MŰEGYETEM 1782

Recognition of dementia types by computer games

Student Research Paper

written by: Nóra Gáspár

M6Y1PX

26 October 2012

Supervisors

dr. Péter Hanák, Healthcare Technologies Knowledge Centre / Department of Process Control and Informatics, BME

dr. Gábor Csukly, Department of Psychiatry and Psychotherapy, Semmelweis University

Table of content

Introduction

When an average European adult is asked about cancer we can nearly be sure that he already has some knowledge about the illness, he is well aware of the risks, and he has already attended some medical screening. Unfortunately when it comes to Alzheimer or dementia, we can not be that confident to receive the same answers.

With the significant increase in average life span, the age related mental decline and the Mild Cognitive Impairment (MCI) which typically precedes it affect an increasing number of people. Although this illness can not be cured yet, an early diagnosis can slow down the process, helping the patient with dementia to live a quality life.

Unfortunately a huge amount of elderly people live with undiagnosed dementia, either because they have never attended any official medical screening, or because they had been misdiagnosed.

The M3W project[1]

The main objective of the M3W (Maintaining and Measuring Mental Wellness) project is to develop a mental wellness toolset for self usage (i.e. for the individuals and their families), and only to a lesser extent for the medical experts (psychiatrists, psychologists, carers, etc.). The project's goal is to measure and visualize mental changes, tendencies in an entertaining way, and to give indications, warnings, alarms or reports, to the affected people and their environment that it is advisable to visit a physician. [4]

The project develops many different games that can be played independently through a framework program. This framework collects data, and sends it to the server. In the "Early Pilot"

[1] The M3W (Maintaining and Measuring Mental Wellness) project has started in December 2012 with the financial support of the Ambient Assisted Living Joint Programme and the national research, development and innovation agencies – in Hungary, the National Innovation Office and the National Development Agency – and lasts till November 2014. The project is led by the Healthcare Technologies Knowledge Centre of the Budapest University of Technology and Economics, with the contribution of the Department of Measurement and Information Systems of the same university, the Department of Psychiatry and Psychotherapy of the Hungarian Semmelweis University, the Hungarian Gaudiopolis Elderly Home, the Hungarian Silver Publishing Ltd., the Swiss Zurich University of Applied Sciences, the Luxembourgian Actimage Co. and the Greek Frontida Zois Home Care company.

stage, game-developers prepared different games with different approaches to the topic in question. Some of the games are more entertaining (for example an implementation of the Solitaire game), while others are more close to a psychological test.

Early Pilot

In the summer, we had our first Early Pilot tests, where my task was to teach the games to mentally healthy elderly users at the Gaudiopolis Retirement Home. In this essay I have used many experiences from these tests. In the following I will refer to these experiences, as the “test situations”.

My objectives

My objective in relation to this project is to create test games that are based on famous psychological tests, so their results can be used as standards.

In the previous semester in my Project Laboratory, I prepared a game based on the well-known psychological test named ”Corsi-cubes”, which measures the capacity of the spatial working memory.

As the decay in working memory is a typical symptom of cognitive impairments, this memory test can be helpful, but in order to categorize the types of the dementia, we need to test the executive functions as well.

This semester, as a completion to the memory tests, I have created three new computer games to test the executive functions, based on the principles of well-known psychiatric tests, such as the Trail Making test. In my paper, after presenting the theoretical background of the research, I am going to describe Corsi cubes, its new features, and all the three new games.

Theoretical background of the research

Importance of Alzheimer and other types of Dementia

According to the World Alzheimer Report 2009 [1] commissioned by the „Alzheimer's Disease International" organisation, there were an estimated 35.6 million people with dementia worldwide, and they expect this number to nearly double every 20 years, to an estimated 65.7 million in 2030, and 115.4 million in 2050. This illness must be a priority question not just because of the high percentage in the population, but also because of its economic relevance.

The total estimated worldwide cost of dementia was US$604 billion in 2010. About 70% of the costs occur in Western Europe and North America. This figure includes costs attributed to informal care (unpaid care provided by family and others), direct costs of social care and the direct costs of medical care. [2] There are some examples to illustrate the importance of economic issue:

If dementia care were a country, it would be the world's 18th largest economy, ranking between Turkey and Indonesia. If dementia care were a company, it would be the world's largest one by an annual revenue exceeding Wal-Mart (US$414 billion) and Exxon Mobil (US$311 billion). [2]

Despite the importance of the illness, research shows that most people currently living with dementia have not received a formal diagnosis. In high income countries, only 20-50% of dementia cases are recognised and documented in primary care.

According to the World Alzheimer Report 2011 [3], studies in high income countries show that only one-fifth to one-half of cases of dementia are routinely recognised and documented in primary care. This 'treatment gap' is much greater in low and middle income countries, with one study in India suggesting 90% of the cases remain unidentified. The National Dementia Strategy in England highlights three factors that result in inactivity in seeking and offering help: the ***stigma of dementia*** that prevents open discussion, the ***false belief*** that changes were a ***normal part of ageing,*** and the ***false belief*** that ***nothing could be done.***

Of course, the impact of a dementia diagnosis depends greatly upon how it is made and imparted.

Earlier diagnosis can be a serious advantage, because it allows people with dementia to plan ahead while they still have the capacity to make important decisions about their future care. In addition, they and their families can receive practical information, advice and support in time. Only through receiving a diagnosis can they get access to available drug and non-drug therapies that may improve their cognition and enhance their quality of life. And, they can, if they choose, participate in research for the benefit of future generations.

Importance of making difference between dementia types

The importance of the right diagnosis can be best illustrated through a personal story. Some weeks ago a friend of mine had his grandmother diagnosed with Parkinson and Alzheimer's disease. She also got some medications, and the doctor promised her, that soon she will feel better. Unfortunately her conditions did not improve in the following weeks so she returned to the doctor. This time it became clear that she was misdiagnosed, and instead of Alzheimer's and Parkinson disease she suffers from a less common subtype of dementia, called Dementia with Lewy bodies (see below).

As it became clear from this story as well, even though some symptoms are similar in these diseases, the necessary treatment can be absolutely different.

How can we facilitate the early diagnosis?

Nowadays, there are plenty of tests available in the psychological and psychiatric literature, that are able to measure the cognitive capacities of the subject. Some of these test are computerized, for example the clinically validated CANTAB tests (Cambridge Neuropsychological Test Automated Battery). This test battery contains tests about visual memory, executive functions, attention, semantic/verbal memory, social cognition, etc. Even though these tests can help recognise several types of dementia, their usage is limited due to their price and availability.

Another example of computerized neuropsychological test batteries are available for free at the PEBL webpage. These tests were programmed in PEBL (Psychology Experiment Building Language) programming language and execution environment. This software helps psychologists design experiments, or use ready-made ones freely. I have also used some ideas, and information from this webpage for my games.

Apart from many advantages, all systems listed above have some disadvantages as well. The CANTAB tests are fairly expensive, and only available for clinical usage. Both the CANTAB and the PEBL tests are less entertaining. There are plenty of less reliable but more entertaining games, available online, for a reasonable price or for free. Some of them even offer to improve or measure mental wellness, but their methods and results are rarely verifiable.

