

Разработка алгоритма для борьбы с игровой зависимостью

Д. Ю. Сычугов*, У. В. Ильиных

ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова», г. Москва, Российская Федерация

Адрес: 119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1

* sychugov@cs.msu.ru

Аннотация

Зависимость от компьютерных игр – проблема современных детей и подростков. В статье описан новый способ борьбы с ней. В рамках целевой аудитории проведён эксперимент, целью которого было создание базы данных с информацией о количестве и среднем времени нажатия каждой клавиши во время игр. Собранные данные позволили сделать вывод, что на компьютере игровую деятельность можно однозначно отличить от любой другой. Большинство игр отличаются многократными нажатиями четырёх клавиш, которые используются для управления персонажем. Реализован программный продукт Gamebreaker-2, способный однозначно определять игровую деятельность и прерывать её. В качестве алгоритма выбран метод машинного обучения – классификатор случайного леса. Для разработки использованы язык программирования Java и свободно распространяемая бесплатная библиотека JNativeHook для считывания ввода пользователя. При запуске программа создаёт объект случайного леса и обучает его на основе данных из созданной в результате эксперимента базы. После этого программа начинает делать сканирование ввода пользователя каждые 20 секунд и отправляет данные в классификатор. Если объект случайного леса решает, что это игра, то программа определяет игровой процесс и прерывает его. Полученный продукт протестирован на целевой аудитории, на играх, для которых характерны частые нажатия четырёх основных клавиш. Программа показала высокую эффективность и полное отсутствие ложных срабатываний при качественном обучении нейронной сети. Выделены направления дальнейшей разработки программы с целью доведения до вида коммерческого продукта.

Ключевые слова: компьютерные игры, видеоигры, игровая зависимость, борьба с игровой зависимостью, динамика нажатий клавиш, ввод пользователя с клавиатуры, отслеживающая программа, разработка программного обеспечения, разработка на Java, нейронная сеть, машинное обучение, классификатор случайного леса

Авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Сычугов Д. Ю., Ильиных У. В. Разработка алгоритма для борьбы с игровой зависимостью // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 1. С. 202-211. doi: <https://doi.org/10.25559/SITITO.18.202201.202-211>

© Сычугов Д. Ю., Ильиных У. В., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Development and Implementation of the Algorithm Designed to Struggle with Videogame Addiction

D. Y. Sychugov*, U. V. Ilinykh

Lomonosov Moscow State University, Moscow, Russian Federation
Address: 1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation
* sychugov@cs.msu.ru

Abstract

Game addiction appears to be a widespread disease among young people. Nowadays methods, such as parent control programs, have their own disadvantages. In this work a new method for controlling game activity is researched. It depends on analyzing user's keyboard input. Research includes experiment in the order to collect information about key pressing dynamics during gaming. It turns out that gamers have their own unique pattern in how they press the keys. The most widespread type of the games are the games using four main buttons to control characters. Such games clearly differ from other types of computer activity such as typing or writing code. The main goal of this work is to develop and implement the algorithm which uses this pattern to detect games and stop computer process which runs the game. Machine learning methods appeared to be good for these purposes. The experiment allowed us to create a big database with input scans. Records are used to train a random forest classifier. Java programming language is used for implementing algorithm as a program product. JNativeHook open library is used for scanning process. At the start, the program creates a random forest classifier and trains it with static data. After this the main cycle of the program starts: every 20 seconds a new scan is performed and analyzed. If the program determines that there is currently game activity, then it will try to detect the game computer process and to stop it. During testing program showed great results for game with described above pattern: it detected the game and stopped its process for ten times in a row without any false effects.

Keywords: computer games, video games, game addiction, struggling with the game addiction, key pressing dynamics, keyboard user input, keylogger program, software development, Java development, neural network, machine learning, random forest classifier

The authors declare no conflict of interest.

For citation: Sychugov D.Y., Ilinykh U.V. Development and Implementation of the Algorithm Designed to Struggle with Videogame Addiction. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2022; 18(1):202-211. doi: <https://doi.org/10.25559/SITITO.18.202201.202-211>



1. Введение

1.1. Проблема игровой зависимости

Игровая зависимость появилась в 90-х годах в США. Это был пик развития игровой индустрии. Благодаря серии игр-шутеров (от англ. shooter – игра, в которой основным действием является стрельба) «Doom» и их интеграции с платформой Windows (которая только набирала обороты) игры вышли на абсолютно новый уровень интерактивности, стали доступными для каждого. Раньше они были «скучными» – пиксельными, с ограниченным количеством цветов, с примитивной двумерной графикой. Теперь же игры стали трёхмерными, в них можно было использовать фотореалистичные изображения и динамическое освещение. Они стали походить на реальный мир, чем и покорили сердца многих подростков [1]. Появление сети Интернет тоже немало способствовало этому – программное обеспечение можно было распространять по всему миру. Игры взламывали, появился пиратский рынок. В Польше в двухтысячном подростки зарабатывали немалые деньги, взламывая американские игры, переводя их на польский язык и перепродавая множество копий по более дешевой цене. Буквально за десятилетие (1990-2000) игры из какого-то роскошного и малодоступного развлечения стали частью культуры, доступной для любого человека, у которого есть компьютер с выходом в Интернет. Сами по себе игры не являются чем-то плохим. Они, как и книги или картины, являются результатом творческой работы отдельного человека или целой команды. Игры – объект культуры (что в конце 20-го века официально признали в США).

