An Adaptive Robot Game

Hansen, Søren Tranberg; Svenstrup, Mikael; Dalgaard, Lars

Published in:
Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)

Publication date:
2010

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Hansen, S. T., Svenstrup, M., & Dalgaard, L. (2010). An Adaptive Robot Game. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)* (pp. 1-8). IEEE Press.

An Adaptive Robot Game

Søren Tranberg Hansen
Centre For Robot Technology, Danish Technological Institute
Forskerparken 10, 5230 Odense, Denmark

Mikael Svenstrup
Automation and Control, Aalborg University
Fredrik Bajers Vej 7, 9220 Aalborg Ø, Denmark

Lars Dalgaard
Centre For Robot Technology, Danish Technological Institute
Forskerparken 10, 5230 Odense, Denmark

Abstract

The goal of this paper is to describe an adaptive robot game, which motivates elderly people to do a regular amount of physical exercise while playing. One of the advantages of robot based games is that the initiative to play can be taken autonomously by the robot. In this case, the goal is to improve the mental and physical state of the user by playing a physical game with the robot. Ideally, a robot game should be simple to learn but difficult to master, providing an appropriate degree of challenge for players with different skills. In order to achieve that, the robot should be able to adapt to the behavior of the interacting person. This paper presents a simple ball game between a single player and a mobile robot platform. The algorithm has been validated using simulation and real world experiments.

1 Introduction

Based on the demographic development in most western countries, it has been predicted that the number of people with mental and/or physical disabilities will increase while the amount of people to take care of them will decrease [19], [1]. Digital games hold a significant promise for enhancing the lives of seniors, potentially improving their mental and physical wellbeing, enhancing their social connectedness, and generally offering an enjoyable way of spending time [9]. It has been shown that mental and physical health can be improved through a small amount of physical exercises [17], [7], and e.g. Nintendo Wii has been suggested as a means to increase physical activity among elderly [2], [13].

In this paper we introduce a physical game which is facilitated and initiated using a mobile robot. A principal question is how to design a robot based game which ensures engagement of the participating players. Many known games are derived from a pursuit-evasion scenario e.g. the child games robbers and cops and the game of tag [15]. In this paper, we describe a simple pursuit and evasion problem played between a single player and a mobile robot. The robot will initiate the game by searching for a potential player in a room and hand over a ball. After that, the player should try to hand back the ball, while the robot should try to avoid receiving the ball.

Motivating elderly to move physically by playing a game is related to Persuasive technology which is defined as technology designed to change attitudes or behaviors of the users through persuasion and social influence, but not through coercion [5]. A similar term is Captology, which is an acronym for computers as persuasive technologies [6]. This term however, is not used as often as Persuasive Technology or Persuasive Design which is the term we will use here. Successful games are often characterized using the concept Flow, as proposed by Csíkszentmihály [4]. Flow is a mental state which can occur when there is an appropriate balance between challenge and skill. As the cognitive and physical capabilities of the users are expected to vary from each person, the robot should adapt the difficulty of the game to the end user.

The goal is to motivate elderly to do physical exercises in a fun and social manner by facilitating and initiating a simple physical game using a robot. The robot automatically initiates the game by autonomously approaching the user. This is a difference from using video games like Nintendo Wii, which facilitate games but does not itself initiate a game.

In this paper, we first outline the theory about persuasive design and the concept of flow. Next we explain the game algorithm, and demonstrate how it works when implemented in a physical robot.

2 Theory

The fundamental concept of Persuasive Design (PD) is persuasion, which is defined by Fogg as an attempt to change attitudes or behaviors or both (without using coercion or deception) [5]. The theoretic background is based on Computer Science and Social Psychology and has been developed mainly through empiric studies. Results from HCI, has shown that e.g. a computer can act as a social character, because it has some characteristics which make us behave as if it was a real person. We know it is piece of a technology, but can still feel happy about it or get angry with it [14]. According to Fogg, this effect is amplified if the system shows social signals we know from interaction with other people, and even more so if the system has a personality the reminds us of our own.

