

Semi automatic highlights identification in a RTS videogame

Albert Porteros Villar

Resum– En aquest projecte proposem una eina d'ajuda a investigadors de didàctica de les matemàtiques, que consisteix en l'anàlisi semi automàtic d'accions fetes per un jugador mentre fa una partida a un videojoc d'estratègia en temps real. En primer lloc, es proporciona de forma automàtica un conjunt de gràfiques i taules que estructurin les dades de l'estudi, extretes prèviament mitjançant un eyetracker. En segon lloc, s'extreuen i s'analitzen de forma automàtica un conjunt de patrons d'accions duts a terme al llarg de la partida, ja sigui mostrant quins són els més repetits durant tota una partida, o cercant-ne de concrets definits de forma manual. En tercer lloc, es realitza una classificació semisupervisada de quins patrons poden ser beneficiosos per l'objectiu pedagògic proposat per l'equip investigador. Finalment, es fa una comprovació dels resultats amb el fitxer analitzat manualment per part d'un professor, el qual anomenem *Ground Truth*, i utilitzant el vídeo editat per un altre estudiant en el qual s'intersecten les dades de l'Eye Tracker amb la partida real.

Paraules clau– Vector Tower Defense II, Python, Eye Tracker, Excel, VBA, Vectoids, Fixations, Clicks, Ground Truth, Script, Naive Bayes, patrons, anàlisi, probabilitats, Python.

Abstract– In this project we propose a tool for helping researchers in mathematics education, which consists in semi-automatic analysis of actions taken by a player while playing a game of strategy in real time. First of all, a set of graphs and tables that structure study data, previously extracted through an eyetracker, is automatically provided. Secondly, a set of patterns of actions carried out throughout the game is automatically analyzed, either by showing which are the most repeated during a whole game, or by looking for specific patterns manually. Thirdly, there is a semi-supervised classification of which patterns can be beneficial for the pedagogical objective proposed by the research team. Finally, a check of results with the file manually analyzed by a teacher, which we call *Ground Truth*, and using the video edited by another student in which the Eye Tracker data is intersected with the actual game.

Keywords– Vector Tower Defense II, Python, Eye Tracker, Excel, VBA, Vectoids, Fixations, Clicks, Ground Truth, Script, Naive Bayes, patrons, analysis, probability, Python.



en profunditat les oportunitats d'aprenentatge matemàtic que pot aportar un videojoc d'estratègia en temps real a les aules de secundària.

1 INTRODUCCIÓ

ELS videojocs cada vegada estan més en la vida quotidiana de la gent jove i és un dels motius pels quals s'estan investigant diferents aplicacions educatives relacionades amb els videojocs. L'objectiu del projecte principal, per part d'un equip d'investigadors del departament de didàctica de les matemàtiques, és estudiar

El videojoc seleccionat és el Vector Tower Defense II (figura 1), un videojoc d'estratègia a temps real (RTS) en el que el principal objectiu és que els enemics no aconseguixin arribar al final d'un camí, i per impedir-ho disposem d'estructures que els hi faran mal. Les diferents estructures tenen diferents formes d'atacar, com per exemple poden ser a un sol enemic, en àrea, augmentant el mal contra certs enemics, etc.

- E-mail de contacte: albert.porteros@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Aura Hernández Sabaté (Departament de Ciències de la Computació)
- Curs 2018/19

També disposem d'un *Eye Tracker*, que consisteix en una càmera que monitorea els moviments de l'ull del jugador

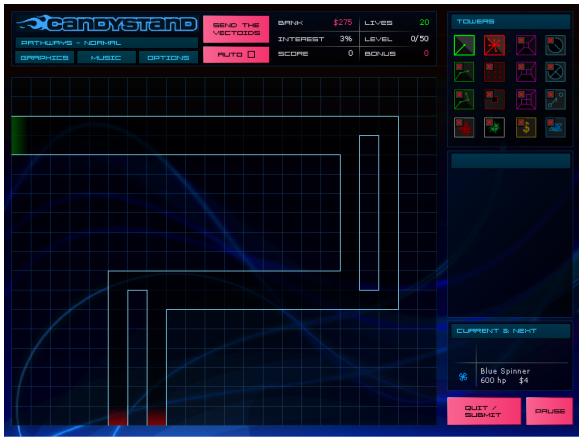


Fig. 1: Imatge del videojoc Vector Tower Defense II

per tal de saber en tot moment que està veient el jugador i on està prestant especial atenció per poder fer una anàlisi a posteriori. Aquest dispositiu ens retorna la informació al final de la partida en forma d'un fitxer amb extensió *.txt*.

Per aquest treball, i altres ja realitzats anteriorment, hem dividit la pantalla del joc en diferents regions (figura 2), 10 en concret. També els hi hem assignat un nom i un número a cadascuna de les regions per tal de poder treballar amb elles. Les regions són les següents:

- 1 - Game Details
- 2 - Gameplay Area
- 3 - Information
- 4 - Pause
- 5 - Quit
- 6 - Sell
- 7 - Send Vectoids
- 8 - Tower Selection
- 9 - Upcoming enemies
- 10 - Upgrade

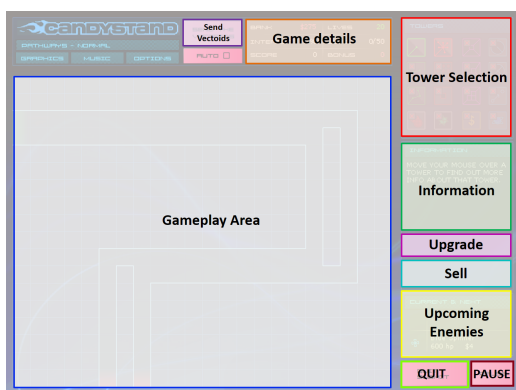


Fig. 2: Regions d'interès de la pantalla (AOIs)

