Rendszermodellezés: házi feladat bemutatás

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group

Motiváció

- Modellezési tapasztalat/jártasság:
  - (szoftver) komponens **határainak** definiálása
  - rendszer **állapotainak** felírása
  - **események** és **átmenetek** modellezése

  → **Specifikáció megértése, modellalkotás!**
- Ami megtanulható:
  - Rendszer tervezése, absztrakt mérnöki gondolkodás
  - Korszerű eszközben
  - Alkalmazásfejlesztés kódolás nélkül
  - Validálás/ellenőrzés: szimuláció + tesztelés +…

Házi feladat: digitalis sakkóra

Sakkóra viselkedési (állapot) modellje

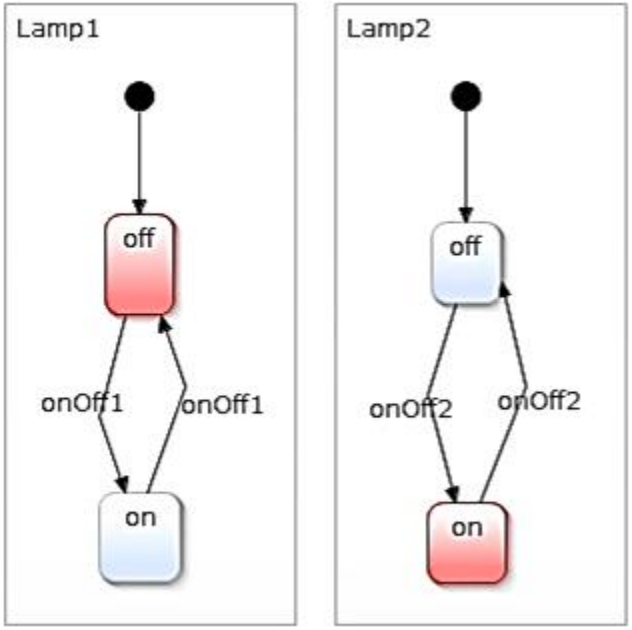
- A játékosok **hátralévő** idejének kijelzése
- Játék végének **(időtúllépés) jelzése**
- **Állítható paraméterekkel** (pl kezdeti idő)

Yakindu bevezető

- Állapotmodellező eszköz
  - Egyszerű felület
  - Platformfüggetlen modell

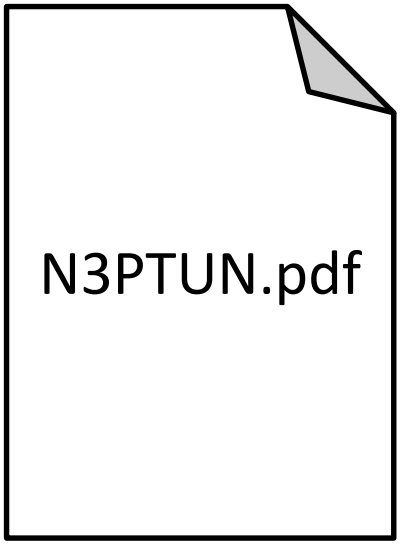
- Szerkesztés
- Szimuláció
- Automatikus kódgenerálás: model →