Figure 1 shows how PD has been defined as a field where persuasion and computer technology overlap. The model was introduced in 1997 and has been continuously enhanced as new technologies emerge [6]. The focus of the theory is technology designed with persuasive intention.

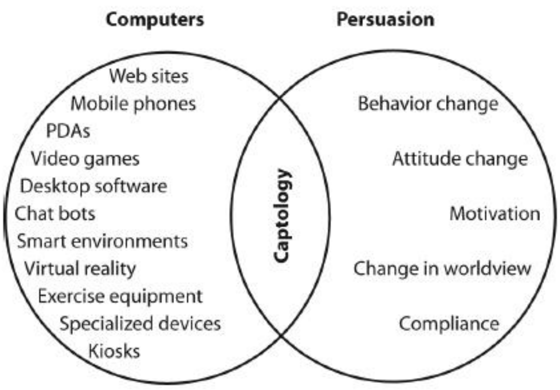

Figure 1: Illustration of how Captology (or PD) is defined as an overlap between computers and persuasion

We here extend the list of technologies in Figure 1 to also include robots. The robot should act a social character which invites the users to play a physical game. By doing this, the user will be more mentally and physically active than would be the case without the robot. In order for people to be motivated to play the game it should appeal to the specific user. Ideally the player should be in the state of flow while playing, being a feeling characterized by great absorption and engagement as proposed by Csíkszentmihály [4]. As illustrated in Figure 2, flow cannot occur if the task is too easy or too difficult.

In the state T1, your skills are not developed, but the challenge is not impossible. The difficulty of the challenge is in an appropriate relation to your (undeveloped) skills, and you are in a state of flow. T2 is the situation where you develop your skills to a level where the challenge becomes too easy and therefore boring. In T3, the difficulty of the challenge is higher than your skills. This leads to discontent and frustration. Common for the state T2 and T3 is that in the long run, they are unsatisfying. In order to enter the state of flow, the difficulty of the challenge has to be changed or you have to improve your skills. T4 is also a flow state, but in a more complex situation than in T1. It is not a stable state because your skills will keep developing [10].

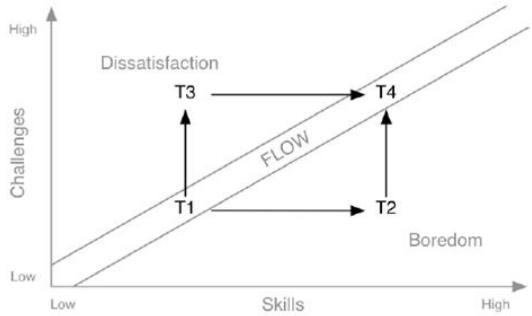

Figure 2: Illustration of the relation between skill and challenge. In state T1 and T4, there is a balance between skill and challenge and the player is in the state of flow. In state T2 and T3 there is no balance, and the player is either bored or frustrated, correspondingly

3 Implementation of the game

The game presented here, is based on a simplified pursuit and evasion scenario with a single pursuer (a human player) and a evader (a mobile robot). The player should try to hand over a ball to the robot, while the robot should try to avoid receiving the ball. Depending on the skill of the player, the robot should make it more or less difficult to hand the ball back. During a game, the robot can be in the following states:

- Roaming. If no player is detected, the robots should search for a player by moving randomly around until a person is spotted.
- Approach. When a player is detected, the robot invites to play a game by approaching the player from the front.
- Avoid. When the player has accepted to play a game by picking up the ball from the robot, the robot initially moves fast backwards away from the player for 3 seconds.
- Evaluating. The robot keeps avoiding the player with a distance and velocity that corresponds to the estimated skill of the player. When the ball has been handed back, the game is complete and the robot will go to the state Avoid and thereafter Roaming.

A more detailed outline of the game algorithm is sketched in Algorithm I.

