Determining competencies and interests from behavior in computer games

(Bachelorproject)

Alex Hamelink, s1798014, j.a.hamelink.1@student.rug.nl,
Marco Wiering* and Gidi van Liempd†

April 7, 2017

Abstract

Choosing the right education is an important decision for students. Currently there exist a wide variety of tests to determine which education matches the student's interests and competencies. Current education choice tests rely on filling in questionnaires to determine the student's personal properties. The goal of this research is to determine whether it is possible to get this information from game behavior. Specifically if behavior in games can predict competencies and interests. Subjects were asked to play three games to test their dexterity, spatial visualization ability and concentration. This behavior, as represented by game events was linked to responses to a personality test and their actual major choice. There was a successful classification on self reported math skill with an error rate of 28%, and also on whether the user was an AI student with an error rate of 38%.

1 Introduction

Choosing the right education is an important part of one's academic career, but it is a difficult decision for most students. Between the year 2002 and 2010 the number of students switching majors in dutch universities fluctuated between six and eight percent. The number of students leaving the university was between fifteen and eighteen percent (VSNU, 2012). A common tool to help in making the decision which major to choose is the use of questionnaires.

There are multiple career assessment tests available. One of the most widely accepted and most used is the SII (Strong Interest Inventory). According to research by Miller (2010) this test is valid:

> the Occupational Scales (OSs) of the SII will predict a participants exact academic major [...] for at least 35% of the sample.

Filling in these forms however, can be a boring and tedious process. To make this process more fun it was proposed to use games to determine which education choice matches the student, or at least which competencies and interests the student has. The website "Kids Future Plaza" with three games were created by Klassewijzer for this research. Because games are appealing to children and students this is seen as a good way to test their interests.

Besides questionnaires, other research on the choice of an education, specifically higher education majors, focuses on the influence of background. Factors like race and income of parents are taken into account. The most relevant for this research is how personality affects the choice of a major. Using Holland's personality model, Porter and Umbach (2006) found that personality and political liberalism are very strong predictors of student major choice. Other factors that are good predictors of major choice are self-efficacy (Bandura, 1997, 1986) and the belief that the student will be successful in the major that he has chosen (Eccles, 1987).

To summarise: choice of major can be predicted by these personal properties, and some external factors. If games can predict personality or skill has not been researched much besides the link between agression and games. However, determining player

*University of Groningen, Department of Artificial Intelligence
†Gee!Design, Briltil, gidi@geedesign.com

properties from games has been researched in the past. Research by Chen and Hong (2007) shows that intruders on a game account can be detected by comparing idle time distribution. This demonstrates the concept of inferring player properties from features extracted from game data. Unfortunately this is aimed at single users and does not apply to behavior in real life. Research on distinguishing bots from humans in online role-playing games has been done by van Kesteren, Langevoort, and Grootjen (2009). Their algorithm was very successful at this task. Even though the differentiation between humans and computers is different from distinguishing people with different skills and interests, their approach to use feature extraction can be useful in the context of this research.

One practical application of games in relation to real-world skills is using games as a learning tool. Current research is mostly on how well a game can contribute to skills and properties of gamers instead of assessing their skills from games, but this shows that games and skills applicable in the real world overlap.

The aim of this research is to find a possible link between game behavior and real-life behavior, so the research question is:

> Can behavior in games be linked to real-life competencies and interests, specifically expressed in major choice?

Traditional questionnaires make use of questions that aim to gather information about one specific interest or skill. When inferring this information from a stream of gameplay events it is much less clear what a specific action means, or what it says about the skill being assessed. For example: moving quickly in a game could be because a player is hasty in nature, or because he is very skilled at the particular game. Therefore a different approach was chosen. Many different features were extracted from the gameplay data and used to make a classification using a Support Vector Machine (SVM) (Cortes and Vapnik, 1995). This will be further explained in the methods section.

In section 2 the games that were used to gather the data are explained. The features that were extracted from this data are explained in section 2.5. In section 3 the results from statistics and classification with an SVM are reported. Section 4 covers the conclusions that can be drawn from the results and gives suggestions for future research.

2 Methods

Gameplay data was gathered from three games. Different games were selected to measure different skills and properties of the subjects. Behavior in games is a very broad concept, therefore different games with much variety between them were preferred to cover as much of the range of possible behavior in games. The look and feel of the games was designed to be like a futuristic testing environment: a clean, distraction-free room with easily distinguishable colored objects. The games also needed to be able to run in a web browser. The games were designed with Unity (Higgins, 2010).

