Automatic classification of actions in a RTS videogame

Mohamed Yassin El Hajouji El Hayadi

Resum–Els videojocs són una manera entretinguda de passar el temps però també poden oferir-nos bones oportunitats d'aprenentatge on els jugadors hauran d'utilitzar accions planificades per poder guanyar. En aquest treball es farà una conversió de dades qualitatives, obtingudes per un *eye-tracker*, a dades quantitatives per poder classificar les accions que fan els jugadors en un videojoc *RTS* comercial. L'objectiu es classificar automàticament els diferents tipus d'accions segons el seu potencial educatiu.

Paraules clau– videojoc, RTS, vector tower defense II, classificació automàtica, accions, oportunitats d'aprenentatge, potencial educatiu, Naive Bayes, SVM, eye-tracker

Abstract– Video games are an entertaining pastime but they can also provide us with good learning opportunities, as players must perform clever actions in order to surpass them. In this paper a conversion from qualitative data, obtained by an eye-tracker, to quantitative data is done in order to classify the actions performed by the players in a commercial RTS videogame. The objective is to be able to automatically classify the different types of actions by their educative potential.

Keywords– videogame, RTS, vector tower defense II, Automatic classification, actions, learning opportunities, educative potential, Naive Bayes, SVM, eye-tracker

1 INTRODUCTION

A real-time strategy or RTS is a genre in video games in which players must perform the actions, as the name indicates, in real-time and not in turns, as for example chess. In this paper we will classify player actions obtained from a commercial video game, **Vector Tower Defense II** (CandyStand, 2008).

The game consists in defending the entrance from enemies by placing towers or upgrading them and provides their players with situations where decisions or actions must be made, while the enemies are approaching, in order to achieve the goal. The enemies will show up in rounds and will follow a predetermined path until they reach the entrance. The player must place towers along the outer of this path to destroy the enemies and avoid them from reaching their destination.

We have been asked to analyse the information, obtained by means of an eye-tracker, of students from ten to sixteen years old playing this game in order to be able to classify their actions with strong educational potential. It is important to note that this game is not a game meant to be an educational game, as commercial games are designed for entertainment. The use of these type of games from an educative approach not only comes from their entertaining interest but also because some of this games can be an intellectual challenge to the players [1].

As stated before the information to analyse is obtained by means of an eye-tracker system. This system provides the movements of the eye, the position of the screen the player is focusing at and the clicks (s)he makes during the play. This data is processed and handed to us with information such as the region of the game the player is looking at. This qualitative data must go through a previous process as the information comes in a format convenient for the eye-tracker system and not to be analysed.

2 OBJECTIVES

As stated previously the main goal of this project is to classify the actions performed by the players. We can differentiate three objectives:

1. **Converting qualitative data into quantitative:** This

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

step is required in order to be able to classify and distinguish the different actions. We have to take in count the two different characteristics that can differentiate an action:

- **Pedagogical:** The action differentiates itself from others by its educative potential. This information will be provided by the teacher of the students as he will decide which actions have this potential.
- **Computational:** The action differentiates itself from others by its temporal features.

2. **Automatic classification of the actions by their type:** Once we know which types of actions to classify and what features will be distinguishing them we can continue on to deciding which type of algorithms we will be using to classify them. Since the types of actions are provided by the teacher the algorithms used will be supervised learning algorithms.

3. **Validation:** The results obtained will be analysed in order to evaluate the success of the algorithms, obtain conclusions and seek ways to improve them.

3 State of the art

This project uses the conclusions made from two previous works in the same line of obtaining learning opportunities when playing Vector Tower Defense II.

The first one proposes a qualitative study of the mathematical learning opportunities this videogame can offer [1].

The second one combines the player actions done during the play with the position the player is looking at extracted from an eye-tracker system [2].

3.1 Classification

As stated, supervised learning algorithms will be used. These algorithms use data already labelled by an expert in order to learn how to label new data. Many of them have already been used to classify human actions. The most popular ones being Decision Trees, k-Nearest Neighbors, Naïve Bayes, Support Vector Machine and Neural Network although other algorithms such as Hidden Markov Model and Multiclass Logistic Regression have also been used [3]. In this project we have decided using Naive Bayes and Support Vector Machine as they require small time to learn [4].

4 Previous considerations

In order to explain how the results have been obtained each step will be explained inside the context of the objectives previously explained (Section 2). But before an explanation about which data we have and what makes an action is needed.

4.1 Preliminary Data

We have been provided two types of qualitative data. The raw data of the games of each player and data about where the desired actions of two players could be.

- **Gameplay Information:** We have the raw data of each player obtained from the processed information obtained by the eye-tracker system.

  The eye-tracker detects the clicks made into the game and the movements of the eye of the player such as:

  - Blinks
  - Saccades: quick movements of the eye made to change focus.
  - Fixations: the player is observing a region of the game, these will be the only ones with the area of interest cell filled.

