|  |  |  |
| --- | --- | --- |
|  | **NEUMANN JÁNOS**  **INFORMATIKAI KAR** | NIK_cimer.jpg |

|  |  |  |
| --- | --- | --- |
| **OE-NIK**  **2022** | Hallgató neve:  Hallgató törzskönyvi száma: | **Kozma Ferenc**  **T/008876/FI12904/N** |

**Log-alapú context free gyanúgenerálás**

**mesterséges intelligenciára alapozva**

Óbudai Egyetem

Neumann János Informatikai Kar

Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT

FELADATLAP

**Ide majd a kitöltött feladatlapod első oldalát kell beilleszteni!**

**Ide majd a kitöltött és intézetigazgató által aláírt feladatlapod második oldalát kell beilleszteni!**



Neumann János Informatikai Kar

eumann Faculty of Informatics

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, Év. hónap nap.

|  |  |
| --- | --- |
|  | hallgató aláírása |



Neumann János Informatikai Kar

eumann Faculty of Informatics

**KONZULTÁCIÓS NAPLÓ**

**Ide majd az első félév belső konzulens által aláírt projektlabor1. „konzulensi naplót” kell beilleszteni!**

****

Neumann János Informatikai Kar

eumann Faculty of Informatics

**KONZULTÁCIÓS NAPLÓ**

**Ide majd az második félév belső konzulens által aláírt projektlabor2. „konzulensi naplót” kell beilleszteni!**

****

Neumann János Informatikai Kar

eumann Faculty of Informatics

**KONZULTÁCIÓS NAPLÓ**

**Ide majd az harmadik félév belső konzulens által aláírt projektlabor3. „konzulensi naplót” kell beilleszteni!**

****

Neumann János Informatikai Kar

eumann Faculty of Informatics

**KONZULTÁCIÓS NAPLÓ**

**Ide majd az negyedik félév belső konzulens által aláírt záródolgozati porjekt. „konzulensi naplót” kell beilleszteni!**

Tartalomjegyzék

[1. BEVEZETÉS 11](#_Toc103183658)

[1.1. Problémafelvetések 11](#_Toc103183659)

[1.2. Célkitűzések 11](#_Toc103183660)

[1.3. Feladatok 11](#_Toc103183661)

[1.4. Célcsoportok és hasznosság 12](#_Toc103183662)

[1.5. Motiváció 12](#_Toc103183663)

[1.6. Dolgozat szerkezetéről 12](#_Toc103183664)

[2. ELMÉLETI HÁTTÉR 14](#_Toc103183665)

[2.1 Gyanúgenerálás és logelemzés 14](#_Toc103183666)

[2.2 eSport 14](#_Toc103183667)

[2.3 Counter Strike Gloval Offensive 15](#_Toc103183668)

[2.4 Reverse engineering 18](#_Toc103183669)

[2.5 Osztályozás: True vs False és Positive vs Negatív 19](#_Toc103183670)

[3. CSALÁSOK 19](#_Toc103183671)

[3.1. Aimbot 19](#_Toc103183672)

[3.1.1. Aimbot elemzése gyanúgenerálással 20](#_Toc103183673)

[3.1.2. Aimbot True Positive és True Negativ 23](#_Toc103183674)

[3.2. Speedhack 23](#_Toc103183675)

[3.3. Wall Hack 23](#_Toc103183676)

[4. JAVASLAT A GYANÚGENERÁLÁS HASZNÁLATÁRA 24](#_Toc103183677)

[5. VITA 24](#_Toc103183678)

[6. KÖVETKEZTETÉS 24](#_Toc103183679)

[7. MESTERSÉGES INTELLIGENCIA JÖVŐJE A CSALÓK DETEKTÁLÁSÁBAN 24](#_Toc103183680)

[8. ÖSSZEGZÉS 24](#_Toc103183681)

[9. SUMMARY 24](#_Toc103183682)

[10. Mellékletek 25](#_Toc103183683)

[10.1. Rövidítések és fogalomtár 25](#_Toc103183684)

[10.2. Ábrajegyzék 26](#_Toc103183685)

[10.3. Irodalomjegyzék 26](#_Toc103183686)

## BEVEZETÉS

### Problémafelvetések

Mai világban elterjedtek az interneten játszató többszemélyes játékok (pl: CSGO, LOL, FIFA, Hearthstone).[[1]](#footnote-1) Az internetes játékok mögé önálló iparág fejlődött évi több milliárd dolláros bevétellel, ami a számítások szerint 2021-re elérte a $23,817m dollárt.[[2]](#footnote-2) Ezt a felhasználói bázist csalók is kihasználják[[3]](#footnote-3) pl. a könnyű pénzszerzési lehetőség érdekében, megkeserítve a békés, kikapcsolódásra vágyó játékosok életét. Mai napig folyamatosan próbálkoznak visszaszorítani a csalásokat mindenféle szoftveres megoldással (pl: VAC, PunkBuster, VALORANT Anti-Cheat), de még mindig nem teljes a siker, s talán soha nem is lehet teljes sikerről beszélni.[[4]](#footnote-4) Így fel lehet tenni a kérdést, hogy van-e megoldás a csalások minimalizálására/visszaszorítására?