Но вместе с ними появились и психологические проблемы. Споры о жестокости игр не прекращаются с 1999 года, со стрельбы в школе «Колумбайн»¹. Подростки, устроившие это, были фанатами серии шутеров «Doom». Когда одного из них спросили, зачем же он это сделал, то он ответил, что хотел «также круто, как в Doom». Нападение было отретперировано в самой игре, на специальной карте, копировавшей школу [2]. Родители встревожились и начали подавать в суд иски с требованием запретить жестокость в играх. Но прямая связь между развитием агрессии у детей и играми не доказана (есть только масса противоречащих друг другу исследований о вреде и о пользе). Так как игры считаются объектами искусства и находятся под особой защитой, удалось добиться только установления возрастного рейтинга для игр со сценами насилия [3].

Самой главной проблемой стала замена реальной жизни на виртуальную. Для детей, которые в чем-то отстают от сверстников, испытывают коммуникативные неудачи, имеют комплексы касательно характера и внешности, обычный

мир кажется слишком сложным и трудным. Вместо того, чтобы сталкиваться с проблемами лицом к лицу, учиться преодолевать препятствия, дети уходят в себя². В виртуальном мире всё проще – школу заменяют приключения, общение с остальными остаётся в форме текстовых сообщений, не нужно вербального ответа. Дети вырастают, а комплексы не уходят. В особо запущенных случаях, описанных психологом Михаилом Лабковским³, взрослым людям виртуальное общение способно заменить дружбу и романтические отношения, а страх перед реальным взаимодействием становится непреодолимым.

1.2. Методы борьбы

На данный момент существуют следующие способы бороться с игровой зависимостью при помощи программного обеспечения:

1. Программы, имеющие базу данных с названиями игр и не позволяющие запускать процесс, если его название есть в списке. Этот способ тяжело реализовать – такая база данных нуждается в ежедневном обновлении, а подобный алгоритм легко обмануть, сменив название игры на компьютере;

2. Программы родительского контроля. Сейчас это наиболее распространённый способ контролировать игровую деятельность на компьютере. Родители просто могут установить время, в течение которого можно пользоваться компьютером, или наблюдать дистанционно, в какой программе сидит ребёнок. Можно наложить фильтры и ограничения на установленные приложения. Пример такой современной программы – Kaspersky Safe Kids⁴;

3. Программы, анализирующие ввод пользователя и по вводу пытающиеся определить игру.

3-ий тип программ ещё не изучен, но их существование реально. Это доказывают многие исследования на тему почерка пользователя при нажатии клавиш. Самый известный эксперимент в этой области – эксперимент Гейнса, Лисовски, Пресса и Шапиро 1980 года – показал, что людей можно отличить друг от друга безошибочно, анализируя их динамику нажатия клавиш⁵. Точно также и играющего человека можно отличить от человека, который набирает текст или сидит в интернете.

Справедливость этого доказывает и сбор данных в рамках этой работы. В качестве примера рассмотрим несколько типичных для определенных видов деятельности последовательностей нажатий клавиш [4; 5]. Ниже представлена типичная динамика нажатий клавиш при игровой деятельности (рисунок 1):

¹ Brown J. Doom, Quake and mass murder: Gamers search their souls after discovering the Littleton killers were part of their clan [Электронный ресурс] // Salon. April 23, 1999. URL: <https://www.salon.com/1999/04/23/gamers> (дата обращения: 10.12.2021).

² Шашмулин А. А. Проблема игровой зависимости и ее диагностика у студентов и школьников [Электронный ресурс] // Физиологические, психофизиологические, педагогические и экологические проблемы здоровья и здорового образа жизни : сб. ст. VII межвуз. студ. НПК. Екатеринбург: РГППУ, 2014. С. 120-129. URL: http://elar.rsvpu.ru/bitstream/123456789/1890/1/fpz_2014_23.pdf (дата обращения: 10.12.2021).

³ Лабковский М. Хочу и буду: принять себя, полюбить жизнь и стать счастливым. М.: Эксмо, 2017. 220 с.

⁴ Программа родительского контроля [Электронный ресурс] // Лаборатория Касперского, 2021. URL: <https://www.kaspersky.ru/safe-kids> (дата обращения: 10.12.2021).

⁵ Gaines R. S., Lisowski W., Press S. J., Shapiro N. Authentication by keystroke timing: Some preliminary results. Technical Report. No. RAND-R-2526-NSF. Santa Monica CA: Rand Corporation, 1980.





