Telemetry and machine learning to speed-up the measure of intelligence through video games

Mingxiao Guo[0000−0003−3627−5981], Pablo Gutiérrez-Sánchez[0000−0002−6702−5726], Alejandro Ortega-Álvarez[0000−0000−0000−0000], Pedro A. González-Calero[0000−0002−9151−5573], María Angeles Quiroga[0000−0003−4542−2744], and Pedro P. Gómez-Martín[0000−0002−3855−7344]

Universidad Complutense de Madrid, Spain
{mingxguo,pabgut02,alejor01,pagoncal,maquirog,pedrop}@ucm.es

Abstract. Recent research has shown a high correlation between the g factor (or general intelligence factor) and overall performance in video games. This relationship is held not only when playing *brain games* but also other generalist commercial titles. Unfortunately, these off-the-shelf games do not allow automatic extraction of in-game behavior data. As a result, researchers are often forced to manually register the game sessions metrics, reducing the gathered information and, as a consequence, the results.

The aim of our work is to help to improve the data collection process used in those studies by: (1) reimplementing a small subset consisting of three of the games used in these former studies; (2) developing a telemetry system to automate and enhance the recording of in-game user events and variables; and (3) deploying a web platform to conduct an online experiment to collect such data. With the obtained data, we attempt to predict a player's final score in a given game from truncated play logs (up to a certain point in time) using neural networks and random forest. This later analysis could potentially allow future studies to shorten experiment times, thus increasing the viability of game-based intelligence assessment.

Keywords: Serious games · Intelligence and video games · Computerized assessment · Game Telemetry.

1 Introduction

Since the 1980s, research has been conducted on the use of video games to measure intelligence [8,13]. [10] showed that intelligence could be measured using brain training games and later [12] showed that intelligence could also be assessed through commercial games of other genres.

This last study uses a preselected battery of commercial video games and carries out measurement protocols where a human evaluator observes the subject playing and monitors their performance, noting the results: mainly the level reached by the subject in the game over a fixed amount of time.

The aim of the work described here is to determine to what extent this process of assessing intelligence through video games can be further improved if we can instrumentalize the games to have a finer measure of the performance in the game. In particular we are interested to see whether we are able to speed up the measurement process through the use of machine learning techniques.

The rest of the article is structured as follows. Next Section summarizes related work on the use of video games for intelligence measurement. Next, we present the games used in our experiments. In Section 4 we describe the telemetry system developed for the games and the data we have collected for each of the 3 games under consideration. In Section 5, we discuss the results of applying neural networks and random forest to predict the outcome with truncated versions of the game traces. Finally, Section 6 proposes future work and concludes the paper.

2 Intelligence and games

The notion of "intelligence" referred to in this paper is the one proposed by Gottfredson in [7], defined as "a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings — 'catching on', 'making sense' of things, or 'figuring out' what to do."

According to [11], studies on intelligence in video games have been developed and researched since 1986, using a variety of the existing titles that require reasoning, planning, learning and performing logical tasks or challenges. All these characteristics are included in the definition of intelligence mentioned above.

We take as our starting point a study already carried out by researchers in Psychology [12], which systematically describes an investigation based on correlations at a latent level between intelligence and video games. Its main objective was to analyze whether performance in video games can be correlated with the results obtained in conventional intelligence tests. To do this, data from 134 people who volunteered to complete 10 games of different genres within a controlled environment were used.

According to [11], these games focus on the study of the following three cognitive skills within the second stratum of the CHC model (a theory on the structure of human cognitive abilities, we refer to [9] for more details):

- **Gf (fluid reasoning)**: The use of deliberate and controlled procedures (often requiring focused attention) to solve novel, "on-the-spot" problems that cannot be solved by using previously learned habits, schemas, and scripts. We can observe this skill in games that require the player to come up with techniques or recognize patterns in order to advance in the resolution of new problems, for example, escape or puzzle games.
- **Gv (visuospatial ability)**: The ability to make use of simulated mental imagery to solve problems — perceiving, discriminating, manipulating, and

recalling nonlinguistic images in the "mind's eye". Sokoban or most platform games (both 2D and 3D) would be examples of video games where this skill could be assessed.
– **Gs (processing speed)**: The ability to control attention to automatically, quickly, and fluently perform relatively simple repetitive cognitive tasks. Processing speed may also be described as attentional fluency or attentional speediness. Some examples of videogames in which this skill plays a central role could be Tetris or Flappy Bird.