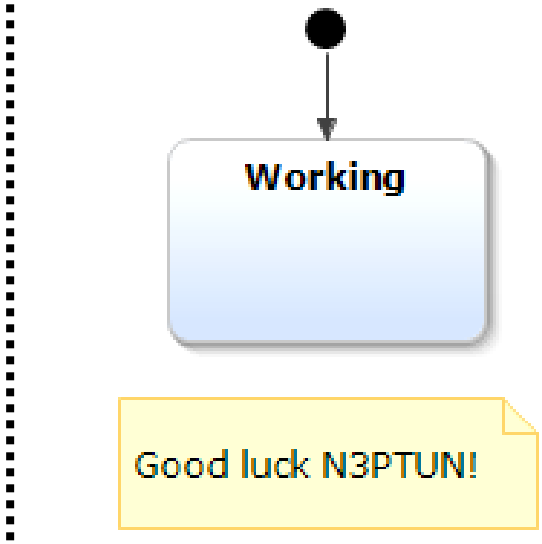
  C/C++, Java, LEGO MindStorms



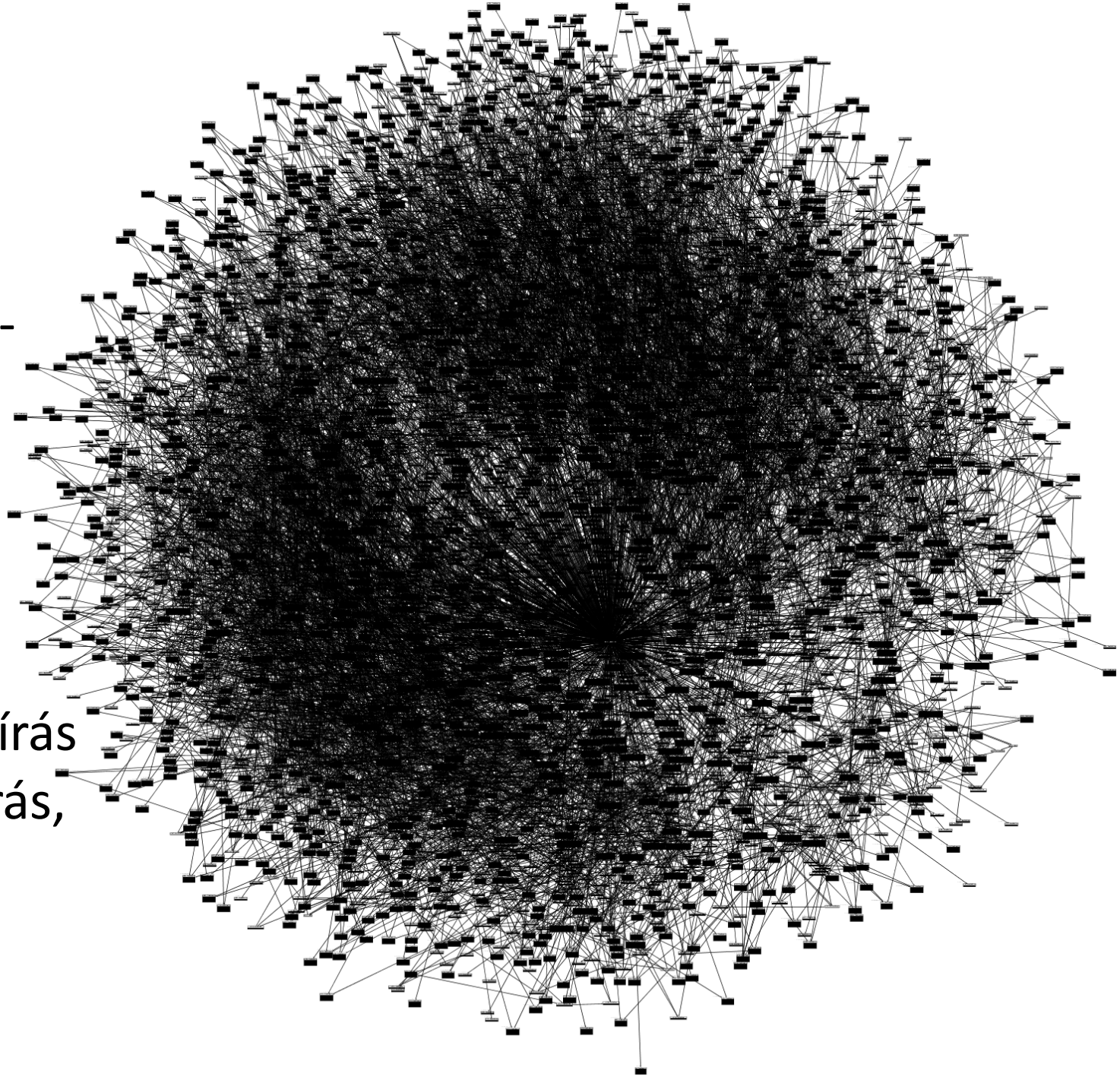
Házi feladat áttekintő

- **Egyedi variánsok** a sakkórára
  - Feladatkiírás, kiindulási modell
  - Feladat: specifikációnak megfelelő modell megalkotása
- Cél: jelentős különbségek legyenek, nehézség állandó

