

	Manuel de l'utilisateur PIAD Détection de triche dans les jeux en temps réel		
Activité	PIAD de Master d'Informatique	Type	MU
Objet	Manuel de l'utilisateur PIAD :Détection de triche dans les jeux en temps réel		
Destinataires	M. Pierre-Henri Willemin	Pages	5

Version	Statut	Auteur	Date	Description
1.0	Création	Larroque Stephen	09/05/2013	Création
1.1	MAJ	Larroque Stephen	09/05/2013	Ajout de l'usage en module python

Table des matières

1 Introduction.....	3
2 Usage en ligne de commande.....	3
2.1 Phase 1 : Apprentissage des paramètres.....	3
2.2 Phase 2 : Détection d'anomalies.....	4
3 Usage en module Python.....	4
4 GUI Notebook.....	5

1 INTRODUCTION

OACS (Open Anti-Cheat System) est un système de détection comportementale de triche opensource.

Le système fonctionne en modélisant le comportement des joueurs honnêtes, puis en déduit les anomalies qui sont probablement des cas de triche.

Pour les utilisateurs expérimentés, le système fonctionne en utilisant des sous-modules, que vous pouvez charger et configurer à votre convenance (plus d'informations dans le manuel du développeur). Si vous ne connaissez encore pas bien le logiciel, vous pouvez simplement utiliser la configuration fournie.

2 USAGE EN LIGNE DE COMMANDE

L'usage de OACS se fait en 2 étapes : tout d'abord il faut apprendre les paramètres à partir d'un jeu de données, puis ensuite avec ces paramètres vous pourrez détecter les anomalies.

N'oubliez pas qu'à tout moment, vous pouvez utiliser la commande suivante pour afficher l'usage de base du système :

```
python oacs.py -help
```

2.1 Phase 1 : Apprentissage des paramètres

Pour apprendre les paramètres, vous aurez besoin :

- de spécifier votre fichier de configuration
- d'un fichier data.txt
- d'un fichier types.txt
- du fichier en sortie pour enregistrer paramètres (parameters.txt par défaut)

Ensuite, il suffit de lancer OACS en mode apprentissage avec l'argument `--learn` :

```
python oacs.py --learn -c config.json --datafile data.txt  
--typesfile types.txt -p parameters.txt
```

Cela lancera OACS en mode apprentissage sur le jeu de données spécifié. Ensuite il suffit d'attendre un peu le temps que l'apprentissage se termine (proportionnel à la taille du jeu de données data.txt en entrées), et OACS enregistrera les paramètres dans le fichier des paramètres (parameters.txt ici).

2.2 Phase 2 : Détection d'anomalies

Une fois les paramètres appris avec le mode apprentissage ou en récupérant un fichier de paramètres auprès d'un autre utilisateur du système, vous pourrez utiliser le système afin de détecter les anomalies.

Pour cela, il suffit de donner les mêmes paramètres que pour l'apprentissage, sauf que le jeu de données data.txt doit être le fichier log généré en temps réel par le jeu.

Vous avez également la possibilité de spécifier une table de correspondance des joueurs pour produire des messages de détection avec davantage d'informations d'identification du joueur (comme l'adresse IP ou le pseudonyme).

Il faut également spécifier un fichier en sortie d'historique des détections (detectionlog).

Voici un exemple de commande :

```
python oacs.py -c config.json --datafile data.txt --typesfile
types.txt -p parameters.txt -pt playerstable.txt -dlog
detectionlog.txt
```

Notez ici qu'il n'y a plus l'argument `-learn`.

Une fois lancé en mode Détection d'anomalie, le système OACS va tourner en boucle, scruter toutes les actions des joueurs et détecter toutes les anomalies qui apparaissent.

Vous pouvez interrompre le programme à tout moment en appuyant sur CTRL + C.

3 USAGE EN MODULE PYTHON

Il est également possible d'utiliser le logiciel OACS complet comme un simple module Python. Vous pouvez donc créer votre propre logiciel en utilisant OACS lui-même comme une brique logiciel (ou n'importe lequel de ses modules d'ailleurs).

Pour cela, il suffit d'utiliser les commandes suivantes dans votre script Python :

```
import oacs.main
runner = oacs.main.main(['--script'])
```

Notez que dans les arguments de `main()` vous pouvez passer n'importe quel argument de ligne de commande.