2.1 Participants

Subjects were asked to play the games on their own computer and play them again at a later time to get as much data as possible per subject. Three movie vouchers were offered as a reward for the highest score in each game, in addition to one randomly awarded to one of the participants. There were 40 participants. Eight of them were students from Alpha College in Groningen, 27 were students at Rijksuniversiteit Groningen. Seventeen of these 27 students studied Artificial Intelligence. Five respondents were affiliated with Klassewijzer. Figure 1 shows that 29 participants were men and 11 were women. Figure 2 shows age varied between 18 and 48 years with a median of 23.

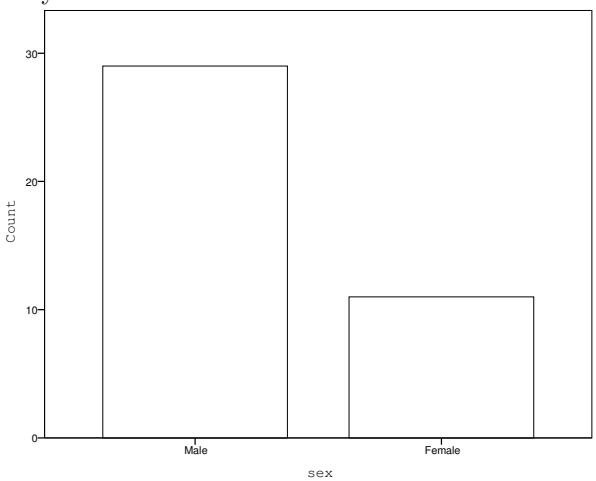


Figure 1: Number of participants by sex

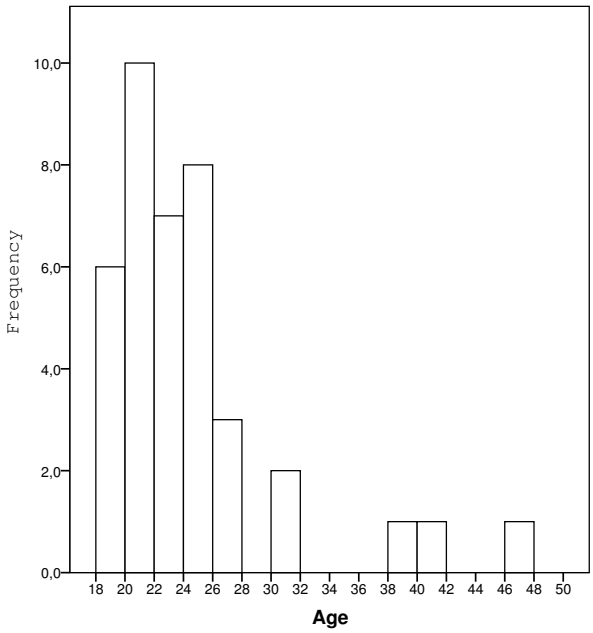


Figure 2: Number of participants by age

2.2 Dexterity game: Block avoidance

In this game the objective for the player is to keep a sphere from colliding with moving blocks by moving it across the screen. The sphere is controlled by pressing the arrow keys on a keyboard. The blocks that are to be avoided appear at the bottom of the screen and move upward at a continually increasing speed. The player starts with three lives. When the sphere controlled by the player collides with a block the player loses a life. At the start of the next life the speed remains the same as it was before the life was lost. After three lives the score, i.e., the number of seconds the player survived, is shown. This game is designed to test the dexterity of the player. In the screenprint of Figure 3 the player has been playing for 22,77 seconds.

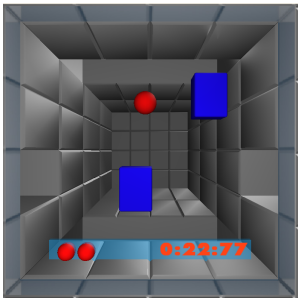


Figure 3: Dexterity game: Block avoidance

2.3 Puzzle game: Based on Rush Hour

The goal of this game is to move the red block to the exit of the board at the edge of the screen. To achieve this, the player has to move other blocks out of the way. This can be done by clicking and dragging them with the mouse. The blocks can only be moved in the direction normal to their shortest side. After completion of the goal the next level can be played. This game is included to test spatial awareness of the player as well as problem solving skill. See Figure 4 for a screenshot of the game.