Based on these two major approaches, the M3W project [4] has tried to find a third way; maybe the most similar example to its goal is shown on the "www.happy-neuron.com" webpage. On the one hand, it aims to provide entertaining games that are available from home while, on the other hand, with these games it intends to filter out MCI at the earliest stage possible.

In the next chapter, I am going to shortly describe and characterize the subtypes of dementia. My goal is to present and classify the symptoms that should be measured by the computer games I have recently developed.

Dementia and its most common subtypes

‘Dementia’ is an umbrella term, used to describe the symptoms that occur when the brain is affected by specific diseases and conditions. Symptoms of dementia include loss of memory, confusion and problems with speech and understanding. [5] Apart from cognitive decline, dementia can cause psychological problems such as anxiety and depression, as well as behavioural disorders like agitation and wandering. In this paper, however, I am going to focus on the decline of cognitive capacities, as these are the symptoms that can be objectively measured by computerized tests.

The most common subtypes of dementia are Alzheimer’s disease, Vascular dementia, dementia with Lewy bodies, and the Frontotemporal dementia.

In the followings I am going to describe these illnesses in details. These informations were collected from several different sources, but most of them are from the Alzheimer’s Society’s webpage. [2]

Alzheimer’s disease

Alzheimer's disease, first described by the German neurologist Alois Alzheimer, is a physical disease affecting the brain. During the course of the disease, protein 'plaques' and 'tangles' develop in the structure of the brain, leading to the death of brain cells. People with Alzheimer's also have a shortage of some important chemicals that are involved with the transmission of messages within the brain. [6]

This illness represents over 50% of all dementia cases, and affects 15 million people worldwide. AD prevalence doubles every 5 years after the 60th years of age. [7]

Alzheimer's is a progressive disease, which means that gradually, over time, more parts of the brain are damaged. As this happens, the symptoms become more severe.

[2] http://www.alzheimers.org.uk/site/scripts/documents.php?categoryID=200362

In the early stage of Alzheimer’s, lapses of memory and problems with finding the right word are typical symptoms. As the disease progresses, patients may experience confusion (forget the names of people, recent events etc.)

The cause of the disease has not been identified by one single factor so far. It is likely that a combination of factors, including age, genetic inheritance, environmental factors, lifestyle and overall general health, are responsible for the progression of Alzheimer’s.

Vascular dementia

Vascular dementia [8] is the second most common form of dementia, which occurs due to impaired circulation of the brain.

The illness affects different people in different ways, and the speed of the progression varies from person to person. Typically, the symptoms of vascular dementia begin suddenly, for example after a stroke. Vascular dementia often follows a 'stepped' progression, with symptoms remaining at a constant level for a time and then suddenly deteriorating. Some symptoms may be similar to those of other types of dementia, such as Alzheimer's disease.

However, people with vascular dementia may particularly experience cognitive decline such as problems with speed of thinking, concentration and communication, periods of severe (acute) confusion, and memory problems (although this may not be the first symptom).

Vascular dementia can be diagnosed by cognitive tests that measure attention, the speed of thinking, and the ability of planning.

Dementia with Lewy bodies [9]

Dementia with Lewy bodies (DLB) is a form of dementia that shares characteristics with both Alzheimer's and Parkinson's diseases. Even though it amounts to ten per cent of all cases of dementia in older people, this illness tends to be under-diagnosed.

This type of dementia is a progressive disease, and it progresses at about the same rate as the Alzheimer's disease, typically over several years. A person with DLB will usually have some of the symptoms of Alzheimer's and Parkinson's diseases (muscle stiffness, trembling of the limbs).

Apart from the symptoms that are common in Alzheimer’s and DLB, there are some that may help us to differentiate between these two subtypes of dementia.

Even though memory is often affected, it is typically less so than in Alzheimer's disease, however the most important symptoms are the difficulties with 'executive functions'.

The term ‘executive function’ refers to a complex cognitive construct, including the whole set of processes underlying controlled and goal directed responses to novel or difficult situations. The process of reasoning, planning, and switching between tasks are all considered to be executive functions.

In the case of DLB the typical executive function problems are difficulties in planning ahead and co-ordinating mental activities.

Frontotemporal dementia

Frontotemporal dementia [10] is a relatively uncommon form of dementia. It is caused by a damage of the frontal lobe and/or the temporal parts of the brain. These areas are responsible for our behaviour, emotional responses and language skills. The main symptoms of frontotemporal dementia are: executive dysfunction, prominent disinhibition (loss of ability to empathise with others, inappropriate behaviour) and apathy, but a relative memory sparing (poor retrieval, but preserved recognition).[5]

Frontotemporal diseases can be best diagnosed by the Wisconsin card sorting test, which requires flexible shifts between strategies.

In the end stages, it is difficult to distinguish from Alzheimer’s.

Mild cognitive impairment

The term mild cognitive impairment (MCI) [11] has been applied recently describing the condition when an individual has difficulty remembering things or thinking clearly but the symptoms are not severe enough to warrant a diagnosis of Alzheimer's disease. Recent research has shown that individuals with MCI have an increased risk of developing Alzheimer's disease. However, the conversion rate from MCI to Alzheimer's is low (about 10-20 per cent each year), and consequently a diagnosis of MCI does not always mean that the person will go on to develop Alzheimer's.

Conclusion

As a conclusion, the two main cognitive functions being affected by dementia are executive functions and memory. In the following table I have collected the typical symptoms of the dementia subtypes in terms of these cognitive functions.

	memory	**executive function**
Mild Cognitive Impairment	difficulty remembering things	less affected
Alzheimer’s	lapses in memory, word finding problems	less affected in the early stages
Vascular Dementia	memory problems are typical, but might not be the first symptoms	confusion, problems with attention, the speed of thinking, and the ability of planning
Dementia with Lewy Bodies	less affected than in Alzheimer’s	difficulty in planning ahead and co-ordinating mental activities
Frontotemporal Dementia	poor retrieval, but preserved recognition	Executive dysfunction (can’t shift between strategies)

As it is visible in the table, in order to recognise the dementia subtype, apart from the Corsi cubes memory test, there should be games that measure the ability of planning, the speed of thinking, and the ability to shift between strategies.

Games

Introduction of the games

In the following, I am going to introduce all the four games (Corsi cubes, Footprints, Planarity and Tricky boxes), their algorithms, psychological and game developmental background, their functions to be measured, and their output.

All games were developed in the Eclipse software development environment, in Java language. For the graphical user interfaces I used Java Swing and Java Awt widget toolkits. The main reasons for these choices are that Java is a free, open source, cross-platform programming language, available for many devices with different operating systems. Also the fact that games developed in Java can also be played offline was an important issue.

All four games share some design elements, and their class-structure is quite similar (going to be discussed later).

Basic game developmental principles applied in all games

Ergonomical requirements

Regarding to the special topic, and to the targeted user group, the following special principles had to be applied to all games:

Simple design

As our target group are elderly people the game has to be simple in design and in rules as well. Instead of spectacular and colorful animations it should rather contain simple graphics.