L'objectiu d'aquest treball és el de facilitar la feina a l'equip investigador mitjançant la creació d'una eina d'anàlisi de les dades que extreu aquest *Eye Tracker*, ja que aquestes dades estan en un format intel·ligible. Per aconseguir això començarem amb una programació en llenguatge *Visual Basic for Applications*, un llenguatge utilitzat per programar en Excel i altres programes d'Office, per tal d'extreure un primer resultat en forma de taules i gràfiques resum, les quals podran modificar els mateixos

usuaris ja que són taules i gràfiques dinàmiques i per tant podran posar especial èmfasi en les dades que prefereixin. A continuació, procedirem amb l'anàlisi dels patrons per tal que l'usuari pugui veure un resum dels patrons seguits pel jugador, i també vegi si aquests patrons són patrons positius o negatius. Aquesta anàlisi el realitzarem utilitzant el llenguatge de programació de *Python*. L'usuari final podrà trobar detalls dels patrons que vulguin buscar, com per exemple nombre de repeticions o instants en els quals es realitzen. Finalment, seguint amb el mateix llenguatge, realitzarem una anàlisi utilitzant un algorisme de *Naive Bayes* combinat amb *Laplace smoothing* per tal de poder classificar els diferents patrons com a positius o negatius depenent de partides prèviament analitzades i classificades. El principal objectiu és que tot això es pugui realitzar automàticament i que l'usuari final hagi de realitzar uns passos molt bàsics per tal de disposar de tota la informació.

Les diferents parts de les quals es compona aquest treball són:

- Taules i gràfiques dinàmiques en *Excel*.
- Anàlisi de patrons general.
- Anàlisi d'un patró en concret.
- Classificació semi-supervisada dels patrons utilitzant *Naive Bayes*.

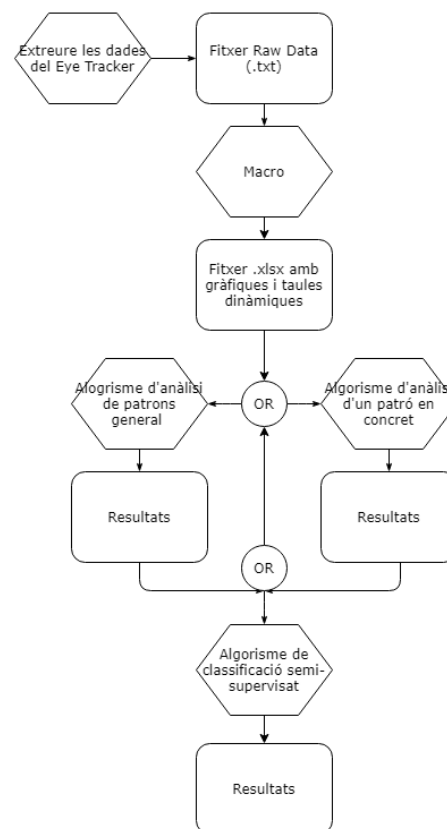


Fig. 3: Diagrama de flux

Tal com podem observar en el diagrama de flux de la nostra eina (figura 3), en primer lloc haurem d'extreure les dades del Eye Tracker. A continuació, executarem la macro (arxiu programat en VBA) que s'explicarà en la secció d'obtenció i manipulació de dades (Secció 2 d'aquest document). Una vegada executada la macro, disposarem d'un fitxer d'Excel amb unes gràfiques i taules dinàmiques. En el pròxim pas, l'usuari final haurà de seleccionar si prosseguir amb l'algorisme d'anàlisi de patrons general (Secció 3 d'aquest document) o amb l'algorisme d'anàlisi d'un patró en concret (Secció 4 d'aquest document). A continuació disposarem dels resultats de l'algorisme i podrem escollir si tornar a executar un dels algorismes o executar l'algorisme final de classificació semisupervisat (Secció 5 d'aquest document).

2 OBTENCIÓ I MANIPULACIÓ DE DADES

Les dades amb les quals treballem són extretes de l'*Eye Tracker* en un fitxer en format *.txt* com ja hem comentat anteriorment. Aquest fitxer és molt difícil d'interpretar degut a la quantitat de dades que conté, moltes d'elles innecessàries per l'objectiu del treball. Per tant, el primer que hem fet ha estat seleccionar les columnes necessàries i afegir algunes columnes combinades per tal de poder ser utilitzades posteriorment. Tot això ha estat automatitzat mitjançant codi *VBA*.

Les principals columnes que hem utilitzat per a aquest projecte han estat la columna de *Category*, la qual ens defineix el tipus d'acció que ha realitzat el jugador en un instant. Aquesta columna ens interessa especialment per seleccionar la categoria de *Fixation*, la qual referència les accions en les quals el jugador està mirant fixament una àrea. També hem utilitzat la categoria de *Clicks*, que gràcies a les columnes de *MousePositionX_px_* i *MousePositionY_px_* adquirirem la posició per píxels on l'usuari ha realitzat un click amb el ratolí. També utilitzarem les columnes de *TrialStartRawTime_ms_* *EventStartRawTime_ms_* per fer un càlcul i saber exactament l'instant en el qual s'ha realitzat l'acció.