### Célkitűzések

Célom bemutatni a log-bejegyzésekből kinyerhető nyers adatokból (pl.: játékosonkénti logbejegyzések, koordináta pontok, játkosok közti kapcsolatok) a gyanúgenerálás adta lehetőségekkel milyen következtetéseket (pl.: gyanús játékos, királycsináló játékos, játék kombinatorikai tere, gyanús logbejegyzések) tudtam levonni a játékosokról.

### Feladatok

Először is találnom kell egy rendszert, amit jól ismerek és szívesen elemzek. Ezért esett a választásom a játékszoftverekre. Utána felderítem, hogy létezik-e hozzá logolási funkció, ha esetleg nem, hogyan tudok log-bejegyzéseket kinyerni a játékszoftverből. Ezután a kinyerhető nyers adatot kezelhetővé teszem pl. Excelben. Végül az Excel és a gyanúgenerálás módszerének a segítségével a nyers log-adatból értelmezhető információt készítek, ideális esetben ezen gyanú-indexeket a játékba visszacsatolhatóvá teszem a játékfejlesztőkkel együtt.

### Célcsoportok és hasznosság

Mindenkinek ajánlom az automatizált, objektív gyanúgenerálás témakörét, akit érdekel az eSport világa. Főleg a Counter Strike rajongókat tervezem megszólítani, de szerintem más játékot játszóknak is ötletet adhat a gyanúgenerálás ezen levezetési/felhasználási módja. Továbbá játékfejlesztőknek is érdemes lehet elolvasni a szakdolgozatot, mert úgy gondolom, megfontolandó ötlet, hogy már a játék tervezése során a gyanúgenerálást is beletervezik a játék motorjába, főleg, ha többszemélyes funkcióval is rendelkezik a szóban forgó játék.

### Motiváció

Kiberbiztonság szakra járok, ill. érdekelnek a számítógépes játékok, és a munkán során sok log-bejegyzéssel találkozom, amit tudnom kell értelmezni. Azért a gyanúgenerálást választottam elemzési módszernek, mert hatékonyan lehet implementálni pl. az Excel programba (vö. Solver), így az Excel vizualizációs képességeinek köszönhetően könnyedén be tudom mutatni a logokból kinyert információt is.

### Dolgozat szerkezetéről

Amivel foglalkozni fogok a dolgozatban az a log-elemzés gyanúgenerálás segítségével és hogy az Excel vizualizációkkal mit tudok elmondani az eredményről. Továbbá bemutatom az elterjedtebb csalásoknak a technikai hátterét hogyan lehet előhozni őket a Counter Strike játékban. Más játékokkal azért nem foglalkozom, mert az a dolgozat terjedelmi határain túlmutatna.

Mivel a gyanúgenerálás egy adatelemzési módszer nem szoftverfejlesztési módszertan, ezért nem fogok foglalkozni a konkrét csalásnak a kódelemzésével, új csalás létrehozásával, saját csalás elleni detektor fejlesztésével, ill. a dolgozatnak nem része a lineáris programozás részletes matematikai bemutatása sem.

## ELMÉLETI HÁTTÉR

### Gyanúgenerálás és logelemzés

*” Eltekintve egy komoly nyelvészeti bizonyítástól, nagy valószínűséggel nyugodtan kijelenthető, hogy az emberiség legősibb, ösztönösen adott, mindenkor rendelkezésre álló gondolata a hasonlóság gondolata (volt).”[[5]](#footnote-5)*

Az idézetből látható a gyanúgenerálás alapja a hasonlóság gondolata, ami már a számítógépes világ előtt is érdekelte az emberiséget. De a hasonlóságnak nem az a vetülete releváns a gyanúgenerálásban, amit például a geometria (háromszögek hasonlósága) vagy a genetika (részleges azonosság) vizsgál, hanem az optimalizálható, automatizálható, antidiszkriminatív matematikai apparátus.

*„Az alábbiakban bemutatásra kerül a leggyakrabban oktatási célra használt modell-demonstráció annak érdekében, hogy a Hasonlóságelemzés, mint olyan a mérnöki kihívások kapcsán azonnal részévé válhasson az Olvasó értelmezési intervallumának.”[[6]](#footnote-6)*

A dolgozatban látható, hogy oktatási és mérnöki használati célokon kívül, loglemezésre és esportoló játékosok viselkedés elemzésére is használható.