According to this principle, all the four games have black background, with light-coloured large elements. The black background makes the mouse to be more visible, while the large elements, and the large menu might be important to help visually impaired players.

In the test situations in the summer, some keyboard issues became clear. Even those players who use computer on a daily or weekly basis, have had problems with finding the right button at the keyboard. As a solution, we avoid the usage of the keyboard in the games. Players only have to use mouse.

Understandability

As most of our players are not that familiar with computer games, and they must be able to play the games from home without help, all games must be simple, understandable, and easy to learn.

Playability, enjoyability

Even though these games have more psychometrical functions, they must be enjoyable to make players eager to play. To achieve this, the game should have a clear and certain aim, and many different ways to complete it better (Faster, less steps, etc.).

In order to make it more motivating, and enjoyable, Footprints game records the game time, Corsi cube records the highest level reached, while Planarity game records the number of steps, and the length of the game. When the player solve a level, there is a dialog box congratulating, and showing the results. This way players will have the motivation to play it again, and improve their performance.

Re-playability

In order to collect right and useful data about their cognitive capacities, players should use the games at least once a week. To avoid distortions in the results caused by the repetition, all games have to test skills, that are not „memorisable", just improvable (until a certain extent) through practice. This way the games will measure more accurately, and they will be more enjoyable. Also the randomization of the data sets are really important in all the games. In the Planarity, and in the Footprints games this problem is solved by random locations for the elements, while in the Corsi cube test, and in the Tricky boxes game randomized animations take care of replayability.

Culture independency

As this is an international project, we considered culture independency to be a major issue. None of my games require any culture-related skills (such as verbal tests).

Device independency

Currently all the games are being played on PC's or laptops, but the operation systems, and the screen sizes might be different. This difference might cause some unexpected distortion in the results. (For example the time for moving a mouse from one side to another might be really different in a 10’’ netbook with touchpad, or in a 16 ’’ PC with mouse.)

Elimination or minimalization of external distracting factors

As most of the cases the games will be played at home without guidance or supervision, there might some distortion in the results caused by external distractions. Even though the game can not eliminate all of it, it has to be minimalized.

Typical problems that might occur, and their possible elimination:

- Player is disturbed by someone or something

  Unfortunately this might be the most common problem, and it can only be solved in the Planarity and the Footprints games, as they don’t require continuous attention.

  The time extension caused by the distraction can be measured in a way that the time difference between every two mouse click is measured. From these data the program calculates not just the total time, but also the average, and longest time passed between two clicks. If one click is extremely long, we can conclude that there was some disturbance there.

- The game is being played on different computers with different screen resolution.

Three out of the four games (Corsi cubes, Footprints, and Planarity) are being played with a fixed screen size, that is convenient for small laptops, but still useful for bigger screens as well. The Tricky boxes game has no fixed screen size, but as the possible locations of the boxes are fixed, the screen size doesn’t really bother the results.

Psychometric criteria

At the tests in the summer, paper based neuropsychological tests were taken along with the computer-based games, so psychometric criteras of the games can be examined through statistical methods on their results. However as only the Corsi cubes game was tested with patient yet , these criteras can not be defined at this stage of the research, but they have to be kept in consideration.

1. **Validity** [12]

   „Are we measuring the real meaning of the concept under consideration?”

- *Construct Validity* refers to the ability of a test to measure the psychological construct, such as dementia, that it was designed to measure. One way this can be assessed is through to examine whether a test can give results similar to other tests of the same construct and different from tests of different constructs.
- *Content Validity* refers to the ability of a test to sample adequately the broad range of elements that compose a particular construct.
- *Criterion-related Validity* refers to the ability of a test to predict someone's performance on something. [13] For example if the player is diagnosed with dementia with Lewy bodies, he is predicted to reach less points in an executive function test, than a healthy person.

2. **Reliability** [14]

„Can we apply the same technique on the same object repeatedly, and gain consistent results?"

- *Internal Consistency Reliability* refers to how well all the test items relate to each other.
- *Test-retest Reliability* refers to how well results from one administration of the test relate to results from another administration of the same test at a later time.13

When it comes to game development, internal consistency reliability can be ensured if equal levels provide equally difficult tasks, while test-retest reliability can be ensured if the levels are not memorisable.

3. **Precision and Accuracy**

„Is the resolution of our method good enough to get a precise result?"

The step between two levels should be not too small, and not too high to get a proper resolution. We also have to design the steps with consideration to avoid ceiling effect (no one can reach better results than a certain level), or floor effect (everyone can reach the highest level).

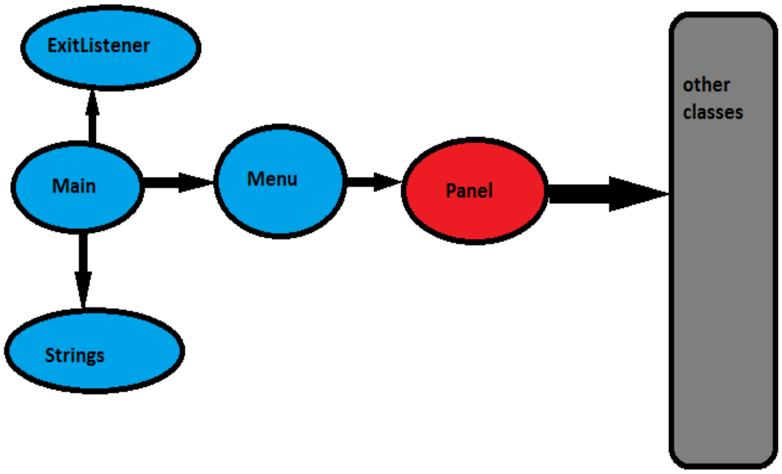
Class structures

The following four classes are nearly identical in every game:

- Main: An interface to the framework. The language and the test options of the game can be set through this class.
- Menu: For the development, I used Java Swing components. In the menu, players can select some settings: the level, in some cases the speed, and they can start the game. The functions connected to the Start button can be seen in the attachments.
- Strings: This class reads the language files into a HashMap according to the chosen language.
- Exit Listener: This class is an extension of the WindowAdapter class, and its only purpose is to dispose the current frame when the window is being closed. This way when the game is finished, the player can return to the framework instead of closing the whole system.

Apart from these four, every game contains some other classes. One of them is an extensions of the JPanel class, and is used for the GUI, while other(s) are for generating datasets, and to compute the results.

All the games share the following class structure:



Special game development solutions

From a developer’s perspective, the four games can be categorised into two types.

The first type includes the Planarity and the Footprints. They are based on algorithms where points are placed at random locations, and the player has to click on them, and/or move them.

These types of games need a method to allocate randomly generated coordinates to the points or the pictures, and in this method it is really important to make sure that the components will not overlap.

As the Planarity game has a more complicated program, in the followings I am going to describe its algorithms in more details (and the setpoints() method can be seen in the attachments), while the Footprints game’s algorithms will be less detailed.