The execution of the original study consisted in recording, for each of the 10 games involved, the variables that were thought could influence each of these cognitive factors and to conduct measurements and calculations on them. After completing the games, the participants had to undergo 6 different aptitude tests (two for each of the aptitude or cognitive factors considered) and a video game habit questionnaire to normalize the results based on prior game experience.

Performance in the games was correlated with standard measures of each of these factors. The results revealed a correlation value of 0.79 between latent factors representing general intelligence (g) and general video game performance (gVG). This finding led to the conclusion that intelligence tests and video games were both backed by shared cognitive processes, and that mind games are not the only genre capable of producing performance measures comparable to standardized intelligence tests.

3 The games

This paper partially replicates the aforementioned study [12] by partially reimplementing three of the ten games previously used, and extending those three games with a telemetry system. Through the telemetry system we can record every action of the player in the game, and thus avoid the need for a human evaluator to collect data. This eases the sampling and let us go forwards in the video game possibilities to intelligence assessment. The selected games are:

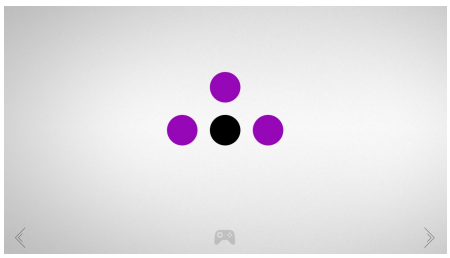
– Blek [1], mobile version for the study of fluid reasoning (Gf).
– Edge [2], PC version for the analysis of visuospatial ability (Gv).
– Unpossible [3], mobile version for the assessment of processing speed (Gs).

For the three selected games, game mechanics and levels required to replicate the original study were developed from scratch using Unity 3D^{TM}.

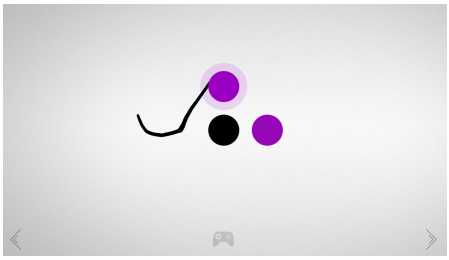
3.1 Blek

Blek [1] is presented as a "canvas" where the player draws (with the mouse on PC, or with the finger, on touch devices) a short line or stroke that "comes to life", repeating itself in a loop until it goes out of the screen or hits an obstacle.

In each level there is a set of circles or coloured balls that must be picked up by the line as it moves through the screen. To complete a level, the line must



(a) Level example with one obstacle and three collectibles.



(b) Trace in motion picking up the second collectible.

Fig. 1: Blek

collect all these items before finishing its path by leaving the screen from the top or the bottom. If the player touches one of these target circles while creating the stroke, it immediately comes to life and the user cannot continue drawing. The main game mechanic becomes on the player defining a short line and anticipating its movement. To add more variety, some special objects are introduced that act as obstacles, always represented as black elements. When the line touches any of them, its progress stops immediately. The line also stops if the user starts drawing a new line while the previous one is still moving on the screen. A final peculiarity is that the trace "bounces" when it comes into contact with the left and right sides of the level.

As the player advances through the levels, new types of elements appear, such as balls with projectiles, which shoot a set of small colored balls when touched by the stroke. These small balls have the ability to pick up exactly one other colored ball from the level when they touch it, with both of them disappearing in the process (noting here that these projectiles can freely pass through any black obstacle). The game is clearly defined in a puzzle category, and is a particularly attractive candidate for the evaluation of fluid reasoning (Gf).

3.2 Edge

The game Edge [2] consists of several levels in which the player controls a rolling cube that can move in 4 possible directions. The world has an isometric perspective and is composed of discrete squares along which the cube advances. The player's objective is to reach the final square of the level in the shortest time possible while collecting a set of prisms distributed all over the map. The strong spatial component of this game makes it an appealing candidate when it comes to evaluating a player's visuospatial ability (Gv).