### eSport

A számítógépes játékok a 70-es években kezdtek megjelenni. Ez időben kevés embernek volt számítógépe, így a játékok iránti érdeklődés a 90-es években ugrott meg a személyi számítógépek elterjedésével (valamiféle idősoros statisztika a számítógépes játékok forgalmáról? - ábra?). Versenyek már ezekben az időkben is voltak, de lokális hálózaton kevés érdeklődővel, mivel ez az egész téma kevésbé volt a köztudatban. A komolyabb internetes versenyek a 2000-es évektől indultak a szélessávú internet elterjedésével. Mostanra egy egész iparág felépült a játékok és a játékosok köré. Ezt mutatja a 2021es DOTA II nyereményalapja $40,018,400.00 dollár volt, így egy mai eSportoló kereshet annyit, mint egy fizikai élsportoló.[[7]](#footnote-7)

A legelterjedtebb játék típusok:

* First person shooter (FPS): Counter Strike, PUBG, Call of Duty
* Multiplayer online battle arena (MOBA): League of Legends, Dota 2
* Fighter games: Mortal Kombat 11, Super Smash Bros
* Sport games: FIFA, NBA2K, Madden
* Collectible Card Games (CCG): Hearthstone, Legends of Runeterra

Mostanra felnőtt úgy egy generáció, hogy a fent említett játékok közül legalább eggyel játszott vagy hallott róla. Ebből lehet következtetni, hogy az esportok iránti érdeklődés továbbra is nőni fog az évek során.[[8]](#footnote-8) Egy elemzés szerint 2022-re elérheti a 645 milliós nézettséget világszerte az esport.[[9]](#footnote-9) Ez köszönhető annak, hogy a mostani generáció inkább szeret a számítógépe előtt ülni és annak is, hogy a streaming technológiák folyamatosan fejlődnek.

### Counter Strike Gloval Offensive

A Counter Strike egy csapatalapú FPS játék. 1999-ben jelent meg a Half Life játék kiegészítőjeként. A siker azonnal megmutatkozott, mivel megjelenése után szinte azonnal a legelterjedtebb online számítógépes akciójáték lett. A megjelenése óta eltelt 12 év alatt a világ egyik legismertebb esport játékává nőtte ki magát.[[10]](#footnote-10)

Miről is szól a játék? A játékban a hivatalos meccseken kettő darab öt fős csapat küzd meg egymás ellen. Az egyik csapat az úgynevezett terrorista, akiknek a célja egy bomba elhelyezése és felrobbantása a pályán kijelölt ponton. A másik csapat az úgynevezett counter terrorista, akiknek a célja, hogy megakadályozzák a másik csapatot a bomba elhelyezésében vagy, ha ez nem sikerült, időben hatástalanítsák az elhelyezett bombát. A játék elején minden csapat bevásárol fegyverekből és egyéb harci eszközökből. Emiatt a játékban van egy gazdasági rendszer, amiből következik, hogy a pontszerzésért játékban pénzt is kap egy játékos, így fontos ilyen fejjel is átgondolni a csapat stratégiáját. A játékot az a csapat nyeri, aki a 30 körös meccsből 16 kört megnyer.[[11]](#footnote-11),[[12]](#footnote-12)

Mostanra felmerülhet a kérdés, hogy miért pont a Counter Strike játékot választottam? Egyszerűen azért, mert ismerem és legfontosabb a gyanúgenerálás szempontjából, hogy tudom kezelni a fejlesztői konzol felületét, ahol a logolást be lehet állítani.

A Counter Strike a Source Engine grafikus motorra épült. Ezt a motort a Valve fejlesztette 2004-ben. Ez a motor számtalan funkcióval rendelkezik – köztük egy fejlett logolási rendszerrel. Ahhoz, hogy ezt előhívjuk engedélyezni szükséges a konzol felületet, amit a „-console” kapcsolóval tudunk megtenni a játék indítási opciókban vagy a játék beállításokban hozzárendelhetünk gombot is a ki-be kapcsolásához. Ahhoz, hogy mindenféle fejlesztői beállítással tudjunk dolgozni futtassuk az „sv\_cheats 1” parancsot. Ezzel engedélyezzük a csalásokat is így nem kell attól félnünk, hogy a játék csalónak néz minket és esetleg kitilt a játékból. A részletes logbeállításokat az „mp\_logdetail 3” paranccsal tudjuk elérni. Végül a logolást a „log on” paranccsal tudjuk elindítani, ami egy szöveges fájlba elmenti a logbejegyzéseket a „log off” parancs futtatása után.[[13]](#footnote-13),[[14]](#footnote-14),[[15]](#footnote-15)