Algorithm I
Main()

```
loop
    Roam()
    Approach()
    if Ball just picked up then
        Avoid()
```

```
6:     end if
7:     GameResult = Evaluate()
8:     UpdateCbrDatabase(GameResult)
9:  end loop
Roam()
10: while Person not detected do
11:     drive randomly around
12: end while
Approach()
13: while Ball is on the robot do
14:     Approach player to invite to a game
15: end while
Avoid()
16: Avoid player for 3 seconds
Evaluate()
17: PSI = 0.5
18: while Time not expired do
19:     Move according to PSI value
20:     Update PSI using CBR database
21:     if Ball returned to robot then
22:         Avoid()
23:         return Positive
24:     end if
25: end while
26: return Negative
```

3.1 Player Skill Indication (*PSI*)

In order for the robot to adapt the challenge to the individual player, is should have a have an estimate of the player's skill but also information about the specific player's style of playing, i.e. the physical behavior pattern of the player. The skill of a player is annotated using the parameter *PSI*, which means Player Skill Indication. $PSI \in [0;1]$ is a fuzzy predicate, which gives an indication of the skill of the current player. When $PSI \approx 1$, the robot believes the player is skilled, and that the player is likely to complete a game within a fixed evaluation period. When *PSI* is close or equal to 0, the robot thinks the player is less skilled, and thereby less likely to complete the game within the time period. *PSI* is updated continuously throughout the game (line 20 in Algorithm I) , so when a specific player gets better at playing *PSI* will increase.

The rate by which *PSI* increases or decreases depends not only on the skills of the player, but is a function of the learning rate parameter, L. The learning rate L is set high if you the game should adapt quickly to changes in the player's skill, but low otherwise.

3.2 Learning usning Case Based Reasoning (CBR)

The robot should learn about the specific player's style of playing with the robot, and therefore the skill is associated with the physical spatio-temporal behavior of the person. To incorporate the ability to learn the behavior pattern of the player, we have selected to use Case Based Reasoning (CBR). CBR allows recalling and interpreting past experiences, as well as generating new cases to represent knowledge from new experiences [12]. CBR has been proven successful solving spatial-temporal problems in robotics in [11] and is characterized by its adaptiveness, making it well suited for implementing an adaptive behavior on a human interactive robot. The CBR system is basically a database describing each encounter. Specifying a case in CBR is a question of determining a distinct and representative set of features, in our case *PSI*, position, pose and the id of the person. While playing, cases are continuously inserted, retrieved and updated. In other words, the robot adjusts the challenge to how the player moves around the robot when playing.

The behavior of the player is evaluated through a continuous registration of the players position and orientation of the body, which is inferred from 2D laser range measurements as explained in [16]. To detect persons the robot rely on the scans from the laser range finder using the leg detection algorithm presented in [18]. The algorithm is further supported by a Kalman filter for tracking and estimation of the person pose [16]. A more detailed description of the CBR database implementation can be found in [8].

3.3 Adaptive Robot Motion

The robot's navigation system is modeled using a person centered potential field, where the robot seeks towards the lowest values using a gradient descent. The potential field is calculated by the weighted sum of four Gaussian distributions of which one is negated. The covariance of the distributions are used to adapt the potential field according to *PSI*. When the player is inexperienced, the *PSI* values registered for a player will be closer to 0 and the robot will try to approach the player so he/she can hand over a ball to the robot. In the extreme case with $PSI = 0$ (an unskilled player), the potential field will like look Figure 3, and the robot will enter the dark blue space right in front of the player. On the other hand, if the player is skilled, *PSI* will be closer to 1 and potential file will look like illustrated in Figure 4. The robot will try to avoid the player by moving towards the dark blue area away from the player, making it more difficult for the player to hand over the ball.

Since the potential field is person centered, it moves with the player. If e.g. a skilled player starts moving towards the robot, the robot will eventually be in the yellow or red area in front of the player. The result is that the robot will start moving backwards towards the dark blue area, thus avoiding the player.