Una vegada finalitzada la fase d'extracció i manipulació de dades, l'equip d'investigació ens vàrem reunir. Després de discutir sobre el que necessitaven, vàrem arribar a la conclusió que estaria molt bé realitzar un estudi de les accions dels jugadors per rondes, per això vàrem afegir la columna de rondes al fitxer. Després de realitzar un primer esbós sobre com tindriem la informació organitzada en les gràfiques, vàrem voler expandir el projecte i vàrem voler obtenir la informació no només de fixacions, sinó també de clicks, i per això vàrem fer la columna addicional de la zona en la qual es feia click. Creiem que és molt interessant tenir aquesta informació perquè així es pot veure l'evolució d'un jugador al llarg de la partida, com per exemple, que al principi llegeix molt la descripció de les estructures i al final no tan a causa del fet que ja les coneix, o que al principi no fa click mai a la zona per millorar o vendre les estructures i cap a les últimes rondes sí.

A continuació tenim una taula amb les columnes originals i modificades del fitxer i un breu resum de com s'han

creat cadascuna de les diferents columnes:

Columnes utilitzades del fitxer		
Nom de la columna	Original	Creada
Category	X	
TrialStartRawTime_ms_	X	
EventStartRawTime_ms_	X	
FixationPositionX_px_	X	
FixationPositionY_px_	X	
MousePositionX_px_	X	
MousePositionY_px_	X	
FixationPositionX_px_2		X
FixationPositionY_px_2		X
AOIName2		X
AOI		X
Tiempo		X
MouseX		X
MouseY		X
Columna1		X
Ronda		X
ClickArea		X

- **AOIName2** - S'ha creat mitjançant una columna que ens retornava la posició on mira el jugador (*FixationPositionX_px_* i *FixationPositionY_px_*) i la seva funció és etiquetar de forma alfabètica a quina àrea de la pantalla està mirant el jugador.
- **AOI** - S'ha creat etiquetant **AOIName2** numèricament i la raó és que treballarem numèricament amb les àrees, etiquetades del 1 al 10.
- **Tiempo** - Retorna el temps en el qual comença l'acció i ens servirà per poder analitzar els instants concrets en el algorisme de *Python*.
- **Columna1** - Comproba si al fer click, els píxels són exactament en el botó de *Send Vectoids* i ens servirà per contar les rondes.
- **Ronda** - Mira si a **Columna1** hi ha valor i realitza un comptador de quina ronda ens situem.
- **ClickArea** - Realitza la mateixa funció que **AOIName2** però en comptes d'agafar les *fixation position* utilitza les columnes de *MousePositionX_px_* i *MousePositionY_px_*.
- **MouseX, MouseY, FixationPositionX_px_2** i **FixationPositionY_px_2** - Les 4 columnes són simplement unes còpies de les seves respectives columnes, però arreglant el format numèric.

Una vegada tenim totes les columnes necessàries creades, es generaran dos fulls extres en el mateix fitxer d'*Excel*. En un tindrem dues taules dinàmiques que ens indicaran respectivament els clicks fets en cada ronda i les *fixations*, que són instants en els que el jugador s'ha quedat mirant una posició en concret, en cada ronda. En el segon full es realitzaran unes gràfiques de barres dinàmiques amb la mateixa informació (figura 4). Aquestes disposaran de filtres en l'àmbit d'objecte per modificar la vista, posar-li més èmfasis en una ronda en concret, buscar una àrea per analitzar, eliminar una àrea del joc, etc.

Fixations X Ronda

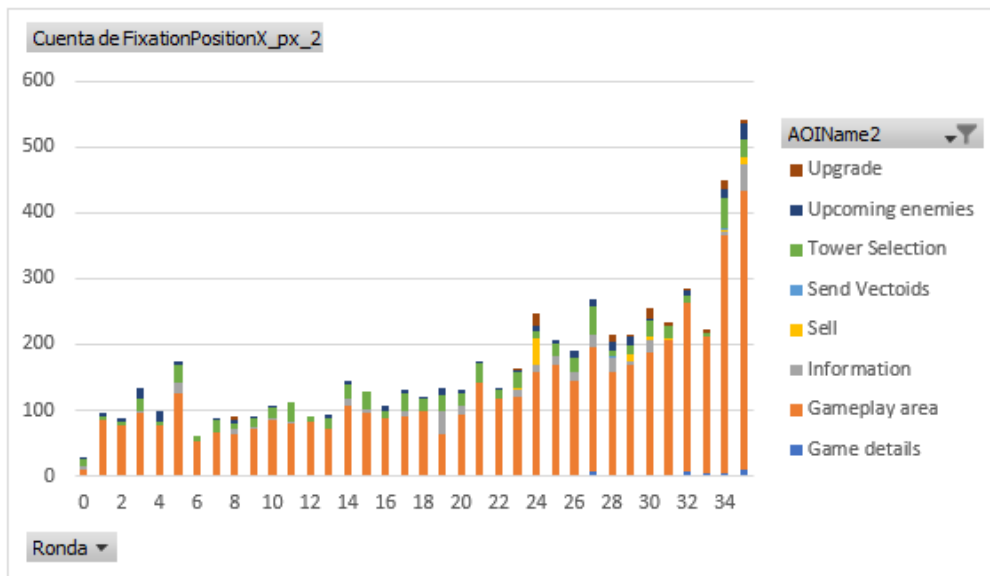


Fig. 4: Exemple de gràfica dinàmica

3 ANÀLISI DE PATRONS GENERAL

3.1 Resum del funcionament

Que és un patró? Per a aquest treball, hem decidit que els patrons són conjunts d'accions en l'àmbit de *fixations* que realitza un jugador. Per tant si un jugador mira la *Tower Selection* àrea, després la *Information* àrea i després la *Gameplay Area*, seria un patró de tres accions.