  Then this data gets processed to obtain information such as the region of the game the player is looking at.

  After that the information was provided to us in a table inside a .txt file. The table contains 45 columns and each row represents an event detected by the system like a movement made by the eye or a click. This table contains the information needed in order to obtain an action made by the player but since it also contains irrelevant information and the format was inconvenient a filtering and extraction process must be made.

- **Teacher data:** This information was provided by the teacher and indicates to us which types of actions have the educational potential to be classified and where they could be. The format of this information was provided in a table in Word with the name of the student, the start time and the end time of the action and the type of action made by the student in each row. This qualitative data was again not convenient for a classification process as it must be associated with the actions performed in the time span that each row of the table indicates.

4.2 Defining actions

We consider an action as the sequence of steps made by a player in order to reach a certain goal. Since the information we have comes from eye-tracking system we will define a step as the region of the game the player is looking at a certain moment. With this definition we have that an action is the sequence of regions the player has observed in order to reach a goal.

Therefore with the information provided by the eye-tracking system we obtain 11 areas of interest: Game Details, Gameplay Area, Information, Overall Area, Pause, Quit, Sell, Send Vectoids(enemies), Tower Selection, Upcoming Enemies and Upgrade. As we can see in Figure 1.

And from the information provided by the teacher about where could the desired actions be we obtain three types of action:

- **Reading:** In this action the player will read information displayed in the screen.
- **Anticipation:** The player anticipates the movement of the enemies and places a tower or upgrades one, in other words, the player performs an action based on information from the game. These actions have the most educative value of all three.

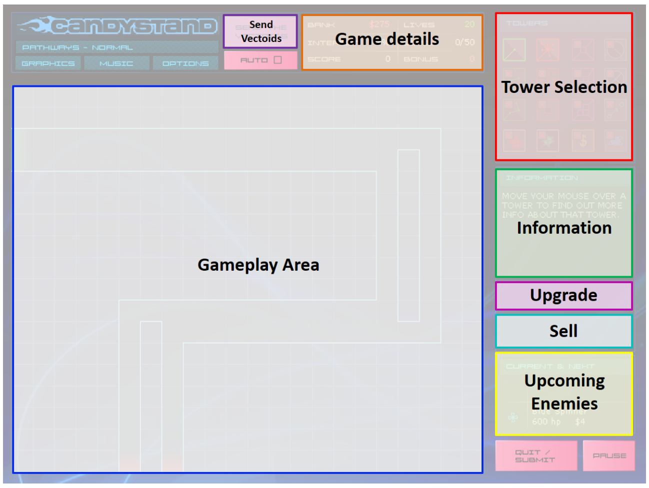

Fig. 1: Areas of interest the player will observe in the game

- **Tower Placing:** The player places a tower in the gameplay area or a bonus item on a tower.

5 Methodology

As previously seen we have the information distributed in two different files and this information must go through a process in order to transform this qualitative data into quantitative data. Therefore two processes will be performed alongside the two first objectives already explained (Section 2).

5.1 Forming an Action

The first step is to define the actions by means of a conversion from qualitative to quantitative. As explained, the player data provided by the eye-tracking system contains extra information and is based by its own system events and not on actions, this is why we go through four different processes. As we can see in Figure 2.

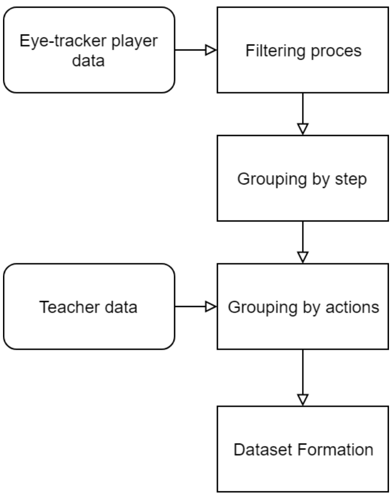

Fig. 2: The process in order to form an action

We will be focusing mainly on four columns of the forty five provided in the player data. The three columns indicating the start time,the end time and duration of an event and the column indicating the Area of Interest the player is looking at. By focusing on the events that contain an Area of Interest columns we filter out all the other events we do not need in order to form an action.

5.1.1 Filtering

First we perform a filtering process. In this process we fix three inconveniences from the player data file:

1. **Irrelevant Areas of Interest:** These areas are removed from the Area column so the following steps do not take them in consideration. These areas are Send Vectoids, Quit and Pause. (Figure 1)
2. **Click information:** A click made by the player is not be considered a step of an action but this information is useful since in order to place a tower, a boost or perform an upgrade a click is needed. This is why in order to add this information in the data-set we add it to the Area column.
3. **Start and end time of an event:** Each event has its start time and end time representing when the event has started and ended. This time is based on trials hence the time-stamps reset to zero in every trial and are not continuous like the teacher data is is.

