

Prog1, C++ bevezetés

Magasszintű programozási nyelvek BSc előadás

Dr. Bátfai Norbert

egyetemi adjunktus

<http://www.inf.unideb.hu/~nbatfai/>

Debreceni Egyetem, Informatikai Kar,
Információ Technológia Tanszék

batfai.norbert@inf.unideb.hu

Skype: batfai.norbert

Prog1_5.ppt, v.: 0.0.4, 2011. 03. 16.

<http://www.inf.unideb.hu/~nbatfai/#p1>

Az óra blogja: <http://progpater.blog.hu/>

A Nokia Ovi store-ban is elérhető: <http://store.ovi.com/content/100794>

Felhasználási engedély

Bátfai Norbert

Debreceni Egyetem, Informatikai Kar, Információ Technológia Tanszék

<nbatfai@inf.unideb.hu, nbatfai gmail com>

Copyright © 2011 Bátfai Norbert

E közlemény felhatalmazást ad önnek jelen dokumentum sokszorosítására, terjesztésére és/vagy módosítására a Szabad Szoftver Alapítvány által kiadott GNU Szabad Dokumentációs Licenc 1.2-es, vagy bármely azt követő verziójának feltételei alapján. Nem változtatható szakaszok: A szerzőről.

Címlap szövegek: Programozó Páternoszter, Bátfai Norbert, Gép melletti fogyasztásra.

Hátlap szövegek: GNU Jáváccka, belépés a gépek mesés birodalmába.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being: A szerzőről, with the Front-Cover Texts being: Programozó Páternoszter, Bátfai Norbert, Gép melletti fogyasztásra, and with the Back-Cover Texts being: GNU Jáváccka, belépés a gépek mesés birodalmába.

Célok és tartalom

Előadás

- a) Referenciatípus
- b) OO alapelvek
- c) Dinamikus tárkezelés

Labor

- a) BogoMIPS
- b) Qt (fraktálok, genom ábrázolások, életjáték)

Laborkártyák

- a) Példás kártyák

Otthoni opcionális feladat

- a) A japán világbajnok HELIOS csapat szoftvereinek otthoni tanulmányozása.

Kapcsoldó videók, videómagyarázatok és blogok

http://progpater.blog.hu/2011/02/26/tan_csodallak_amde_nem_ertelek_de_kepzetem_hegyvolgyedet_bejarja

http://progpater.blog.hu/2011/03/03/feagyvert_a_nepnek

http://progpater.blog.hu/2011/03/05/figyelem_ez_nem_gyakorlat

http://progpater.blog.hu/2011/03/05/szonyegen_a_human_genom

http://progpater.blog.hu/2011/03/06/utam_a_csucsra

Az írásbeli és a szóbeli vizsgán bármi (jegyzet, könyv, forráskód, számítógép mobiltelefon stb.) használható! (Az írásbeli vizsgán beszélni viszont tilos.) Hiszen az én feladatomban az lesz, hogy eldöntsem, jól felkészült programozóval, vagy mennyire felkészült programozóval állok szemben.

Minimális gyakorlati cél

A hallgató meg tudja írni (másolás alapján) és tudja módosítani a bevezető Qt példákat:

svn co svn://hallg.inf.unideb.hu:2005/bevezetes
(az elsoQt/{Frak|Sejtauto|GenAblak})

Minimális elméleti cél

- 1) C++ típusok, paraméterátadás
- 2) OO alapelvek (egységbezárás, adatrejtés, öröklés)
- 3) Egyszerű osztályok (például a PolarGen) elkészítése
- 4) Dinamikus tárkezelés: new, new[] alapvető használata
- 5) Az Aranycsapat felállása az évszázad mérkőzésén

Referenciatípus

```
#include <stdio.h>

void
csere (int *a, int *b)
{
    *a = *a - *b;
    *b = *a + *b;
    *a = *b - *a;
}

int
main (void)
{
    int a = 5, b = 7;

    csere (&a, &b);
    printf ("a=%d b=%d\n", a, b);

    return 0;
}
```

```
[morpheus@zion morpheus]$ gcc cs.c -o cs
[morpheus@zion morpheus]$ ./cs
a=7 b=5
```

Referenciatípus

```
#include <stdio.h>

void
csere (int &a, int &b)
{
    a = a - b;
    b = a + b;
    a = b - a;
}

int
main (void)
{
    int a = 5, b = 7;

    csere (a, b);
    printf ("a=%d b=%d\n", a, b);

    return 0;
}
```

```
[morpheus@zion morpheus]$ g++ cs2.c -o cs2
[morpheus@zion morpheus]$ ./cs
a=7 b=5
```


Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
[morpheus@zion morpheus]$ g++ c2.cpp -o c2
```

```
[morpheus@zion morpheus]$ ./c2
```

alma

alma

```
#include <iostream>

int
main (void)
{
    char c;
    while (std::cin >> c)
        std::cout << c;
}
```

```
#include <stdio.h>
int
main (void)
{
    int c;
    while ((c = getchar ()) != EOF)
        putchar (c);
    return 0;
}
```

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
[morpheus@zion morpheus]$ g++ r2.cpp -o r2
```

```
[morpheus@zion morpheus]$ ./r2
```

```
10 a #include <iostream>
6 6 #include <iomanip>
14 e #include <cstdlib>
14 e #include <ctime>
8 8 int
1 1 main (void)
14 e {
0 0     std::srand (std::time(NULL));
10 a     for (int i = 0; i < 10; ++i)
14 e     {
        int r = (int) (16.0 * std::rand () / (RAND_MAX + 1.0));
        std::cout <<
            std::setw (3) << std::dec << r <<
            std::setw (3) << std::hex << r << std::endl;
    }
}
```

```
#include <stdio.h>
#include <stdlib.h>
int
main (void)
{
    int i, r;
    for (i = 0; i < 10; ++i)
    {
        r = 1 + (int) (10.0 * rand () / (RAND_MAX + 1.0));
        printf ("%d ", r);
    }
    printf ("\n");
    return 0;
}
```

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
[morpheus@zion morpheus]$ g++ a2.cpp -o a2
```

```
[morpheus@zion morpheus]$ ./a2
```

2.2

1

2.8

```
2[morpheus@zion morpheus]$
```

```
#include <stdio.h>
#include <stdlib.h>
int
main (void)
{
    int i = 0;
    double r, s = 0.0;
    while (scanf ("%lf", &r) > 0)
    {
        ++i;
        s += r;
    }
    printf ("%lf\n", s / i);
    return 0;
}
```

```
#include <iostream>

int
main (void)
{
    int i = 0;
    double r, s = 0.0;
    while (std::cin >> r)
    {
        s += r;
        ++i;
    }

    std::cout << s / i;
}
```

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
#include <stdio.h>
#include <math.h>
int
main ()
{
    FILE *gyok2f;
    double gy2 = sqrt(2.0);
    gyok2f = fopen ("GyokKetto.txt", "w");
    fprintf (gyok2f, "%f", gy2);
    fclose (gyok2f);
    return 0;
}
```

```
#include <iostream>
#include <fstream>
#include <cmath>
int
main ()
{
    const double gy2 = std::sqrt (2.0);
    std::fstream gyok2f ("GyokKetto.txt", std::ios_base::out);
    gyok2f << gy2;
    gyok2f.close ();
    return 0;
}
```

[morpheus@zion morpheus]\$

[morpheus@zion morpheus]\$

[morpheus@zion morpheus]\$ more GyokKetto.txt

1.414214

[morpheus@zion morpheus]\$ g++ k1.cpp -o k1

[morpheus@zion morpheus]\$./k1

[morpheus@zion morpheus]\$ more GyokKetto.txt

1.41421

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
#include <stdio.h>
int
main ()
{
    FILE *gyok2;
    double gy2;
    gyok2f = fopen ("GyokKetto.txt", "r");
    fscanf (gyok2f, "%lf", &gy2);
    fclose (gyok2f);
    printf ("%lf\n", gy2);
    return 0;
}

#include <iostream>
#include <fstream>
int
main ()
{
    double gy2;
    std::fstream gyok2f ("GyokKetto.txt", std::ios_base::in);
    gyok2f >> gy2;
    gyok2f.close ();
    std::cout << gy2 << std::endl;
    return 0;
}
```