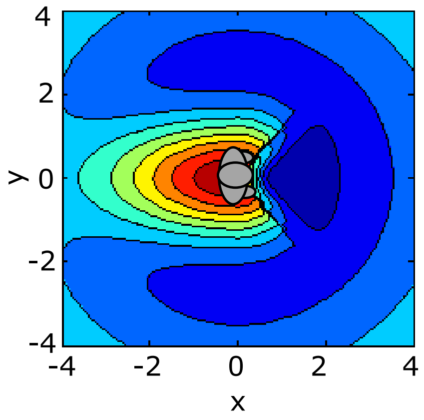

Figure 3: The player's skill $PSI = 0$, and the robot will seek the dark blue area in front of the player

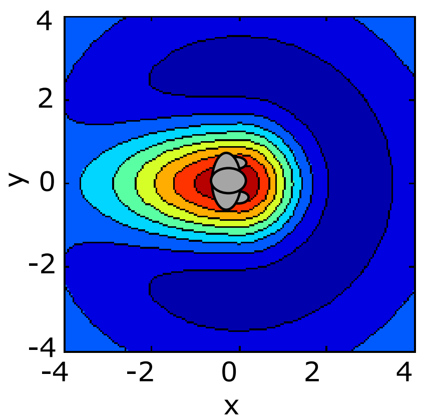

Figure 4: The player's skill $PSI = 1$, and the robot will seek the dark blue area a bit away from the player

4 Experiments

A series of experiments have been designed to demonstrate the following features of the implemented system:

1. The learning capability of the CBR database. The generation of cases in the database.
2. Adaptiveness of the system. How the CBR database adapts to the skill of a player.
3. The estimate of the *PSI* of a player.
4. Adaptive Navigation. How the motion of the robot depends on the player skill.
5. Effect of Learning Rate. How the learning rate affect the *PSI*.

4.1 The learning capability of the CBR database.

This experiment should show that cases are actually created in the database and that these cases reflect the behavior of the player. To avoid a huge amount of repetitive playing time, simulations using the Player/Stage environment have been used to train the database. Using an empty CBR database, first the database is trained by a skilled player. Afterwards, a new database is created which is trained by an unskilled player.

4.2 Adaptiveness of the System

This experiment have been done using a combination of simulations and real world experiments. The two databases from the former experiment are used in a real world setting, where a test person is playing against the system. To show the system is capable of adapting to a new situation, an unskilled player plays with the system trained for a skilled player and vice versa. The average value of *PSI* in the whole database is logged continuously during the experiments.

4.3 Estimate of *PSI*

To show that the robot is able to estimate the player *PSI*, and hereafter adapt the motion accordingly, the trained databases are used again in the real world setting. This time the motion of both the person and the robot are recorded.

4.4 Effect of Learning Rate

A central parameter of the game algorithm, is the learning rate L which is a numeric value in the interval $0 - 100$ used to control how fast *PSI* should adjust the estimate of the player's skill *PSI*. A simulation has been designed, such that in the first 10 games, the player needs 4 evaluation periods before he/she manages to hand the ball back to the robot. The effect should be that most of the cases in the database have a relatively low *PSI* value reflecting an unskilled player. In the last 40 games, the simulation has been changed so the player hands back the ball to the robot within one evaluation period representing the behaviour of a skilled player. To demonstrate the effect of changing the learning rate, the same simulation setup has been tried with the learning rate set to $0, 20, 40, 60, 80$ and 100.

4.5 The robot platform

The robotic platform, which forms the basis of the experiment, is shown in Figure 5. The robot is FESTO and is called Robotino. The robot is equipped with a head having 126 red diodes (see Figure 5) which enables it to express different emotions. The robot is 1 meter high, and has mounted an URG-04LX line scan laser placed $35cm$ above ground level, scanning 220 degrees in front of the robot. In order to get feedback from the test person and find out when the robot has the ball, a cup with an on/off switch in the bottom, has been placed just below the robot's head, $75cm$ above ground level. The software framework Player [3] is installed on the platform and used for control of the robot and implementation of the CBR system.