The other type of games includes the Tricky Boxes, and the Corsi cubes. They are both based on animation., that was carried out with Java Swing Timers. For the same reason than in the first type, I am going to introduce more details about the Tricky Boxes game’s algorithms, and less about the Corsi cubes.

Game or test modes

All games have a game and a test mode. In game mode, all data sets, as for example the chosen rectangles at the Corsi cubes, or the generated graphs at the Planarity game, are randomized, and the games can have several settings changed (for example the speed).

In test mode, however, these data are predefined constants, and settings are limited. The test mode is very important for clinical test purposes, while the game mode makes playing more enjoyable with the game.

Output

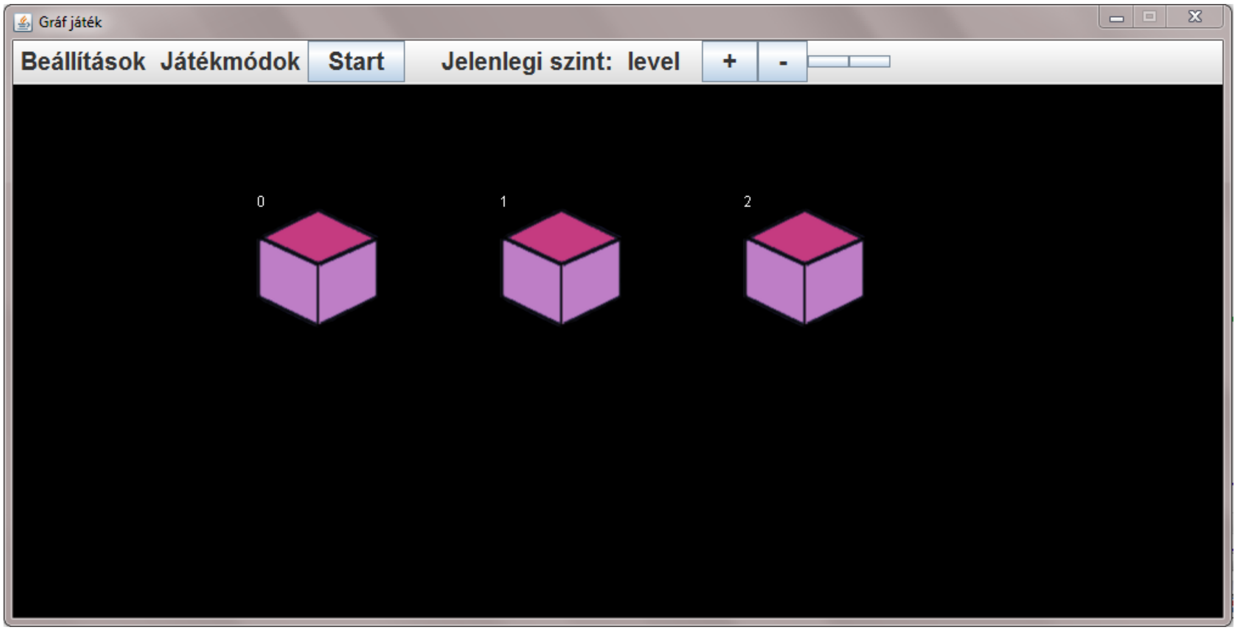
Each game generates an output file in XML format, containing the personal identifier of the player, the date and time of the test, the identifier of the game, independent variables (those that are being set by the level), dependent variables (those that are being measured), and the results.

The generated XML file is sent to the server by the framework.

An example structure of the generated XML file:

```
<m3wgame>
<userid>0</userid>
<version>corsicube-5-201-585</version>
<game id='1'>
<turn level='2' reverseorder='false' datetime='2012-10-23 08:57:58.592' >
<try id='1' >
...
</try>
</turn>
</game>
<interrupted msg='exit'/>
</m3wgame>
```

Tricky boxes game



This game, in different realisations, is one of the world's most famous games played in pubs, or on the streets. Despite of the fact that it is mostly played by gamblers and it is usually involves fraudulence, it is a really interesting game that can measure one's attention. In the beginning, the player sees three identical boxes, and has to choose one. A red ball will be placed into that box. Then the player sees an animation, where all the boxes change place.

The player's task is to find the box that contains the red ball.

Programming issues, classes, and algorithms

Even though this game has a simple graphic design, it contains a quite complicated mathematical background in order to reach the randomized elliptical routes, and the nice animation.

Classes

In the project, there is a class with three instances for the locations (“Location”), another class with three instances for the boxes (“Box”), and the third one for mathematical functions (“Maths”).

While the instance of the Maths class calculates the routes from one location to another, boxes change their coordinates according to the chosen route. I used several mathematical algorithms in order to make the boxes change place in a correct way, and to make them move in the specified route. In the followings, I am going to introduce them.

Elliptical route

When a box moves from one location to another, it changes its place along an elliptical route, defined in advance, that makes the animation as smooth as possible.

In the Maths class, I have defined 50 points of a half ellipse, and their x and y coordinates with the following function.

$$y=\pm b * \sqrt{1-\frac{x^2}{a^2}}$$

In all cases, “a” is defined as the half of the distance between the start and the aim location's center point, while b is a semi-random height: *“b=100+ 20*r.nextInt(3);”*

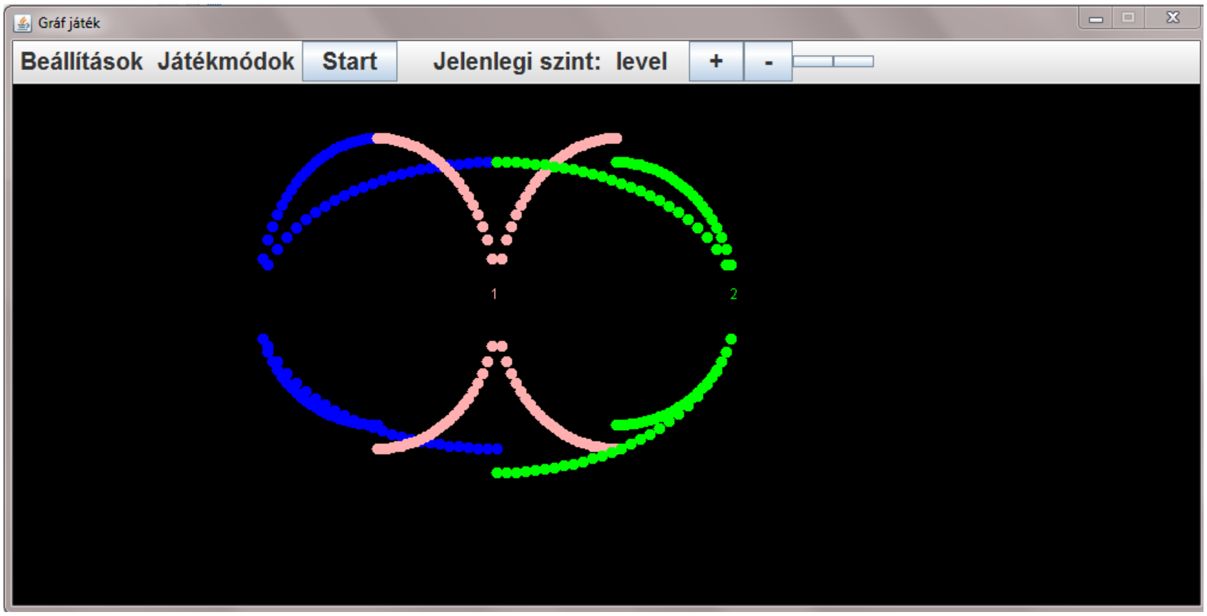
The origo point is the middle point between the two locations, and the coordinates are: *(x0=location1.x+a,y0=location1.y)*

Also the position of the half ellipse has to be defined: if the box goes on the negative side, the y (relatively to y0) is negative, so the sign in the equation above is also minus.