Р и с. 1. 5-минутная запись количества нажатий при игровой деятельности
F i g. 1. Comparison of the number of clicks. 5-minute recording of the number of clicks during gaming activities

А вот динамика нажатий при написании этой работы (рисунок 2):



Р и с. 2. 5-минутная запись количества нажатий при работе с текстом
F i g. 2. 5-minute record of the number of clicks when working with text

Наконец, рассмотрим типичную динамику нажатий клавиш при написании кода (рисунок 3):



Р и с. 3. 5-минутная запись количества нажатий при написании кода
F i g. 3. 5-minute record of the number of clicks while writing code

Отличие первого графика от остальных двух ощутимо и, как далее покажет практика, легко обнаруживается методами машинного обучения.

Разумеется, рисунки, представленные выше, являются всего лишь примерами и для подтверждения гипотезы об отличии игровой деятельности от прочих требуется большой объем данных. Такие данные были собраны во время испытания отслеживающей программы на целевой аудитории. Шесть учащихся 11-го московской школы № 1251 установили себе программу, отслеживающую нажатия клавиш, и запускали её во время игровой деятельности на своих персональных устройствах. В результате было собрано более трёх тысяч записей, анализ которых позволил говорить о достоверности высказанных гипотез, и на их основе создать программный продукт, распознающий игровую деятельность и пресекающий её.

2. Цель исследования

Можно выделить следующие цели работы:

1. Собрать статистический материал в таком количестве, чтобы можно было подтвердить гипотезу о том, что игровая деятельность однозначно определяется по вводу пользователя;
2. Выбрать методы для построения и обучения нейронной сети, которая будет распознавать игры⁶;
3. Разработать и проверить программный продукт, который собирает ввод пользователя, анализирует его, решает, играет пользователь или нет. Если играет, то пытается определить игровой системный процесс и прервать его;
4. Провести испытания созданного продукта на разных простейших играх и оценить его эффективность.

3. Основная часть

3.1. Выбор инструментов для разработки программы

При разработке такого программного обеспечения следующие факторы важнее остальных:

1. Производительность. Программа будет работать в реальном времени и слушать ввод пользователя постоянно, при этом запуская анализ собранных данных через фиксированные промежутки времени. Это серьёзная нагрузка на систему, реализовать её эффективно можно не на каждом языке программирования;
2. Кроссплатформенность. Целевая аудитория для такого программного обеспечения – все играющие дети/молодые люди. Значит, нам нужно, чтобы программа работала в любой системе, не требовала серьёзных изменений при появлении новой платформы и больших трудовых затрат на изменение кода от платформы к платформе;
3. Простота разработки. Рынок игр меняется, сами игры меняются, следовательно, программное обеспечение требует поддержки и развития. Хороший фактор программируемой системы – её расширяемость, то есть возможность добавлять новые компоненты, не перерабатывая существующие фрагменты. Поддержка расширяемости достижима не на каждом инструменте.

⁶ Steinberg R. 6 areas where artificial neural networks outperform humans [Электронный ресурс] // VentureBeat. December 8, 2017. URL: <https://venturebeat.com/2017/12/08/6-areas-where-artificial-neural-networks-outperform-humans> (дата обращения: 10.12.2021).



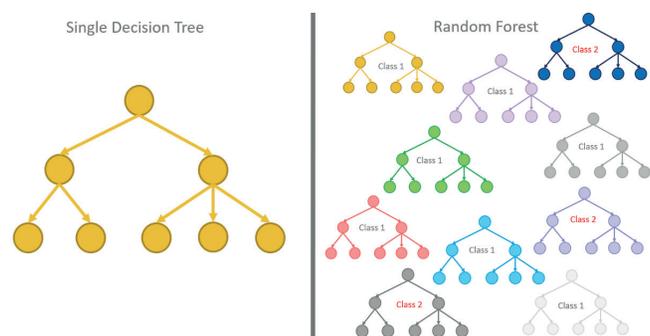
Учитывая эти факторы, было решено, что лучшим инструментом для разработки будет Java 11 от поставщика Azul Community⁷. Java как язык программирования удовлетворяет факторам из списка выше следующим образом:

1. В основе Java лежит C++ – самый производительный язык программирования высокого уровня. Java позволяет подключать дополнительные инструменты (библиотеки), написанные исключительно на C++;
2. Проекты на Java не требуют компиляции. Они преобразовываются в исполняемые файлы, которая может распознать и исполнить любая виртуальная машина Java на компьютере. Это означает, что один и тот же код на Java может исполняться на любой платформе и не требует дополнительного переписывания;
3. Java – объектно-ориентированный язык программирования. Весь код разделён на фрагменты, называемые классами. Эта структура позволяет проектировать системы таким образом, чтобы изменения в одном компоненте не вызывали проблем в других.

Для связи с системными инструментами, позволяющими работать с вводом пользователя, была использована библиотека JNativeHook⁸. Она также кроссплатформенная.