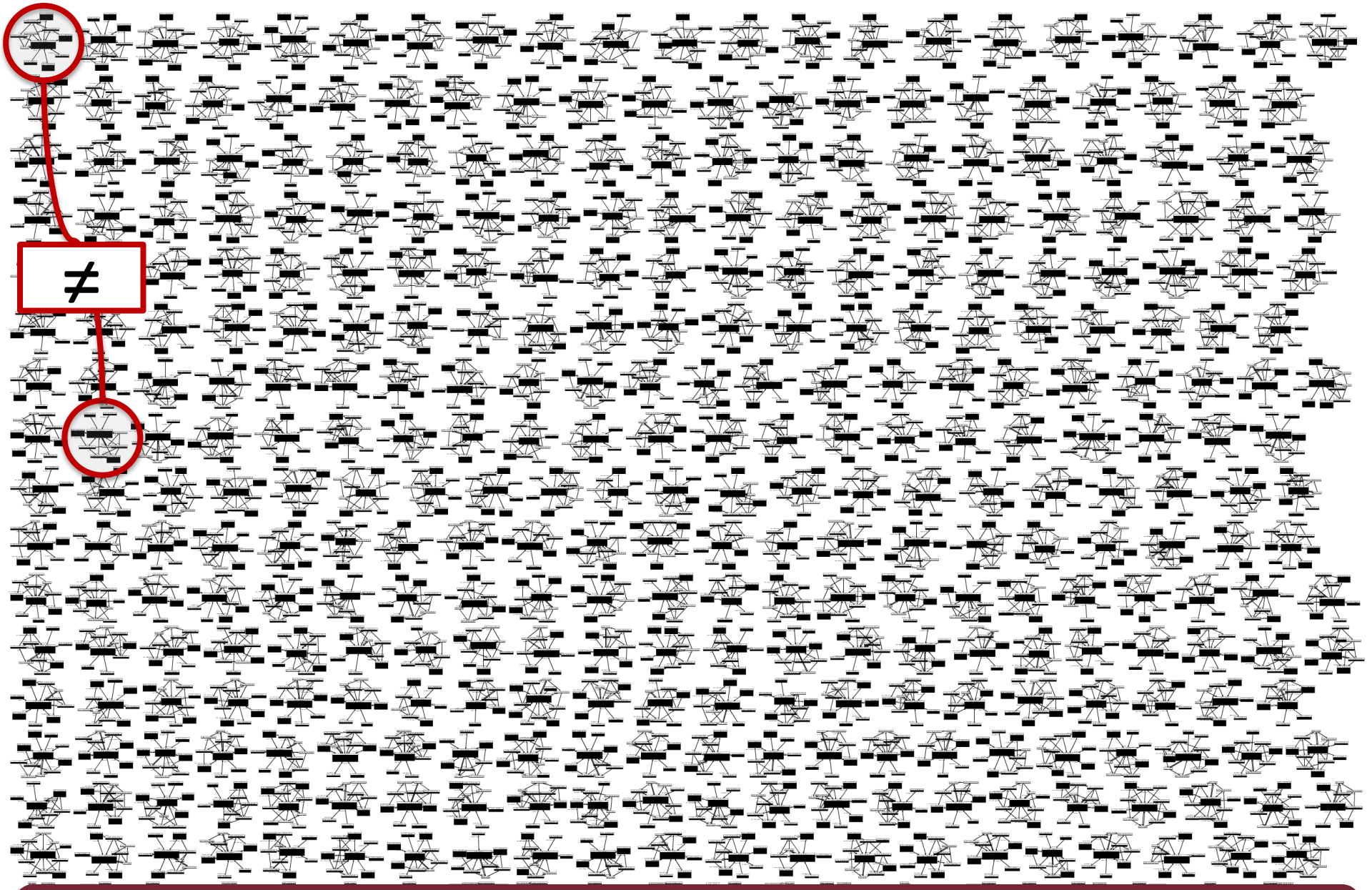




Háttér:
Minden házi feladathoz egyedi követelmény-modell