In order to get the correct route for the boxes, the created sequence of points have to be rearranged in accordance with the direction of the movement.

It was also an important issue to make all the boxes start and arrive in the same time, while the length of their journey is not exactly the same. This problem was solved relatively easily: since each route is composed of 50, evenly arranged points, in those cases when the two locations are not adjacent (i.e. if the box goes from *location[0]* to *location[2]*), the animation looks a bit faster.

In the following picture, all the possible elliptic routes are visible (in the middle of their possible route). Blue ellipses are derived from *location[0],* while pink comes from *location[1],* and green from *location[2]*.



Randomization

As there are three possible stationary locations for the boxes, there are two opportunities to change from one arrangement to another. (In this case, I use the word 'arrangement' to describe a certain situation, where every box is placed to one of the possible stationary locations, and all of them have different locations.)

Each box has two possible routes to go from one location to another (with positive or negative y related to the original point).

As a conclusion, this means 2^4=16 options to change arrangement.

In addition, one level includes at least one, but several change of arrangements, so it means 16 different possible animation in the first, 16^2 in the second, 16^3 int the third level (see below).

To conclude, with these algorithms, the game generates nice and clear random curves, so playing the game can be more enjoyable.

Levels

In every level, the length and the speed of the animation change.

The number of the arrangement changes is directly proportional to the level, while the delay of the timer, and through this, the speed of the animation changes according to the following equation: *"speed=20-level*2"*

In game mode, after every level the game offers the player to go on to the next level, start again, or cancel the game. In clinical test mode, every level can be tried three times (similarly to the Corsi cubes), and if two out of the three answers are good, the game can be continued.

Psychological background, and functions to be measured

This game can measure attention span and slowness, however, it is more likely to be motoric slowness than cognitive. Even though the results in themselves, and the measured functions are less relevant to the topic, the game's output can still be very important in comparison with other results, as for example with the results of the Corsi cubes.

In the test situation in the summer I met many players whose memory did not seem to be as impaired as bad their result was in the Corsi test. According to their reflections and my experiences their problem was to a lesser extent related to remembering the rectangles; their bad results were more likely due to the failure of paying attention to the animation.

Independent variables

- level

Dependent variables

- initial location of the chosen rectangle, final location of the chosen rectangle
- whether the answer was right

The Output

As the game has only a few variables, the generated XML file is also quite short.

It saves the level, the date and the time of the test, the speed and the length of the animation, and whether the answer was right or wrong.

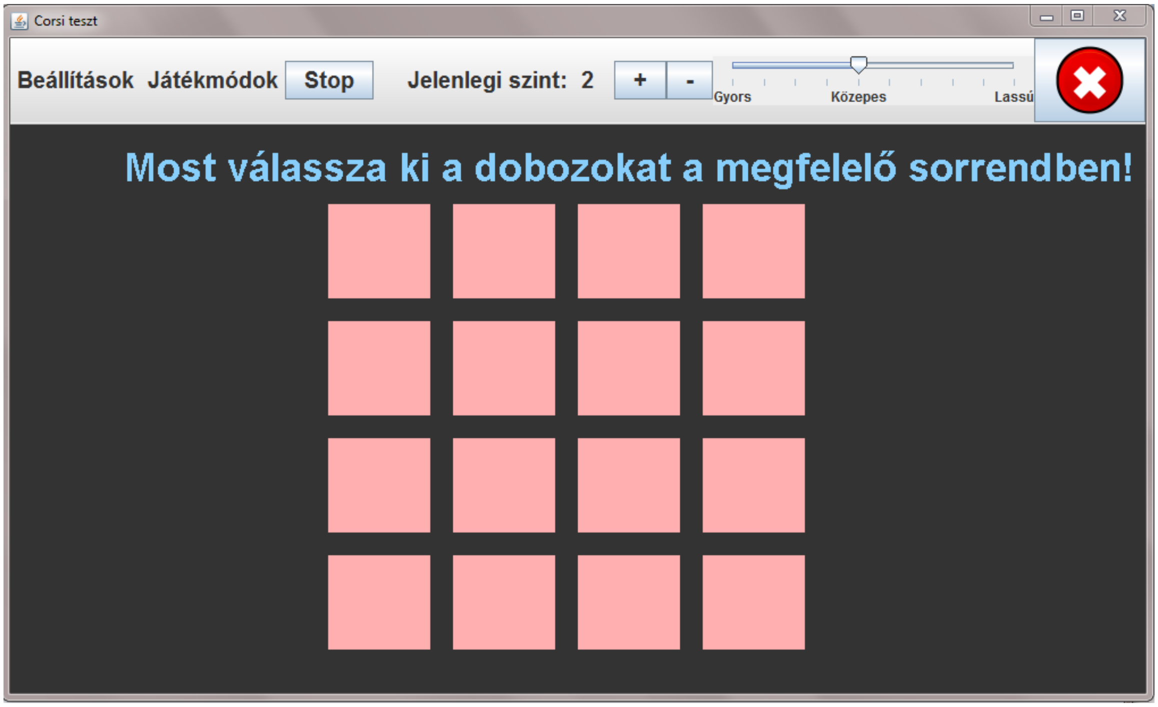
```
<game id='1' gamemode='game'>
    <turn level='1' datetime='2012-10-23 11:16:22.820' speed='18' length='1' >
        <result='true' />
    </turn >
```

Corsi cubes

The Corsi block tapping test is a revised version of the Knox Cubes Imitation test. The latter was originally devised by Knox (1913) to diagnose mental retardation in potential immigrants to the United States, while the former has been specifically designed to be used in neuropsychological practice by Corsi (1972). The game has many computerized versions, as it is one of the most famous and useful tests for measuring the capacity of the visuospatial working memory.[22]

The game

In my version of the game, in the beginning the player sees sixteen identical boxes, and then a blue dot appearing for a short time on some of them in a certain sequence. The task is to click on the marked boxes in the correct order. This order might be forward or reverse, depending on the player's choice.