3.2. Выбранный алгоритм

Такая система должна уметь регулярно изменяться в соответствии с развитием игр, значит, должна легко обучаться. Лучше всего с этим справляются методы машинного обучения⁹. Для данной программы выбран алгоритм Random Forest (случайный лес) [6]-[8]. Анализируя данные, он составляет деревья вопросов, вложенные друг в друга¹⁰. Совокупность этих деревьев позволяет однозначно определить, к какому классу относится подаваемый на вход объект (рисунок 4).



Р и с. 4. Схематичное представление алгоритма случайного леса
F i g. 4. Schematic representation of the random forest algorithm

3.3. Этапы разработки

3.3.1 Написание отслеживающей программы

Кейлоггер (англ. keylogger) – отслеживающая программа, собирающая данные о вводе пользователя. Традиционно кейлоггером называется простейший вирус, который запоминает все клавиши, нажатые пользователем во время интернет-сёрфинга, и пытается определить, что из введённой информации является паролем. Кейлоггер обращается в систему и достаёт из неё данные – код клавиши, время её нажатия и остальные характеристики. Обращение непосредственно к системному хранилищу этой информации вместо драйвера устройства, с которого нужно считывать ввод – важный аспект. Это гарантирует кроссплатформенность и независимость от устройств. Такую программу невозможно обмануть, поменяв устройства или подключив несколько [9]-[13].

Кейлоггер ложится в основу двух программных продуктов, необходимых для реализации алгоритма, – lodger и Gamebreaker-2.

Logder при запуске начинает непрерывно сканировать ввод пользователя и каждые 20 секунд делать записи в формате .txt (рисунок 5).

```

1   KeyCode: 38, times pressed: 3, total press time: 3
2   KeyCode: 33, times pressed: 1, total press time: 1
3   KeyCode: 21, times pressed: 3, total press time: 3
4   KeyCode: 31, times pressed: 1, total press time: 1
5   KeyCode: 20, times pressed: 4, total press time: 4
6   KeyCode: 57, times pressed: 8, total press time: 8
7   KeyCode: 32, times pressed: 5, total press time: 5
8   KeyCode: 48, times pressed: 2, total press time: 2
9   KeyCode: 14, times pressed: 4, total press time: 52
10  KeyCode: 42, times pressed: 1, total press time: 1
11  KeyCode: 19, times pressed: 1, total press time: 1
12  KeyCode: 16, times pressed: 1, total press time: 1
13  KeyCode: 37, times pressed: 2, total press time: 2
14  KeyCode: 36, times pressed: 4, total press time: 4
15  KeyCode: 22, times pressed: 2, total press time: 2
16  KeyCode: 35, times pressed: 1, total press time: 1
17  KeyCode: 39, times pressed: 1, total press time: 1
18  KeyCode: 49, times pressed: 1, total press time: 1
19  KeyCode: 46, times pressed: 2, total press time: 2
20  KeyCode: 44, times pressed: 1, total press time: 1
21  KeyCode: 18, times pressed: 2, total press time: 2
22  KeyCode: 26, times pressed: 1, total press time: 1

```

Р и с. 5. Текст внутри записи нажатия клавиш
F i g. 5. Text inside the keystroke record

⁷ Azul JDKs [Электронный ресурс] // Azul, 2021. URL: <https://www.azul.com/downloads/?package=jdk> (дата обращения: 10.12.2021).

⁸ Barker A. JNativeHook [Электронный ресурс] // GitHub, Inc., 2021. URL: <https://github.com/kwhat/jnativehook> (дата обращения: 10.12.2021).

⁹ Воронцов К. В. Лекции по линейным алгоритмам классификации [Электронный ресурс] // MachineLearning.ru, 2009. 43 с. URL: <http://machinelearning.ru/wiki/images/6/68/Voron-ML-Lin.pdf> (дата обращения: 10.12.2021).

¹⁰ Черезов Д. С., Тюкачев Н. А. Обзор основных методов классификации и кластеризации данных // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2009. № 2. С. 25-29. URL: <https://www.elibrary.ru/item.asp?id=13040050> (дата обращения: 10.12.2021); Ситникова Н. В., Парингер Р. А., Куприянов А. В. Сравнительное исследование алгоритмов классификации больших объемов данных // Информационные технологии и нанотехнологии (ИТНТ-2016): материалы Межд. конф. и молодежной школы. Самара: Самарский университет, 2016. С. 1096-1099. URL: <https://elibrary.ru/item.asp?pff=1&id=26715455> (дата обращения: 10.12.2021).



3.3.2 Использование отслеживающей программы для сбора базы данных среди целевой аудитории

Целевая аудитория для программного обеспечения – молодые люди, часто играющие в компьютерные игры. В эксперименте, целью которого был сбор данных о том, какие клавиши нажимаются во время игровой деятельности, приняли участие учащиеся ГБОУ «Школа № 1251 имени генерала Шарля де Голля». Они установили себе программу lodger и запускали её одновременно с играми. Сам процесс установки подтвердил правильность выбора инструментов разработки. Программу удалось запустить на разных платформах с разными версиями, не меняя её кода. Результат – большая база данных с нажатиями клавиш. Она содержит 3887 записей длительностью 20 секунд.