```
[morpheus@zion morpheus]$ ./k2
1.414210
```

```
[morpheus@zion morpheus]$ g++ k2.cpp -o k2
[morpheus@zion morpheus]$ ./k2
1.41421
```

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <math.h>
int
main ()
{
    int gyok2f;
    double gy2 = sqrt (2.0)
    gyok2f = open ("GyokKet
                S_IRUSR
    write (gyok2f, (void *)
    close (gyok2f);
    return 0;
}
```

```
#include <iostream>
#include <fstream>
#include <cmath>
int
main ()
{
    const double gy2 = std::sqrt (2.0);
    std::fstream gyok2f ("GyokKetto", std::ios_base::out);
    gyok2f.write ((char *) &gy2, sizeof (double));
    gyok2f.close ();
    return 0;
}
```

```
[morpheus@zion morpheus]$ g++ k3.cpp -o k3
```

```
[morpheus@zion morpheus]$ ./k3
```

```
[morpheus@zion morpheus]$ more GyokKetto
```

```
Íf ö?
```

```
[morpheus@zion morpheus]$
```

```
[morpheus@zion morpheus]$
```

```
[morpheus@zion morpheus]$ more GyokKetto
```

```
Íf ö?
```

```
[morpheus@zion morpheus]$ file GyokKetto
```

```
GyokKetto: data
```

Visszatekintés

Az első előadás laborkártyáinak C++ átirata

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int
main ()
{
    int gyok2f;
    double gy2;
    gyok2f = open
    read (gyok2f,
    close (gyok2f)
    printf ("%lf\n"
    return 0;
}
```

```
#include <iostream>
#include <fstream>
int
main ()
{
    double gy2;
    std::fstream gyok2f ("GyokKetto", std::ios_base::in);
    gyok2f.read ((char *) &gy2, sizeof (double));
    gyok2f.close ();
    std::cout << gy2 << std::endl;
    return 0;
}
```

```
[morpheus@zion morpheus]
[morpheus@zion morpheus]
1.414214
```

```
[morpheus@zion morpheus]$ g++ k4.cpp -o k4
[morpheus@zion morpheus]$ ./k4
1.41421
```

00

Alapelvek:

- a) egységbezárás
- b) adatrejtés
- c) öröklés

Komplex osztály, komplex tömb
elemei komplex objektumok?

Zbl 0997.68170

Bátfai, Norbert; Erdei, Szabolcs; Ispány, Márton

Developing a numerical library Maja in Java

<http://www.zentralblatt-math.org/zmath/en/search/?q=an:0997.68170&format=complete>

Javaslatom az első Java osztályra

<http://www.tankonyvtar.hu/informatika/javat-tanitok-1-1-1-080904-1>

(kb. a lap közepén a PolárGenerátor osztály)

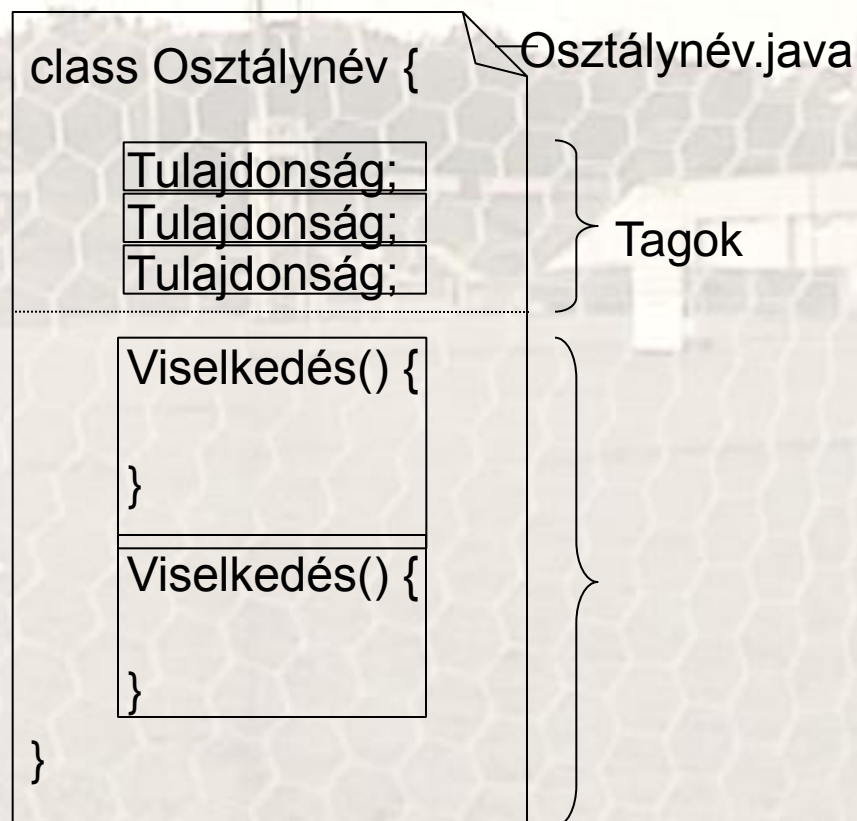
```
public class PolárGenerátor {

    boolean nincsTárolt = true;
    double tárolt;

    public PolárGenerátor() {
        nincsTárolt = true;
    }

    public double következő() {
        if (nincsTárolt) {
            double u1, u2, v1, v2, w;
            do {
                u1 = Math.random();
                u2 = Math.random();
                v1 = 2 * u1 - 1;
                v2 = 2 * u2 - 1;
                w = v1 * v1 + v2 * v2;
            } while (w > 1);
            double r = Math.sqrt((-2 * Math.log(w)) / w);
            tárolt = r * v2;
            nincsTárolt = !nincsTárolt;
            return r * v1;
        } else {
            nincsTárolt = !nincsTárolt;
            return tárolt;
        }
    }

    public static void main(String[] args) {
        PolárGenerátor g = new PolárGenerátor();
        for (int i = 0; i < 10; ++i) {
            System.out.println(g.következő());
        }
    }
}
```



Forrás: Bátfai Norbert: Nehogy már a mobilod nyomkodjon Téged! DEENK 2008.

[http://download.java.net/openjdk/jdk6/
openjdk-6-src-b12-28_aug_2008.tar.gz](http://download.java.net/openjdk/jdk6/openjdk-6-src-b12-28_aug_2008.tar.gz)

Lister - [C:\Documents and Settings\Bátfai Norbert\Dokumentumok\OS\OpenJDK\jdk\src\share\classes\java\util\Random.java]



83 %

```
}  
/ (double)(1L << 53);  
}
```

```
private double nextNextGaussian;  
private boolean haveNextNextGaussian = false;
```

Lister - [C:\Documents and Settings\Bátfai Norbert\Dokumentumok\OS\OpenJDK\jdk\src\share\classes\java\util\Random.java]

Fájl Szerkesztés Beállítások Súgó

```
synchronized public double nextGaussian() {  
    // See Knuth, ACP, Section 3.4.1 Algorithm C.  
    if (haveNextNextGaussian) {  
        haveNextNextGaussian = false;  
        return nextNextGaussian;  
    } else {  
        double v1, v2, s;  
        do {  
            v1 = 2 * nextDouble() - 1; // between -1 and 1  
            v2 = 2 * nextDouble() - 1; // between -1 and 1  
            s = v1 * v1 + v2 * v2;  
        } while (s >= 1 || s == 0);  
        double multiplier = StrictMath.sqrt(-2 * StrictMath.log(s)/s);  
        nextNextGaussian = v2 * multiplier;  
        haveNextNextGaussian = true;  
        return v1 * multiplier;  
    }  
}
```

```
/**  
 * Serializable fields for Random.  
 *  
 * @serialField    seed long  
 *                 seed for random computations  
 * @serialField    nextNextGaussian double  
 *                 next Gaussian to be returned  
 * @serialField    haveNextNextGaussian boolean  
 *                 nextNextGaussian is valid  
 */  
private static final ObjectStreamField[] serialPersistentFields = {  
    new ObjectStreamField("seed", Long.TYPE),  
    new ObjectStreamField("nextNextGaussian", Double.TYPE),  
    new ObjectStreamField("haveNextNextGaussian", Boolean.TYPE)  
};
```

Ha megírta a hallgató, akkor erre érdemes felhívni a figyelmét, s arra is persze, hogy ugyanezt találja a Sun-os JDK src.zip-ben is.

00

Alapelvek:

- a) egységbezárás
- b) adatrejtés
- c) öröklés

```
#ifndef POLARGEN__H
#define POLARGEN__H

#include <cstdlib>
#include <cmath>
#include <ctime>

class PolarGen
{
public:
    PolarGen ()
    {
        nincsTarolt = true;
        std::srand (std::time (NULL));
    }
    ~PolarGen ()
    {
    }
    double kovetkezo ();

private:
    bool nincsTarolt;
    double tarolt;
};