Jelenlegi szint: 2

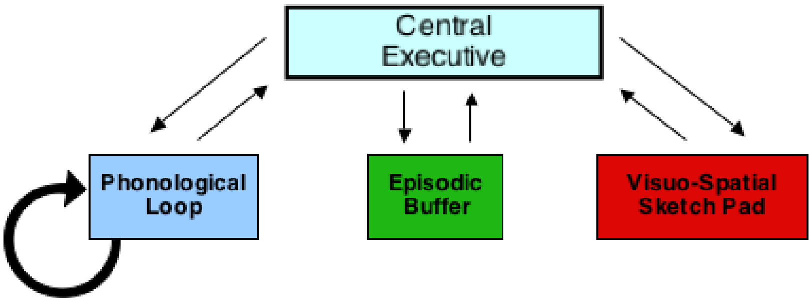
Most válassza ki a dobozokat a megfelelő sorrendben!

Levels

On the first level, the sequence consists of two dots. When a level is completed, the next level consists of one more dots than the previous one. Every level contains three different test sequences, and the level is considered to be completed when at least two out of the three were solved flawlessly. The game goes on until a certain level is failed.

Neuropsychological background, and functions to be measured

The dynamic concept of “working memory”—as opposed to the passive nature of a simple information store—is at the heart of the Baddeley-Hitch model, a system that consists of three short-term stores and a control system. The three short term stores are the phonological loop, the episodic buffer, and the visuospatial sketchpad.



[21]

As I have already mentioned before, lapses in the working memory might be a typical symptom of Alzheimer’s, or MCI. This game mostly measures the capacity of the fourth component, the visuospatial sketchpad, and the condition of the working memory might be concluded from it.

Independent variables (those that are being set)

- level
- game mode (is it a game, or a clinical test?)
- order (does the player have to answer in reverse or in forward order?)
- coordinates of the chosen rectangles in the order of choices

Dependent variables(those that are being measured)

- total time
- coordinates of the answer
- was the answer right or wrong?

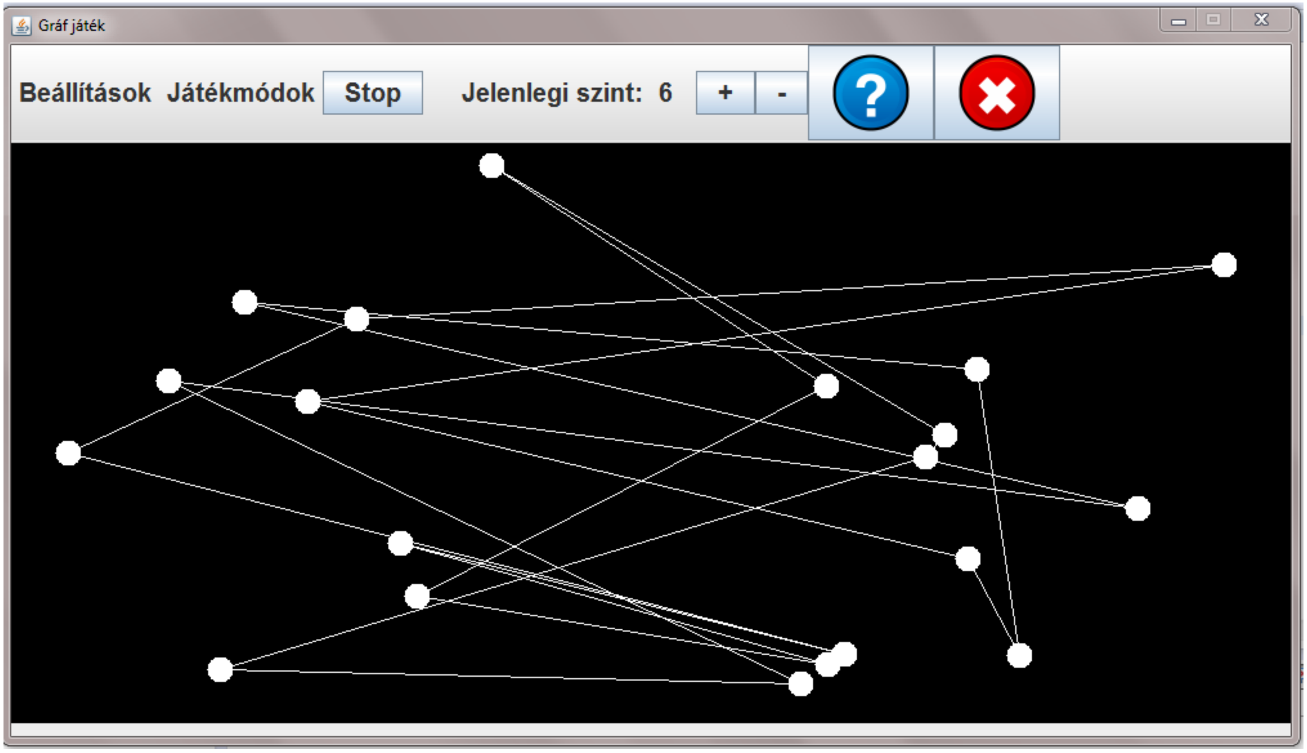
An example output

In the following example output, the “turn” tag means an entire level, which consists of three “try” tags. The “rectangle” tag saves the x and y reference coordinates of the rectangle (which is the point in the upper left corner), and the rectangle identifier. This identifier indicates the order of the rectangles. The “answer” tag can be paired to the rectangles according to this identifier.

```
<turn level='2' reverseorder='false' datetime='2012-10-23 08:57:58.592' >
<try id='1' >
<rectangle id='1' x='500' y='173' />
<rectangle id='2' x='500' y='379' />
<answer id='1' x='559' y='219' answer='true' />
<answer id='2' x='419' y='321' answer='false' />
</try>
<try id='2' >
<rectangle id='1' x='280' y='70' />
<rectangle id='2' x='390' y='173' />
<answer id='1' x='329' y='135' answer='true' />
<answer id='2' x='406' y='196' answer='true' />
</try>
<try id='3' >
<rectangle id='1' x='280' y='276' />
<rectangle id='2' x='500' y='173' />
<answer id='1' x='448' y='305' answer='false' />
</try>
<result try1='0' try2='1' try3='0' sum='level failed' time='4100' />
</turn>
```

Graph game-Planarity

This game is based on the popular online game called „Planarity” that is based on a concept by Mary Radcliffe at Western Michigan University [15], and was first created by John Tantalo [16]. In each level, players receive a complicated graph, and they have to rearrange the dots in a way that no lines intersect (except for their end points).



Programming issues, and algorithms

The mathematical background of this game is planar graphs. According to the definition a *planar graph* is one that can be drawn on a plane in such a way that there are no "edge crossings," i.e. edges intersect only at their common vertices. [17]

According to Kuratowski's Theorem, a finite graph *G* is planar if and only if it has no subgraph that is homeomorphic to the complete graph in five vertices (*K*5) or the complete bipartite graph *K*3, 3. [16]

The creator of the first implementation of the game has also published his algorithm of generating non-planar embeddings of planar graphs, so that every level can be completed. In my implementations, I have avoided using this algorithm, because the generated graphs might be too complicated for the target player groups.

Instead of John Tantalo's algorithm, in my version every dot is only connected with two others, so this way none of the generated graphs can contain complete subgraphs, which also means, that all the generated graphs are planar.