La principal funció d'aquest algorisme és la de fer un recompte del número de vegades que apareixen tots els patrons de la llargada escollida per l'usuari, ordenar-los i mostrar-los per pantalla.

3.2 Preparació

Per tal de poder realitzar l'anàlisi de patrons, cal primer de tot haver executat la *macro* programada amb *VBA* per tal que les columnes necessàries estiguin presents en el nostre document. Una vegada l'arxiu està preparat i guardat en la mateixa carpeta que el fitxer *patrons.py*, podem executar l'algorisme preparat.

3.3 Extracció de dades i creació de diccionaris

Aquest algorisme està basat en llistes i diccionaris, utilitzant el mòdul de *pandas*. Primer de tot fa una llista amb els camps que necessitem, que en aquest cas com que estem fent l'estudi general dels patrons només serà el camp '*AOI*' que ens retorna la posició que estava mirant el jugador en cada instant etiquetat de manera numèrica de l'1 al 10. Una vegada hem importat el camp, netejarem els valors pel fet que a la *raw data* no només tenim les *fixations*, també tenim els clics i altra informació no rellevant per a aquest

projecte. També eliminarem valors consecutius del mateix camp, pel fet que no ens aporta informació.

Una vegada realitzada la neteja preliminar de valors, l'algorisme demana a l'usuari si vol un llistat dels patrons de 3 o 4 accions i l'usuari final ho insereix per teclat. Un cop decidit la mida del patró, el programa recorre totes les accions que ha pres el jugador mirant-les en grups de la llargada delimitada pel patró.

Per evitar valors que sortien moltes vegades repetits a causa del fet que el jugador estava dubtant, hem establert que les 3 o 4 accions han de ser diferents entre si, que per exemple mirar a la *Tower Selection* àrea, després a la *Gameplay Area* i una altra vegada a la *Tower Selection* àrea no és un patró.

3.4 Resultats

Una vegada tenim el diccionari amb tots els patrons i el número de vegades que es repeteix cadascun d'ells, els ordenem per tal de mostrar per pantalla els més repetits i només mostrem els que tinguin 3 repeticions o més per tal que l'usuari pugui veure els que realment són importants.

Tal com podem observar en la imatge (figura 5), l'algorisme ens retorna un resultat en format: ((patró dividit en elements), núm. de repeticions). Les posicions estan etiquetades numèricament per facilitar la feina a l'hora de treballar amb els valors, però es disposa d'una guia que relaciona les posicions a nivell numèric amb la seva *AOI*. Per exemple, el primer resultat seria el patró *Tower Selection - Information - Gameplay Area - Upcoming enemies* s'ha repetit 5 vegades al llarg de la partida.

```

Vols el patró de 3 o 4 moviments?4
-----
((8, 3, 2, 9), 5)
((9, 2, 8, 3), 4)
((1, 2, 8, 3), 4)
((8, 3, 2, 1), 3)
-----
Guía de posicions:
1-Game Details
2-Gameplay Area
3-Information
4-Pause
5-Quit
6-Sell
7-Send Vectoids
8-Tower Selection
9-Upcoming enemies
10-Upgrade

```

Fig. 5: Resultat algorisme patrons general

Una vegada hem fet aquest primer anàlisi, el programa guarda automàticament un fitxer *.csv* amb tots els patrons que s'han realitzat, la freqüència d'aquests patrons i el valor que li hem donat a la partida (1 si és positiva i 0 si és negativa). La diferenciació entre partides positives i negatives hauria de ser molt clara per tal d'entrenar l'últim algorisme que hem creat, el de classificació, per tant haurien de ser partides de professors o gent experimentada etiquetades com a 1, i partides de nens que és la primera vegada que juguen a aquest joc com a 0. El professor es pot ajudar amb les taules i gràfiques extretes en la primera part del treball per decidir si una partida és positiva o negativa també.

4 ANÀLISI D'UN PATRÓ EN CONCRET

4.1 Resum del funcionament

La principal funció d'aquest algorisme és la de fer un estudi més profund d'un patró en concret escollit per l'usuari. La informació que l'algorisme extraurà d'aquest patró serà: nombre de repeticions, rondes en les quals es realitza el patró i els instants en els quals es realitza aquest.

4.2 Extracció de dades i preparació de les llistes

Per tal de realitzar una anàlisi més exhaustiu d'un sol patró, haurem d'importar més d'una columna del fitxer prèviament preparat amb *VBA*. En aquest cas exportarem les columnes: '*AOI*', '*Tiempo*' i '*Ronda*'. Tornarem a treballar amb conjunts de llistes aquesta vegada. Primer de tot, el programa demanarà a l'usuari que introdueixi el nombre d'elements que tindrà el patró. El nombre d'elements no està limitat de cap manera, per tant, si l'usuari sabes que hi ha un patró de 20 accions i el vol analitzar, només ho hauria d'indicar en el programa i aquest funcionaria. Una vegada l'usuari ha triat la longitud del

patró al programa, aquest li demana el valor de cada una de les posicions (recordem que les *AOI* estan etiquetades de manera numèrica per tal que no hi hagi problemes a l'hora d'escriure'ls o operar amb ells).