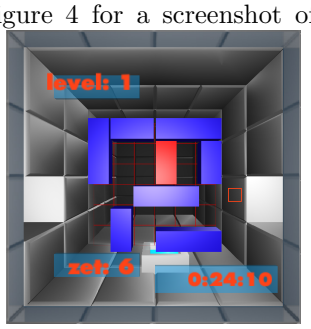
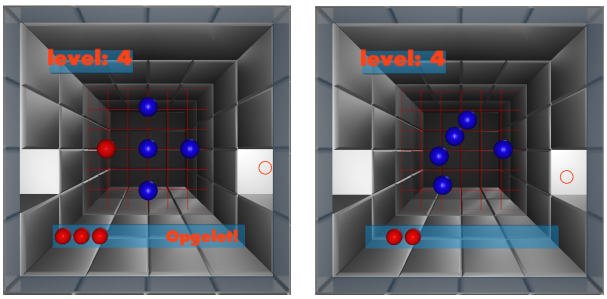


Figure 4: Puzzle game

2.4 Tracking game: Based on Shell game

This game tests how well a player can follow one moving sphere among other moving spheres. The game starts with a number of spheres on the screen. One sphere is highlighted in red, the others are blue. After a short period of time the red sphere stops being highlighted and becomes blue like the rest. The player must now track the sphere as it is being switched around. Switches occur between random spheres with the possibility of a fake switch for extra difficulty. In a fake switch the spheres are

not actually switched, but are moved towards each other slightly and are then quickly returned to their initial position. After a set amount of iterations of shuffling the spheres the player has to click the sphere that was highlighted at the start. The difficulty increases after each level by increasing the number of iterations and thereby the duration, the speed at which the spheres move and the percentage of fake switches. See Figure 5 for a screenshot of the game.



(a) At the start of the game the sphere to track is highlighted. In this case the leftmost.

(b) The spheres move around.

Figure 5: Tracking game: Based on Shell game

2.5 Features

When an important game event occurs, like the completion of a level or a game over, gameplay data is stored in the database. Features are derived from this data. The block avoidance game has the most user input because the player has to almost constantly move the sphere. Therefore many features were derived from the player input in this game. Each time a life is lost by the player 41 features are stored. These include time, average position and speed, number of times a key was pressed and collisions with walls. In total there are 180 features extracted from this game. For a more detailed description of these features see Appendix A. For the puzzle game there are 13 features extracted, for example the number of moves on each level and the time taken to solve the level. In the tracking game there are 27 features including the number of times the game was played, the highest level that was achieved and the number of times the player took too long to answer and was timed out.

2.6 Questions

In order to get a complete picture of the personality of the player 14 questions were asked. The list of questions used is described in Table 1. These questions were selected from a list of questions created by Porter and Umbach (2006) for their research on college major choice. The questions were from four personality types as theorized by Holland (1973). Questions for the investigative personality type were: 1, 3, 5 and 11. For social 2, 9 and 10, for artistic 7, 12 and 13. Finally for enterprising 4, 6, 8 and 14. Table 1 also shows the frequency of answers given to those questions. To be able to make a classification, questions with an approximately equal number of positive and negative responses are preferred. This means that questions 2, 3, 6, 7, 11 and 12 are not very useful for classification. The disagree and strongly disagree are added together to form one group and agree and strongly agree are added together to form the other group.

2.7 Support vector machine

A support vector machine was used to classify subjects. The main advantage of using an SVM is that many features can be used for the classification simultaneously. Subjects were divided into two classes based on their answers on the questions, and based on their college major. The questions where the number of students that agreed and disagreed were most similar were used. Question 1, 4, 8 and 13 have a distribution of responses that makes it good enough to use this question as a classification criterium. The features derived from their gameplay are mapped to a point in n-dimensional space, where n is the number of features. A linear kernel was used to compute the similarity between vectors. The SVM then finds an optimal linear separator between those points – the one that has the largest margin between it and the positive examples on one side and the negative examples on the other. Russell and Norvig (2003) describe the algorithm and formulas to accomplish this in their book. Cross-validation was used with 60% of the data in the training set, and the remaining 40% as the test set. After classification this returns an error rate between 0 and 1 that represents how many subjects are incorrectly classified.