5.1.2 Grouping by Step

Once the filtering process has placed all relevant information in the Area column when can proceed with the first grouping process. We group the events based by their Area column in order to form the step that will form an action. Also the time information is going to be updated so the data in consistent. This process is needed considering that fixations on areas performed by the player can be interrupted by other events such us blinks, saccades and clicks. The eye-tracking system separates this information in different events but this behaviour is inconvenient for our purpose as the player may still be focusing on the same area even if (s)he blinks.

5.1.3 Grouping by Action

At this point each row of the data represents an Area of Interest of the game or a click made by the user. Meaning that each row represents either a step of an action or a click.

The next step is grouping these steps in order to form an action to classify. We can distinguish two types of actions to form:

- **Actions defined by the teacher data:** As stated previously each row of the teacher data has a start and an end time associated to an action. We use this information in order to group which of the steps will form an action. At the same time we label the action with the information provided by the teacher.
- **Other actions:** The other actions that are not defined by the teacher data do not have a predefined start or end time of formation therefore we are going to define a way of grouping the steps to be able to form an action.

Steps, like actions, have a starting and an ending time.

Trial	...	Category	StartTime_ms_	EndTime_ms_	...	AOIName
Trial001		Fixation	0	8.5		Send Vectoids
Trial001		Fixation	9.7	184.8		Quit
Trial002		Fixation	0.5	8.5		Pause
Trial002		Left Click	8.6	-		-

⇓

Trial	...	Category	StartTime_ms_	EndTime_ms_	...	AOIName
Trial001		Fixation	0	8.5		-
Trial001		Fixation	9.7	184.8		-
Trial002		Fixation	185.3	193.3		-
Trial002		Left Click	193.4	193.4		Left Click

Table 1: FILTERING PROCESS

Trial	...	Category	StartTime_ms_	EndTime_ms_	DurationTime_ms_	...	AOIName
Trial001		Fixation	0	8.5	8.5		Overall Area
Trial001		Fixation	9.7	184.8	175.1		Overall Area
Trial001		Fixation	184.8	223.8	39		Tower Selection
Trial001		Fixation	223.8	225	1.2		Tower Selection

⇓

Trial	...	Category	StartTime_ms_	EndTime_ms_	DurationTime_ms_	...	AOIName
Trial001		Fixation	0	184.8	184.8		Overall Area
Trial001		Fixation	184.8	225	40.2		Tower Selection

Table 2: GROUPING BY STEP PROCESS

The steps that will form the first type of action are already defined by the teacher data. Any steps ending, starting or containing a teacher data action will form that action, which means if a step span intersects an action span it will be forming part of that action.

On the other hand, the other actions not taken in count by the teacher data can go through two types of formation.

- **Fixed number of steps:** A number of steps for each action will be chosen. Then each step that does not form a teacher defined action will be grouped until the action has reached the number of steps chosen or a teacher data action starts.
- **Variable number of steps:** We will chose certain steps to be the ones that will end an action. In this case each step will be grouped into an action until it reaches this type of step or a step that would be grouped into a teacher data action. These steps that will decide when to end an action will be chosen by their duration.

The idea of selecting which steps will end an action comes from the following observation: There are more steps with a *small* duration than those steps with a *higher* duration. As we can see in Figure 3 there are more steps below the 2 seconds threshold than above.

We think that the reason behind this comes from how actions are carried out by the player. When an action has been performed, (s)he will start considering his/her next approach or observing the results of his/her last action, therefore the steps with higher duration will represents this end of action.

When grouping all the steps, even the ones defined by the teacher data, based on that criteria we obtain that the grouping made by the teacher data nearly matches with this way of grouping steps in most cases (Figure 4).

In conclusion this process will go as follows:

- We will go over all the steps based on their start time.
- Each step will be forming part of an action until one of the 3 conditions take place.
- The conditions being:
  - The step span intersects a teacher data action span.
  - The action has the same number of steps or more than the desired maximun of steps for each action.
  - The step duration is equal or surpasses the defined threshold
- When a condition is met the action will stop acquiring steps and a new action will begin its formation.

5.1.4 Forming the Data-set

Now we have the actions formed but we still have not decided which information to use as features. We use the sum of durations of each step to represent an action.

