

Att välja och planera ett projekt



Steg 3: Utforska, skapa ett eget projekt!

- I projektet ska ni:
 - **Designa och implementera** ett OO-program
 - **Utan detaljerad styrning** från en fördefinierad uppgift
 - **Lära er** av erfarenheten – framgångar och motgångar
 - **Visa upp** vad ni lärt er – *vid demo/inlämning*

D1, kogvet

Arbete över 2 perioder,
tid att låta idéer mogna

Före tentaperioden:

Tid avsatt till att *hitta på* ett projekt,
skriva en *projektbeskrivning*

Tentaperioden:

Vi granskar, ger kommentarer
om det verkar finnas *uppenbara problem*

Inga garantier att vi hittar *allt!*

Eget ansvar för genomförande.

Får baseras på exempelprojekten,
måste i så fall vidareutvecklas
med mer detaljerad beskrivning

U1

Arbete över 1 period,
mindre tid att arbeta fram egna projekt

Projektförslag kommer att finnas:

Outline med milstolpar,
inspiration som ni kan bygga vidare på

Inför projektstart:

Delar av projektbeskrivningen
fyller ni i för er egen skull

Förslagen visar inte allt,
ska vidareutvecklas under eget ansvar!

Kan (gärna) välja helt eget projekt,
skicka in fullständig projektbeskrivning

Allmänt: Arbeta gärna i par



Ny gruppindelning, grupper om 2 rekommenderas –
cirka 80 timmar arbete per person

Kan samarbeta över klassgränser: [D1a/b/c+ Kogvet], [U1a/b]

Jag skulle också säga att valet att jobba själv nog ändå bör undvikas, även om det finns fördelar så tror jag att det är bättre att jobba i grupp. Att ha en projektpartner att diskutera med är lärorikt.

Att jobba själv har fördelar och nackdelar. En fördel är att det är lätt att hitta tider att jobba men samtidigt finns nackdelen att det kan vara svårt att faktiskt göra det då det inte finns någon annan som förväntar sig att man ska jobba.

Att välja projekt

- Projekt väljs fritt, men:
 - **Inte** fortsättning på Tetris
 - **Inte** Android-projekt (för mycket Android-specifikt)
- Visa vad ni har lärt er om objektorienterad programmering
 - Undvik projekt som kräver alltför mycket:
 - "Lågnivåprogrammering", matrismatematik, ...
 - Grafik, ljud, animering
 - ...

- Fundera på **stegvisa utökningar!**
 - Liten "kärna" som kan implementeras snabbt
 - Många finesser som kan implementeras efter hand
 - Möjlighet att stoppa när tiden är slut!

- → Lista på milstolpar.

- 1. ...

- 2. ...

- 3. ...

- ...

- 10. ...

- 11. ...

- ...

- ...

- 25. ...