Una vegada l'usuari ha inserit el patró desitjat, l'algorisme es guarda les indexacions de les posicions inútils de la llista (les que no són *fixations*) per després poder-les eliminar no solament a la llista de *AOIs* sinó també en les llistes de '*Tiempo*' i '*Ronda*' per tal que segueixin alineades. A continuació realitzarem una transformació en la llista de '*Tiempo*' per tal que el temps sigui verídic. Això és necessari pel fet que el temps extret en la llista de '*Tiempo*' és un temps relatiu a l'inici de la gravació, i la gravació es reinicia cada 10 minuts segons els paràmetres establerts, per tant, hem realitzat un arranjament que diu que si el temps actual és més petit que el temps anterior (o sigui si hem canviat de gravació i per tant ha tornat a començar) que li sumi 600000, ja que la unitat de control de temps són mil·lisegons. Això ho hem aplicat en forma de *while* per tal que si han passat 2 cicles, li sumi 1200000 i així successivament. A continuació hem transformat els mil·lisegons en format '*minuts:segons*'.

4.3 Execució de la cerca

Una vegada tenim totes les llistes ben ordenades, amb només els elements importants disponibles i les unitats tal com les desitgem, ja podem executar l'algorisme de busca. Aquest algorisme el que realitza una cerca del patró desitjat en cadenes de la mateixa llargada del patró i cada vegada que el troba, li suma 1 a un comptador i es guarda en una llista l'instant de temps i la ronda en el qual l'ha trobat. A continuació podem veure part del codi d'aquest algorisme (figura 6).

```

count = 0
time = []
ronda = []
x=0
for i in range(len(listSepalWidth)-len(pattern)+1):
    tmp = listSepalWidth[i:(i+len(pattern))]

    if tmp == pattern:
        count +=1
        time.append(listSepalWidth2[i])
        ronda.append(listSepalWidth3[i])
        x=x+1

```

Fig. 6: Part de l'algorisme d'anàlisi d'un patró concret

En la figura que podem observar, *listSepalWidth* és la llista de '*AOI*', *listSepalWidth2* és la llista de '*Tiempo*' i *listSepalWidth3* és la llista de '*Ronda*'.

4.4 Resultats

Una vegada ha acabat amb l'anàlisi del patró, imprimirà per pantalla la informació més rellevant d'aquest, que és el patró, el número de vegades que es repeteix, els instants en els quals es repeteix, les rondes en les quals és du

a terme i el total d'accions realitzades pel jugador (figura 7).

```

El seu patró és:
[8, 3, 2, 9]

Apareix 5 vegades

En els instants: ('minuts:segons')
['02:02', '05:37', '19:00', '20:09', '34:24']

Que pertanyen a les rondes:
[3, 9, 26, 26, 35]

I el total d'accions realitzades pel jugador és: 1109

```

Fig. 7: Resultat de algorisme d'anàlisi d'un patró en concret

5 ALGORISME DE CLASSIFICACIÓ SEMI-SUPERVISAT

5.1 Resum del funcionament

A continuació, hem realitzat un algorisme de classificació semisupervisat basat en un algorisme que varem veure l'any passat en l'assignatura de *Coneixement, raonament i incertesa* el qual aplicava un algorisme classificador de *Naive Bayes*. Aquest algorisme s'alimenta amb l'algorisme descrit en la secció 3, l'algorisme d'anàlisi de patrons general, i extreu una llista en un fitxer .csv amb les probabilitats que tenen cadascun dels patrons de ser positius o negatius segons les etiquetes prèviament seleccionades en l'algorisme de patrons general.

5.2 Naive Bayes

Un classificador de *Naive Bayes* és un classificador basat en el teorema de *Bayes* que diu que per tal de poder classificar un total en positiva o negativa, es pot fer mirant a quina categoria pertanyen cadascuna de les seves parts i amb això acabant calculant el resultat final del total. Per exemple per saber si una frase és positiva o negativa es calcularia cadascuna de les paraules que conté quina possibilitat tenen d'estar en una frase positiva o negativa i amb això podrien determinar si la frase és positiva o negativa (figura 8).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability

Posterior Probability
Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Fig. 8: Funcionament de l'algorisme de Bayes

5.3 Algorisme propi

Per tant, hem intentat recrear un algorisme basat en el teorema de *Bayes* utilitzant les paraules citades anteriorment com a patrons. Per fer això, l'investigador que utilitzi l'eina haurà d'etiquetar les partides com a positives (1) o negatives (0), depenent del seu criteri i recolzant-se en les dades que hem extret en format taules i gràfiques, en l'algorisme esmentat prèviament com anàlisi de patrons general (Secció 3). Un cop tenim els patrons etiquetats hem estat fent proves i finalment hem decidit que la millor manera per fer-ho ha estat amb una predicció de LaPlace. Així i tot hem deixat comentat en el codi la predicció normal per si de cas amb grans quantitats de dades fiable obtinguéssim millors resultats. Finalment, com que pensem que etiquetar partides senceres no era molt útil, el predictor acabar predient si els patrons són positius o negatius i et treu un resum dels millors patrons analitzats en les partides que has passat per l'algorisme anterior (Anàlisi de patrons general) i també dels pitjors. També pots buscar un patró concret per saber quantes vegades es repeteix en partides positives i negatives i veure la probabilitat que té de ser un patró positiu o de ser un patró negatiu. A continuació veurem el resultat d'una execució (figura 9).

```

Escriuiu el patró que voleu analitzar en el següent format:(x, x,
x, x),(8, 2, 9)