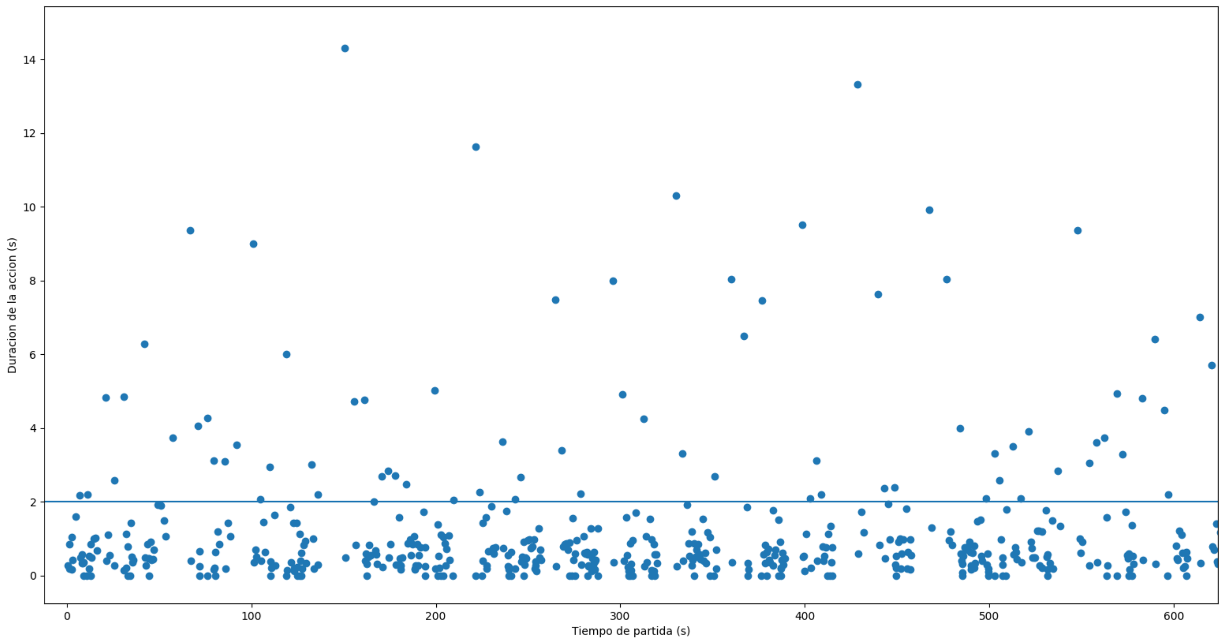

Fig. 3: Each point represents a step made by the user. The line would represent a threshold of two seconds. The points above the line would be the ending steps of the actions formed with the steps under the line.

Trial	...	Category	StartTime_ms_	EndTime_ms_	DurationTime_ms_	...	AOIName
Trial001		Fixation	0	184.8	184.8		Overall Area
Trial001		Fixation	184.8	225	40.2		Tower Selection
Trial001		Fixation	225.6	300.6	75		Overall Area
Trial001		Fixation	300.6	325.6	25		Tower Selection

⇓

Action Type	Trial	...	Category	StartTime_ms_	EndTime_ms_	DurationTime_ms_	...	AOIName
Reading	Trial001		Fixation	0	184.8	184.8		Overall Area
	Trial001		Fixation	184.8	225	40.2		Tower Selection
Anticipation	Trial001		Fixation	225.6	300.6	75		Overall Area
	Trial001		Fixation	300.6	325.6	25		Tower Selection

Table 3: FORMING ACTION PROCESS

With this each action is represented by the amount of time the user looks at each Area of the game. Meaning that the order of the steps performed by the player are not taken in count in the classification.

We think this is a good approach as the same action can be performed in distinct ways and still obtain the same result at the cost of a longer action, considering the number of steps.

For example, the action of placing a tower can take place in 3 different ways, just placing the tower, look at how much currency remains and then placing it or placing a tower and then look how much currency is still left.

All these 3 actions would correspond to the action of placing a tower, as they have the same result, but the order and the number of steps differs from each action, however the time spend on each action should be similar in each action.

In order to be able to distinguish which actions have a strong educative potential from the ones that do not we must add another type of action in to the data-set. This new type will represent the actions that do not have educative potential and we will call them NEPA (Non-Educative Potential Action). This is necessary as the teacher data only provides information about some actions that the teacher thought have educative potential. In other words, the teacher data does not provide information about which are NEPA actions and there may be other actions with educative potential that the teacher may have missed when checking the data.

As to reduce the number of mislabelled actions we will label as NEPA an action in between two actions given by the teacher data, an action without a label. The reasoning behind that comes from the idea that if the teacher did not label an action between two actions it may be because that action had no educative potential as the teacher has already paid attention to that zone.

5.2 Classification Process

The data provided by the teacher specifies which types of action we will be classifying. Therefore we use a supervised learning algorithm.