#endif
```

src/polargen.h

00

Alapelvek:

- a) egységbezárás
- b) adatrejtés
- c) öröklés

```
#include "polargen.h"

double
PolarGen::kovetkezo ()
{
    if (nincsTarolt)
    {
        double u1, u2, v1, v2, w;
        do
        {
            u1 = std::rand () / (RAND_MAX + 1.0);
            u2 = std::rand () / (RAND_MAX + 1.0);
            v1 = 2 * u1 - 1;
            v2 = 2 * u2 - 1;
            w = v1 * v1 + v2 * v2;
        }
        while (w > 1);

        double r = std::sqrt ((-2 * std::log (w)) / w);

        tarolt = r * v2;
        nincsTarolt = !nincsTarolt;

        return r * v1;
    }
    else
    {
        nincsTarolt = !nincsTarolt;
        return tarolt;
    }
}
```

src/polargen.cpp

00

Alapelvek:

- a) egységbezárás
- b) adatrejtés
- c) öröklés

```
#include <iostream>
#include "polargen.h"

int
main (int argc, char **argv)
{
    PolarGen pg;

    for (int i = 0; i < 10; ++i)
        std::cout << pg.kovetkezo () << std::endl;

    return 0;
}
src/polargenteszt.cpp
```

```
nbatfai@hallg:~/masodikC++$ cmake .
```

```
-- The C compiler identification is GNU
```

```
...
```

```
-- Build files have been written to: /home/nbatfai/masodikC++
```

```
nbatfai@hallg:~/masodikC++$ make
```

```
Scanning dependencies of target polargen
```

```
[ 50%] Building CXX object
```

```
src/CMakeFiles/polargen.dir/polargen.cpp.o
```

```
[100%] Building CXX object
```

```
src/CMakeFiles/polargen.dir/polargenteszt.cpp.o
```

```
Linking CXX executable polargen
```

```
[100%] Built target polargen
```

```
nbatfai@hallg:~/masodikC++$ src/polargen
```

```
0.681487
```

```
0.791811
```

```
-0.452172
```

```
0.305843
```

```
-1.1143
```

```
-2.67287
```

```
0.923875
```

```
0.690822
```

```
-0.816956
```

```
-0.422164
```

```
nbatfai@hallg:~/masodikC++$
```

00

Alapelvek:

- a) egységbezáras
- b) adatretjtés
- c) öröklés

```
#ifndef FRAKSZAL_H  
#define FRAKSZAL_H
```

```
#include <QThread>  
#include "frakablak.h"
```

```
class FrakAblak;
```

```
class Frakszal : public QThread  
{
```

```
#ifndef FRAKABLAK_H  
#define FRAKABLAK_H
```

```
#include <QtGui/QMainWindow>  
#include <QImage>  
#include <QPainter>  
#include "frakszal.h"
```

```
class Frakszal;
```

```
class FrakAblak : public QMainWindow  
{
```

```
    Q_OBJECT
```

```
public:
```

```
    FrakAblak(double a = -2.0, double b = .7, double c  
              double d = 1.35, int szelesseg = 600,
```

```
double c, double d,  
massag, int iterac
```

00

Alapelvek:

- a) egységbezárás
- b) adatrejtés
- c) öröklés

The screenshot shows a Mozilla Firefox browser window with the title "rcsc::SoccerAgent Class Reference - Mozilla Firefox". The address bar shows the file path "file:///home/batfai/RoboCup/Doxygen/html/". The page content includes navigation tabs for "Main Page", "Related Pages", "Namespaces", "Classes", and "Files". Below these are "Class List", "Class Hierarchy", and "Class Members" tabs. The main heading is "rcsc::SoccerAgent Class Reference". The text describes it as an "abstract soccer agent class" and includes a preprocessor directive: "#include <soccer_agent.h>". An inheritance diagram for "rcsc::SoccerAgent" is shown, with "rcsc::CoachAgent", "rcsc::PlayerAgent", and "rcsc::TrainerAgent" as direct subclasses, and "SampleCoach", "SamplePlayer", and "SampleTrainer" as further subclasses. Below the diagram is a link for "List of all members." and a section for "Public Member Functions" starting with "SoccerAgent ()".

http://fersml.blog.hu/2011/01/05/ismerkedes_a_japan_helios_csapat_szoftvereivel_avagy_hany_trivialis_hello_vilag

Labor

Qt példák

http://progpatner.blog.hu/2011/03/06/utam_a_csucsra

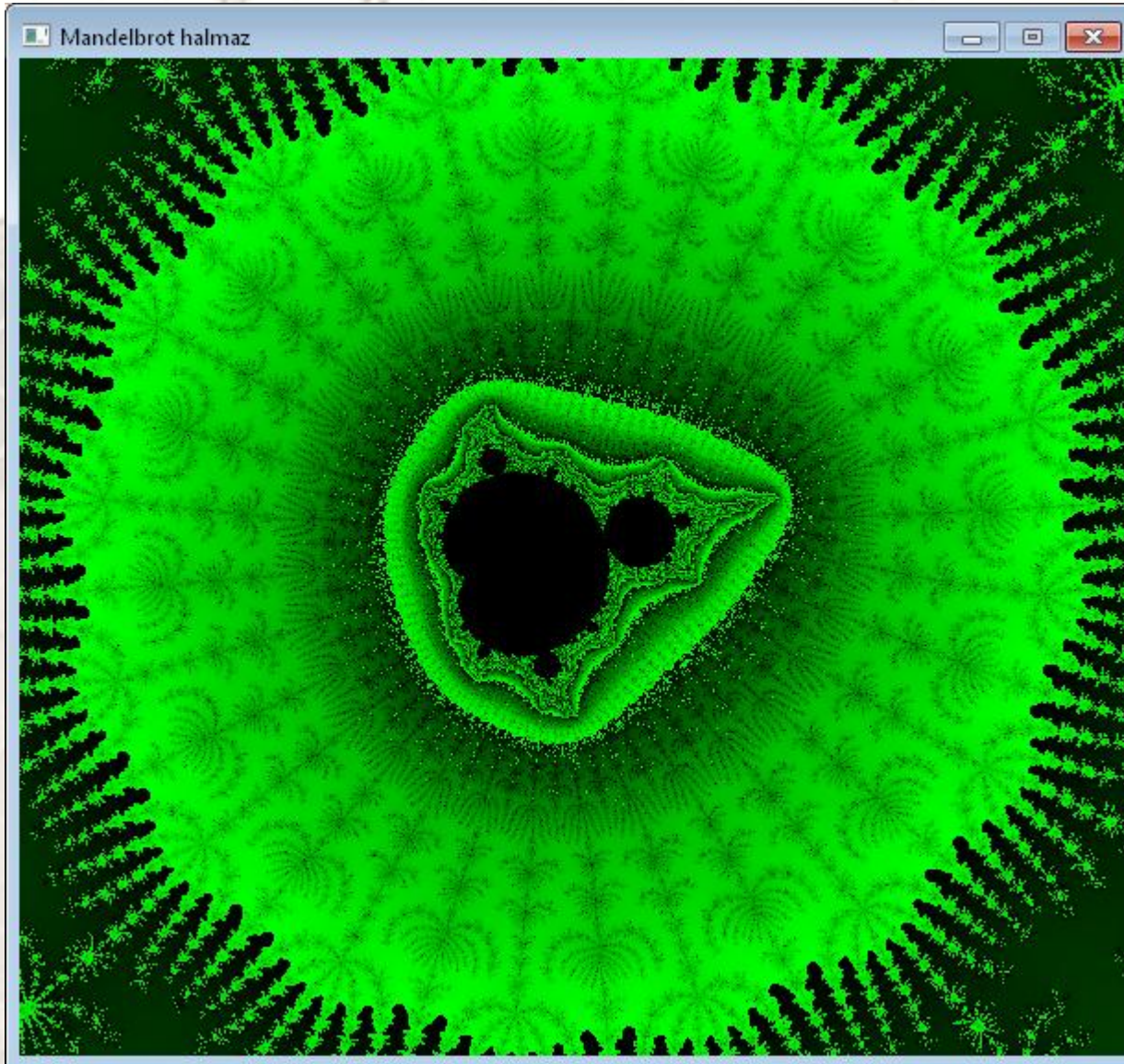


http://progpatner.blog.hu/2011/03/06/utam_a_csucsra

A videó lejátszása.

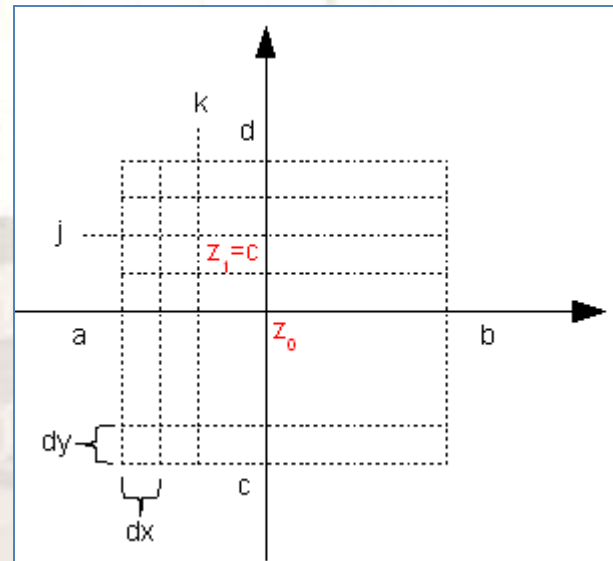
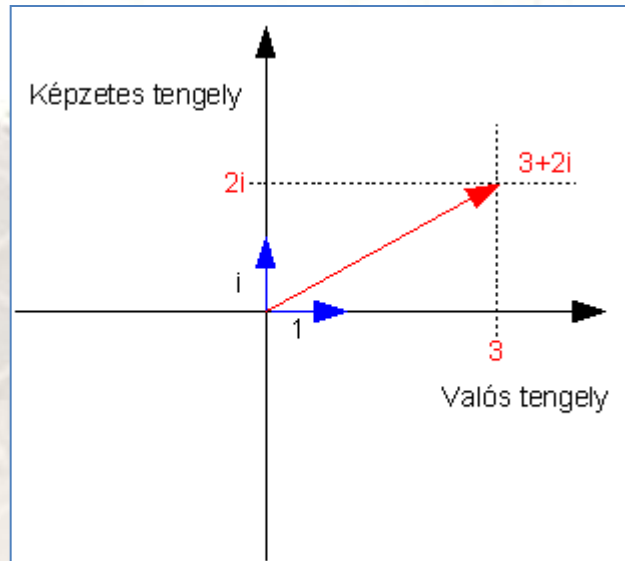
Mandelbrot halmaz

http://progpater.blog.hu/2011/02/26/tan_csodallak_amde_nem_ertelek_de_kepzetem_hegyvolgyedet_bejarja



Mandelbrot halmaz

http://progpater.blog.hu/2011/02/26/tan_csodallak_amde_nem_ertelek_de_kepzetem_hegyvolgyedet_bejarja