**Enkelt att implementera,
ändå testbart**

**Lagom stora steg
att utöka med**

**Hit hinner vi säkert inte!
(Men vem vet...)**

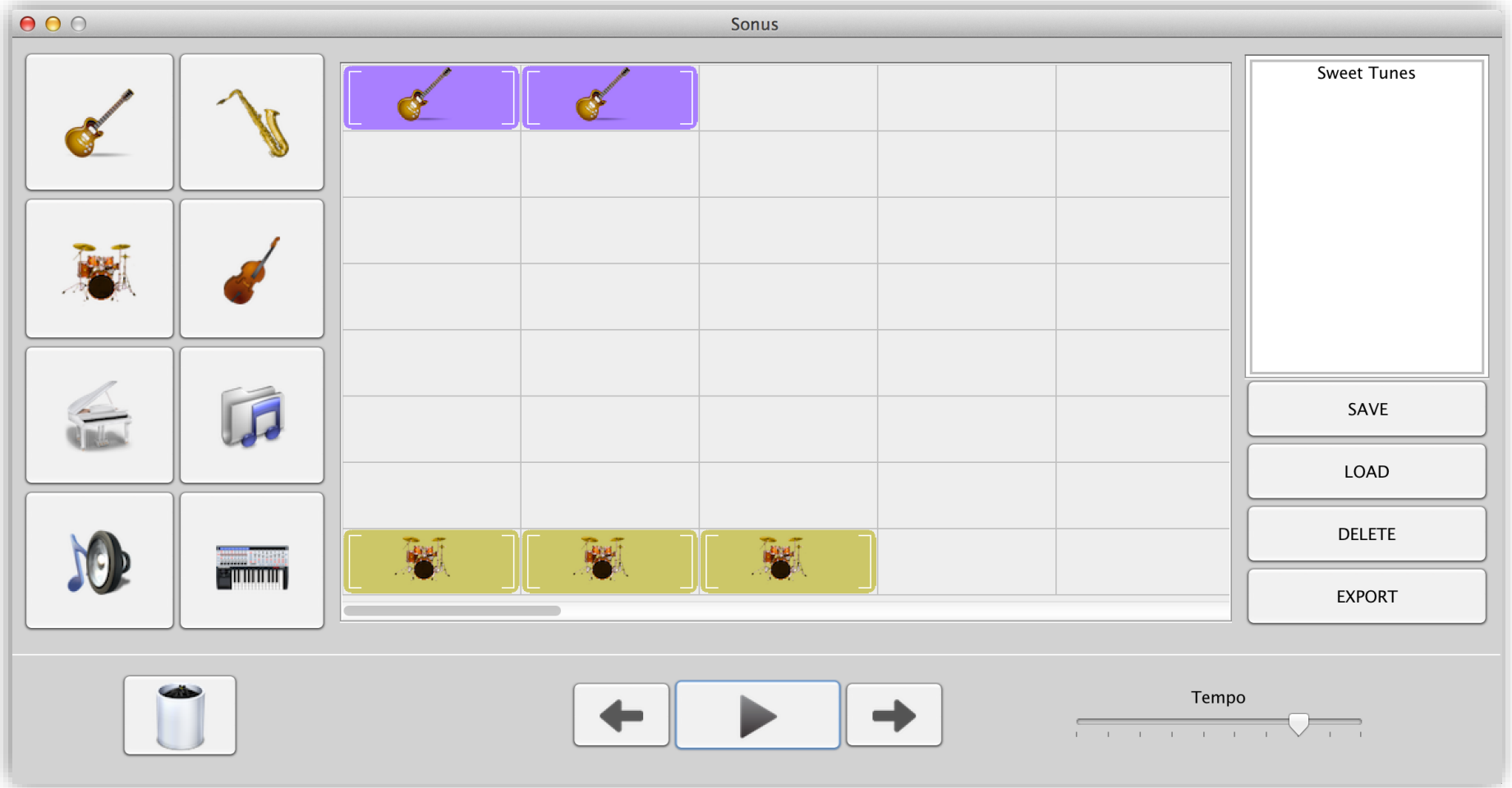
**Exempelprojekt
har redan en
generell outline**