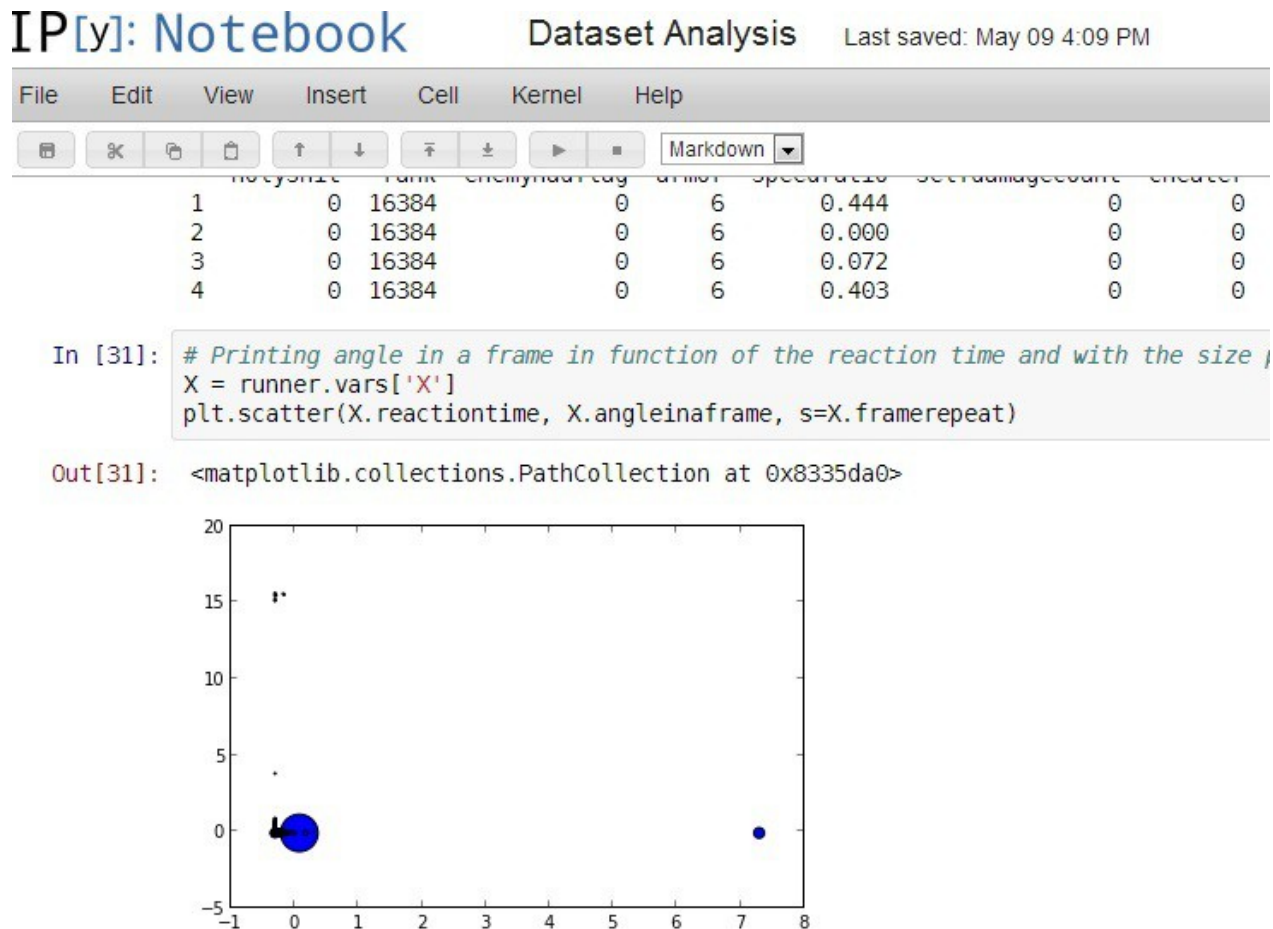
Notez également qu'il est nécessaire d'utiliser l'argument « `--script` » afin d'activer la compatibilité en inclusion dans un script Python.

4 GUI NOTEBOOK

Une interface graphique interactive est également disponible.

Celle-ci est basé sur Ipython Notebook et vous fournit une interface dans laquelle vous pouvez facilement expérimenter et commenter vos essais.

Avec cette interface, des instructions vous sont données pour vous accompagner et vous pouvez juste cliquer sur « Lancer » pour lancer les différentes fonctionnalités de l'application.



Cette interface est également très pratique et extrêmement conseillée pour les utilisateurs expérimentés.