```
void FraksZal::run()
{
    // A [a,b]x[c,d] tartományon milyen sűrű a
    // megadott szélesség, magasság háló:
    double dx = (b-a)/szelesseg;
    double dy = (d-c)/magassag;
```


Mandelbrot halmaz

```
class FraksZal : public QThread
{
    Q_OBJECT

public:
    FraksZal(double a, double b, double c, double d,
             int szelesseg, int magassag, int iteraciosHatar, FrakAblak *frakAblak);
    ~FraksZal();
    void run();

protected:
    // A komplex sík vizsgált tartománya [a,b]x[c,d].
    double a, b, c, d;
    // A komplex sík vizsgált tartományára feszített
    // háló szélessége és magassága.
    int szelesseg, magassag;
    // Max. hány lépésig vizsgáljuk a  $z_{n+1} = z_n * z_n + c$  iterációt?
    // (tk. most a nagyítási pontosság)
    int iteraciosHatar;
    // Kinek számolok?
    FrakAblak* frakAblak;
    // Soronként küldöm is neki vissza a kiszámoltakat.
    int* egySor;
};
```

frakszal.h

Mandelbrot halmaz

```
mandelbrot = new Frakszal(a, b, c, d, szelesseg, magassag, iteraciosHatar, this);  
mandelbrot->start();
```

```
}
```

frakablak.cpp

```
        ++iteracio;  
  
    }  
    // ha a < 4 feltétel nem teljesült és a  
    // iteráció < iterációsHatar sérülésével lépett ki, azaz  
    // feltesszük a c-ről, hogy itt a  $z_{n+1} = z_n * z_n + c$   
    // sorozat konvergens, azaz iteráció = iterációsHatar  
    // ekkor az iteráció %= 256 egyenlő 255, mert az esetleges  
    // nagyítások során az iteráció = valahány * 256 + 255  
  
    iteracio %= 256;  
  
    //a színezést viszont már majd a FrakAblak osztályban lesz  
    egySor[k] = iteracio;  
}  
// Ábrázolásra átadjuk a kiszámolt sort a FrakAblak-nak.  
frakAblak->vissza(j, egySor, szelesseg);
```

```
}
```

frakszal.cpp

Mandelbrot halmaz

```
void vissza(int magassag , int * sor, int meret) ;

protected:
    void paintEvent(QPaintEvent*);

private:
    QImage* fraktal;
    FrakSzal* mandelbrot;

};
```

frakablak.h

```
void FrakAblak::paintEvent(QPaintEvent*) {
    QPainter qpainter(this);
    qpainter.drawImage(0, 0, *fraktal);
    qpainter.end();
}

void FrakAblak::vissza(int magassag, int *sor, int meret)
{
    for(int i=0; i<meret; ++i) {
        QColor szin = QColor(0, 255-sor[i], 0);
        fraktal->setPixel(i, magassag, szin);
    }
    update();
}
```

frakablak.cpp

Mandelbrot halmaz

Dinamikus memóriakezelés

```
class FrakSzal {
public:
    FrakSzal(double a, double b, double c, double d,
             int szelesseg, int magassag, int iteraciosHatar, FrakAblak *
             ablak) {
        this->a = a;
        this->b = b;
        this->c = c;
        this->d = d;
        this->szelesseg = szelesseg;
        this->iteraciosHatar = iteraciosHatar;
        this->frakAblak = ablak;
        this->magassag = magassag;

        egySor = new int[szelesseg];
    }

    ~FrakSzal() {
        delete[] egySor;
    }
};

// soronként küldöm is neki vissza a kiszámoltakat.
int* egySor;
```

frakszal.cpp

frakszal.h

A John Horton Conway-féle életjáték

http://progpater.blog.hu/2011/03/03/feagyvert_a_nepnek



OO szempontból teljesen ua., mint az előző.

A Conway-féle életjáték

http://progpater.blog.hu/2011/03/03/fegyvert_a_nepnek

```
/**
 * A sejtter időbeli fejlődése a John H. Conway féle
 * életjáték sejtautomata szabályai alapján történik.
 * A szabályok részletes ismertetését lásd például a
 * [MATEK JÁTEK] hivatkozásban (Csákány Béla: Diszkrét
 * matematika)
 */
void Sejtter::Eljaras()
{
    bool elottelek;
    bool utatek;

    for (int i=0; i<magassag; ++i)
    {
        for (int j=0; j<szelesseg; ++j)
        {
            int elok = szomszedokSzama(racsElotte, i, j, SejtAblak::ELO);

            if (racsElotte[i][j] == SejtAblak::ELO)
            {
                /* Élő élő marad, ha kettő vagy három élő
                 szomszedja van, különben halott lesz. */
                if (elok==2 || elok==3)
                    racsUtana[i][j] = SejtAblak::ELO;
                else
                    racsUtana[i][j] = SejtAblak::HALOTT;
            } else {
                /* Halott halott marad, ha három élő
                 szomszedja van, különben élő lesz. */
                if (elok==3)
                    racsUtana[i][j] = SejtAblak::ELO;
                else
                    racsUtana[i][j] = SejtAblak::HALOTT;
            }
        }
    }
}
```

A Conway-féle életjáték

Dinamikus memóriakezelés

```
public:
    SejtAblak(int szelesseg = 100, int magassag = 75, QWidget *parent = 0);
    ~SejtAblak();
    // Egy sejt lehet élő
    static const bool ELO = true;
    // vagy halott
    static const bool HALOTT = false;
    void vissza(int racsIndex);

protected:
    // Két rácsot használunk majd, az egyik a sejtter állapotát
    // a t_n, a másik a t_n+1 időpillanatban jellemzi.
    bool ***racsok;
    // Valamelyik rácsra
    // [2][][][]
    // állítjuk fel a
    bool **racok;
    // Megmutatjuk a
    int racsIndex;
    // Pixelbeosztás
    int cellaSzelesseg;
```

