CPU

La «intel·ligència» de la màquina comença al mètode `gestionarMaquina()` de la classe `Tirada.java`. En aquesta es fa un recorregut per a totes les posicions disponibles (incloent els angles\*) el qual utilitza el senzill mètode d'agafar la posició de punts màxima per tal d'assignar-la. Aquests punts es passen conjuntament amb una regió d'una fitxa a on ficar el seguidor.

En cas que en totes les opcions els punts siguin 0 fa una tirada aleatòria.

Per a calcular els punts de cada posició es crida el mètode `simularPunts(...)` de la classe `Tauler`. Aquesta genera les fitxes que envolten a la fitxa actual i les regions més properes d'aquestes. Amb aquesta informació va obtenint i sumant els punts per cada regió de la fitxa actual i a més comprova quina regió és la que obté més punts per a tal d'assignar un seguidor.

A més el mètode `getPuntsRegio(...)` atorga dos estats a la IA del programa: tancar i crear:

- Tancar: Quan el jugador té 3 seguidors o menys aquest intenta tancar totes les possessions possibles per tal de recuperar els seguidors. En cas de no poder es mira d'ampliar una possessió existent.

- Crear: Aquesta es duu a terme sempre que el seguidor té més de 3 seguidors i consisteix en buscar possessions pròpies per ampliar o bé crear-ne de noves. En cas de no poder obtenir cap punt es fa una crida recursiva per tal de reconsiderar la opció de poder tancar alguna possessió ja existent.

Tot i ser una IA molt senzilla hem demostrat que és bastant eficaç ja que competint contra humans ha arribat a guanyar bastantes partides contra nosaltres mateixos i familiars i amics nostres. Per això hem considerat que seria un nivell òptim per competir amb ella amb una certa dificultat.

També es té en compte el fet de que un seguidor no es pot ficar a totes les regions possibles, si no que està limitat depenent de l'entorn o bé dels propietaris d'altres possessions.

Això ho gestionem mitjançant la crida del mètode `getPosicionsDePossessio(...)` i comprovant abans de col·locar una fitxa a una posició que la regió no pertanyi a un altre jugador.

## Millors del programa

En la Gui i tauler, una de les millors més importants que haurien d'haver es la limitació del espai en el tauler, les dimensions del tauler el vam limitar a 10x10 estàtic perquè teníem la limitació per desconeixement de com fer un tauler en la interfície que sigui scrollable a través de petició de l'usuari.

Per temes de dificultat i falta de temps no vam acabar de fer la gestió de les possessions Camp en la classe tauler, s'hauria de acabar de implementar aquesta possessió.

Una de les millors en quan a IA que haguéssim pogut considerar és el fet de tenir en compte la opció d'evitar que els altres jugadors aconseguixin punts.

Actualment cada jugador màquina només considera la seva situació independentment de com progressen els altres. Una millora important podria ser que per exemple si veu que col·locant una casella a un punt estratègic pot evitar que un jugador sumi una quantitat important de punts, la col·loqui per tal de que aquest no pugui completar la possessió.

De fet es podria millorar casi fins l'infinit aquest punt.

## Dificultats obtingudes

Una de les dificultats que hem tingut ha sigut la gestió de les possessions en la classe tauler.

No vam acabar de fer la possessió de Camp ja que ho vam intentar però se'ns escapava molt de les mans la gestió de com es guardava les possessions de camp en el tauler, i per falta de temps i complicació d'aquest no vam aconseguir fer funcionar la possessió camp.

Vam començant a gestionar la classe de possessió de fitxes, però després ens vam donar compte que només una llista de fitxes no bastava per gestionar-la sinó que també ens feia falta per cada fitxa una llista de regions i vam acabar fent una llista de parelles de fitxes i llista de regions.

Hem tingut també amb tirades quan jugaven màquines algunes dificultats.

El problema va ser que a la primera ronda cadascú tenia la seva tirada però el jugador màquina no acabava mai la seva tirada, per tant en cas que el jugador 1 fos humà i el 2 màquina, l'humà agafava la tirada del jugador 2 ja que aquest no l'havia acabat. Ho vam solucionar afegint un botó a la part superior de la Gui que quan s'apreta el jugador màquina acaba el torn.

Aquesta opció també en va permetre poder veure més detalladament les passes que realitza cada jugador màquina al seu torn corresponent. Per tal de resoldre'l ens vam ajuntar els 3 i amb una tarda tots investigant el mateix problema el vam trobar i com es pot veure la resolució d'aquest va ser senzilla.

A les subclasses de la classe possessió a l'hora de calcular els punts de cada possessió, hem tingut de pensar per cada fitxa i les seves regions que tenia per saber quin format de fitxa era i com afectava a l'hora calcular els punts per si la possessió estava tancada o no, tema si una possessió està tancat amb la classe ciutat és la que més dificultat hem tingut ja que és la classe tipus de possessions que més variables has de tenir en compte per saber si està tancat o no.

El principi no sabíem perquè servia la classe estructura, ja que totes les subclasses tenien mètodes diferents.

A la classe fitxa hem tingut dificultats per rotar la fitxa ja que no teníem en compte que la fitxa la podíem rotar per els dos sentits, sentit horari i antihorari.

Una cosa que no hem aconseguit fer funcionar es fer un jar del projecte, vam estar intentant generar-lo per no vam acabar de fer-lo funcionar, d'un costat ens el generava però no el podíem executar, per l'altre no ens el generava per un error de que faltaven unes llibreries que no podíem obtenir, i al final no vam poder generar un jar que poguéssim executar correctament el joc.