	Question	Strongly disagree	Disagree	Agree	Strongly agree	Total
1	I am good at maths	5	5	12	13	35
2	I like spending my time helping others	0	1	18	16	35
3	I am intelligent	0	1	14	20	35
4	I like to tell others what to do	6	8	16	5	35
5	I strive to get to the top in my career	2	7	17	9	35
6	I like it when others show their appreciation for what I do	0	0	11	23	34
7	I want to use my creativity in my job	0	3	10	22	35
8	I work mainly to make money	10	10	13	2	35
9	I like to cooperate with others	1	4	18	12	35
10	I find it important to do work that is useful to society	2	7	18	8	35
11	I like learning new things	1	0	6	28	35
12	I like to give unexpected solutions	0	0	8	27	35
13	I have a new idea about something every day.	5	13	12	4	34
14	I want to be my own boss so I can decide what to do myself	2	8	19	6	35

Table 1: Questions and their frequencies

3 Results

Subjects were asked to play games on their own computers. In Figure 6 a histogram of the number of times that the dexterity game was played is shown. Most subjects only played the game once or twice. Specifically eighteen subjects played the game two times or less. Figure 7 shows a histogram of the total number of levels played in the puzzle game. It is possible to replay levels to finish them in less moves or less time. The figure shows that some subjects replayed levels since their total amount of levels played is above 40, which is the number of levels in the game. Figure 8 shows a histogram of the number of times that the tracking game was played. Note this is the number of times a new game was started, not the amount of levels played. Most subjects played each game once, only a few tried several times to achieve the high score.