```
        racsok = new bool**[2];
        racsok[0] = new bool*[magassag];
        for(int i=0; i<magassag; ++i)
            racsok[0][i] = new bool [szelesseg];
        racsok[1] = new bool*[magassag];
        for(int i=0; i<magassag; ++i)
            racsok[1][i] = new bool [szelesseg];
```

sejtablak.cpp

sejtablak.h

A Conway-féle életjáték

Dinamikus memóriakezelés (malloc()-os ismétlődő ábra)

bool ***racsok

bool **

bool **

tár

racsok[0]

```
racsok = new bool**[2];  
racsok[0] = new bool*[magassag];  
for(int i=0; i<magassag; ++i)  
    racsok[0][i] = new bool [szelesseg];  
racsok[1] = new bool*[magassag];  
for(int i=0; i<magassag; ++i)  
    racsok[1][i] = new bool [szelesseg];
```

racsok[0][1]

0.

1.

2.

magassag-1

bool*

bool*

bool*

...

bool*

tár

bool

bool

...

bool

bool

bool

...

bool

tár

0.

1.

szelesseg-1

A Conway-féle életjáték

Dinamikus memóriakezelés

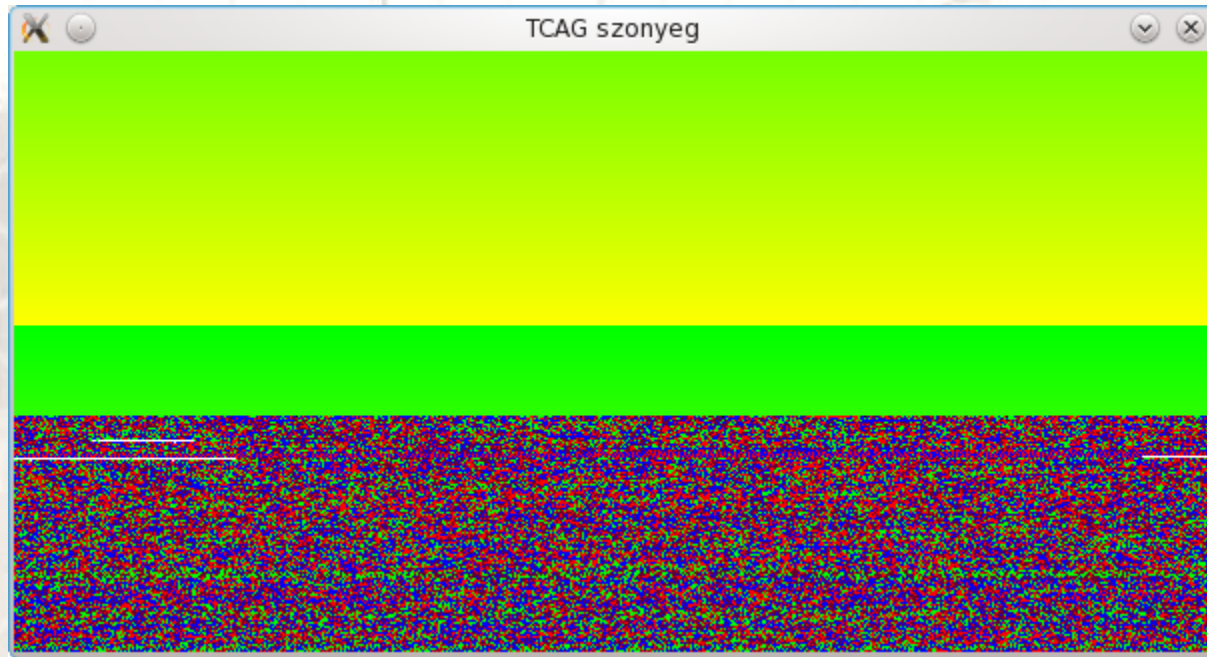
```
racsok = new bool**[2];  
racsok[0] = new bool*[magassag];  
for(int i=0; i<magassag; ++i)  
    racsok[0][i] = new bool [szelesseg];  
racsok[1] = new bool*[magassag];  
for(int i=0; i<magassag; ++i)  
    racsok[1][i] = new bool [szelesseg];
```

sejtablak.cpp

```
SejtAblak::~SejtAblak()  
{  
    delete eletjatek;  
  
    for(int i=0; i<magassag; ++i) {  
        delete[] racsok[0][i];  
        delete[] racsok[1][i];  
    }  
  
    delete[] racsok[0];  
    delete[] racsok[1];  
    delete[] racsok;  
}
```

Szőnyegen a humán genom

http://progpater.blog.hu/2011/03/05/szonyegen_a_human_genom



OO szempontból teljesen ua., mint az előző.

Szőnyegen a humán genom

http://progpatet.blog.hu/2011/03/05/szonyegen_a_human_genom

```
class GenSzal : public QThread
{
    Q_OBJECT

public:
    GenSzal(const char *fajlNev, int szelesseg, GenAblak *genAblak);
    ~GenSzal();
    void run();

protected:
    int szelesseg;
    GenAblak *genAblak;
    int *egySor;
    QFile *fajl;
};
```

genszal.h

<http://>

```
#include "g  
  
GenSzal::Gen  
{  
    this->s  
    this->g  
  
    egySor =  
  
    fajl =  
    fajl->op  
  
}
```

```
void GenSzal::run()  
{  
    while(true) {  
        QThread::msleep(80);  
  
        char c;  
        for(int i=0; i < szelesseg && (fajl->getChar(&c) != -1); ++i) {  
  
            if(c==0x0a)  
                continue;  
  
            switch(c)  
            {  
                case 'T':  
                    egySor[i] = qRgb (255, 0, 0);  
                    break;  
                case 'C':  
                    egySor[i] = qRgb (0, 255, 0);  
                    break;  
                case 'A':  
                    egySor[i] = qRgb (0, 0, 255);  
                    break;  
                case 'G':  
                    egySor[i] = qRgb (100, 0, 50);  
                    break;  
                default:  
                    egySor[i] = qRgb (255, 255, 255);  
                    break;  
            }  
        }  
        genAblak->vissza(egySor, szelesseg);  
    }  
}
```

genszal.cpp

Kezd ezzel, ha még nem tetted mással

http://progpater.blog.hu/2011/03/05/figyelem_ez_nem_gyakorlat

```
#ifndef ARECIBOABLAK_H
#define ARECIBOABLAK_H

#include <QtGui/QMainWindow>
#include <QtGui/QPainter>

class AreciboAblak : public QMainWindow
{
    Q_OBJECT

public:
    AreciboAblak(QWidget *parent) : QMainWindow(parent) {}

protected:
    void paintEvent(QPaintEvent *event) {
        QPainter qpainter(this);

        for(int i=0; i<magassag; ++i) {
            for(int j=0; j<szelesseg; ++j) {
                if(uzenet.at(i*szelesseg+j) == '1')
                    qpainter.fillRect(j*cellaSzelesseg, i*cellaMagassag,
                                       cellaSzelesseg, cellaMagassag, Qt::black);
                else
                    qpainter.fillRect(j*cellaSzelesseg, i*cellaMagassag,
                                       cellaSzelesseg, cellaMagassag, Qt::white);

                qpainter.setPen(QPen(Qt::gray, 1));
                qpainter.drawRect(j*cellaSzelesseg, i*cellaMagassag,
                                 cellaSzelesseg, cellaMagassag);
            }
        }

        qpainter.end();
    }
};

#endif // ARECIBOABLAK_H
```

Labor

PP 277

XIII.5.1 BogoMIPS

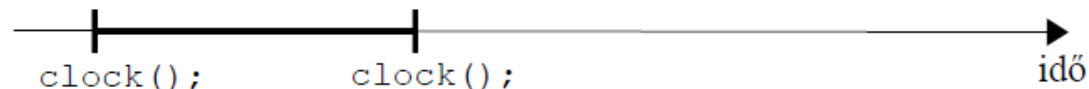
A Jeff Tranter által (Linus Torvalds Linux kernel-beli kódja alapján) írt standalone BogoMIPS izgalmas mérésekre ad lehetőséget.

Letöltés:

<http://packages.debian.org/stable/utils/sysutils.html>

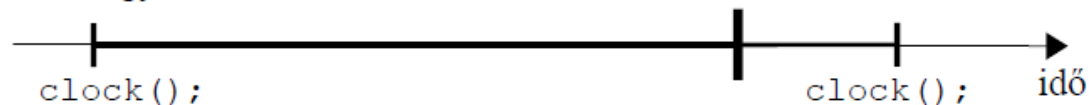
A működés vázolata:

A `delay(loops_per_sec);`
végrehajtásához szükséges
idő, legyen pl. 2^{26}



```
ticks = clock();  
delay(loops_per_sec);  
ticks = clock() - ticks;  
azaz a ticks a CLOCKS-ok száma itt.
```

A `delay(loops_per_sec);`
végrehajtásához szükséges
idő, legyen most akkor 2^{27}