As for the controls and mechanics of the game, in the computer version, the cube is moved using the arrows on the keyboard, or the WASD keys. The cube can climb up steps, but only one at a time and only if there are no obstacles above it. It can also be moved and pushed by moving obstacles or platforms, some of which can be activated with triggers. Other game features include activators

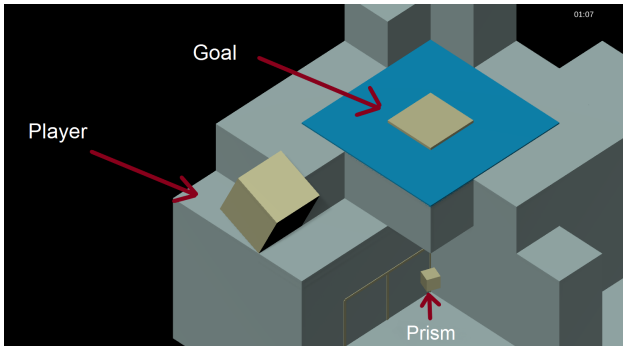


Fig. 2: Example of a game scene from Edge.

that push the cube a number of squares in a particular direction and brittle floor blocks that collapse soon after they are stepped on.

At certain levels the player can shrink and become a mini-cube. This mini-cube is controlled in the same way and has the added feature that it can climb walls and access places that the normal sized cube could not. The player must revert to the initial size at some point to be able to complete the level. The player dies when it falls over the edge of the scene, after which it respawns at the last checkpoint reached on the map.

3.3 Unpossible

Unlike the other two games, the mechanics of Unpossible are much simpler: the player rides on the outside of a curved tube in space and tries to hold on as long as possible by dodging all the obstacles they encounter. The forward movement is not controlled by the player, and all they can do is turn left or right on the tube. Every time players hit an obstacle, they die and are "respawned" at the beginning of the corresponding level.

As the game progresses, the sequence of obstacles becomes more complex and demanding, requiring the player to react increasingly faster and concentrate constantly on controlling their movements. Therefore, it is a very promising candidate to measure the processing speed (Gs), and probably also a certain amount of visuospatial capacity (Gv).

4 Telemetry System

In order to record the traces of player interaction with our games, a telemetry system was developed in accordance with the following design decisions and fundamental requirements:

– The telemetry system must be event-driven in order to: (1) be as general as possible in terms of the kinds of games it is capable of supporting, and

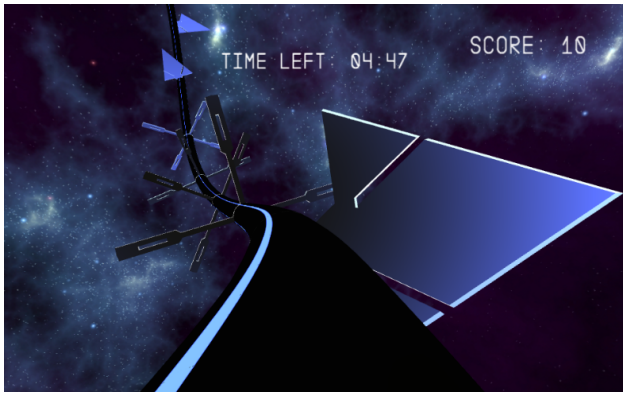


Fig. 3: A scene from Unpossible

(2) be versatile enough to allow the inference of different types of metrics in further processing, even those that were not initially considered. We note that condition (1) refers to a design consideration in the infrastructure of the system itself, while (2) is intended to act as a general guideline in selecting which events to send from games via that system.

- The events, given the nature of the experiments to be conducted, must have as mandatory fields, at least, identifiers for the *game* in which they occurred and the *user* who produced them, a *timestamp* marking the instant (following the UNIX format) in which they happened, a representative *name* that differentiates it from other events in the game, and a dictionary or list of possible additional *parameters* that may be necessary for the handling and contextualization of these events.
- Sending events should be as simple a task as possible from the perspective of the programmer of each game, ideally resembling the way it is used in the case of Unity Analytics. This means that the instrumentalization in the code must be clear and intuitive to use. Furthermore, the corresponding code must be reusable from any additional game implemented in the same engine without additional adaptations.

Different available telemetry systems were analysed (Unity, Firebase and Google Analytics) but they were finally rejected because although they provide dashboards to analyse aggregated data, they do not allow to get the raw events for further analysis. So, following the above guidelines, two systems for sending events to the server were developed, one for use from mobile devices/desktop applications, and the other adapted to WebGL. The latter was finally used for experimental data collection.