As already stated, we have decided using Naive Bayes(NB) and Support Vector Machine(SVM) as they re-

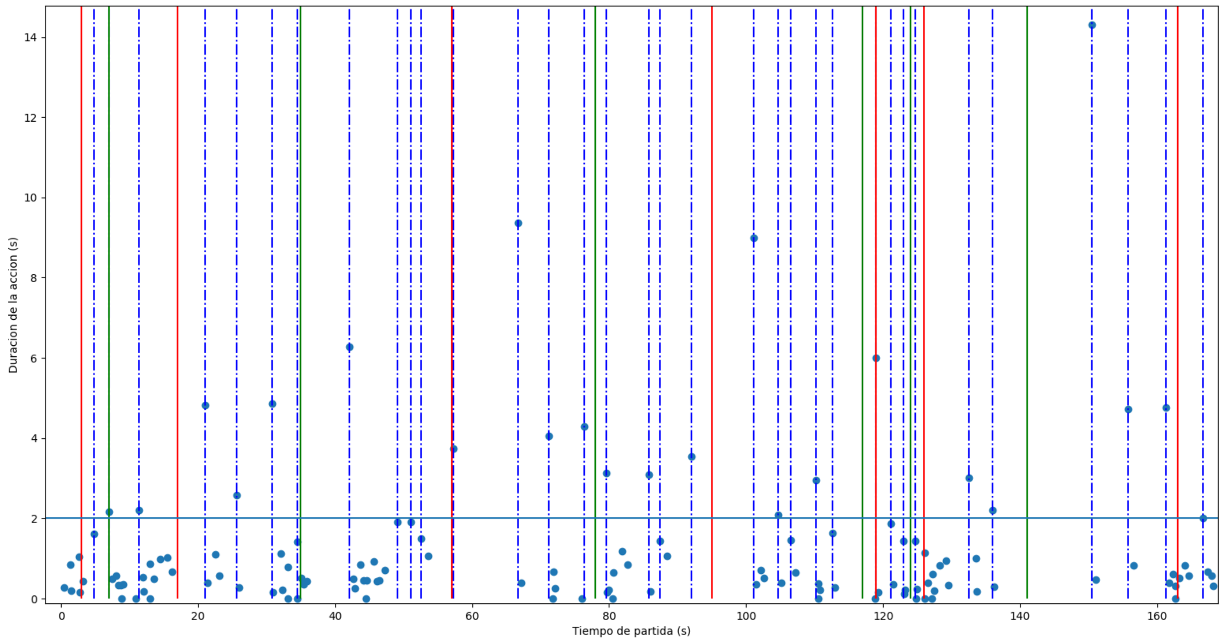

Fig. 4: Each point represents a step made by the user. The horizontal line would represent a threshold of two seconds. Vertical lines represents the start and end of an action. Red vertical lines represent the start of an action of the teacher data and green vertical lines its end. Blue dotted vertical lines are the ones made by the process forming an action with a variable number of steps.

Action Type	Trial	...	Category	StartTime_ms_	EndTime_ms_	DurationTime_ms_	...	AOIName
Reading	Trial001		Fixation	0	184.8	184.8		Overall Area
	Trial001		Fixation	184.8	225	40.2		Tower Selection
Anticipation	Trial001		Fixation	225.6	300.6	75		Overall Area
	Trial001		Fixation	300.6	325.6	25		Tower Selection

⇓

Action Type	Game Details	Gameplay Area	Information	Left Click	Overall Area	Sell	Tower Selection	Upcoming Enemies	Upgrade
Reading	0	0	0	0	184.8	0	40.2	0	0
Anticipation	0	0	0	0	75	0	25	0	0

Table 4: PROCESS OF FORMING THE DATA-SET

quire small time to learn [4].

5.2.1 Naive Bayes

Naive Bayes is an algorithm based on the Bayes Theorem that assumes independence between the features [5]. The idea behind this algorithm is to calculate the probability of a certain feature to appear based on its classification, in order to obtain the probability of the feature to belong to a certain class. With this information the algorithm calculates the probability of a given combination of features to belong to each one of the classes and it will classify the combination of features by labelling it with the class that has the highest probability. In our case the features would be the duration the player observes each Area and the classes to label by would be the four types of action (reading, anticipation, tower placing and NEPA).

5.2.2 Support Vector Machine

The Support Vector Machine is a learning machine algorithm used for classification problems [6]. This algorithm places will represent the data to classify with a point in the space, where to place this point comes from the values the features have. By doing this similar information with similar feature values will be placed next to each other and the information with different features between them will be placed far away from each other.

The algorithm will then try to divide the data with hyperplanes respecting the separation between them so the groups of data can be classified by this dividing line.

Figure 5 shows a graphical example.

The two processes, forming an action and classifying it, have been implemented in Python 2.7 [7] using the Numpy [8] and scikit-learn [9] libraries for both NB [10] algorithm and SVM [11].

6 RESULTS

In this section we will analyse the results obtained by the algorithms in order to obtain conclusion of the effectiveness of the process and possible improvements.

The following results come from a 80/20 partition from the data-set. Meaning that 80% the data was used for training and 20% was used for validating them. The actions are formed using a fixed number of steps, 5 steps (Section 5.1.3).

The data-set is made with the data of one player.