During development of the game, I have faced two algorithmical problems.

First of all, there are no built-in Java methods to find the point where two lines intersect, only an „intersect(Line2D line2)” method, which returns true (a boolean value) if the two lines have any point in common. However, in my graph, all the lines have a common point with two others, which are their endpoints.

In order to solve this problem I had to find the intersection point of two lines, so this way the program can differentiate between real intersections, and common end points.

For the „protected Point2D FindLineIntersection(Line2D a, Line2D b)” method I made a Java version of a code that I had found on the internet [18].

The second problem was the playability of the generated levels. In every level I had to make sure that the graph has at least one intersection, so the player will not receive a pre-solved solution.

Functions to be measured, and the output of the game

This game measures many complex executive functions, such as strategic planning and problem solving. A healthy player would create a strategy, and instead of moving the dots semi-randomly, he or she would solve the level according to this strategy, and his results would be faster by practice.

Independent variables

- number of dots, calculated with the following formula: “points=2+level*3”
- initial number of intersections in the graph

Instead of defining the number of intersections in every level, I found it more useful to generate the coordinates randomly, and count the generated points. The “countintersections()” method counts every line’s intersection with every other line, and divide the results by two (as all the common points were counted twice).

Dependent variables

- number of steps taken to solve the problem
- total time
- average time passed between two clicks
- longest time passed between two clicks

The game measures the longest and the average time passed between two steps in order to minimise external distractions.

I set also a fixed screen size to eliminate distortions caused by different sizes of the screen. The chosen size is dimensioned to a 10’’ screen.

Output

```
“<game id='4' gamemode='game'>
<turn level='4' datetime='2012-10-19 17:26:23.508' points='14'  intersections='4' >
<steps='24' time='23.0' average='0.9772727272727273' maximum='12.9'/></turn>”
```

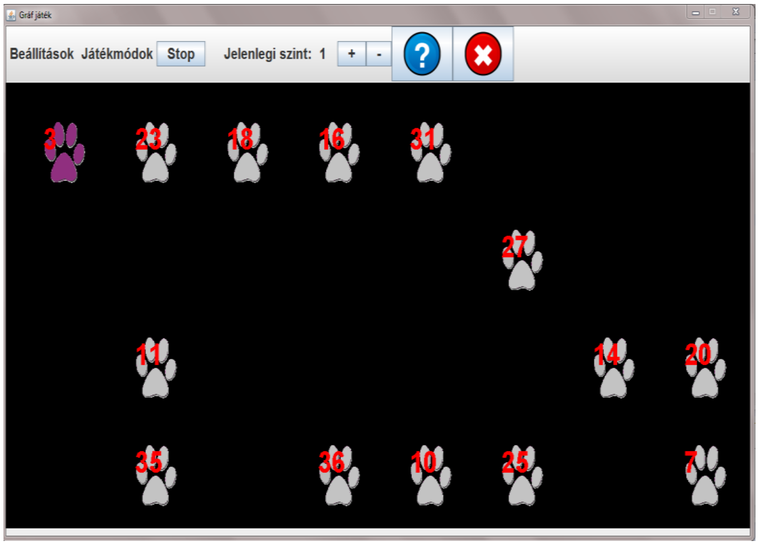
Footprints

This game is a more enjoyable version of the well-known trail making neuropsychological test. The original test consists of two parts. In part A, the player sees numbers, and the task is to connect these numbers in ascending order. In part B, there are numbers and letters as well. The task is to connect the numbers in ascending and the letters in alphabetical order, alternatingly. [20] (For example: 1-a-2-b etc.)

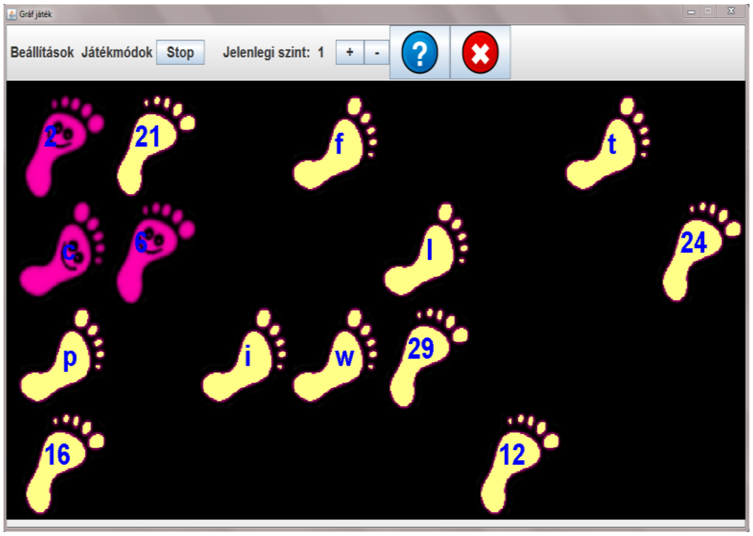
In my version, the player sees footprints on the screen. In game B, numbers are on the left footprints, and letters are on the right ones, while in game A only numbers are shown on symmetrical animal footprints. The task is to “walk through” the screen, by clicking on the footprints. Those footprints that have already been clicked correctly, change color, to make the game more clear, and easier.

<u>Programming solutions</u>

Most of the game structure is similar to the Planarity game. The only real difference is the oval class that stores the image, the character, the number, and the rule if the a character, or a number should be drawn on the Panel.



Jelenlegi szint: 1



Levels

Levels are separated according to the number of footprints. The minimum number is 14, while the maximum is 50, and the formula is the following: “points=10+level*4”

This way ten different levels can be played.

Psychological background, and functions to be measured

The Trail Making Test (TMT) is one of the most popular neuropsychological tests. It is included in most test batteries, such as in our test situation in the summer (on paper, as part of the standard Addenbrook’s Cognitive Examination). The TMT provides information on visual search, scanning, speed of processing, mental flexibility, and executive functions.[19]

While game A measures speed of processing and visual search greatly, game B can measure the flexibility to shift between tasks, which can measure Frontotemporal dementia.

Independent variables

- level: number of footprints visible
- game A, or game B version?

Dependent variables

- total time
- average time passed between two clicks
- longest time passed between two clicks
- number of mistakes

The output

The generated output code looks like the following one:

```
<game id='4' gamemode='game'>
<turn level='2' datetime='2012-10-25 17:28:23.508' points='18'  mode=”B”>
<mistakes=’3’ time='23.0' average='0.9772727272727273' maximum='12.9'/></turn>
```

Further plans

In the current state of the games there are three equally important directions to improve.

First of all, most of the games are not tested yet, so we have no information, whether they meet psychometric test criteria. The first and most important improvements should include the test of reliability and validity of the games, and to reset the difference between levels according to the preferred accuracy and precision.

The second important direction of improvements should be user experience advancement. Currently these games are not enjoyable enough to keep players interested for a longer period.

The third main direction of improvement is data processing. While the tests try to save as much information about the games as possible, their data procession is not prepared yet. In the future I am planning to create a program that can read and process data, and can even restore the whole game process for the physician if needed.