A tisztított\_LOG.txt file-ban összegyűjtöttem azokat a logbejegyzéseket, amiket elemezni érdemes gyanúgenerálással. A szerver beállítások azért fontosak, mert ha be tudjuk mutatni, hogy egy játékos eltér ettől, az valószínű, hogy csalást használ. Például a „sv\_maxspeed = 320” de mi egy játékosnál 320 fölötti értéket mérünk az a játékos csalást használ. A probléma ott kezdődik a dolgozat szempontjából, amikor a csalásfeltáró szabályokat nem lehet egyszerűen pl. küszöbértékek formájában statikus szakértői rendszerként értelmezni, implementálni, hanem a csalást, mint mintázatot kell tetszőlegesen sok dimenziós log-adattérben felismerni.

Én 10 körös meccseket elemzek, de érdemes elkülöníteni a köröket egymástól, ha körönként akarjuk kiértékelni a játékosokat. Ezért gyűjtöttem ki a „Kör legelső indulása”, „Bevásárlás és kör indulása”, „Kör végi győzelmek” logbejegyzéseket. Ezek semmit sem mondanak egy játékos teljesítményéről, de hasznos, ha a köröket jó be akarjuk határolni. A „Támadások” bejegyzésekkel tudjuk nyomon követni a támadó és a védekező közti eseményt. Megtudjuk, ki honnan támadott kit. A védekező melyik pozíción állt, milyen fegyverrel támadták, mennyi sérülést szenvedett és végül mennyi életereje és pajzsa maradt. Fontos, itt még nem történt pontszerzés. A pontszerzéssel a „Pontszerzés” bejegyzések foglalkoznak. Ez hasonló a „Támadások” bejegyzésekhez annyi különbséggel, hogy a végén a pontszerzés típusát tudjuk meg. A „semmi” típus az egy sima pont. A „headshot” az a fejen történő találat, ami azonnali pontszerzést ad a védekező pedig azonnal kiesik abból a körből. A „penetrated” az egy objektumon történő átlövés következtében szerzett pont. A „throughsmoke” füstön történő átlövés következtében szerzett pont. Az „attackerblind” pedig vakon történő pontszerzés. Érdemes megjegyezni, hogy ebben a három esetben a játékos nem láthatta az ellenfelet, mégis így szerzett pontot. Tehát csaló lenne? Nem tudhatjuk, mert előfordulhat véletlen, plusz tapasztaltabb játékosok ismerik az elterjedtebb pozíciókat így vakon is célba találhatnak. Utolsó a „noscope”, ami annyit jelent, hogy mesterlövész puska használatakor nem használta a távcsövet, de mégis célba talált. Ezeket a pontszerzési típusokat lehet variálni tehát előfordulhat egy „headshot penetrated noscope throughsmoke attackerblind” típusú pont. A „Harci eszközök” bejegyzések az egyéb eszközökkel használatát mutatja. Megtudhatjuk: ki, mit használt és az eszköz melyik koordináta pontot érvényesült. Végül a „Disconnect” bejegyzés az, ami lezárja a logfilet.[[16]](#footnote-16)

### Reverse engineering

Csalás készítéséhez nyilván ismerni szükséges a számítógép felépítését. A legfontosabb a processzor működésének ismerete. A processzor a számítógép agya, itt történnek a számítások. Ahhoz, hogy egy processzort működtetni tudjunk, alacsony szintű programozási nyelvvel az assemblyvel parancsokat kell megadni. Szükséges magasabb szintű nyelveket is ismerni: például C, C++, azért, mert a legtöbb játék motort ezekben a nyelvekben írták. Ezekkel a nyelvekkel is el lehet érni a processzort, de már tartamaz hatékonyabb eszközöket: például a ciklusokat és a döntési szerkezeteket.

A játék program működését is tudni kell csalás készítéséhez. Egy mai játék komplex, de ezek a főbb elemei:

* Grafikai elemek
* Hangok
* Bemeneti perifériák kezelése
* Fizika szimulálása
* Játék logika

Ezeket a játék program a számítógép memóriájába tölti a RAM-ba. Ebből következik, hogy szükséges egy eszköz, amivel a memóriába történő írás-olvasást tudjuk nyomon követni. Ezeket a programokat memory scannernek hívjuk. Fontos tudni, mit akarunk keresni. Például, ha a játékos pénzét akarjuk befolyásolni, elképzelhető, hogy ugyan ilyen elnevezésű változót keresünk a memóriában. Miután megtaláltuk és módosítottuk a változót, egyszerűen visszatérünk a játékba és láthatjuk, hogy az kiolvassa az általunk átírt értéket.[[17]](#footnote-17),[[18]](#footnote-18),[[19]](#footnote-19)

### Osztályozás: True vs False és Positive vs Negatív

Szakdolgozatomban a True Positive és a True Negatív elemzést fogom bemutatni. [[20]](#footnote-20)

True Positiv (TP) azt jelenti, hogy tudom ki a csaló és a gyanúgenerálás elemzése is kimutatja, hogy ki a csaló.