Action Type	Game Details	Gameplay Area	Information	Left Click	Overall Area	Sell	Tower Selection	Upcoming Enemies	Upgrade	SVM	NB
Tower	0.	0.	3.124	0.	0.	0.	0.3137	0.	0.	Reading	Reading

Table 5: TOWER ACTION LABELLED AS READING WITH FEATURES THAT SUGGEST THIS IS A READING ACTION

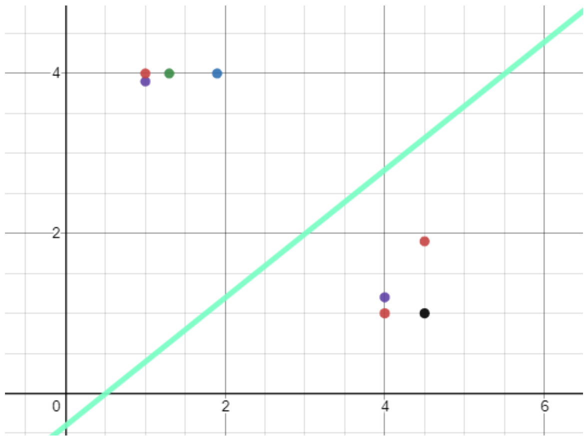

Fig. 5: Simplification of what SVM does. The upper points would represent a class and the lower points another class.

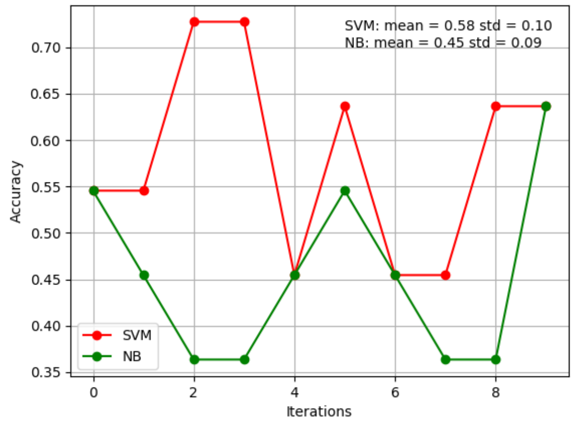

Fig. 6: Accuracy scores of each iteration

The data-set used for this process contains only the labelled information that we posses. Which means we only use actions from the teacher data and the actions labelled as NEPA by us.

Each cell of the confusion matrix represents the number of actions in its classification.

With this we have 52 labelled actions:

- **16 NEPA:** These are the actions added by us following the process explained in Section 5.1.4.
- **36 teacher data:** The actions labelled manually by the teacher:
  - 17 Anticipating
  - 13 Reading
  - 6 Tower placing

With the partition made we obtain 41 actions to be trained with and 11 actions to be tested on. To see how the algorithm performs we select the 11 actions to test on randomly and observe the performance given. We repeat this process ten times and compare the results.

Figure 6 shows the results obtained using this process. As we can see SVM provides much better scores for the same data than NB and overall its performance is higher.

Now we will see the confusion matrix of the highest score of each algorithm in order to understand the reason behind this results.

Firstly we will check the results obtained by the SVM in Table 6.

		Predicted			
		Anticipation	Reading	NEPA	Tower
Actual	Anticipation	3	0	0	0
	Reading	0	2	1	0
	NEPA	0	0	3	0
	Tower	0	1	1	0

Table 6: CONFUSION MATRIX OBTAINED BY THE SVM CLASSIFICATION ALGORITHM.

In this matrix we can see that the algorithm has no problem in classifying Anticipation and NEPA actions instead has very poor results with Tower Placement action.

Upon observing one of the mislabelled tower actions (Table 5) we observed the following. An action that would normally been classified as Reading as the player is observing the Information Area is labelled as Tower selection by the teacher.

Poor results on Tower action classification may be due that this class features are being blurred by the features of Reading class, in other words this class lacks of features or characteristics that would make it special.

We can also observe that the classifier mislabels a Reading and Tower action in favour of NEPA actions. This may be due the big amount of NEPA actions in the data-set and because since these actions were selected from the non labelled data, some of these NEPA actions may be in fact educational potential actions which makes the results given by the algorithm worse.

Now we will check Table 7 which contains the results obtained by the NB.

		Predicted			
		Anticipation	Reading	NEPA	Tower
Actual	Anticipation	5	0	0	0
	Reading	1	2	0	0
	NEPA	0	2	0	0
	Tower	1	0	0	0

Table 7: CONFUSION MATRIX OBTAINED BY THE NB CLASSIFICATION ALGORITHM.

As before the Anticipation action has a high success this may be due the high amount of this type of action on the data-set, also we can observe a Reading and a Tower action are being mislabelled in favour to the Anticipation action, which gives more strength to this idea. On the other hand, this time the NEPA actions are being mislabelled in favour to Reading actions.