Р и с. 6. 20-секундная запись среднего времени нажатия при игровой деятельности

F i g. 6. Comparison of average pressing time. 20 second recording of average click time during gaming activity

```
[SCAN] Duration: 20 seconds
[5] was pressed 0 times, average press duration = 0.0
[8] was pressed 0 times, average press duration = 0.0
[1] was pressed 1 times, average press duration = 1.0
[2] was pressed 0 times, average press duration = 0.0
[3] was pressed 0 times, average press duration = 0.0
[4] was pressed 0 times, average press duration = 0.0
[5] was pressed 0 times, average press duration = 0.0
[6] was pressed 0 times, average press duration = 0.0
[7] was pressed 0 times, average press duration = 0.0
[8] was pressed 0 times, average press duration = 0.0
[9] was pressed 0 times, average press duration = 0.0
[0] was pressed 0 times, average press duration = 0.0
[-] was pressed 0 times, average press duration = 0.0
[=] was pressed 0 times, average press duration = 0.0
[~] was pressed 0 times, average press duration = 0.0
[Q] was pressed 0 times, average press duration = 0.0
[W] was pressed 2 times, average press duration = 1.5
[E] was pressed 0 times, average press duration = 0.0
[R] was pressed 0 times, average press duration = 0.0
[T] was pressed 0 times, average press duration = 0.0
[Y] was pressed 0 times, average press duration = 0.0
[U] was pressed 0 times, average press duration = 0.0
[I] was pressed 0 times, average press duration = 0.0
[O] was pressed 0 times, average press duration = 0.0
```

Р и с. 7. Текст внутри 20-секундной записи

F i g. 7. Text inside a 20-second recording

Данные собраны по следующим простейшим играм:

1. Lord of the Rings Online¹¹;
2. Blasphemous¹²;
3. Apex Legends¹³;
4. Grand Theft Auto V¹⁴;
5. 7 Days to Die¹⁵;
6. Наборы браузерных игр.

Одна 20-секундная запись любой из этих игр представлена в виде графиков среднего времени и количества нажатий (рисунок 6) и в виде файла .txt (рисунок 7).

3.3.3 Анализ двух характеристик клавиш. Решения о том, что использовать при обучении системы

В компьютере характеристики клавиш представлены двумя состояниями:

1. Клавиша нажата;
2. Клавиша не нажата.

Lodger преобразует их в следующие:

1. Время нажатия клавиши;
2. Время, в течение которого клавиша зажата;
3. Время, когда клавишу отпускают.

В итоге чёткие количественные характеристики, которые пригодны для обучения системы:

1. Количество нажатий клавиши в течение определённого промежутка времени;
2. Среднее время, которое была зажата клавиша в течение определённого промежутка времени.



Р и с. 8. 20-секундная запись количества нажатия клавиш

F i g. 8. 20 second record of keystroke count

Как показал эксперимент, количество нажатий будет лучшей характеристикой. На графике количества нажатий (рисунок 8) чётко выделяются четыре основные клавиши. Наличие четырёх основных кнопок характерно для большинства игр. Как правило, они означают четыре команды для управления игровым персонажем: пойти вперёд, назад, вправо или влево. Чаше

¹¹ The Lord of the Rings Online [Электронный ресурс] // Valve Corporation, 2021. URL: https://store.steampowered.com/app/212500/The_Lord_of_the_Rings_Online (дата обращения: 10.12.2021).

¹² Blasphemous [Электронный ресурс] // Valve Corporation, 2021. URL: <https://store.steampowered.com/app/774361/Blasphemous> (дата обращения: 10.12.2021).

¹³ Apex Legends [Электронный ресурс] // Electronic Arts Inc., 2021. URL: <https://www.ea.com/games/apex-legends> (дата обращения: 10.12.2021).

¹⁴ Grand Theft Auto V [Электронный ресурс] // Rockstar Games, 2021. URL: <https://www.rockstargames.com/ru/gta-v> (дата обращения: 10.12.2021).

¹⁵ 7 Days to Die [Электронный ресурс] // Valve Corporation, 2021. URL: https://store.steampowered.com/app/251570/7_Days_to_Die (дата обращения: 10.12.2021).



всего это клавиши W, A, S, D, как показано на графиках (рисунки 8 и 9), Q, W, E, R, или клавиши-стрелки, но необязательно¹⁶. Распределение среднего времени нажатия клавиш (рисунок 9) не будет использоваться. Оно не настолько характерно из-за клавиш, которые используются в игре для дополнительных действий, что создаёт лишние проблемы для обучаемой системы [14]-[19].