True Negatív (TN) azt jelenti, hogy tudom ki a korrekt játékos és a gyanúgenerálás elemzése is kimatatja, hogy rendesen játszik a játékos.

False Positive (FP) azt jelentené, hogy nincsen csaló egyáltalán, de a rendszer mégis csalónak nézi a korrekt játékosokat. Ehhez rengeteg tesztet lenne érdemes futtatni, hogy lássuk, mikor mehet félre a gyanúgenerálás ezért időhiány miatt ezzel nem foglalkozom a szakdolgozatomban.

False Negatív (FN) azt jelentené, hogy hogy van csaló a játékban, de a gyanúgenerálás nem veszi észre, hogy csalás történik. Ehhez mindenféle csalásokat kéne telepíteni és tesztelni, amit jogi dolgok és időhiány miatt most nem fogok bemutatni a szakdolgozatomban. Plusz felvetődhet az a kérdés mi van, ha a csalás nem is segíti a játékost tehát használ csaló programot, de ugyan úgy vagy esetleg még rosszabbul is játszik. Ez a kérdés nagyon messzire vezethet így ennek a feltárása is kimarad a szakdolgozatból.

## CSALÁSOK

### Aimbot

Az aimbot egy célzást segítő eszköz. Ennek a segítségével a játékos mindig célba talál. A csalás úgy van leprogramozva, hogy mindig a másik játékos fejére célozzon, így a headshot pontokat könnyen össze lehet gyűjteni.

A játék konzoljában a „ent\_fire cs\_bot AddOutput "ModelScale 0"” paranccsal lehet beállítani az aimbot-ot. Ez azt csinálja, ha a kurzort a másik játékos modelljére irányítjuk, automatikusan headshot adunk, ezzel pontot szerzünk. Érdemes lehet beállítani a „enable\_skeleton\_draw 1” parancsot, mivel így a program jobban kirajzolja az ellenfél modelljét, így könnyebben észrevehető.[[21]](#footnote-21)

### Aimbot elemzése gyanúgenerálással

Az alábbi ábrán (1.) látható, hogy a domináns pontszerzés típusa a headshot. Nyilván az aimbot csalás használatakor más típus nem fog keletkezni. Láthatjuk, hogy a Csaló elnevezésű játékos szerezte a legtöbb headshot pontot. Továbbá kimagasló értékekkel zárta a meccset a damage és damage\_armor területen. Fontos vajon a damage kimagasló értéke? Igen, mert képzeljünk el egy játékmódot, ahol csak headshot-tal lehet pontot szerezni, ilyenkor nyilván a legtöbb pontot szerző játékosnak csak headshot pontja lesz. Mivel most a játékosunk csalást használt így, szlengesen mondva: telibe találta az ellenfelét, vagyis sok damage pontot szerzett.

A képen asztal látható

Automatikusan generált leírás

1. ábra: Támadás, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, OAM, B2-K13) jelmagyarázat: Minden érték darabban értendő

A következő (2.) ábrán az a feltűnő, hogy nincsen rajta a Csaló nevezetű játékos. Ez azért van, mert a csalást használó játékos olyan gyorsan cselekedett az aimbot miatt, hogy a többi játékosnak esélye sem volt támadással válaszolni. Így a csaló egyszer sem esett ki a körből, nem volt szüksége a védekezésre.

A képen asztal látható

Automatikusan generált leírás

2. ábra: Védekezés, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, OAM, B17-J27) jelmagyarázat: Minden érték darabban értendő

Ezen az (3.) ábrán rangsoroltam az eredményeket. A piros egyes jelenti a leggyanúsabb játékost, a zöld 10-es (rangsorszám a becstelenség versenyben) a legbecsületesebbet. Már ebből az ábrából is láthatjuk, hogy a Csaló játékos öt oszlopban is az első helyen szerepel.

A képen asztal látható

Automatikusan generált leírás

3. ábra: Rangsorolás, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, Solver, A3-H13) jelmagyarázat: Minden érték rangsorban értendő

További elemzéssel láthatjuk (vö. 4. ábra), hogy a játék 648 kombinatorikai térből áll. Tehát 648 szituáció fordult elő ezen a meccsen. Ebből a damage armor az, ahol a legnagyobb a váltás a piros gyanús és zöld becsületes értékek között. Ez annyit jelent, hogy itt derül ki, ki a csaló viszont odafigyeléssel itt lehet a legjobban elbújnia egy csalónak.