```

```

-----
El patró (8, 2, 9) té una probabilitat de ser negatiu/positiu
del: -1.6187949489716504 / -1.3860722548467759
Ha aparegut 20 vegades en partides negatives i 45 vegades en
partides positives
Per tant, la predicció sortirà positiva ja que la probabilitat de
positiu es més gran que de negatiu
S'han fet un total de 166 prediccions
Aciertos: 121.0
El patró més negatiu és el: (8, 2, 7) amb una relació 0
Ha aparegut 0 vegades en partides positives i 5 vegades en
partides negatives
El patró més positiu és el: (10, 2, 6) amb una relació
0.1932141528412008
Ha aparegut 7 vegades en partides positives i 0 vegades en
partides negatives

```

Fig. 9: Resultat algorisme de classificació

5.4 Funcionament del algorisme a nivell tècnic

Per realitzar aquest algorisme ens hem basat en un sistema de diccionaris, tuples i comptadors amb els mòduls de *pandas* i *numpy*. El primer que fa aquest algorisme és demanar a l'usuari que introdueixi el patró que vol analitzar, en el cas que sigui algun en concret, si no, simplement pot prémer *enter* i deixar que l'algorisme s'executi sol. Aquest algorisme obre el fitxer .csv guardat amb l'anterior algorisme, l'algorisme d'Anàlisi de patrons general, i genera un diccionari on els valors són tuples. Llegeix tots els valors emmagatzemats en el fitxer .csv i busca la seva clau en un nou diccionari. Si la clau existeix mira si l'entrada que li arriba del fitxer .csv és positiva o negativa (està etiquetada amb un 1 o un 0). En cas que sigui positiva afegeix a la primera posició de la tupla el valor que té en l'arxiu .csv i si és negativa, l'afegeix en la segona posició. En cas que aquesta clau no existeixi en el diccionari, la

genera amb el valor que té en el fitxer .csv en la primera posició en cas que sigui positiva o en la segona posició en cas que sigui negativa. A continuació realitza un recompte del total de patrons que hi ha tant positius com negatius.

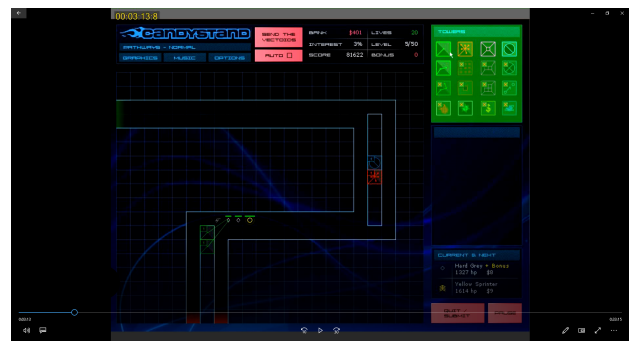
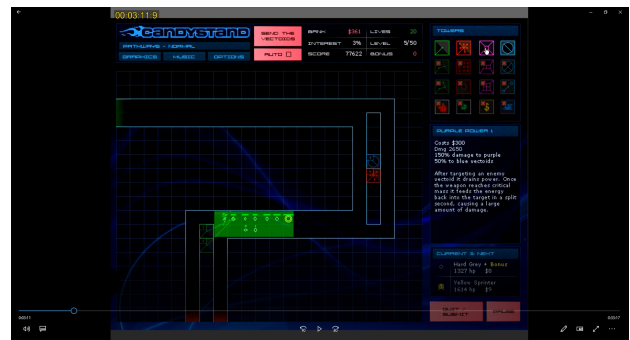
Una vegada tenim el total de patrons que hi ha, marquem la probabilitat de negatius base com a $\log_{10}(n^{\circ} \text{negatius} / \text{total de patrons} + 1)$ i el total de positius base com a $\log_{10}(n^{\circ} \text{positius} / \text{total de patrons} + 1)$. A continuació calculem la probabilitat que té cada patró de ser positiu o negatiu. Això ho hem fet de dues maneres i finalment hem decidit quedar-nos amb el mètode anomenat *Laplace smoothing*. Tot i això hem deixat la manera de treure la probabilitat normal comentada en el codi per si finalment amb grans quantitats de dades funciona de manera més eficaç.

L'algorisme es guarda el patró amb la diferència probabilística més alta de ser positiu i el de ser negatiu, i més endavant ens el mostra. També ens mostra les probabilitats que té el patró que li hem demanat i el número de vegades que es repeteix aquest patró en partides positives i negatives. També podem observar el nombre de prediccions que ha realitzat i el número d'encerts que ha tingut. Les dades de training/test es pot calibrar tot i que ara mateix no està calibrat a causa de la manca de dades i per tant la poca fiabilitat d'aquests resultats.

6 RESULTATS I AVALUACIÓ/REFLEXIÓ SOBRE AQUESTS

Per assegurar-nos que l'aplicació funcionava perfectament hem hagut de demanar un *Ground Truth* a en Lluís, investigador en didàctica de les matemàtiques, i recolzant-nos amb el projecte de en Daniel Calvo, que realitza una edició en el vídeo de la partida per tal de marcar en quina zona està mirant en tot moment el jugador, hem pogut confirmar que els patrons vists en el nostre algorisme coincidien amb el vídeo editat i amb les definicions que tenim en el *Ground Truth*.