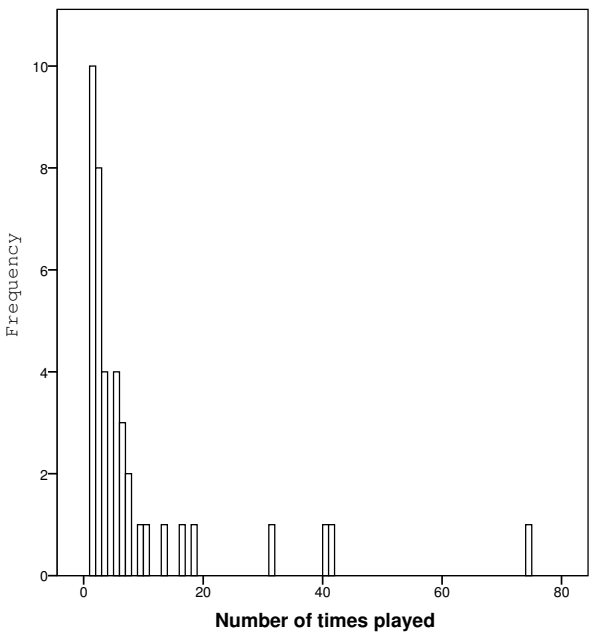


Figure 6: Histogram of times played the dexterity game

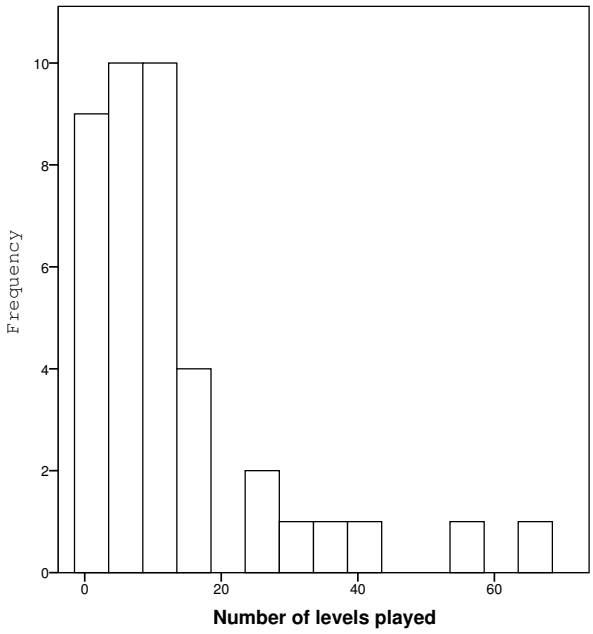


Figure 7: Histogram of levels played in the puzzle game

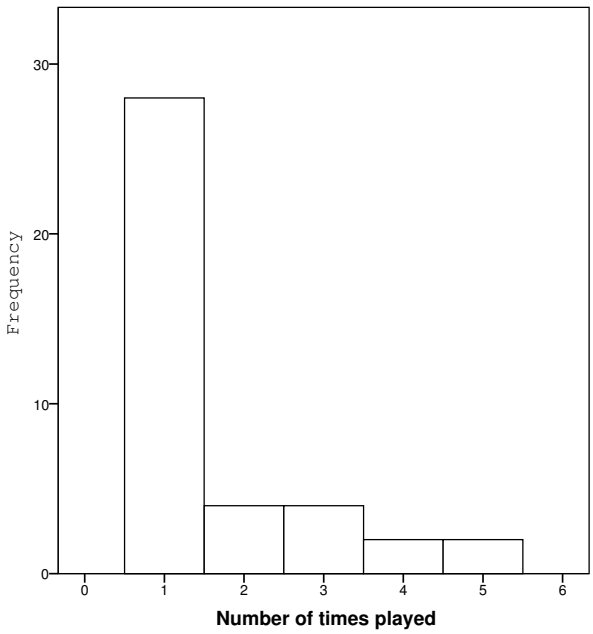


Figure 8: Histogram of times played the tracking game

3.1 Differences in game behavior between subject groups

Using students t-test we found a significant difference in game behavior for subjects who rate themselves as good at maths and those who do not. The number of times that the dexterity game was played, the number of levels and highest achieved level in the puzzle game and the highest achieved level in the tracking game yielded significant results. It was expected that subjects who rated themselves good at maths would perform better on the games because the games were expected to appeal more to beta-major students. Players that rate themselves as good at maths will play the dexterity game more often, score higher on the puzzle game, will play the puzzle game more often and are better in the tracking game. However, what we found was the opposite for all of these hypotheses. See Table 2 for the results of t-tests. We also hypothesized that subjects who say they work mostly for money, will play games more often. and subjects who say they have a new idea every day will reach higher levels on the puzzle game. For these hypotheses t-tests did not yield significant results. How well subjects rate themselves to be good at math seems to be a good predictor of a few types of game behavior. Subjects who said they worked mainly to make money (see question 8 in Table 1) did not differ significantly from subjects who said they did not in the hypothesized features. There also was no significant difference between subjects who said they have a new idea every day and those who said they did not.

3.2 Support vector machine

A support vector machine algorithm was used to classify players using the gameplay features. Users were classified using the features of each game separately using three classification variables: Whether they majored in Artificial Intelligence or something else, whether they rated themselves to be good at math or not and whether they had a new idea every day or not. The number of simulations used is 1000. Two sets of runs were done for each combination of features and classification variable. One run with learning enabled, and one with learning disabled to determine a null error rate to compare the SVM result to. In Table 3 the results are shown. The SVM classification shows that whether a subject reports

	Good at maths		Not good at maths		t-test results	
	mean	SD	mean	SD	T	p
Number of times played in the dexterity game	4.65	5.16	9.07	13.64	-2.71	0.01
Number of levels played in the puzzle game	5.1	4.09	17.12	16.08	-3.47	0.002
Highest achieved level in the puzzle game	4.8	3.7	15.16	11.86	-3.91	0.001
Highest achieved level in the tracking game	10.7	5.16	15.36	3.08	-3.308	0.02

Table 2: Results of t-tests

Table 3: Results SVM

Features used	Classification variable	Learning Enabled Mean Error ± S.D.	Learning disabled Mean Error ± S.D.	T	p
Dexterity game	AI or other	0.55 ± 0.12	0.5 ± 0	13.18	< 0.001
	good at maths	0.45 ± 0.1	0.5 ± 0	-15.81	< 0.001
	new idea	0.28 ± 0.09	0.28 ± 0.09	0	1
Puzzle game	AI or other	0.4 ± 0.1	0.5 ± 0	-31.62	< 0.001
	good at maths	0.37 ± 0.1	0.56 ± 0.03	-57.55	< 0.001
	new idea	0.57 ± 0.11	0.57 ± 0.04	0	1
Tracking game	AI or other	0.32 ± 0.1	0.5 ± 0	-56.92	< 0.001
	good at maths	0.28 ± 0.09	0.5 ± 0	-77.3	< 0.001
	new idea	0.56 ± 0.11	0.5 ± 0	17.25	< 0.001
All games	AI or other	0.38 ± 0.1	0.5 ± 0	-37.95	< 0.001

being good at maths is the best predictor across all games with error rates between 0.28 and 0.45. This was an expected result from the t-tests that were run earlier. Classifying AI students and non-AI students gave better results than chance level for the puzzle game, the tracking game and combined features of all games. In the dexterity game there was a mean error rate of $> 50\%$ which is unsuccessful. Classifying whether the subjects had a new idea every day was not successful for the features of any of the games.

4 Conclusion and Future Work

So can competencies and interests be determined from features that describe behavior in computer games? The results for the good and not good at math groups show that there is a difference in player behavior based on self reported math skill. The good at math group was expected to perform better and play more often, but results show a significant difference suggesting the opposite. The results from the SVM classification on the good at maths question confirm the difference in player behavior with the best error rate being 28%.