A képen asztal látható

Automatikusan generált leírás

4. ábra: Konszolidált kombinatorikai tér, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, Solver, A16-H28) jelmagyarázat: Minden érték gyanúpontban értendő

Ez az (vö. 5. ábra) ábra a határt mutatja, ahol a csaló csalóvá válhat. Itt is szintén a damage armor oszlopra mutat az eredmény. Fontos megjegyezni, hogy a damage oszlopban lévő zöld egyes nem mutat semmit, az Excel program kerekítési hibájából fakadóan lett ilyen színű (1.000 <> 0.9999999).

A képen szöveg, mérőszalag, képernyőkép látható

Automatikusan generált leírás

5. ábra: Egy lépcsős függvény karakterisztikus pontjai, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, Solver, J17-Q26) jelmagyarázat: Minden érték gyanúpontban értendő

Végül az utolsó táblázatban (vö. 6. ábra) láthatjuk az eredményt: ki mennyi gyanúpontot kapott az elemzés során. Láthatjuk, hogy a leggyanúsabb játékos az Elite12, pedig tudjuk, hogy a Csaló játékos az, aki csalást használt. Mi történt? Rosszul számoltunk?

A képen asztal látható

Automatikusan generált leírás

6. ábra: Gyanúbecslés rétegei, forrás (<https://miau.my-x.hu/miau/280/kozma.xlsx>, Solver, A31-K44) jelmagyarázat: Minden érték gyanúpontban értendő

Ilyenkor meg kell vizsgálni, hogy létezik-e a játékosok között úgynevezett „királycsináló”, egy olyan játékos, aki annyira rosszul játszott, hogy miatta torzult az eredmény. Ez annyit jelent, hogy egy becsületes játékos a királycsináló játékos mellett túl jónak tűnik, tehát csalónak hihetjük. Ennek a kiszámításához össze kell hasonlítani a játékosokat a többi játékossal és megállapítani ki volt mindenben rosszabb a másikhoz képest.

A 7. ábrán láthatjuk a királycsináló nélküli gyanúgenerálás eredményét. Jelent esetben az Elite12 a királycsináló és amint kivettük a számításból ki is bukik, hogy valójában a Csaló játékos a csaló.

A képen asztal látható

Automatikusan generált leírás

7. ábra: Gyanúbecslés rétegei királycsináló nélkül, forrás (URL, munkalap, cellatartomány) jelmagyarázat: Minden érték gyanúpontban értendő

### Aimbot True Positive és True Negativ

Látható a 7. ábrából, hogy a Ture Positive sikeres lett mivel a Csaló nevezetű játékost tudatosan állítottam be csalásra és az elemzések is őt mutatták ki csalónak.

A True Negatív is sikeres lett mivel a többi játékost úgy állítottam be, hogy rendesen játszanak és az elemzések is azt mutatják, hogy nem csalók.

A képen asztal látható

Automatikusan generált leírás

8. ábra: Aimbot kontengencia, forrás (URL, munkalap, cellatartomány) jelmagyarázat: Minden érték Főben értendő

### Speedhack

Ez a fejezet fejlesztés alatt.

### Wall Hack

Ez a fejezet fejlesztés alatt.

## JAVASLAT A GYANÚGENERÁLÁS HASZNÁLATÁRA

A VAC a Valve által fejlesztett csaló detektáló szoftver. Úgy működik, mint egy vírusírtó tehát a játék elindításakor a VAC végig szkenneli a számítógépet és ha talál csaló programot eltiltja a játékost a játéktól. [[22]](#footnote-22)

Bár fejlesztők napra készen tartják a VAC-ot, hogy felismerje a legelterjedtebb csaló programokat a hackerek mégis egy lépéssel előbbre járnak. Logikusan következik, hogy csak azt a vírust tudja detektálni a vírusírtó, amit már egyszer leprogramoztak és használtak.

Hogyan lehetne a gyanúgenerálást használni a VAC mellett? Én úgy gondolom az lenne a leghasznosabb megoldás, ha a gyanús játékos számítógépét alaposan átszkenneli a VAC nem csak a játék elején, hanem közben is akár valós időben. Így gyorsabban detektálni lehet a csalót nem kell megvárni amíg a többi játékos jelenti, hogy szabályellenesen játszik a játékos.

## VITA

Ez a fejezet fejlesztés alatt.

## KÖVETKEZTETÉS

Ez a fejezet fejlesztés alatt.