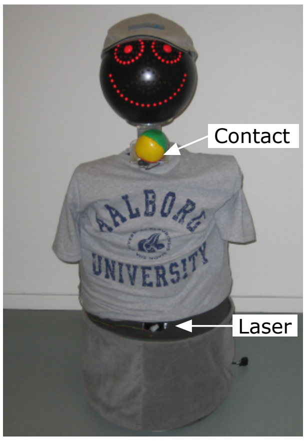

Figure 5: The modified FESTO Robotino robotic platform.

5 Results

5.1 Trag the CBR System

Figure 6 shows a plot of the CBR database, when the robot has been trained by an unskilled player. The position of the robot is $(0, 0)$. Each case in the database is represented by a short vector extending from a black dot in the figure. The color of the vector represents the value of *PSI* for the corresponding case. The pose and position of the player is represented by the corresponding position of the dot and angle of the vector. In Figure 6, most vectors are in the color span between blue and green which represents *PSI* values between 0 - 0.5, and the average of all *PSI* values is 0.25. This *PSI* range is as expected for an unskilled player, and it shows that the CBR system is capable of being trained for an unskilled player. Furthermore it can be seen that the database the is more densely populated closer to the robot. This is also expected, since a player will start off at a random direction away from the robot and will always move towards a point just in front of the robot.

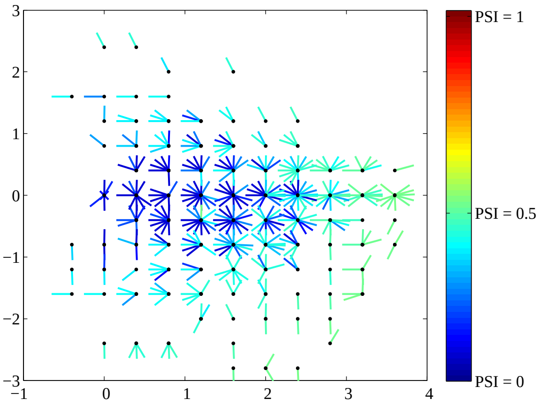

Figure 6: A plot of the trained CBR database for an unskilled player. Each vector represents a case in the database using the features pose and position. The color of the vector denotes the *PSI* value using the color scale to the right. The robot is positioned in $(0, 0)$

Similar results have been obtained, when training with a skilled player. Here, most vectors are in the color span between green and red which represent *PSI* values between 0.5 - 1. This is expected for a skilled player and the average *PSI* of the whole database is 0.74

5.2 Adaptiveness of the System

Using the trained database in Figure 6, a skilled player is set to play in the real world. This makes the CBR database turn into the one shown in Figure 7. It can be seen that the database has adapted to the player's skill, and has started to contain higher *PSI* values. Especially in the areas close to the robot, the *PSI* values have changed which is expected as most case updates happens here.

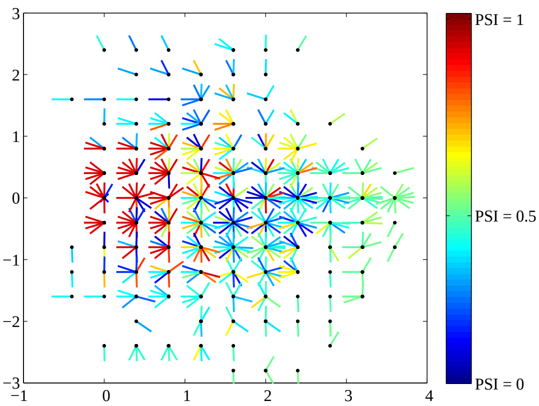

Figure 7: A plot of the trained CBR database in Figure 6 after the player has started to become good. The robot is positioned in $(0, 0)$.

In Figure 8 the development of the average value of *PSI* in the database can be seen. The value is saved for each lookup in the database, i.e. each time Line 20 in Algorithm I is passed. Initially the value is slightly less than 0.5. After the database has been trained by an unskilled player for a while (after around 12000 iterations) the average value has stabilized around $PSI = 0.25$. Now a skilled player starts playing, and it can be seen that the average value starts to increase rapidly, as expected. The noise on the figure is caused by individual trajectory differences.