```
ticks = clock();  
delay(loops_per_sec);  
ticks = clock() - ticks;  
azaz a ticks a CLOCKS-ok száma itt.
```

Tegyük fel, hogy itt a `CLOCKS_PER_SEC` érték, azaz már átléptük a 2 hatványos ciklusokkal. Fejezzük ki innen a másodpercenkénti ciklusokat:

$$\text{loops_per_sec/ticks} = ?/\text{CLOCKS_PER_SEC}$$

Ábra 51 A BogoMIPS működése

BogoMIPS

```
[morpheus@zion bogo-1.2]$ gcc -O2 bogomips.c -o bogomips
```

```
[morpheus@zion bogo-1.2]$ ./bogomips
```

```
Calibrating delay loop..0 2
```

```
0 4
```

```
0 8
```

```
0 16
```

```
0 32
```

```
0 64
```

```
0 128
```

```
0 256
```

```
0 512
```

```
0 1024
```

```
0 2048
```

```
0 4096
```

```
0 8192
```

```
0 16384
```

```
0 32768
```

```
0 65536
```

```
0 131072
```

```
0 262144
```

```
0 524288
```

```
0 1048576
```

```
0 2097152
```

```
10000 4194304
```

```
10000 8388608
```

```
40000 16777216
```

```
50000 33554432
```

```
100000 67108864
```

```
210000 134217728
```

```
420000 268435456
```

```
860000 536870912
```

```
1680000 1073741824
```

```
ok - 1278.00 BogoMips
```

Labor

Módosítsuk tovább!

```
#include <stdio.h>
#include <time.h>

void
delay (unsigned long l
{
    unsigned long long i
    for (j = 0; j < numi
        for (i = 0; i < lo
}
```

```
0 4611686018427387904
0 9223372036854775808
0 18446744073709551616.000000
0 36893488147419103232.000000
0 73786976294838206464.000000
```

```
int
main (void)
{
    unsigned long long int max_loops_per_sec, loops_per_sec = 1, numit = 1;
    unsigned long long int ticks;
    printf ("Calibrating delay loop...");
    fflush (stdout);
    while ((loops_per_sec <= 1))
    {
        max_loops_per_sec = loops_per_sec;
        ticks = clock ();
        delay (loops_per_sec, numit);
        ticks = clock () - ticks;
        printf ("%llu %llu\n", ticks, loops_per_sec);
        if (ticks >= CLOCKS_PER_SEC)
        {
            printf ("ok - %llu BogoMips\n",
                    ((loops_per_sec / ticks) * CLOCKS_PER_SEC) / 500000);

            return 0;
        }
    }
    while ((numit <= 1))
    {
        ticks = clock ();
        delay (max loops per sec, numit);
        ticks, (long double) max_loops_per_sec * numit);
        ER_SEC)

        BogoMips\n",
        (long double) max_loops_per_sec * numit / ticks) *
        ER_SEC) / 500000);

    }
}

printf ("failed\n");
return -1;
}
```

Lab

Teszteljük

iad010.inf.unideb.hu - PuTTY

```
0 256
0 512
0 1024
0 2048
```

norbi : bogomips

Fájl Szerkesztés Nézet Könyvjelzők Beállítások Súgó

```
0 128
0 256
0 512
0 1024
0 2048
0 4096
0 8192
0 16384
0 32768
0 65536
0 131072
0 262144
0 524288
0 1048576
10000 2097152
10000 4194304
20000 8388608
30000 16777216
80000 33554432
140000 67108864
290000 134217728
570000 268435456
1130000 536870912
ok - 950 BogoMips
[norbi@sgu ~]$
```

norbi : bogomips