## MESTERSÉGES INTELLIGENCIA JÖVŐJE A CSALÓK DETEKTÁLÁSÁBAN

Nvidia olyan megoldáson dolgozik, aminek a lényege, hogy a pont szerzés előtti utolsó képkockákat elemzi és ez alapján el tudja dönteni a program, hogy csalással történt a pontszerzés vagy rendesen. Ehhez úgy gyűjtöttek adatot, hogy megvizsgáltak kettő akció játékot és három csaló szoftvert. Az eddigi eredmények azt mutatják, hogy használható ez a megoldás.[[23]](#footnote-23)

Létezik olyan megoldás, ami a perifériák bemeneti válaszidejét (input lagg) és egyén perifáriák által gyűjthető biometrikus adatokat (Pl. billentyűzet kezelése) vizsgálja. Ilyen adat gyűjtése során meg lehet állapítani, milyen gyorsan használja egy játékos a perifériákat és ha pontszerzéskor nagyobb eltérés mutatkozik az gyanúra adhat okot. Egyetlen probléma ezzel a megoldással, hogy a mai játékosok már hobbi szinten is közelítenek a profikhoz tehát honnan tudhatjuk, hogy csaló vagy egyszerűen jól játszik? Várhatóan a mesterséges intelligencia képes lesz minél több részletet kielemezni így ez a kérdés nem jelenthet problémát.[[24]](#footnote-24)[[25]](#footnote-25)

Hálózati adatforgalom vizsgálata is elvezethet a csalók detektálásához. Mivel a konkrét csomagot nehéz megfogni a titkosítási eljárások miatt ezért a csomag tulajdonságait vizsgálták például a csomag mérete, iránya, darabszáma. Ez azért működhet mert specifikálva van a játék adatforgalma így lehet következtetéseket levonni a mért és a specifikációban leírt eltérések alapján.[[26]](#footnote-26)

## ÖSSZEGZÉS

Ez a fejezet fejlesztés alatt.

## SUMMARY

Ez a fejezet fejlesztés alatt.

## Mellékletek

### Rövidítések és fogalomtár

|  |  |
| --- | --- |
| aimbot | Akció játékokban célzást segítő eszköz, amivel a program automatikusan az ellenfél felé irányítja a kurzort így nem szükséges semmilyen reflex a játékhoz. |
| ASM | Assembly gép közeli programozási nyelv. |
| attackerblind | Rövid időre megvakul a játékos mivel belenézett a vakító gránátba. |
| Bevásárlás és kör indulása | Játék elején be kell vásárolni a felszerrelésből itt tered ad a játék gazdasági taktikázásra. |
| C/C++ | Programozási nyelvek. Használnak magasabb szintű eszközöket (if, ciklusok) de még mindig lehet velük gép közeli kódot írni. |
| CCG | Collectible Card Games. Ez egy játék típus, ahol gyűjtögetni lehet kártyákat. A kártyák begyűjtése véletlenszerűen történik nem tudhatjuk mi van a pakliban. |
| console | Fejlesztői konzol. Itt lehet játékot mindenféle paranccsal beállítani. |
| CSGO |  |
| Damage |  |
| Disconnect |  |
| DOTA 2 |  |
| ent\_fire cs\_bot AddOutput "ModelScale 0 |  |
| eSport |  |
| Excel Solver |  |
| False Negatív (FN) |  |
| False Positive (FP) |  |
| FIFA |  |
| Fighter games |  |
| FPS |  |
| Gyanúgenerálás |  |
| Hearthstone |  |
| Játék moddolás/kiegészítő |  |
| királycsinálás |  |
| Kör legelső indulása |  |
| Kör végi győzelmek |  |
| LOL |  |
| memory scanner |  |
| MOBA |  |
| mp\_logdetail 3 |  |
| noscope |  |
| penetrated |  |
| PunkBuster |  |
| Reverse Engineering |  |
| Source engine |  |
| Speedhack |  |
| Sport games |  |
| sv\_cheats 1 |  |
| sv\_maxspeed = 320 |  |
| throughsmoke |  |
| True Negatív (TN) |  |
| True Positiv (TP) |  |
| VAC |  |
| VALORANT Anti-Cheat |  |
| Valve |  |
| Wallhack |  |

### Ábrajegyzék

Ez a fejezet fejlesztés alatt.

### Irodalomjegyzék

<https://www.statista.com/outlook/dmo/digital-media/video-games/online-games/worldwide> (megtekintés: 2021.12.08.)

<https://www.forbes.com/sites/nelsongranados/2018/04/30/report-cheating-is-becoming-a-big-problem-in-online-gaming/?sh=1e965a0a7663> (megtekintés: 2021.12.08.)

<https://en.wikipedia.org/wiki/Cheating_in_video_games> (megtekintés: 2021.12.08.)