The events recorded for each game were selected so that the traces produced during a player's session would give a sufficiently accurate picture of their behavior and performance during the game. In this way, we chose to record the following significant events:

- General events: These are common to most games and serve to represent the player's progression and the course of the session (`tutorial_start`, `level_end`, and so on), as well as various frequent events in generalized games, such as the death of a player (`player_death`).
- Exclusive events: they were specific for each game:
  - *Blek*: the player touches the screen for the first time after a static period (`first_touch`), start of a drawing of a stroke (`begin_drawing`), start of the repetition phase of the stroke (`begin_looping`) and collision with an obstacle in the game (`black_touched`).
  - *Edge*: collectible prism obtained (`got_item`), new progress mark reached (`got_checkpoint`), and an additional parameter (`num_moves`) at the end-of-level events to provide a total of movements made in the level.
  - *Unpossible*: no additional events were considered, but new parameters were added to the `player_death` events with information about the turns and keystrokes made by the user in each direction. Similarly, new parameters were added to determine the point on the curve where the player died in the corresponding attempt.

5 Predicting performance

The question arises as to whether it would be possible to estimate the intelligence of a player through shorter experiments, now that more information is available than in the original study. What we propose is to predict, by means of machine learning algorithms, the final results of a player in each game. We look only at the data corresponding to the first minutes of each session. Indeed, if considering a "truncated" experiment we are able to obtain a robust estimate of a player's final progress, it would be reasonable to assume that it is possible to reduce the duration of the sessions in the experiment without incurring a serious loss of information.

Following this pattern, in each game, we prepare new data from the "raw" events in the database, recording selected variables at certain time intervals. That is, we produce a "timed sequence" comprised of different metrics inferred from a given player's event trace. Since these parameters are game dependent and we tested different models, we will elaborate on each specific case:

- **Blek**: Blek's experiment lasts 10 minutes. We process the data from the first 5 minutes, at 3-second intervals. For each timestamp, we record: the current level, a Boolean variable that indicates whether the player is thinking or not (by "thinking", we refer to any moment/time when the player is not drawing a trace), and the number of curves attempted so far on the current level.
- **Edge**: In the case of Edge, the trial takes up to 12 minutes. We record the first 6 minutes of data, at 5-second intervals, due to the slower pace at which events take place in this game (checkpoints and collectable prisms are more separated than the distance that could be traveled in 5 seconds). The variables recorded at each timestamp are: the current level, the percentage

of checkpoints reached so far in the current level, the percentage of prisms collected so far in the current level, and the cumulative number of deaths so far in the current level.
– **Unpossible**: Of the 5 minutes duration of an experiment session, we collect data from the first 2.5 minutes, at intervals of 2 seconds. At each timestamp, we recorded the following 5 indicators: the number of deaths accumulated so far, the number of times the player has pressed the left/right arrow from the last death to the moment, and the amount of rotation in each direction from the last death to the present. A special aspect of this game that should be mentioned is that these last variables are only reported to the server when the player dies or the experiment is over, so we don't have their real values in each timestamp. To estimate them, we assume that they have a linear behavior, and we calculate the corresponding proportion at each moment.

The dimensions of the processed data are summarized in Table 1, where the first column describes the number of individuals that completed the required play time, the second the resulting number of records generated for every individual, computed from the sampling frequency and sampling time (Blek: 5 minutes, timestamp each 3 seconds; Edge: 6 minutes, timestamp each 5 seconds; Unpossible: 2.5 minutes, timestamp each 2 seconds), and the third column the number of values for every record.

	individuals	timestamps	variables
Blek	68	100	3
Edge	63	72	4
Unpossible	60	75	5

Table 1: Dimensions of the processed data

Table 2 summarizes the variables we are predicting in each game and the range of values they can take. In each case, we consider two baseline models: one based on linear regression and a second model that merely predicts the outcome as an average of the training dataset values. The metric used in this case is the Mean Absolute Error (MAE).