```
ok - 340 BogoMips
[norbi@iam035 ~]$
```

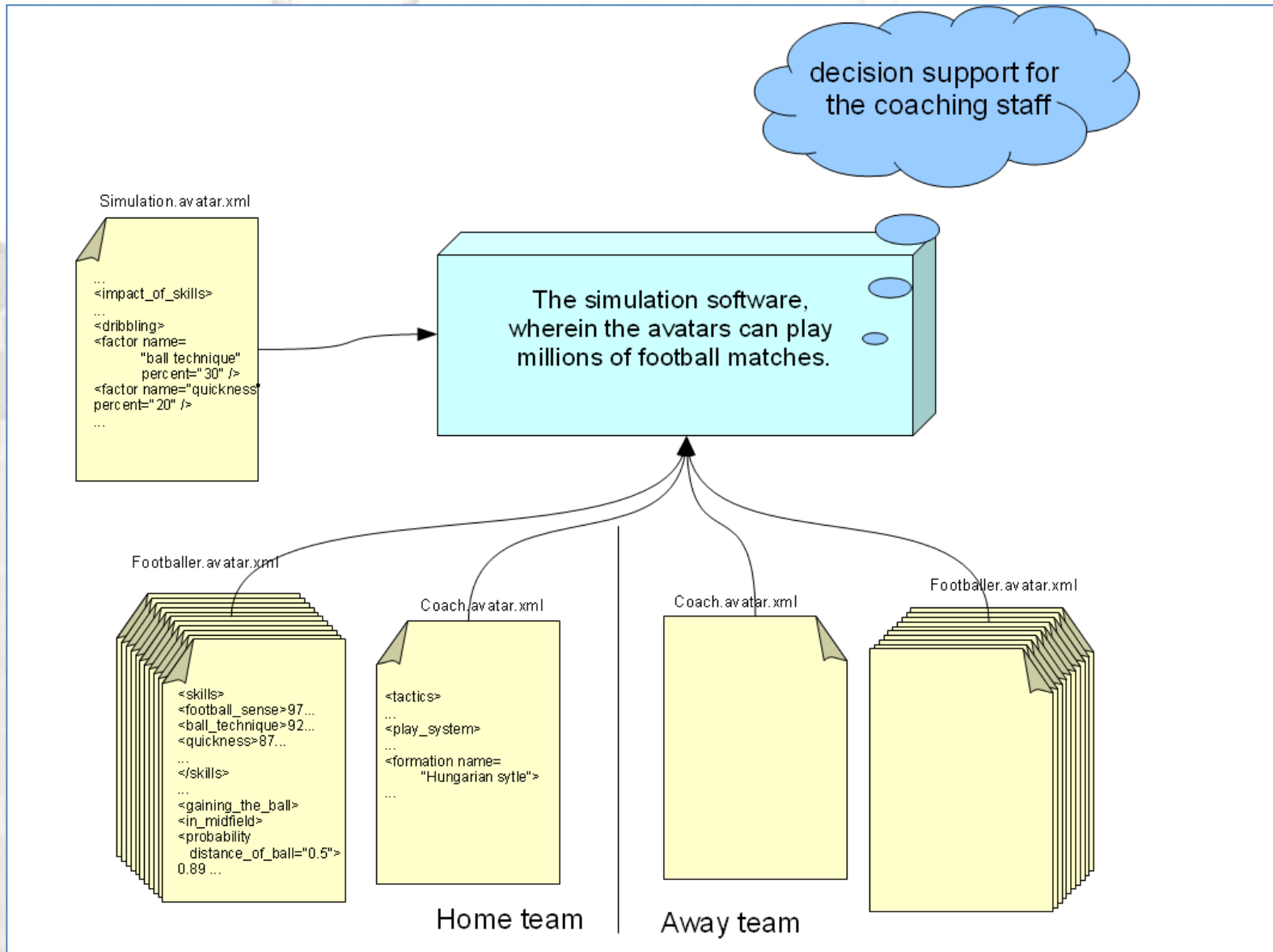
```
ok - 336 BogoMips
[morpheus@zion morpheus]$
```

Középkedés



WM (esetünkben 3-2-2-3), a „6-3”
4-2-4

Ötlet – szimuláció alapú döntéstámogatás



Célok

- Távoli cél: szimuláció alapú szakértői rendszer.
- Közelebbi cél: olyan szimulációs modell kialakítása, melyben a megfigyelt jellemzők ugyanolyan valószínűségi tulajdonságokat mutatnak, mint a valóságban.

A közelebbi cél

Lássunk egy példát! Eddig 10 olyan világbajnokságot rendeztek, amiben az első négy hely négy negyeddöntő, két elődöntő, döntő és a harmadik-negyedik helyért folyó mérkőzéssel alakult ki. Ha ezen a 10 tornán megvizsgáljuk a lőtt gólok számát és ugyanilyen tornákat szimulálunk a FerSML platformon, akkor a jelenlegi modellünket dicséri, hogy a szimulált és lőtt gólok számának eloszlása egyezik.

- Bátfai, N. (2010) *The Soccer Force*, **ArXiv e-prints**, 2010., <http://arxiv.org/abs/1004.2003>

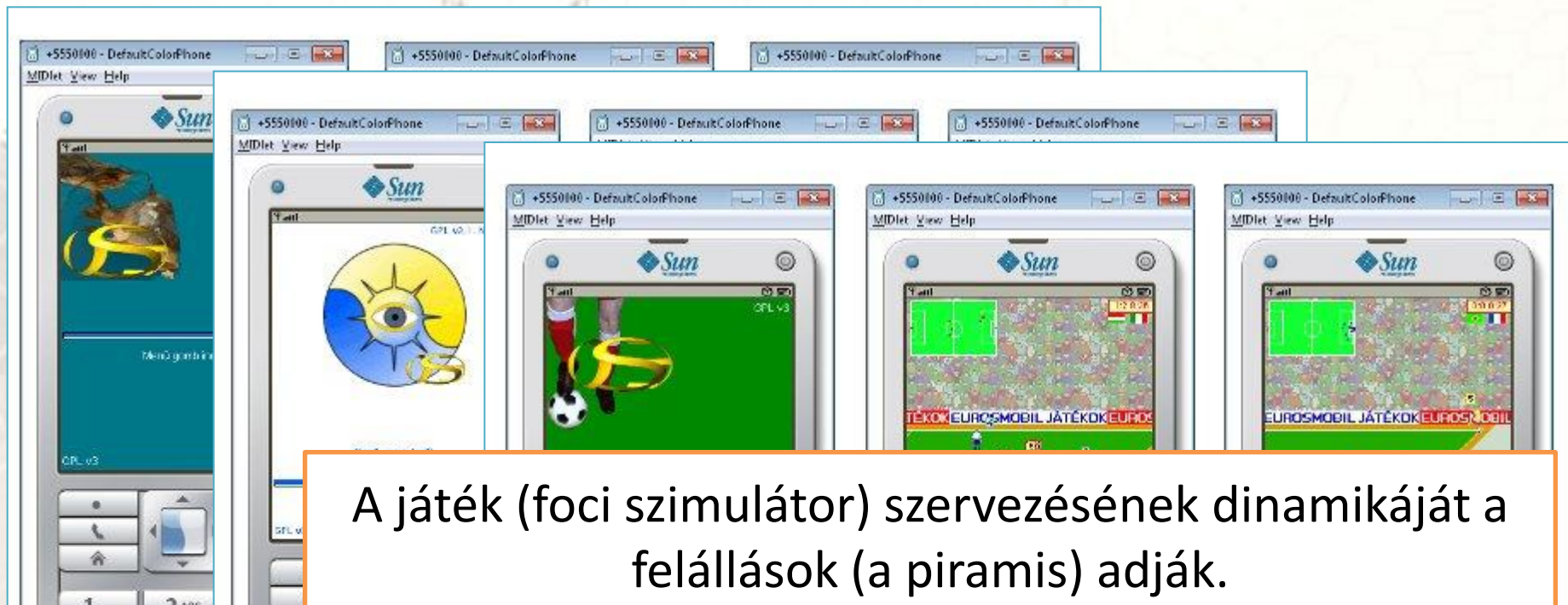
Gyökerek – Jávácska ONE

<https://sourceforge.net/projects/javacska/>

- 2008, Év Informatikai Oktatója (Eurosmobil Open Source bejelentése)
http://www.mvisz.hu/index/oktatoi_dij.html
- Bátfai N., “Mobiltelefonos játékok tervezése és fejlesztése”.
PhD doktori disszertáció. 2010. <http://www.inf.unideb.hu/~nbatfai/phd>
- N. Bátfai, E. Bátfai, I. Pšenáková: *Jávácska One: Open Source Mobile Games to Revolutionize Education of Programming*, **Journal of Computer Science and Control Systems**, Vol. 3., No. 2., pp. 5-10, (2010).
- Bátfai N, Bátfai M. E.: *A mobiljáték-fejlesztés elméleti és gyakorlati momentumai*. In: **Híradástechnikai Szemle**, LX. (2005) 5. p. 34-37.,
http://www.hiradastechnika.hu/data/upload/file/2005/2005_5/HT_0505-7.pdf

Gyökerek – Jávácska ONE

<https://sourceforge.net/projects/javacska/>



A játék (foci szimulátor) szervezésének dinamikáját a felállások (a piramis) adják.



Ötlet – szimuláció alapú döntéstámogatás

- N. Bátfai, „*Footballer and Football Simulation Markup Language and related Simulation Software Development*”, **Journal of Computer Science and Control Systems**, Vol. 3., No. 1., pp. 13-18, (2010).
- Bátfai N. (2010): *Bevezető számítások a labdarúgás szimulációs jelölőnyelv kialakításához*, **Híradástechnika**, LXV: 5-6, 16-20.
- Bátfai N., Ispány M., Terdik Gy., *Döntés előkészítő, információs és szimulációs rendszer a labdarúgásban*, „A Szilícium Mező Regionális Informatikai Klaszter eddigi fejlesztése és további tervei” konferencia, 2010. február 19. , Debreceni Egyetem, Kassai úti Campus, Inkubátorház.

Alternatív tabellák*

a nemzeti bajnokságok vizsgálatához

http://progpater.blog.hu/2011/03/11/alternativ_tabella

2009/2010 Soproni Liga	Pontszám	Alternatív tabella I.	Rangszám
Debreceni VSC	62	Győri ETO	0.0974
Videoton	61	Videoton	0.0823
Győri ETO	57	Zalaegerszegi TE	0.0813
Újpest	55	Ferencváros	0.0779
Zalaegerszegi TE	53	Újpest	0.0726
MTK Budapest	43	Debreceni VSC	0.0726
Ferencváros	41	Budapest Honvéd	0.0677
Szombathelyi Haladás	39	Kaposvári Rákóczi	0.0665
Budapest Honvéd	38	Paks	0.0643
Kecskeméti TE	37	Lombard Pápa	0.0619
Lombard Pápa	35	Budapest Honvéd	0.0611
Kaposvári Rákóczi	32	Újpest	0.0596
Vasas	31	MTK Budapest	0.0559
Paksi FC	31	Kecskeméti TE	0.0507
Nyíregyháza Spartacus	27	Győri ETO	0.0442
Diósgyőr	17	BFC Siófok	0.0431
		Vasas	0.0373
		Szombathelyi Haladás	
		Szolnoki MÁV	

- Bátfai N., Bátfai E. (2010) Nemzeti labdarúgó bajnokságok összehasonlító vizsgálata, Magyar Sporttudományi Szemle, beküldve.

*, http://hu.wikipedia.org/wiki/Alternat%C3%ADv_tabella

Irodalomkutatás és célkitűzés

Sport Science Journals:

Journal of Human Sport and Exercise
Journal of Quantitative Analysis in Sports
Magyar Sporttudományi Szemle

Mesterséges
intelligencia

Sporttudomány
és „coaching”

Robot foci
(2D szimulációs liga)

Az általunk fejlesztendő terület

Kitekintés – robotfoci

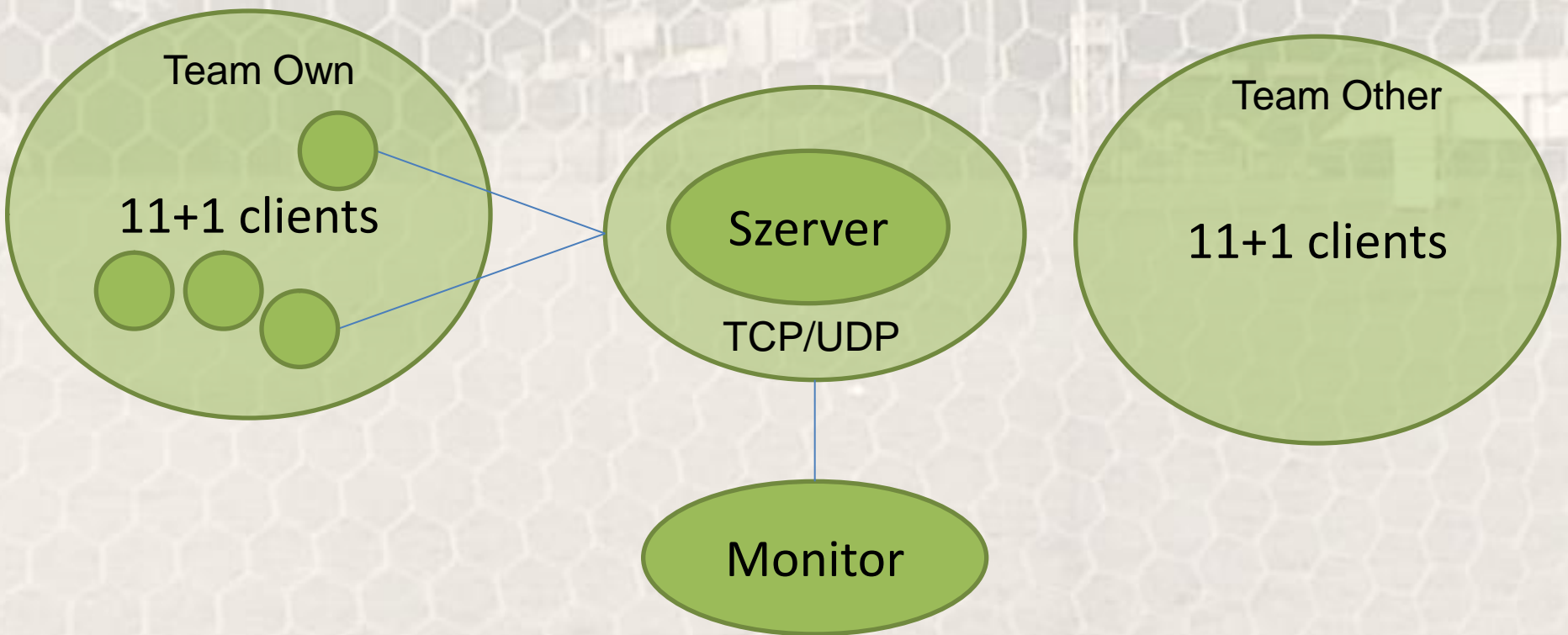
Miért alkalmatlan a mi céljainkra?

Jóval magasabb absztrakciós szinten mozgunk: például a piramis alapú üzem nálunk egy belépési pont, amit a robot focinál kialakítani már komoly eredmény. (Mi nem akarunk a semmiből egy olyan játékos ágenst kialakítani, aki rendelkezik a pálya és a játék egy belső reprezentációjával, így képes intelligens viselkedésre, mert triviálisan feltesszük, hogy ez adott.)

(Aki elkészíti saját robotfocis csapatát, tapasztalni fogja, hogy ugyanaz a fejlesztői élmény, mintha csak egy LEGO robotot programozott volna.)

Kitekintés – RoboCup Soccer 2D Simulation League

Alapcikk: Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. 1997. RoboCup: *The Robot World Cup Initiative*. In Proceedings of the first international conference on Autonomous agents (AGENTS '97). ACM, New York, NY, USA, 340-347.



The RoboCup Soccer Simulator

rcssserver

<https://sourceforge.net/projects/sserver/>

Egy szimulációs lépés:



6000 lépés 10 percben.

Laborkártyák