<https://medium.com/ax1al/introduction-to-game-hacking-fb70e29de60f> (megtekintés: 2021.12.08.)

<https://gamehacking.academy/> (megtekintés: 2021.12.08.)

<https://miau.my-x.hu/miau/280/kozma.log> (megtekintés: 2021.12.08.)

<https://totalcsgo.com/commands> (megtekintés: 2021.12.08.)

<https://developer.valvesoftware.com/wiki/Source> (megtekintés: 2021.12.08.)

<https://en.wikipedia.org/wiki/Source_(game_engine)> (megtekintés: 2021.12.08.)

<https://en.wikipedia.org/wiki/Esports#Internet_live_streaming> (megtekintés: 2021.12.08.)

<https://blog.counter-strike.net/index.php/about/> (megtekintés: 2021.12.08.)

<https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/> (megtekintés: 2021.12.08.)

<https://en.wikipedia.org/wiki/Counter-Strike_(video_game)> (megtekintés: 2021.12.08.)

https://www.esportsearnings.com/tournaments (megtekintés: 2021.12.08.)

https://learn.g2.com/esports (megtekintés: 2021.12.08.)

<https://miau.my-x.hu/miau/196/My-X%20Team_A5%20fuzet_HU_jav.pdf> (megtekintés: 2021.12.08.)

<https://miau.my-x.hu/miau/277/szakdoga_PM_final.pdf> (megtekintés: 2021.12.08.)

<https://www.prosettings.com/csgo-hacks-cheats/> (megtekintés: 2021.12.08.)

1. https://en.wikipedia.org/wiki/Online\_game [↑](#footnote-ref-1)
2. <https://www.statista.com/outlook/dmo/digital-media/video-games/online-games/worldwide> [↑](#footnote-ref-2)
3. <https://www.forbes.com/sites/nelsongranados/2018/04/30/report-cheating-is-becoming-a-big-problem-in-online-gaming/?sh=1e965a0a7663> [↑](#footnote-ref-3)
4. https://www.wired.com/story/kernel-anti-cheat-online-gaming-vulnerabilities/ [↑](#footnote-ref-4)
5. <https://miau.my-x.hu/miau/196/My-X%20Team_A5%20fuzet_HU_jav.pdf> [↑](#footnote-ref-5)
6. <https://miau.my-x.hu/miau/277/szakdoga_PM_final.pdf> [↑](#footnote-ref-6)
7. <https://www.esportsearnings.com/tournaments> [↑](#footnote-ref-7)
8. <https://learn.g2.com/esports> [↑](#footnote-ref-8)
9. https://en.wikipedia.org/wiki/Esports#Internet\_live\_streaming [↑](#footnote-ref-9)
10. <https://blog.counter-strike.net/index.php/about/> [↑](#footnote-ref-10)
11. <https://store.steampowered.com/app/730/CounterStrike_Global_Offensive/> [↑](#footnote-ref-11)
12. <https://en.wikipedia.org/wiki/Counter-Strike_(video_game)> [↑](#footnote-ref-12)
13. <https://totalcsgo.com/commands> [↑](#footnote-ref-13)
14. <https://developer.valvesoftware.com/wiki/Source> [↑](#footnote-ref-14)
15. <https://en.wikipedia.org/wiki/Source_(game_engine)> [↑](#footnote-ref-15)
16. <https://miau.my-x.hu/miau/280/kozma.log> [↑](#footnote-ref-16)
17. <https://en.wikipedia.org/wiki/Cheating_in_video_games> [↑](#footnote-ref-17)
18. <https://medium.com/ax1al/introduction-to-game-hacking-fb70e29de60f> [↑](#footnote-ref-18)
19. <https://gamehacking.academy/> [↑](#footnote-ref-19)
20. https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative [↑](#footnote-ref-20)
21. https://www.prosettings.com/csgo-hacks-cheats/ [↑](#footnote-ref-21)
22. https://help.steampowered.com/en/faqs/view/571A-97DA-70E9-FF74 [↑](#footnote-ref-22)
23. https://research.nvidia.com/publication/2021-03\_robust-vision-based-cheat-detection-competitive-gaming [↑](#footnote-ref-23)
24. https://www.pcmag.com/news/machine-learning-is-now-being-used-to-cheat-in-multiplayer-games [↑](#footnote-ref-24)
25. https://esports-news.co.uk/2022/03/02/ai-tech-detect-bots-boosters-cheaters-smurfs-valorant-anybrain/ [↑](#footnote-ref-25)
26. https://lifestyle.livemint.com/smart-living/innovation/a-new-ai-innovation-aims-to-stop-cheating-in-multiplayer-video-games-111605612437694.html [↑](#footnote-ref-26)