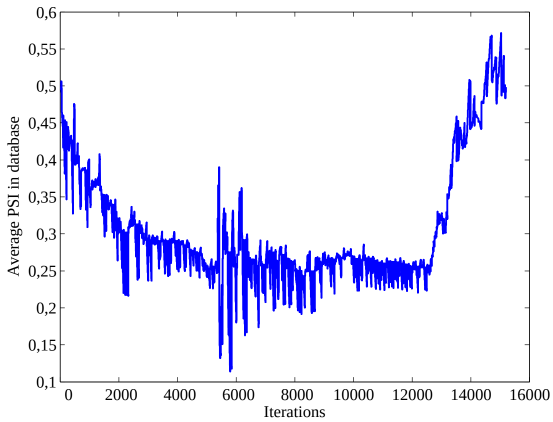

Figure 8: The development of the average value of all *PSI* in the database after each iteration (each look up in the database). Initially *PSI* is around 0.5 and the game is played by an unskilled player. The player slowly improves his skills and the average value of *PSI* increases correspondingly.

A similar result has been obtained when starting with the system, which was trained by a skilled player, and played in the real world by a unskilled player. Here the database values are adapted from relative high values to lower values. This could be the case if a player starts to have more severe physical disabilities caused by e.g. a stroke.

5.3 Adaptive Navigation

Figure 9 shows the trajectory of a person and for the robot, for a game where the robot has been trained by a skilled player. The person starts from the right side, and goes to pick up the ball. Hereafter the robot and person moves away from each other. When the game starts, the robot approaches slowly because it is far away. But as soon the person comes too close (when the trajectory is orange), the robot starts to move away. The player then tries to cheat the robot by moving sideways. This is a behavior the robot has not learned yet, and therefore it lets the person approach a bit more.

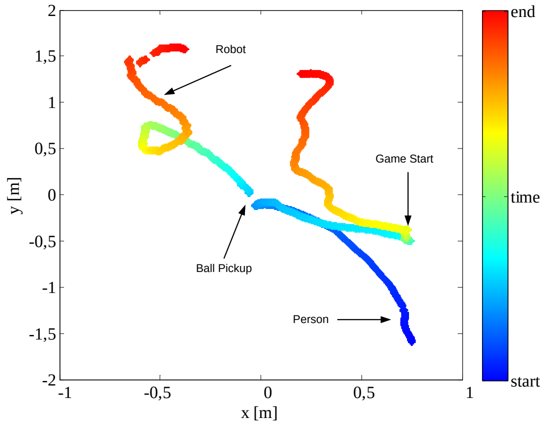

Figure 9: The trajectory of a person and the robot for a game, where the robot has been trained by a skilled player. The color of the trajectory defines the time. Blue is the beginning and red is the end of the game.

Figure 10 shows the trajectory of a person and a robot for a game where the robot has been trained by an unskilled player. The person starts from the lower right corner, and goes to pick up the ball. Hereafter the robot and person moves away from each other. Towards the end of the game, the robot approaches the person to make it easier for the player to hand back the ball. These two experiments show that the system is able to estimate the *PSI* correctly and navigate accordingly.

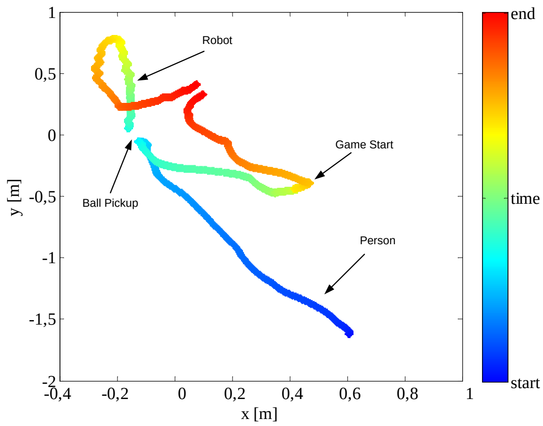

Figure 10: The trajectory of a person and the robot for a game, where the robot has been trained by an unskilled player. The color of the trajectory defines the time. Blue is the beginning and red is the end.