A continuació veure'm un exemple de com hem realitzat les comprovacions pertinents. En aquest exemple hem mirat el vídeo i analitzat manualment i després hem buscat la correlació de patrons. Per exemple a continuació veure'm com en l'instant 3:10 comença el patró de mirar la *Information* àrea, mirar la *Gameplay Area* i després mirar la *Tower Selection* àrea, i en el vídeo processat es pot veure perfectament com és el que el jugador ha realitzat. També podem observar a la cantonada d'adalt de tot a l'esquerra el temps on han succeït les accions i en l'apartat de *Game details*, el nivell que és el 5è, tal com diu l'algorisme (figura 10).



El seu patró és:
[3, 2, 8]

Apareix 27 vegades

En els instants: ('minuts:segons')
['00:08', '03:10', '03:24', '04:46', '05:44', '06:08', '06:40', '10:33', '10:51', '11:58', '12:10', '12:24', '12:45', '12:49', '12:52', '12:55', '13:16', '15:42', '20:18', '20:28', '21:14', '22:21', '26:18', '26:20', '26:31', '26:39', '35:53']

Que pertanyen a les rondes:
[0, 5, 5, 8, 10, 10, 11, 17, 17, 19, 19, 19, 20, 20, 20, 20, 20, 23, 27, 27, 27, 28, 30, 30, 30, 30, 35]

I el total d'accions realitzades pel jugador és: 1109

Fig. 10: Resultat de l'algorisme per el patró 3, 2, 8

Així i tot, hi ha un punt on ens hagués agradat aprofundir més però teníem certes limitacions i no ha sigut possible. Aquest punt és en l'algorisme de classificació de patrons. Ens haguera agradat tenir una gran quantitat de dades, de partides positives (d'adults que saben com va aquest videojoc i per tant siguin *masters*) i negatives (de nens de l'edat per la qual està contemplat aquest projecte, que no tinguin experiència en el joc). El principal problema per no tenir més dades és que el *Eye Tracker* no podia ser extret del *Centre de Visió per Computació* i per tant era impossible generar un gran nombre de proves amb el joc amb usuaris que no haguessin estat mai en contacte amb aquest videojoc. Tot i això, l'algorisme està preparat per rebre grans quantitats de dades i ens agradaria poder

observar els resultats si finalment s'acaba implementant a didàctica, ja que considerem que pot ser molt interessant. Encara que tinguéssim poques dades, ja es van poder començar a veure algunes tendències com per exemple que els usuaris que no miraven quins serien els pròxims enemics que vindrien abans de començar les rondes, no arribaven tan lluny com els que sí que ho feien.

7 CONCLUSIONS

Creiem que hem assolit satisfactòriament la majoria dels objectius proposats amb aquest projecte, ja que hem aconseguit generar una eina molt útil per a l'anàlisi de partides del videojoc *Vector Tower Defense II* dedicat al personal d'investigació en didàctica de les matemàtiques. Aquesta eina és fàcil d'executar (figura 11), genera informació tant de forma gràfica com de forma numèrica i es poden fer anàlisis dinàmics a gran velocitat sobre les partides dels diferents jugadors.



Fig. 11: Interfície de la macro en Excel

Al principi del projecte ens vàrem plantejar un conjunt d'objectius principals els quals s'han acabat complint tots. Hem aconseguit fer una eina de fàcil ús per al professorat, ha estat correctament documentada, i la informació que ens aporta és més que la que esperàvem quan vàrem començar aquest projecte. A continuació farem un petit resum pas per pas sobre els objectius marcats i la manera segons la qual han estat assolits.

- En primer lloc, volíem ser capaços d'entendre la informació que ens aportava el *Eye Tracker*, i no només l'hem entès sinó que hem estat capaços de manipular-la de tal manera que generés unes taules i unes gràfiques dinàmiques en un temps d'execució no superior als 20

segons. L'aparença que obté aquest fitxer és molt simple i intuïtiva per al professorat.

- A continuació, volíem fer una anàlisi dels patrons que repetien els jugadors i hem elaborat un algorisme que no només els compti sinó que els etiqueta, et diu en quin instant de temps han estat realitzant i en quina ronda ha estat.
- També volíem fer un classificador semisupervisat per als patrons, el qual hem acabat realitzant mitjançant *Naive Bayes* i *Laplace Smoothing*. En aquest punt ha faltat tenir una gran base de dades amb moltes partides, però tot i això ha sortit un algorisme que, segons la classificació realitzada anteriorment, et retorna les probabilitats de cadascun dels patrons de ser positiu o negatiu.

Per tant, crec que podríem concloure aquest article dient que el treball ha estat un gran èxit, que l'única mancança i problema que hem tingut al llarg del treball ha estat la poca quantitat de dades i la impossibilitat de treure el *Eye Tracker* del *Centre de Visió per Computador* però que tot i així en quant vàrem detectar aquest problema, la reacció va ser molt bona i ens vàrem focalitzar en altres aspectes del projecte com acabar de perfilar les taules dinàmiques i les gràfiques i fer una anàlisi no només de les *fixations* sinó també dels *clicks*.

A partir d'aquí, aquest projecte és només l'inici d'alguna cosa molt gran que es pot aconseguir. Crec que la millor manera de seguir amb aquest treball seria en primer lloc fer una primera prova de camp amb alumnes per veure com funciona l'algorisme de classificació amb moltes dades fiables. Una vegada acabat això, és podria seguir expandint en diferents direccions com per exemple:

- Realitzar els algorismes de classificació e identificació dels patrons no només amb les *fixations* sinó també amb els *clicks*
- Generar un mapa amb les *Saccades*, que són els moviments de vista bruscs, dels jugadors en cadascuna de les partides.
- Investigar més sobre l'algorisme de classificació per tal de fer que no sigui binari.
- Realitzar una interfície o convertir el treball en una aplicació per tal que sigui més fàcilment manipulable.
- El projecte no és tancat ni molt menys, per tant qual-sevol nova idea que tingués el professorat o el desenvolupador es podria dur a terme.