Now we will use all the data-set as a training set and use another smaller data-set from another player to test the algorithm.

This smaller data-set contains 13 labelled actions:

- **4 NEPA:** These are the actions added by us following the process explained in Section 5.1.4.
- **9 teacher data:** The actions labelled manually by the teacher:
  - 3 Anticipating
  - 4 Reading
  - 2 Tower placing

We obtain that the accuracy is 0.54 for both classifiers.

When looking at the confusion matrix of the SVM classification in Table 8:

		Predicted			
		Anticipation	Reading	NEPA	Tower
Actual	Anticipation	3	0	0	0
	Reading	0	2	2	0
	NEPA	0	2	2	0
	Tower	1	0	1	0

Table 8: CONFUSION MATRIX OBTAINED BY THE SVM AND USING INFORMATION FROM TWO DIFFERENT PLAYERS.

Just like before the Anticipation actions gives the best results and Reading and Tower also get mislabelled in favour to the NEPA.

And in the NB confusion matrix in Table 9:

		Predicted			
		Anticipation	Reading	NEPA	Tower
Actual	Anticipation	3	0	0	0
	Reading	0	4	0	0
	NEPA	0	4	0	0
	Tower	2	0	0	0

Table 9: CONFUSION MATRIX OBTAINED BY THE NB AND USING INFORMATION FROM TWO DIFFERENT PLAYERS.

The actions when mislabelled get on the most dominant classes, Anticipation and Reading, Anticipation as seen previously is the class with the highest score classification due to the amount of actions in the data-set and how well the features distinguish this class. Reading would be the second most dominant class as it has a decent amount of data and NEPA class, due to being selected from unlabelled data, may have less features that would specialise this class.

7 CONCLUSIONS

Based on the results we can obtain the following conclusions:

- Anticipation actions get labelled correctly pretty often.
- Reading and NEPA actions get miss-labelled between them sometimes.
- Tower placing actions get mislabelled.

Anticipation actions have these good results due to having a decent amount of data and a distinctive characteristic: to perform these actions the player must observe Areas that involve playing rather than area to obtain information.

Reading actions get mislabelled sometimes because in order be able to perform a Reading action the player must select a tower, hence observing a the Tower Selection area of the game or the Gameplay Area in order to select a tower already placed. Also we observed that players do not usually spend a considerable time observing an Area with information as the game still goes on when they perform these type of action.

Using unlabelled data as NEPAs does not help in the process of classifying Reading actions. As actions that would be classified in other classes may get classified in this, therefore classes without a strong characteristic may get mislabelled with this class.

The problem with Tower placing actions lays in the fact that the amount of actions of this type provided by the teacher data is low and that to perform these actions the player must follow similar steps that could be performed in Anticipation or Reading actions, steps as looking the Tower Selection area.

7.1 Further work

In order to improve the results we propose the following:

- **Obtaining more teacher data:** A larger number of actions with balanced amounts of actions of each type would provide much better results as less than 18 actions en each class may be not enough to classify certain classes.
- **NEPA Actions:** By having this type of actions already labelled by the teacher we would avoid the risk of mislabelling an action with educative potential as NEPA therefore reducing the noise this class is producing in our results.
- **Tower Placement:** In order to be able to classify this class a special characteristic that represents it is needed. Information such if a tower is being selected or if the amount of currency has been reduced would be a big help in order to correctly classify this class.

ACKNOWLEDGMENTS

I would like to thank my tutor, Aura Hernández Sabaté for guiding me through this project. Also to my family and friends for all the support and help given to me through my university years.

REFERENCES

[1] HERNÁNDEZ-SABATÉ, Aura, et al. Mathematics learning opportunities when playing a tower defense game. International Journal of Serious Games, 2015, vol. 2, no 4, p. 57-71.

[2] HERNÁNDEZ-SABATÉ, Aura, et al. EyeMath: Identifying mathematics problem solving processes in a RTS video game. En International Conference on Games and Learning Alliance. Springer, Cham, 2016. p. 50-59.

[3] BIN ABDULLAH, Mohd Fikri Azli, et al. Classification algorithms in human activity recognition using smartphones. International Journal of Computer and Information Engineering, 2012, vol. 6, no 77-84, p. 106.

[4] MORAES, Rodrigo; VALIATI, JoãO Francisco; NETO, Wilson P. GaviãO. Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications, 2013, vol. 40, no 2, p. 621-633.

[5] HAND, David J.; YU, Keming. Idiot's Bayes—not so stupid after all?. International statistical review, 2001, vol. 69, no 3, p. 385-398.

[6] CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. Machine learning, 1995, vol. 20, no 3, p. 273-297.

[7] Python Software Foundation. Python Language Reference, version 2.7. https://docs.python.org/2/reference/index.html

[8] OLIPHANT, Travis E. A guide to NumPy. USA: Trelgol Publishing, 2006.