Р и с. 9. 20-секундная запись среднего времени нажатия клавиш
F i g. 9. 20 second record of average keypress time

3.3.4 Преобразование данных

Для алгоритма RandomForest из библиотеки weka.classifiers полученные данные нужно преобразовать в обучающий набор, то есть файл .arff (рисунок 10).

```
@RELATION game
@ATTRIBUTE <key name> numeric
...
@ATTRIBUTE <key name> numeric
@ATTRIBUTE class {game, not-game}

@DATA
<key set> <class>
...
<key set> <class>
```

Р и с. 10. Схема кода внутри обучающего файла
F i g. 10. Code diagram inside the training file

3.3.5 Написание и обучение системы

Второй программный продукт – Gamebreaker-2. В его основе также лежит кейлоггер. Gamebreaker-2 работает по следующему алгоритму:

1. Инициализация. Программа подключает все нужные зависимости, в частности, библиотеку JNativeHook для сканирования ввода. Если инициализация не удалась, программа завершается с ошибкой;

2. Создание объекта случайного леса и обучение. Программа считывает статичный, вшитый в проект файл TrainingSet.arff и выстраивает деревья вопросов, по которым будет определять игру;

3. Запуск главного цикла:

3.1 Делать сканирование в течение 20 секунд;

3.2 Отправить результат в объект случайного леса;

3.3 Если объект – игра, определить игровой процесс и прервать его. Продолжить цикл.

Схематически работу программы можно представить в виде блок-схемы [20]-[25] (рисунок 11):



Р и с. 11. Схематичное представление работы программы Gamebreaker-2
F i g. 11. Schematic representation of the operation of the Gamebreaker-2 program

3.3.6 Тестирование системы

В качестве объекта испытания были выбран простейший, самый распространённый тип игр. Это игры, в которых самое частое действие – перемещение персонажа при помощи четырёх клавиш. Ложных срабатываний не наблюдалось. Имеется запись тестирования на игре Stardew Valley. Программе понадобился всего один цикл – 20 секунд – чтобы определить игру и прервать её (рисунок 12).



Р и с. 12. Запись реакции программы на игру Stardew Valley
F i g. 12. Recording the reaction of the program to the game Stardew Valley

¹⁶ Савинов А. Н. Методы, модели и алгоритмы распознавания клавиатурного почерка в ключевых системах : дисс. ... канд. тех. наук : 05.13.19. Йошкар-Ола, 2013. 97 с.



4. Полученные результаты

В результате работы:

1. Собрана большая база данных (более трёх тысяч записей по 20 секунд);
2. Подтверждена гипотеза о том, что на основе ввода пользователя можно однозначно определить игровую деятельность;
3. Разработан программный продукт Gamebreaker-2, отслеживающий ввод пользователя в реальном времени и прерывающий простейшие игры¹⁷;
4. Продукт протестирован на простейших, самых распространённых играх.

Gamebreaker-2 отлично показал себя на таких играх. Первого цикла сканирования (то есть двадцати секунд) достаточно, чтобы определить игру. На испытании на игре Stardew Valley не было ни одного ошибочного срабатывания из десяти попыток подряд.

5. Заключение

Можно довести Gamebreaker-2 до вида коммерческого продукта, чтобы программа была доступна и понятна любому пользователю. Для этого нужно сделать следующее:

1. Добавление интерфейса. Программа станет доступной для загрузки и использования любым пользователем, если будет удобной. Она должна иметь интерактивный интерфейс, через который можно запускать, останавливать, настраивать программу. Она должна откликаться на действия пользователя, информировать его о том, что определила игру и решила её прервать;
2. Увеличение базы данных. Индустрия игр быстро меняется¹⁸. Базу данных нужно обновлять, расширять и поддерживать;
3. Улучшение алгоритма прерывания игры. В процессе разработки программы были рассмотрены три способа её прервать:
 - 3.1. Отключение процесса, оказывающего наиболее высокую нагрузку на центральный процессор или видеокарту. Данный способ оказался нерабочим, так как существует множество программ, необходимых для работы или учёбы, но оказывающих на компьютер нагрузку в разы большую, чем многие простые игры (например, продукты компании Adobe. Программа Adobe Photoshop загружает видеокарту сильнее, чем некоторые игры с реалистичной графикой). Количество ложных срабатываний слишком велико и может мешать работе на компьютере;
 - 3.2. Анализ командной строки, при помощи которой запущен игровой процесс. Сейчас используется именно этот способ. Он рабочий, но очевидно, что слишком зависит от технических особенностей программы и возможны ложные срабатывания, которые никак не пресесть;
 - 3.3. Отправление системного вызова, который сворачивает открытую на рабочем столе программу, окно которой находится на переднем плане. Это однозначно лучший способ, но на данный момент он только изучается. Основная проблема в том, что здесь используется системный вызов. Системные вызовы разные от платформы к платформе. Добавление в программу модуля, который использует конкретный системный вызов, лишит программу кроссплатформенности. Для успешной реализации такого прерывания процесса необходимо написать компонент, который проверяет, на какой именно платформе запускается программа и какую библиотеку системных вызовов необходимо подгрузить и использовать;
4. Тестирование на других типах игр. Существующая реализация Gamebreaker-2 хорошо показала себя на играх, которые используют четыре основные клавиши для управления персонажем. Но есть игры, которые используют другое управление, например:
 1. Игры-платформеры (англ. platformer), которые используют всего две клавиши (как правило, A, D для перемещения);
 2. Игры в браузере, которые используют только мышь;
 3. Игры-стратегии, такие как Civilization, League of Legends, которые используют мышь и несколько регулярных клавиш. Игры категорий 2 и 3 требуют дополнения к программе – компонента, который бы сохранял движения мыши, анализировал их;
 5. Предусмотреть попытки обойти программу. Существующая реализация Gamebreaker-2 может быть обманута переназначением клавиш в системе. Надо улучшить алгоритм так, чтобы он учитывал динамику нажатия клавиш в целом, без привязки к конкретным значениям.