Attachments

Tricky Boxes

Timer function for the animation:

```
public void actionPerformed(ActionEvent arg0) {
if(animation)
{
    counter++;
    if (counter==30){
        //first change in the beginning
        change();
    }
    if(counter>30){
        gamestate=1;
        for(int i=0; i<box.length; i++){
            box[i].setImage(boximg);
        }

        if (counter%3==0) {loc++;}
        if(loc>=Location.points)
        {
            change();
            //restart the loc counter
            loc=0;
        }
        repaint();
    }}
if(gamestate==2)animation=false;
}
```

Maths class

```
public Maths(int x0, int y0, int a, int b, int pos, boolean length) {
        Point[] temp = new Point[points.length];
        this.x0 = x0;
        this.y0 = y0;
        this.a = a;
        this.b = b;
        this.pos = pos;// -1 or 1
        int k = points.length / 2;
        for (int i = 0; i < k; i++) {
            int x = length ? i * 8 : i * 4;
            double vmi = (double) (x * x) / (double) (a * a);
            int y = (int) Math.round(pos * b * Math.sqrt(1 - vmi));
            temp[i] = new Point(x0 + x, y0 + y);}
        for (int i = 0; i < k; i++) {
            int x = length ? -i * 8 : -i * 4;
            double vmi = (double) (x * x) / (double) (a * a);
            int y = (int) Math.round(pos * b * Math.sqrt(1 - vmi));
            temp[k + i] = new Point(x0 + x, y0 + y);}
        for (int i = 0; i < k; i++) {
            if (a < 0) {
                points[i] = temp[points.length - k - i - 1];
                points[k + i] = temp[k + i];
            } else {
                points[i] = temp[points.length - i - 1];
                points[k + i] = temp[i];}}}
```

A method to set the points of the routes

```
public Point[] routepoints(Location b1, Location b2, boolean seged, boolean length){
    Random r=new Random();
    int b=100+10*r.nextInt(3);
    int dir=seged?-1:1;
    int a=(int)(b2.x-b1.x)/2;
    Maths m=new Maths((int)b1.centerx+a,(int)b1.centery,a,b,dir, length);
    return m.points;

}
```

Corsi cubes

Timer function

```
    public void actionPerformed(ActionEvent e) {
      if (start) {
          counter++;

          if (next < level + 1) {
             if ((counter - 1000) % tick == 0) {
                  // in every 1 sec it shows the next ball,
                 // 1000 msec delay before the game starts
                 next++;
                 repaint();
             }
          }
      }
```

Start button's action listener, the start and the stop functions of the menu (same in all the games)

```
    start.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            if (starting) {

                // ha már elindult le lehet állítani
                start.setText(Strings.labels.get("stop").toString());
                m.repaint();
                start();
            } else {
                start.setText(Strings.labels.get("start").toString());
                m.repaint();
                stop();}
        }
    });
public void start() {
        c.instructions=false;
        starting = false;
        //c.start(); // changed by HP 2012-08-22
        c.gamestart();
    }

    public void stop() {
        starting = true;
        c.stop("stop");

    }}
```

Planarity game

setpoints function to set the random coordinates of the points

```
protected void setpoints(boolean gamemode){

        if(gamemode)
            {
            for (int i = 0; i < points; i++) {
            xpoints[i] =10+r.nextInt(x-30);
            ypoints[i] =10+r.nextInt(y-20);
            oval[i] = new Ellipse2D.Double(xpoints[i] - 10, ypoints[i] - 10,
20,
                    20);}}
        else
            {xpoints=testdataX(Level-1);
            ypoints=testdataY(Level-1);
            for (int i = 0; i < points; i++) {
                oval[i] = new Ellipse2D.Double(xpoints[i] - 10, ypoints[i]
- 10, 20,
                        20);}}
    }
```

linecontrol functions to check if the two lines have any intersections

```
protected boolean linecontrol() {
        // check if any of the lines have an intersection
        boolean intersect = false;

        for (int i = 0; i < points - 1; i++) {
            for (int j = 0; j < points; j++) {
                Point2D p = mat.FindLineIntersection(lines[i], lines[j]);
                // if the two lines are not equal, but they have an
intersection
                // except of their end points
```

```
                                if (!lines[i].equals(lines[j])
                                                && p != null
                                                && !(p.equals(lines[i].getP1()) ||
p.equals(lines[i]
                                                                .getP2()))) {
                                        intersect = true;
                                }
                        }
                }
                return intersect;
        }
```

Footprints

constructor of the Oval class

```
public class Oval {
        int id;
        int num;
        char c;
        boolean numchar;
        Image yellowimg;
        Image pinkimg;
        Image img;
        Area area;
        int x;
        int y;
        Color color=Color.WHITE;
        public Oval(ImageIcon image,ImageIcon pinkimage, int x, int y, int num, char
c, int id, boolean n){
                this.x=x;
                this.y=y;
                yellowimg=image.getImage();
                pinkimg=pinkimage.getImage();
                img=yellowimg;
                area=new Area(new Rectangle(x,y,140,140));

                this.id=id;
                this.num=setnum(num,n);
                this.c=setchar(c,n);
                numchar=n;

        }
        public int setnum(int n,boolean numchar){
                int ret=numchar?(n+1):n;
                return ret;}
        public char setchar(char c, boolean numchar){
                char ch=numchar?((char)(c)):((char)(1+c));
                return ch;}
        public void setimage(){
                img=pinkimg;}

}
```

References

[1]http://www.alz.co.uk/research/files/WorldAlzheimerReport-ExecutiveSummary.pdf (page 7)

[2]http://www.alz.co.uk/research/statistics

[3]http://www.alz.co.uk/research/WorldAlzheimerReport2011ExecutiveSummary.pdf

[4] https://m3w-project.eu/

[5]http://www.alzheimers.org.uk/site/scripts/documents.php?categoryID=200360

[6]http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=100

[7]http://www.gov.ns.ca/health/mhs/pubs/seniors/dementia_Diagnosis_seniors.pdf

[8]http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=161

[9]http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=113

[10]http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=167

[11]http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=120

[12]The Practice of Social Research, Earl R. Babbie (page 151.)

[13] http://www.guidetopsychology.com/testing.htm

[14]The Practice of Social Research, Earl R. Babbie (page 148.)

[15]http://en.wikipedia.org/wiki/Planarity

[16]http://www.planarity.net/

[17]http://www.math.lsa.umich.edu/mmss/coursesONLINE/graph/graph5/

[18] http://stackoverflow.com/questions/1119451/how-to-tell-if-a-line-intersects-a-polygon-in-c

[19] http://www.sciencedirect.com/science/article/pii/S0887617703000398

[20] http://doa.alaska.gov/dmv/akol/pdfs/uiowa_trailmaking.pdf

[21] http://generallythinking.com/articles/wp-content/uploads/2009/12/baddeley_and_hitch_working_memory.png

[22]http://generallythinking.com/articles/wpcontent/uploads/2009/12/baddeley_and_hitch_working_memory.png