5.4 Effect of Learning Rate

Figure 11, shows the *PSI* values in the database for simulation set of 50 games with one player using a learning rate set to $L = 40$. In total, 249 cases are stored in the CBR database. Because the cases in the database are stored in the order they were observed, two consecutive cases do not necessarily have anything to do with each other and large fluctuations occur. A moving average gives an overview over what is happening, and as can be seen from the figure, the values of *PSI* decreases due to a gradually better trained database. Then, around case 100, there is a sudden increase in *PSI*. This corresponds to the time when the player changes behavior from being unskilled to skilled and manage to hand the ball back in one evaluation period.

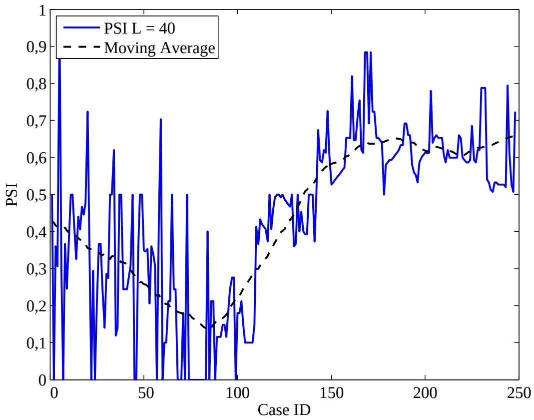

Figure 11: Shows the development of the player skill *PSI* in a simulated set of 50 games with one player using a learning rate on $L = 40$

Table 5.4, shows the same scenario with a learning rate set to $L = \{0, 20, 40, 60, 80, 100\}$. As L increases, the deviation also increases which is expected. When the learning rate is 100, *PSI* is adjusted with every little change of player behavior. The fluctuations of *PSI* becomes high, which makes the game algorithm too varying to be usable. On the other hand, *PSI* stays constant at 0.5 when the learning rate is equal to 0. The robot simply does not learn from its experience, and the game algorithm will never adapt the challenge to the player. The development of *PSI* using a learning rate of 0 and 100 is illustrated in Figure 12. It has been chosen to use a learning rate of $L = 40$ for all experiments, since this value gives an adequate balance of adaptability and stability.

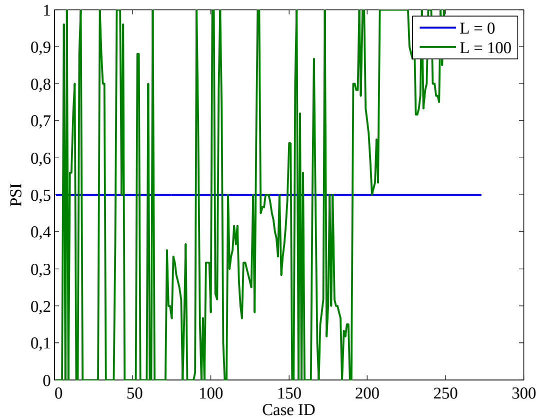

Figure 12: Shows the development of the player skill *PSI* in a simulated set of 50 games with one player using a learning rate on $L = 0$ and $L = 100$

L	n	σ
0	273	0.00
20	248	0.18
40	249	0.23
60	259	0.27
80	230	0.29
100	251	0.38

Table 1: The table shows the development of *PSI* for different learning rates (L) with respect to the number of cases in the database (n) and the standard deviation σ

.

6 Conclusion

In this paper, we have presented the concept of a robot based game. The function of the game is to increase the health of the players by motivating them to do a regular amount of physical exercise in a fun and social manner. The fact that the robot autonomously initiates the game, is a major difference from similar types of technology driven approaches e.g. using Nintendo Wii which has been successfully applied in a nursing home setting. The robot game is based on a simplified pursuit and evasion scenario, where the player should try to hand over a ball to the robot while the robot should try to avoid receiving the ball. Changing the behavior of the user through the use of technology is related to the term Persuasive Technology which is explained in the first section of the paper. The term Flow is often used to describe an ideal user experience in games. As described, one of the primary requisites of Flow is to provide an appropriate relation between game challenge and the user's skills. Based on this fact, a game algorithm has been designed and it is outlined how the algorithm works. The algorithm is implemented in a physical robot and the game is validated in simulation and