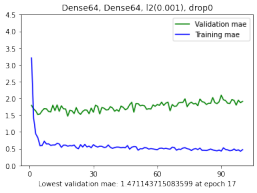
We first explored the use of neural network-based techniques [5, 6]. For each game, we train different models of both recurring and feed-forward networks. It should be noted that for the feed-forward models, the input should be pre-flattened so as to match the accepted data format. We split the data as follows: 75% for the training set and 25% for the test set. A K-fold cross-validation is used to select the optimal configuration. The hyperparameters we tuned include the *learning parameters* of the networks, from one single layer to three layers and from 8 neurons to 64 neurons per layer, considering the dataset's size; a *L2* regularization with coefficients in $\{10^i | -3 \leq i \leq 2\}$; a dropout between 0.2 and

	Variable to predict	Range	MAE linear regression	MAE mean prediction
Blek	no. levels	[1,26]	4.588	5.433
Edge	no. levels	[1,8]	0.605	1.084
Unpossible	no. deaths	[3,25]	5.043	3.627

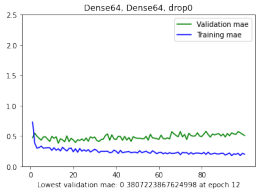
Table 2: Baseline models

0.5; and a learning rate in $\{10^i | -5 \leq i \leq -1\}$. The activation function employed is always *ReLu* and the optimizer *rmsprop*.

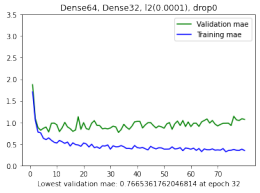
We include the results from the best performing models for each game in Tables 3 and 4 . Figures 4 and 5 show the learning curves corresponding to each model.



(a) Blek: 2 dense layers of 64 units per layer, with $l2(0.001)$ regularization, without dropout



(b) Edge: 2 dense layers of 64 units per layer, without regularization or dropout



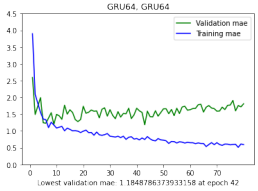
(c) Unpossible: 2 dense layers of 64 and 32 units per layer, with $l2(0.0001)$ regularization, without dropout

Fig. 4: Learning curves of the best performing feed-forward models

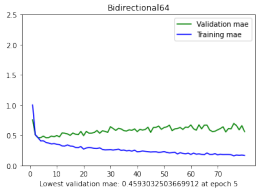
	Best performing feed-forward model	Epochs	MAE val	MAE test
Blek	2 layers, 64 units/layer, reg l2(0.001)	17	1.47	2.99 - 4.86
Edge	2 layers, 64 units/layer, no reg, no dropout	12	0.38	0.66 - 1.14
Unpossible	2 layers, 64,32 units, reg l2(0.0001)	32	0.76	1.53 - 2.17

Table 3: Best performing feed-forward neural network models

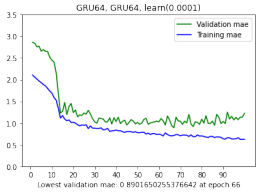
In a general overview, neural networks have greatly improved the reference models we considered, especially in Blek and Unpossible. In Edge the error reduction is not appreciated to a great extent, for the simple reason of it being



(a) Blek: 2 GRU layers of 64 units per layer, without regularization or dropout



(b) Edge: 2 GRU layers of 64, 32 units per layer, without regularization or dropout



(c) Unpossible: 2 GRU layers of 64 units per layer, without regularization or dropout, with 0.0001 learning rate

Fig. 5: Learning curves of the best performing recurrent models

	Best performing recurrent model	Epochs	MAE val	MAE test
Blek	2 GRU, 64 units/layer, no reg, no dropout	42	1.18	2.61 - 4.60
Edge	2 GRU, 64,32 units, no reg, no dropout	5	0.44	0.90 - 1.06
Unpossible	2 GRU, 64 units/layer, learning rate(0.0001)	48	0.88	1.59 - 1.92

Table 4: Best performing recurrent neural network models

already in a relatively small scale with a lower range. In fact, in the test suite a worse prediction is attained than with the linear regression model. By observing the graphs, all the models found show a fairly similar behavior: the learning curve of the training set decreases rapidly, while the validation remains stable after a few epochs, exhibiting a slight overfit. Generally, when the overfit is small and the error in the validation stalls, it is a sign that the model does not have sufficient expressive capacity. However, testing networks with more layers and/or neurons per layer has not resulted in a significant improvement.

Comparing feed-forward network models and recurrent network models, we spot no clear advantage in the use of the latter, which are theoretically better suited for serial data. Only a slight improvement is shown in the case of Blek. However, due to the high variance obtained on the test set, we cannot confidently state the dominance of recurrent networks. As a matter of fact, this large variance may indicate that we are dealing with a much more complicated and heterogeneous dataset than that of the other cases, or that the variables we included in the analysis do not facilitate the prediction.