Благодарности

В эксперименте, целью которого был сбор данных о том, какие клавиши нажимаются во время игровой деятельности, приняли участие следующие учащиеся ГБОУ «Школа № 1251 имени генерала Шарля де Голля»: Игорь Зуенко, Ренат Ким, Анна Красильникова, Виталий Кодочигов, Александр Скороходов, Владимир Сергиев.

Авторы работы выражают благодарность учащимся за помощь.

Acknowledgements:

The following students of the Charles de Gaulle Moscow School No. 1251 took part in the experiment, aimed at collecting data on which keys are pressed during the game activity.

The authors express their gratitude to the students for their assistance: Igor Zuenko, Renat Kim, Anna Krasilnikova, Vitaly Kodochigov, Alexander Skorokhodov, Vladimir Sergiev.

¹⁷ Свидетельство о государственной регистрации программы для ЭВМ № 2018619955 Российская Федерация. Gamebreaker-1-0 : № 2018617519 : заявл. 17.07.2018 : опублик. 15.08.2018 / Д. Ю. Сычугов, А. А. Лукьяница, Д. А. Герасимов ; заявитель МГУ. URL: <https://elibrary.ru/item.asp?id=39300130> (дата обращения: 10.12.2021).

¹⁸ Langlotz A. C. H., Rhode M., Whaley C. Video Games Industry Overview: An Analysis of the Current Market and Future Growth Trends. International Business Project, 2008. URL: http://holgerlanglotz.de/downloads/BU4510_VideoGamesIndustry_LanglotzEtAl.pdf (дата обращения: 10.12.2021).