```
int a;  
int *b;  
int c[5];  
int *d[5];  
int e[3][3];  
int *f ();  
void* (*g)(void *);  
int &h = a;  
void fgv(int& i);
```

Mit takar az a, b, c, d, e, f, g, h, i?

Laborkártyák

Lefordul? Ha igen, mennyit ír ki?

```
#include <stdio.h>

void
csere (int &a, int &b)
{
    a = a - b;
    b = a + b;
    a = b - a;
}

int
main (void)
{
    int a = 5, b = 7;

    csere (5, 7);
    printf ("a=%d b=%d\n", a, b);

    return 0;
}
```

Laborkártyák

Lefordul? Ha igen, mennyit ír ki?

```
#include <stdio.h>

int
main (void)
{
    int a = 5;
    int *ap = &a;
    int &ar = a;
    int *ap2 = &ar;

    ++a;

    ++*ap;

    ++ar;

    ++*ap2;

    printf ("a=%d\n", a);

    return 0;
}
```

Laborkártyák

Lefordul? Ha igen, mit ír ki?

```
#include <stdio.h>

void
csere (int a, int b, int &c, int &d)
{
    c = b;
    d = a;
}

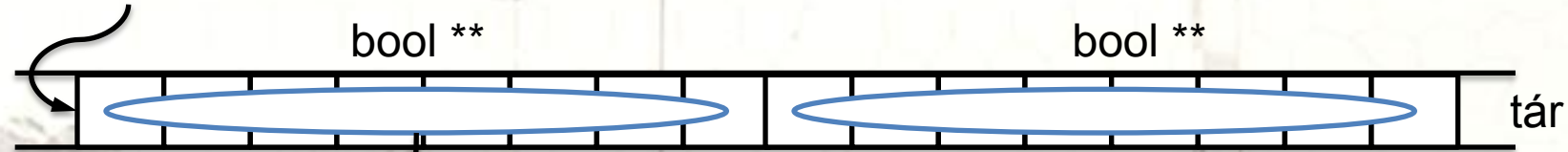
int
main (void)
{
    int a = 5, b = 7;

    printf ("a=%d b=%d\n", a, b);
    csere (a, b, a, b);
    printf ("a=%d b=%d\n", a, b);

    return 0;
}
```

Laborkártyák

`bool ***racsok`

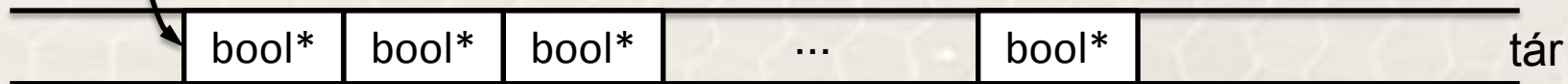


`racsok[0]`

`bool **racs = racsok[racsIndex];` tgyfh `racsIndex = 0`
... `if(racs[i][j] == ELO) ...` melyik cellát vizsgálja, ha
`i=1, j=szelesseg-1` és miért?
(sejtblak.cpp)

`racsok[0][1]`

0. 1. 2. ... magassag-1



0. 1. szelesseg-1



Otthoni opcionális feladat

A robotfoci japán szoftvereinek (librcsc, agent2d) tanulmányozása a KDevelop-ban.

The screenshot displays the KDevelop IDE interface. The main editor window shows the following C++ code snippet from `basic_client.cpp`:

```
int ret = ::select( M_socket->fd() + 1, &read_fds,
                  static_cast< fd_set * >( 0 ),
                  static_cast< fd_set * >( 0 ),
                  &interval );

if ( ret < 0 )
{
    perror( "select" );
    break;
}
else if ( ret == 0 )
{
    // no message. timeout.
    waited_msec += M_interval_msec;
    ++timeout_count;
    agent->handleTimeout( timeout_count,
                        waited_msec );
}
else
{
    //if(M_socket->fd(), &read_fds){
    // received message, reset wait time
    waited_msec = 0;
    timeout_count = 0;
    agent->handleMessage();
}
}
```

The Code Browser at the bottom shows the definition of `M_socket`:

```
boost::shared_ptr<rcsc::UDPSocket> M_socket
Container: BasicClient Access: private Kind: Variable definition
Decl.: basic_client.h :77 Show uses
! udp connection
```

On the right side, a 2D visualization of a soccer field is shown, titled "FerSML_team 0". The field is green with white lines, and several red and yellow robot icons are positioned on it. A black rectangle is visible on the right side of the field.

Otthoni opcionális feladat

A robotfoci japán szoftvereinek (librcsc, agent2d) tanulmányozása a KDevelop-ban.

```
rcsc::BasicClient client;

if ( ! agent.init( &client, argc, argv ) )
{
    return EXIT_FAILURE;
}

/*
    You should add your copyright message here.
*/
std::cout << "*****\n"
            << " This program is modified by FerSML team\n"
            << " Copyright 2011. Norbert Bátfai.\n"
            << " University of Debrecen\n"
            << " All rights reserved.\n"
            << "*****\n"
            << std::flush;

/*
    Do NOT remove the following copyright notice!
*/
std::cout << "*****\n"
            << " This program is based on agent2d created by Hidehisa Akiyama.\n"
            << " Copyright 2006 - 2010. Hidehisa Akiyama.\n"
            << " National Institute of Advanced Industrial Science and Technology\n"
            << " All rights reserved.\n"
            << "*****\n"
            << std::flush;

client.run( &agent );

return EXIT_SUCCESS;
}
```

main_player.cpp

Kötelező olvasmány

Benedek Zoltán, Levendovszky Tihamér: Szoftverfejlesztés C++
nyelven, Budapest, 2007, Szak K.

487-489

1-16

73-80