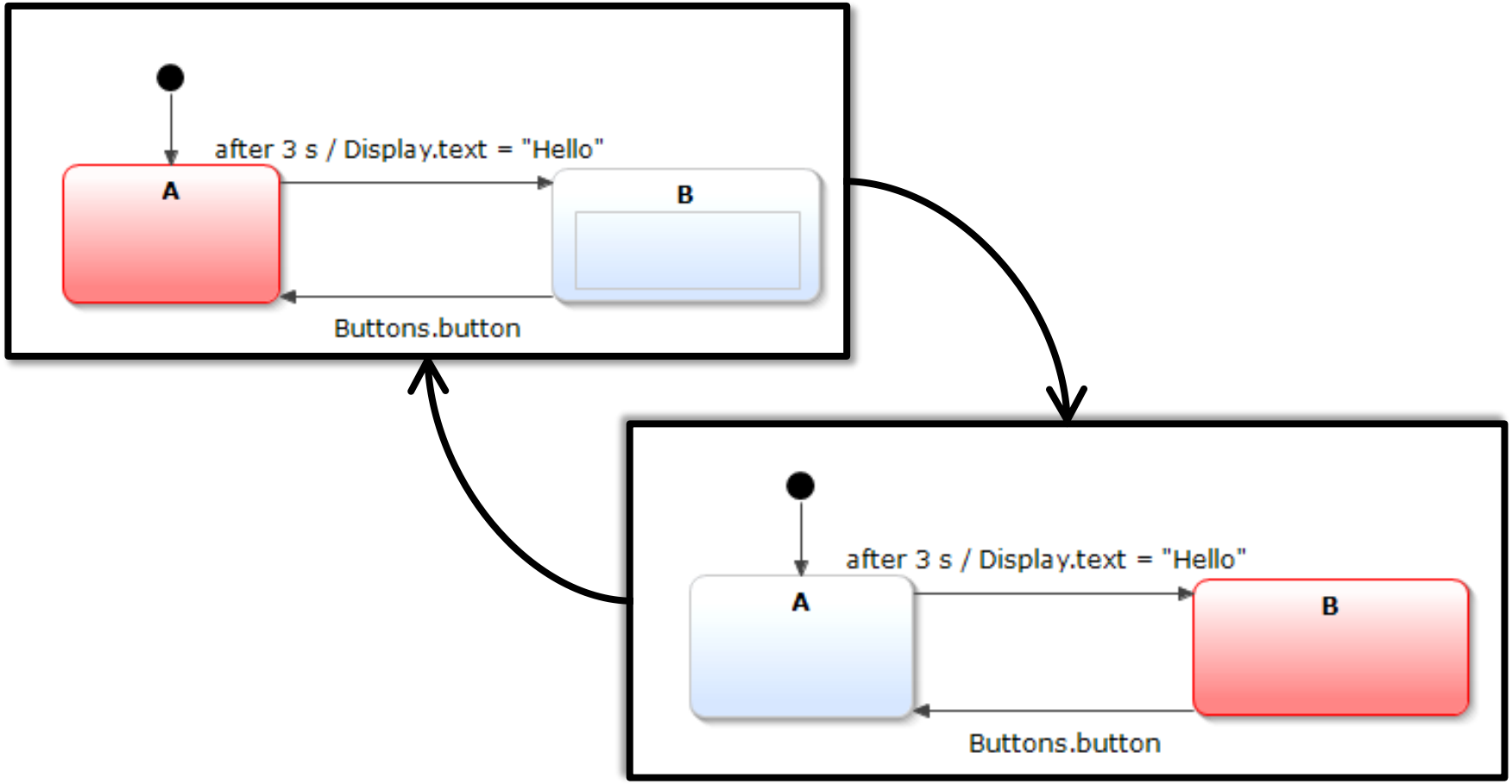
Gráf alapú leírás → feladatkiírás, projektváz, tesztesetek