References

- [1] Tang K.T.Y., et al. Gambling disorder and comorbid behavioral addictions: Demographic, clinical, and personality correlates. *Psychiatry Research*. 2020; 284:112763. (In Eng.) doi: <https://doi.org/10.1016/j.psychres.2020.112763>
- [2] Hong J.S., et al. The social ecology of the Columbine High School shootings. *Children and Youth Services Review*. 2011; 33(6):861-868. (In Eng.) doi: <https://doi.org/10.1016/j.childyouth.2010.12.005>
- [3] Chadha A., Abbas A., Andreopoulos Y. Video Classification With CNNs: Using the Codec as a Spatio-Temporal Activity Sensor. *IEEE Transactions on Circuits and Systems for Video Technology*. 2019; 29(2):475-485. (In Eng.) doi: <https://doi.org/10.1109/TCSVT.2017.2786999>
- [4] Bryan W.L., Harter N. Studies in the physiology and psychology of the telegraphic language. *Psychological Review*. 1897; 4(1):27-53. (In Eng.) doi: <https://doi.org/10.1037/h0073806>
- [5] Bryan W.L., Harter N. Studies on the telegraphic language: The acquisition of a hierarchy of habits. *Psychological Review*. 1899; 6(4):345-375. (In Eng.) doi: <https://doi.org/10.1037/h0073117>
- [6] Carbonell J.G., Michalski R.S., Mitchel T.M. Machine Learning: A Historical and Methodological Analysis. *AI Magazine*. 1983; 4(3):69-79. (In Eng.) doi: <https://doi.org/10.1609/aimag.v4i3.406>
- [7] Fix E., Hodges Jr. J.L. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *Revue Internationale de Statistique = International Statistical Review*. 1989; 57(3):238-247. (In Eng.) doi: <https://doi.org/10.2307/1403797>
- [8] Breiman L. Random Forests. *Machine Learning*. 2001; 45(1):5-32. (In Eng.) doi: <https://doi.org/10.1023/A:1010933404324>
- [9] Gladwell M. Blink: The Power of Thinking Without Thinking. Back Bay Books; 2007. 296 p. (In Eng.)
- [10] Hebb D.O. The Organization of Behavior: A Neuropsychological Theory. Psychology Press; 2002. 335 p. (In Eng.)
- [11] Navarro H., Bennun L. Descriptive examples of the limitations of Artificial Neural Networks applied to the analysis of independent stochastic data. *International Journal of Computer Engineering & Technology*. 2014; 5(5):40-42. (In Eng.)
- [12] Karnan M., Akila M., Krishnaraj N. Biometric personal authentication using keystroke dynamics: A review. *Applied Soft Computing*. 2011; 11(2):1565-1573. (In Eng.) doi: <https://doi.org/10.1016/j.asoc.2010.08.003>
- [13] Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., Fei-Fei L. Large-Scale Video Classification with Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Press, Columbus, OH, USA; 2014. p. 1725-1732. (In Eng.) doi: <https://doi.org/10.1109/CVPR.2014.223>
- [14] Kia S.M., et al. A Novel Scheme for Intelligent Recognition of Pornographic Images. arXiv:1402.5792v3. 2014. (In Eng.) doi: <https://doi.org/10.48550/arXiv.1402.5792>
- [15] Weinstein A.M. Computer and Video Game Addiction – A Comparison between Game Users and Non-Game Users. *The American Journal of Drug and Alcohol Abuse*. 2010; 36(5):268-276. (In Eng.) doi: <https://doi.org/10.3109/00952990.2010.491879>
- [16] McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biology*. 1943; 5:115-133. Available at: <https://www.cs.cmu.edu/~/epxing/Class/10715/reading/McCulloch.and.Pitts.pdf> (accessed 10.12.2021). (In Eng.)
- [17] Peacock A., Ke X., Wilkerson M. Typing patterns: A key to user identification. *IEEE Security and Privacy*. 2004; 2(5):40-47. (In Eng.) doi: <https://doi.org/10.1109/MSP.2004.89>
- [18] Durak H.Y., Demirhan E.K., Cital M. Examining various risk factors as the predictors of gifted and non-gifted high school students' online game addiction. *Computers & Education*. 2022; 177:104378. (In Eng.) doi: <https://doi.org/10.1016/j.compedu.2021.104378>
- [19] Ball C., et al. Gaming the SySTEM: The Relationship Between Video Games and the Digital and STEM Divides. *Games and Culture*. 2020; 15(5):501-528. (In Eng.) doi: <https://doi.org/10.1177/1555412018812513>
- [20] Schölkopf B., Simard P., Smola A., Vapnik V. Prior knowledge in support vector kernels. In: Jordan M., Kearns M., Solla S. (eds.) *Proceedings of the 10th International Conference on Neural Information Processing Systems (NIPS'97)*. *Advances in Neural Information Processing Systems*. Vol. 10. MIT Press, Cambridge, MA, USA; 1997. p. 640-646. Available at: <https://papers.nips.cc/paper/1997/file/01d8bae291b1e4724443375634ccfa0e-Paper.pdf> (accessed 10.12.2021). (In Eng.)
- [21] Sepehr S., Head M. Online Video Game Addiction: A Review and an Information Systems Research Agenda. *Proceedings of the 19th Americas Conference on Information Systems, AMCIS 2013 – Hyperconnected World: Anything, Anywhere, Anytime*. Chicago, Illinois; 2013. Vol. 4. p. 2590-2600. Available at: <https://aisel.aisnet.org/amcis2013/HumanComputerInteraction/GeneralPresentations/3> (accessed 10.12.2021). (In Eng.)
- [22] Shalev-Shwartz S., Ben-David S. Understanding Machine Learning: From Theory to Algorithms. Cambridge: Cambridge University Press; 2014. 410 p. (In Eng.) doi: <https://doi.org/10.1017/CBO9781107298019>
- [23] Vapnik V. The Support Vector Method of Function Estimation. In: Suykens J.A.K., Vandewalle J. (eds.) *Nonlinear Modeling*. Springer, Boston, MA; 1998. p. 55-85. (In Eng.) doi: https://doi.org/10.1007/978-1-4615-5703-6_3
- [24] Ng B.D., Wiemer-Hastings P. Addiction to the Internet and Online Gaming. *CyberPsychology & Behavior*. 2005; 8(2):110-113. (In Eng.) doi: <http://doi.org/10.1089/cpb.2005.8.110>
- [25] Wu Z., Yao T., Fu Y., Jiang Y.-G. Deep Learning for Video Classification and Captioning. *Frontiers of Multimedia Research*. Association for Computing Machinery and Morgan & Claypool; 2017. p. 3-29. (In Eng.) doi: <https://doi.org/10.1145/3122865.3122867>

Поступила 10.12.2021; одобрена после рецензирования 17.02.2022; принята к публикации 25.02.2022.

Submitted 10.12.2021; approved after reviewing 17.02.2022; accepted for publication 25.02.2022.



Об авторах:

Сычугов Дмитрий Юрьевич, профессор кафедры автоматизации научных исследований, факультет вычислительной математики и кибернетики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), доктор физико-математических наук, **ORCID: <https://orcid.org/0000-0002-2844-3845>**, sychugov@cs.msu.ru

Ильиных Ульяна Владимировна, студент кафедры автоматизации научных исследований, факультет вычислительной математики и кибернетики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), **ORCID: <https://orcid.org/0000-0003-1946-0907>**, black.n.pink@yandex.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Dmitry Yu. Sychugov, Professor of the Chair of Automation for Scientific Research, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), Dr. Sci. (Phys.-Math.), **ORCID: <https://orcid.org/0000-0002-2844-3845>**, sychugov@cs.msu

Uliana V. Ilinykh, student at the Chair of Automation for Scientific Research, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), **ORCID: <https://orcid.org/0000-0003-1946-0907>**, black.n.pink@yandex.ru

All authors have read and approved the final manuscript.