through a practical lab experiment setup. Trajectories of the player and the robot in the lab experiment are documented along with plots of the CBR database which form the basis of the learning algorithm. The experiments document the learning capability of the CBR database and the adaptiveness of the system. It also shows how the system estimates the skill of the player and how it adapts depending on the learning rate parameter.
The next step will be to do a real world experiment at an activity center for elderly with the goal of measuring the user feedback and experience. Also we will work on enhancing the game, so the robot game is based on multiple players which will strengthen the social aspect of the game.

References

[1] A. Alzheimer's Disease International. The prevalence of dementia worldwide. Technical report, Alzheimer's Disease International, The International Federation of Alzheimer's Disease and Related Disorders Societies, Inc., 2008.

[2] R. Brown, H. Sugarman, and A. Burstin. Use of the nintendo wii fit for the treatment of balance problems in an elderly patient with stroke: A case report. *International Journal of Rehabilitation Research, Proceedings of the 10th Congress of the European Federation for Research in Rehabilitation*, 32:109, 2009.

[3] T. Collett, B. A. MacDonald, and B. P. Gerkey. Player 2.0: Toward a practical robot programming framework. In C. Sammut, editor, *In Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*, Sydney, Australia, December 2005. http://playerstage.sourceforge.net.

[4] M. Csikszentmihalyi. *Beyond boredom and anxiety*. Jossey-Bass Publishers, 1975.

[5] B. Fogg. *Persuasive Technology. Using Computers to Change What We Think and Do*. Morgan Kaufmann, 2003.

[6] B. J. Fogg. Captology. the study of computers as persuasive technologies. *Proceedings of the CHI 97, Extended abstracts on Human factors in computing systems*, New York: AMC Press:129, 1997.

[7] D. K. R. Fox. The influence of physical activity on mental well being. *Public Health Nutrition*, 2:411, 1999.

[8] S. T. Hansen, M. Svenstrup, H. J. Andersen, and T. Bak. Adaptive human aware navigation based on motion pattern analysis. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, Toyama, Japan, September-October 2009.

[9] W. IJsselsteijn, H. H. Nap, Y. de Kort, and K. Poels. Digital game design for elderly users. *Proceedings of the 2007 Conference on Future Play*, Future Play '07:17–22, Toronto, Canada, November 14 - 17, 2007.

[10] F. L. Jensen. Flow & læringsspil. Master's thesis, Aalborg University, 2006.

[11] I. Jurisica and J. Glasgow. Applying case-based reasoning to control in robotics. In *3rd Robotics and Knowledge-Based Systems Workshop, St. Hubert Quebec*, 1995.

[12] J. Kolodner. *Case-based Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[13] C. Neufeldt. Wii play with elderly people. *international reports on socio-informatics*, 6, 2009.

[14] M. E. Pertou and S. D. Iversen. Persuasivt design i retorisk perspektiv. *RhetoricaScandinavica*, 49/50, 2009.

[15] C. W. Reynolds. Competition, coevolution and the game of tag. In *Artificial Life IV*, 1994.

[16] M. Svenstrup, S. T. Hansen, H. J. Andersen, and T. Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation. ICRA 2009.*, pages –, Kobe, Japan, May 2009.

[17] G. Wendel-Vos, A. Schuit, E. Feskens, H. Boshuizen, W. Verschuren, W. Saris, and D. Kromhout. Physical activity and stroke. a meta-analysis of observational data. *International Journal of Epidemiol.*, 33(4):787–98, 2004 May 27.

[18] J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes. Fast line, arc/circle and leg detection from laser scan data in a player driver. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, pages 3930–3935, 18-22 April 2005.

[19] H. Zlotnik, editor. *World Population Prospects - The 2004 Revision, Highlights*. United Nations, Population Division/DESA at www.unpopulation.org., 2005.