AGRAÏMENTS

Moltes gràcies a la meua tutora, Aura Hernández, per ajudar-me i guiar-me al llarg del projecte, sobretot en els moments més durs quan no sabia per on continuar. Tot això ha estat possible també gràcies a en Lluís, investigador en didàctica de les matemàtiques, del departament de Didàctica de la matemàtica i les ciències experimentals, que també ens ha pogut anar corregint sobre que serà útil en un

futur i on havíem de posar especial èmfasi. També m'han ajudat molt la meua família i els meus amics aportant-me idees sobre com gestionar el treball i la meua parella sobretot per la seva paciència i la seva ajuda a l'hora de l'expressió escrita. També cal agrair la seva gran feina al Daniel Calvo, que va generar els vídeos editats i ens han ajudat a contrastar que els resultats extrets són verídics i coherents.

BIBLIOGRAFIA

- 1 Baldrich i Caselles, R. (2018). Classificació no supervisada. Coneixement, raonament i incertesa.
- 2 Cabrera, A. F. i Osuna, S. H. (2017). Design and implementation of a dynamic spell checker for the intelligent tutoring system, ELE-TUTORA. Revista Signos, 50(95), 385.
- 3 Wikipedia (2019). Naive Bayes classifier. Recuperat el 25 de Maig de 2019 de: https://en.wikipedia.org/wiki/Naive_Bayes_classifier.
- 4 Zadrozny, B. i Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Icml* (Vol. 1, pp. 609-616).
- 5 <http://en.wikibooks.org/wiki/LaTeX>
- 6 <https://www.overleaf.com>
- 7 Hernández-Sabaté, A., Albarracín, L., Calvo, D., Gorgorió, N. (2016, December). EyeMath: Identifying mathematics problem solving processes in a RTS video game. In *International Conference on Games and Learning Alliance* (pp. 50-59). Springer, Cham.
- 8 Hernández-Sabaté, A., Joanpere, M., Gorgorió, N., Albarracín, L. (2015). Mathematics learning opportunities when playing a tower defense game. *International Journal of Serious Games*, 2(4), 57-71.

APÈNDIX

1.1 Com utilitzar correctament l'eina

Per començar, haurem d'obrir el fitxer *Macro.xlsm* i ens trobarem amb una sola fulla d'excel, bloquejada, amb un botó central i unes instruccions. La única acció que ha de prendre l'usuari final del treball és apretar el botó central. Un cop apretat el botó central, ens trobarem amb que ha generat un fitxer *.xlsx* amb 3 fulls els quals són: un full de gràfiques, un full de taules i un amb la *Raw Data*. És molt important que no es manipuli el full amb la *Raw Data* sota ningun concepte però en els altres 2 fulls es pot fer el que es vulgui, filtrar, generar noves taules, canviar les referències, etc. Una vegada ja haguem manipulat tant com haguem volgut aquest fitxer, és important guardar-lo en la mateixa carpeta que tenim els fitxers de *Python BayesClassifier.py* y *EstudiDePatrons.py*.

Una vegada estigui tot en la mateixa carpeta ja podem començar l'anàlisi amb l'eina de *Python*. Primer obrirem l'editor de *Python* del que disponguem, en el meu cas

ha estat creat amb *Spyder*, i obrirem l'arxiu *EstudiDePatrons.py*. En la línia 14 d'aquest document podem canviar el nom del document, però no hem de canviar el *sheet_name* degut a que ja ha estat prèviament configurat per a que així funcioni.

Una vegada iniciat el primer algorisme de *Python*, ens demanarà que escrivim 1 si ja coneixem el patró o 2 si volem buscar patrons. En cas de que es vulgui executar l'algorisme prèviament esmentat com Anàlisi de patrons general haurem de premer 2 i seguidament la tecla *intro*. Si per el contrari volem executar l'algorisme prèviament esmentat com Anàlisi d'un patró en concret, haurem de premer 1 i seguidament la tecla *intro*.

En el cas que haguem executat l'algorisme d'anàlisi de patrons general el programa ens preguntarà si es una partida positiva o negativa, haurem de premer 0 per negativa o 1 per positiva depenent del que interpretem amb les taules i gràfiques dinàmiques. Acte seguit ens preguntarà si volem analitzar patrons de 3 o 4 moviments, haurem de escriure 3 o 4 i acte seguit premer la tecla *intro*. Un cop acabada aquesta selecció l'algorisme començarà a executar-se i crearà el fitxer *.csv* en la mateixa carpeta en el qual guardarà l'informació dels diferents patrons. En el cas de que ja hi hagués un fitxer *.csv* amb aquest nom, el seguiria omplint des de l'última fila que tingués. Per tant, contra més partides analitzem d'aquesta manera, de més dades disposarà l'algorisme de classificació i per tant més fiables seran els resultats.

En el cas que haguem executat l'algorisme d'anàlisi d'un patró en concret el programa ens preguntarà de quants elements està compost el patró que estem buscant. Una vegada introduït el número d'elements, ens demanarà tantes vegades com elements tingui el patró que introduïm un número seguint la llegenda que tenim en pantalla. A continuació l'algorisme s'executarà i et treurà l'informació exacte d'aquell patró que estem evaluant.