Another general phenomenon is that all models perform worse on test data, having achieved quite promising results on the validation set. This difference is notably greater in Blek, whose error of around 1 on the validation data in the best models can reach a value of almost 5 on the test set. This problem may be due to the limited number of observations we have. In this case, the random division of the data into three subsets can greatly affect the results obtained.

For example, in a sufficiently large dataset, the divisions follow roughly the same distribution, presenting similar characteristics. Conversely, splits in a small data set may contain very different patterns that models are not able to learn just by looking at the training data. Therefore, instead of delving further into different neural network configurations, we turn to traditional machine learning techniques, which are known to perform better on a small dataset. The option we chose is the random forest [4].

We follow the same train-test split percentage and employ K-fold cross-validation. In the same way as the previous case, we tuned a subset of hyperparameters, including: the number of samples drawn to train each base estimator (tree) in the bagging method; the number of variables to consider in each split; the maximum depth of the trees; and the number of trees in the forest. The obtained results are presented in Table 5.

	validation MAE	test MAE
Blek	3.24 ± 0.37	2.86 - 3.15
Edge	0.91 ± 0.20	0.88 - 0.95
Unpossible	1.67 ± 0.39	1.42 - 1.74

Table 5: Best performing models of Random Forest

We observe that random forest models behave similarly or even better than neural network models. We can conclude that, with a dataset as small as this one, it is not necessary to resort to such complex deep learning techniques.

6 Conclusions and future work

In this paper we have presented preliminary results on the application of telemetry in video games to speed-up the measure of intelligence which has been previously demonstrated to correlate with performance in those games. Despite the dataset modest size, we were able to predict a player's final score in a given game from truncated play logs using neural networks and random forests. By avoiding the need for a human evaluator to collect data in such experiments we could potentially allow future studies to shorten experiment times, thus increasing the viability of game-based intelligence assessment. In the future, we plan to carry out further experiments in order to further validate this automatic approach.

In addition to the application of this approach to intelligence assessment, we envision the use of similar techniques for the dynamic adjustment of game difficulty. Since we are able to predict game performance, we can adjust the game based on the predicted performance of a particular player.

7 Acknowledgments

This work is partly supported by the Spanish Ministry of Economy, Industry and Competitiveness (TIN2017-87330-R).

References

1. Blek, http://www.blekgame.com/
2. Edge, https://en.wikipedia.org/wiki/Edge_(video_game)
3. Unpossible, https://apps.apple.com/us/app/unpossible/id583577503
4. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
5. Chollet, F.: Deep Learning with Python. Manning Publications Company (2017), https://books.google.es/books?id=Yo3CAQAACAAJ
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
7. Gottfredson, L.S.: Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography. Intelligence **24**(1), 13–23 (Jan 1997). https://doi.org/10.1016/S0160-2896(97)90011-8
8. Jones, M.B., Dunlap, W.P., Bilodeau, I.M.: Comparison of video game and conventional test performance. Simulation & Games **17**(4), 435–446 (1986)
9. McGrew, K.S.: CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. Intelligence **37**(1), 1 – 10 (2009). https://doi.org/https://doi.org/10.1016/j.intell.2008.08.004
10. Quiroga, M.Á., Escorial, S., Román, F.J., Morillo, D., Jarabo, A., Privado, J., Hernández, M., Gallego, B., Colom, R.: Can we reliably measure the general factor of intelligence (g) through commercial video games? yes, we can! Intelligence **53**, 1–7 (2015). https://doi.org/http://dx.doi.org/10.1016/j.intell.2015.08.004
11. Quiroga, M., Colom, R.: Videogames and Intelligence. Chapter 26. In: Sternberg, R.J. (ed.) Cambridge handbook of intelligence. Cambridge University Press, 2 edn. (2020). https://doi.org/10.1017/9781108770422
12. Quiroga, M., Diaz, A., Román, F., Privado, J., Colom, R.: Intelligence and video games: Beyond "brain-games". Intelligence **75**, 85–94 (2019). https://doi.org/https://doi.org/10.1016/j.intell.2019.05.001
13. Rabbitt, P., Banerji, N., Szymanski, A.: Space fortress as an iq test? predictions of learning and of practised performance in a complex interactive video-game. Acta Psychologica **71**(1-3), 243–257 (1989)