Részfeladatok (taskok) kombinációja

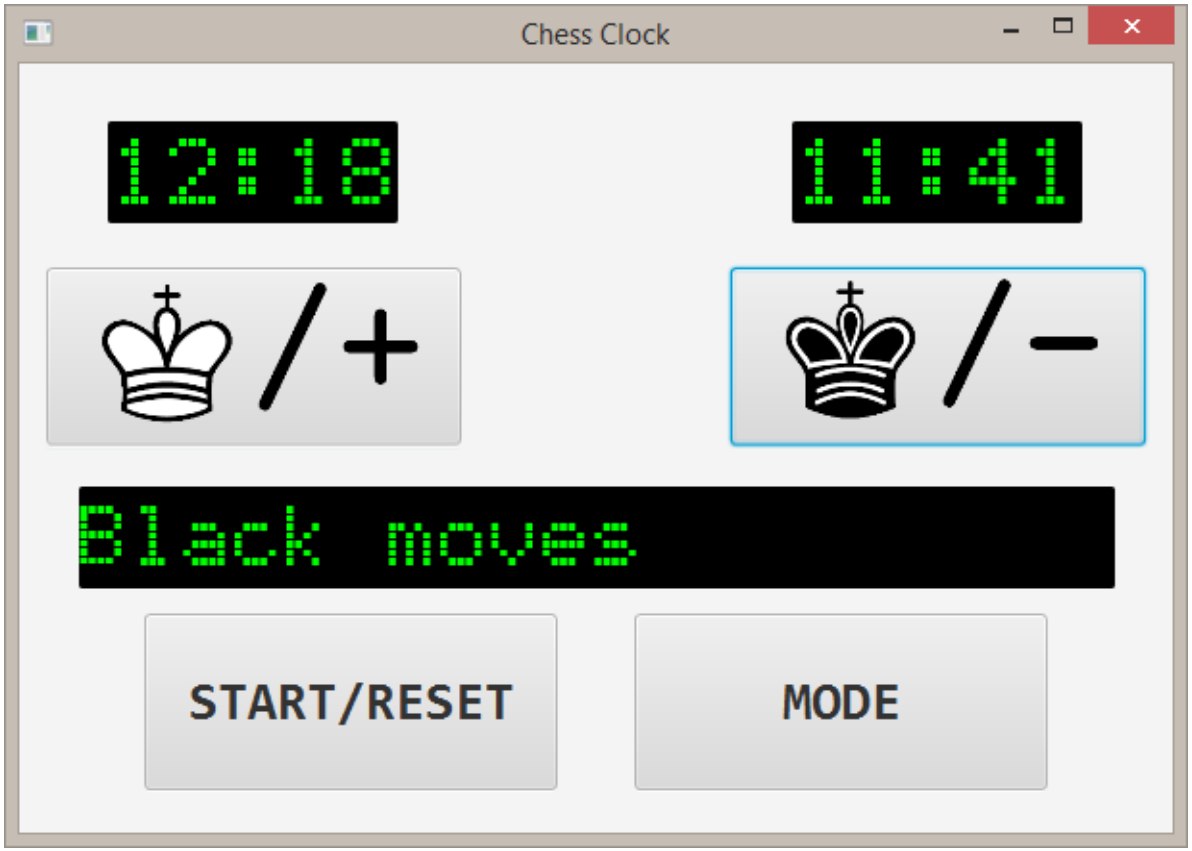
Házi feladat áttekintés: szimuláció

- Az elkészült modell “futtatható”, a szimulált rendszer állapota követhető



Házi feladat áttekintés: grafikus alkalmazás

- Java alkalmazás, közvetlenül a modellből generált kódot hívja meg
- Felület egységes
- Gombok →
  Események
- Változók →
  Kijelzők



Házi feladat áttekintés: ellenőrzés

- Kezdeti tesztesetek→ beadás előtt saját gépen kipróbálható
- Hiba esetén visszajelzés

```
neptun2 Failed:
After pressing a button it shows your Neptun code XXXXXX
----------
 - Button at 0s
 - Failed main display check: expected "XXXXXX" but found "Other text"
```

- Funkcionális tesztek:
  jó megközelítés⇔ kielégíti a követelményeket
- Pontozás kiterjesztett tesztkészlet végrehajtása (+modell megnézése+szóbeli védés) után

Bemelegítő házi feladat

- Egyszerűsített feladat, melynek célja:
  - Eszköz megismerése
  - Tesztelés kipróbálása
  - Beadófelület kipróbálása
  - Független a kötelező házi feladat kiírástól
- Feladatok:
  1. Bemelegítő feladat elolvasása, projektváz letöltése
  2. Egyszerű modell létrehozása
     "Hello World" → Neptunkód → "Hello World" → …
  3. Modell tesztelése
  4. Modell feltöltése → Jutalom pontok☺ (ha jó…)

Határidők és feltételek: bemelegítő feladat

- Elérhető a 4. hét végén
  - feladatkiírás + projektváz + tesztesetek
- Leadási határidő: 5. hét vége (március 15. 23:59)
- Elektronikus feltöltés a honlapon
  - Model (.SCT)
  - NEM kötelező leadni
  - DE a határidő után nem tudjuk elfogadni
- Értrékelés: Max: 5 pont
  - Jutalompont HA a ZH1+ZH2+HF külön-külön megvan

Határidők és feltételek: házi feladat

- Egyedi feladatkiírás elérhető (3. héttől)
- Egyedi projektváz + kiadott tesztesetek +GUI: 5. héttől
- Leadás: 12. hét vége (május 3. 23:59)
  - Modell (.SCT) feltöltése
  - Automatikus ellenőrzés előre ki nem adott tesztekkel is
- Egy javítási lehetőség a leadott feladatokra legfeljebb a 12. hét végéig
- Max: 30 pont
- Szóbeli védés a 13. héten
  - Csak az automatikus szűrésen minimumot elért feladatokra

További segédanyagok

- Yakindu hivatalos tutorial
- Tanszéken készült Yakindu tutorial
- Videó
- QA oldal
- Mindezek összegyűjtve:

https://inf.mit.bme.hu/edu/courses/remo/hazifeladat

Demó

- Feladat
  - Yakindu példa http://statecharts.org/documentation.html
- “The example application we will create during this tutorial is a system for handling of incoming phone calls. After start up, the system is in an Idle state and waits for incoming calls. If a call comes in, the user can either accept the call and open a connection or dismiss it. If the connection is opened, the system tracks the duration of the call and waits for the user to hang up. After hang up, the system displays the total time of call and returns to its idle state. ”