[9] PEDREGOSA, Fabian, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research, 2011, vol. 12, no Oct, p. 2825-2830.

[10] 1.9. Naive Bayes scikit-learn 0.21.1 documentation

https://scikit-learn.org/stable/modules/naive_bayes.html

[11] 1.4. Support Vector Machines scikit-learn 0.21.2 documentation

https://scikit-learn.org/stable/modules/svm.html

APPENDIX

Vídeo	Inici	Final	Comentaris
Víctor	00:03	00:07	Mira la descripció de la torre vermella 1
Víctor	16:43	16:46	Mira si es pot upgradar una torre
Víctor	16:49	17:09	Mira si es pot upgradar una torre

...

Table 10: Teacher data: some Reading actions provided by the teacher

Vídeo	Inici	Final	Comentaris
Víctor	00:17	00:35	Comença seguint els vectoids i va mirant altres zones, quan veu que passaran la torreta, reacciona
Víctor	00:57	01:18	Aquí es veu com apura fins al final per decidir si posa la torre
Víctor	01:35	01:57	Aquí es veu com apura fins al final per decidir si posa la torre

...

Table 11: Teacher data: some Anticipating actions provided by the teacher

Vídeo	Inici	Final	Comentaris
Víctor	06:07	06:18	Posa el bonus verd (augment distància) i busca la millor posició
Víctor	12:19	12:21	Decideix on posar una torre blava 2
Víctor	13:05	13:09	Posa el bonus verd (augment distància) i busca la millor posició

...

Table 12: Teacher data: some Tower actions provided by the teacher

Action	Game Details	Gameplay Area	Information	Left Click	Overall Area	Sell	Tower Selection	Upcoming Enemies	Upgrade
Reading	0	0.4294	2.174	0	0	0	2.1036	0	0
Anticipation	0	0.2147	0	2	16.4242	0	1.7836	0.3964	0
Anticipation	0	3.7409	0	2	20.8353	0	1.3188	0	0
Anticipation	0	0.3964	0	2	24.05	0	2.1802	0	0
Reading	0	0.2111	1.437	0	0.3634	0	1.9853	1.4369	0
Anticipation	0	1.288	0	2	19.7858	0	0.3304	4.03	0
Anticipation	0.5615	0.9909	1.7839	2	26.2575	0	6.847	1.6516	0
Tower	0	0	0	1	0.2147	0	1.1066	0	0
Reading	0	0	3.4155	1	0	0	2.0181	0	0
Anticipation	0	0.2807	0	0	3.0624	0	1.9583	0	0
Anticipation	0	0.5946	0	3	15.9871	0	4.1347	0	0
Reading	0	1.6682	1.8708	2	1.8333	0	0.6276	0	1.3706
Anticipation	0	0.2625	0	2	19.2816	0	1.8089	0	0
Tower	0	0.8242	0.3303	2	8.1933	0	1.8496	0	0
Anticipation	0.8921	0.8092	0	2	20.8475	0	7.9294	0	0
Anticipation	0.7432	4.03	0.2642	1	9.4634	0	0.3469	1.1891	0
Anticipation	0	0	0.6112	2	31.116	0	7.2102	0	0
Reading	0.5945	0	0	1	0	0	0.6276	0	0
Anticipation	0	3.1904	0	2	26.5834	0	0.7432	2.2413	0
Reading	0	0	0	0	6.4081	0	2.1966	0	0
Anticipation	0.6441	1.9328	1.701	1	15.4914	0	6.4413	1.1725	0
Anticipation	0	0	0	0	9.3479	0	0.1817	0	0
Anticipation	0	0	0	0	9.9589	0	0	0	0
Reading	1.0074	0.9417	2.0644	0	3.2699	0	1.4866	0.9581	0
Tower	0	0	3.1214	0	0	0	0.3137	0	0
Tower	0	0	0	0	6.887	0	1.1727	0	0
Anticipation	0	0.1653	0	5	37.8991	0	2.4611	0	0
Anticipation	0.2807	0.8425	0	1	14.8805	0	4.3273	1.2387	0.2148
Reading	0	0	0	0	7.4818	0	0	0	0
Reading	0	2.7415	0.4129	4	8.1887	3.3278	0.7384	1.2885	1.7342
Reading	0	2.3286	0	0	0.9743	0	0	0	0.5285
Reading	0	3.0885	0	5	5.819	1.0901	0	0	0.5285
Tower	2.0645	0	0.4294	0	0.244	0	1.3708	0.7432	0
Reading	0	0	0.2808	1	6.0111	0	0	0	0
Reading	0	0.7139	0	2	5.4143	0	0	1.7674	0
Tower	0	0	1.0074	0	15.5251	0	0	0	0

Table 13: Data-set obtained when the process of forming and action is done