Concerning predicting college major choice, there were too many different majors with too few students to divide subjects in different groups per chosen major, so an SVM classification was run on the categories AI student and non-AI student. The best SVM classification resulted in an error rate of 32%. This was significantly better than chance level. This means that students can be classified as AI students, however 32% is a high error rate for a classifier.

4.1 Suggestions for future work

The games that were used for this research mostly require dexterity skills, spatial awareness and attention. A possible way to improve results would be to add games that require different skills. This way, more aspects of behavior can be recorded, and possibly matched to different majors. Another option could be to use different kinds of games. Modern role playing games usually offer multiple ways to solve problems and achieve goals in the game. Using these different approaches could be more closely

linked to interests, and real world behavior of subjects.

References

A. Bandura. Social foundations of thought and action: A social cognitive theory. 1986.

A. Bandura. *Self-efficacy: The exercise of control.* Worth Publishers, 1997.

K. Chen and L. Hong. User identification based on game-play activity patterns. In *NetGames 07: Proceedings of the 6th ACM SIGCOMM workshop on Network and System Support for Games*, pages 7–12. ACM, 2007. ISBN 978-0-9804460-0-5. doi: http://doi.acm.org/10.1145/1326257.1326259.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

J. S. Eccles. Gender roles and women's achievement-related decisions. *Psychology of women Quarterly*, 11(2):135–172, 1987.

T. Higgins. Unity-3d game engine, 2010.

J. L. Holland. *Making vocational choices.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

S. M. Miller. *The Validity of the Strong Interest Inventory in Predicting College Major Choice and Academic Achievement.* PhD thesis, Washington State University, 2010.

S.R. Porter and P.D. Umbach. College major choice: An analysis of personenvironment fit. *Research in Higher Education*, 47:429–449, 2006.

S. Russell and P. Norvig. Artificial intelligence, a modern approach, the intelligent agent book. *ISBN: 0-13-080302-2*, pages 749–754, 2003.

M. van Kesteren, J. Langevoort, and F. Grootjen. A step in the right direction: Botdetection in mmorpgs using movement analysis. In *Proc. of the 21st Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2009)*, 2009.

VSNU. Prestaties in perspectief, trendrapportage universiteiten 2000-2020, 2012.

5 Appendix A

Feature	Mean	SD
Time in hundredths of seconds	5155,08	2879,03
Average X position of player sphere (greater X means more to the right)	0,08	0,40
Standard Deviation in X position	0,50	0,24
AverageY position of player sphere (greater Y means more to the top of the screen)	2,85	0,57
Standard Deviation in Y position	0,23	0,18
Min x position	-0,80	0,54
Max x position	0,93	0,56
Min y position	2,08	0,17
Max y position	3,00	0,59
Average X speed	0,02	0,10
Standard Deviation in X speed	0,81	0,36
Average Y speed	0,08	0,13
Standard Deviation in Y speed	0,36	0,30
Number of times an Up move was made	1,03	1,01
Average time per Up move (in seconds)	0,40	0,26
Standard deviation in time per Up move	0,03	0,09
Number of times a Down move was made	0,27	0,81
Average time per Down move (in seconds)	0,06	0,16
Standard deviation in time per Down move	0,01	0,04
Number of times a Left move was made	7,62	8,59
Average time per Left move (in seconds)	0,39	0,19
Standard deviation in time per Left move	0,16	0,12
Number of times a Right move was made	7,75	8,93
Average time per Right move (in seconds)	0,40	0,18
Standard deviation in time per Right move	0,16	0,12
X position of player sphere (center) at time of collision	0,14	0,66
Y position of player sphere (center) at time of collision	2,94	0,63
X position of colliding block (center) at time of collision	0,13	0,69
Y position of colliding block (center) at time of collision	2,52	0,85
Number of times a collision with the Top plane occurs	0,60	0,53
Average time per collision (in seconds)	11,45	18,05
Standard deviation in time per collision	0,07	0,75
Number of times a collision with the Bottom plane occurs	0,00	0,07
Average time per collision (in seconds)	0,01	0,12
Standard deviation in time per collision	0,00	0,00
Number of times a collision with the Left plane occurs (Left = as the player looks)	0,37	0,79
Average time per collision (in seconds)	0,36	1,26
Standard deviation in time per collision	0,08	0,47
Number of times a collision with the Right plane occurs	0,60	0,99
Average time per collision (in seconds)	0,88	2,02
Standard deviation in time per collision	0,19	0,90