**Kan behöva
utökas**

**Bör justeras efter
egna önskemål**

Exempel

Sonus



Sweet Tunes

SAVE

LOAD

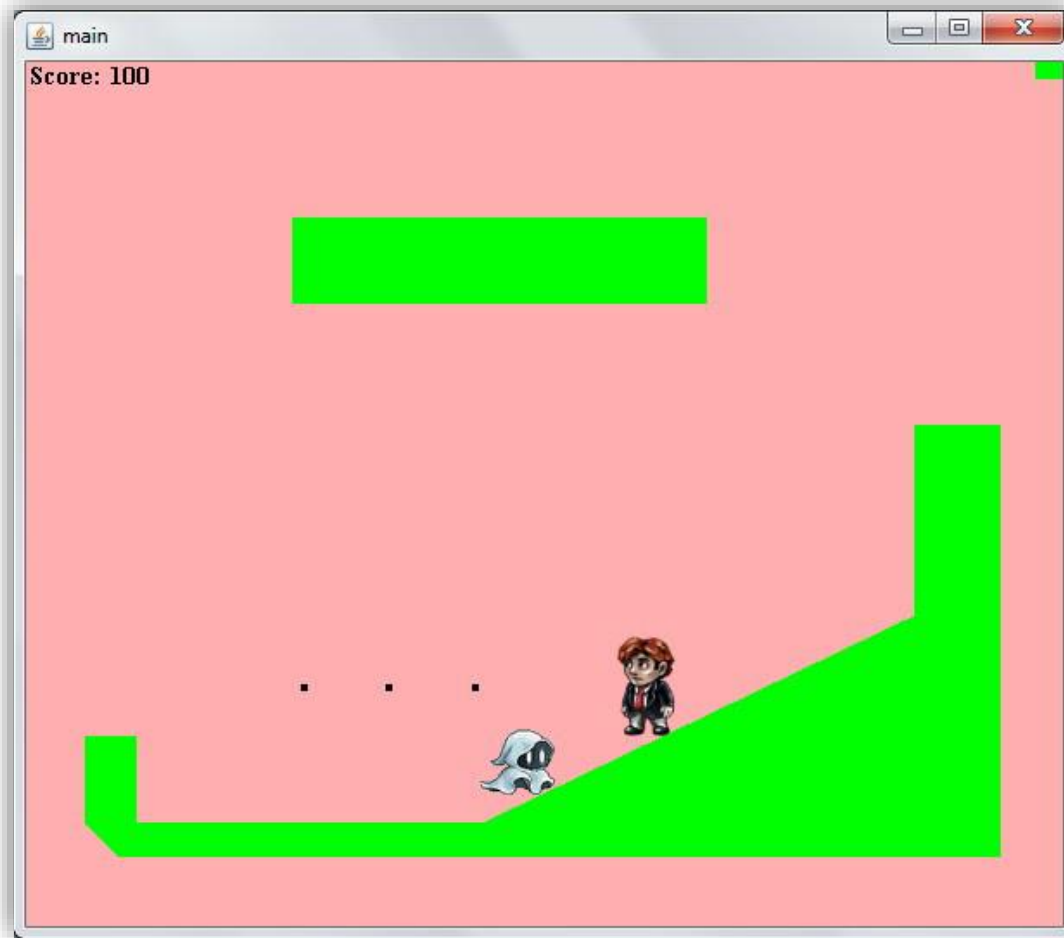
DELETE

EXPORT

Tempo

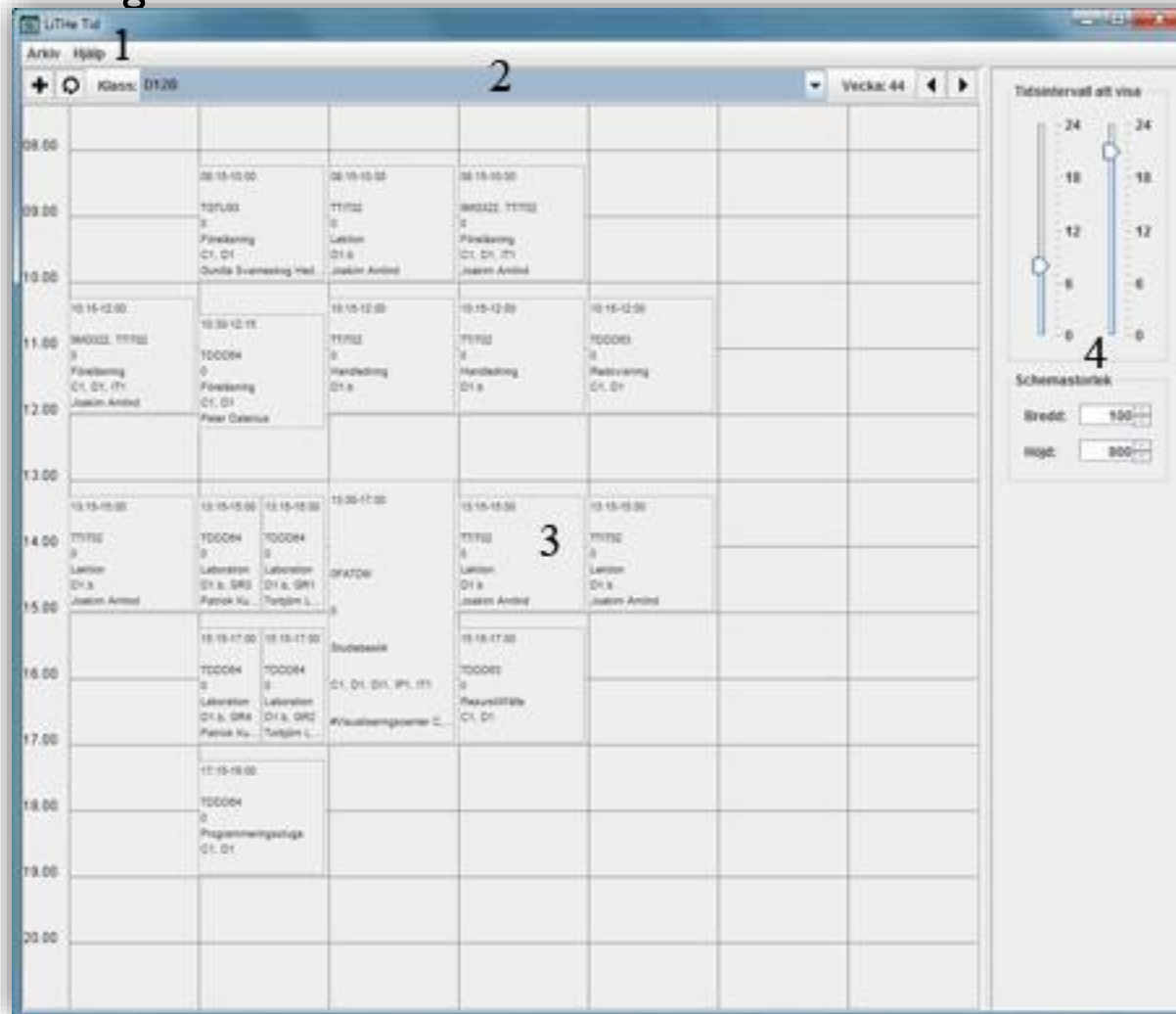
Exempel

- Sidoscrollande spel



Exempel

- Kalenderprogram
 - Kanske med gränssnitt mot TimeEdit



■ Tower Defence

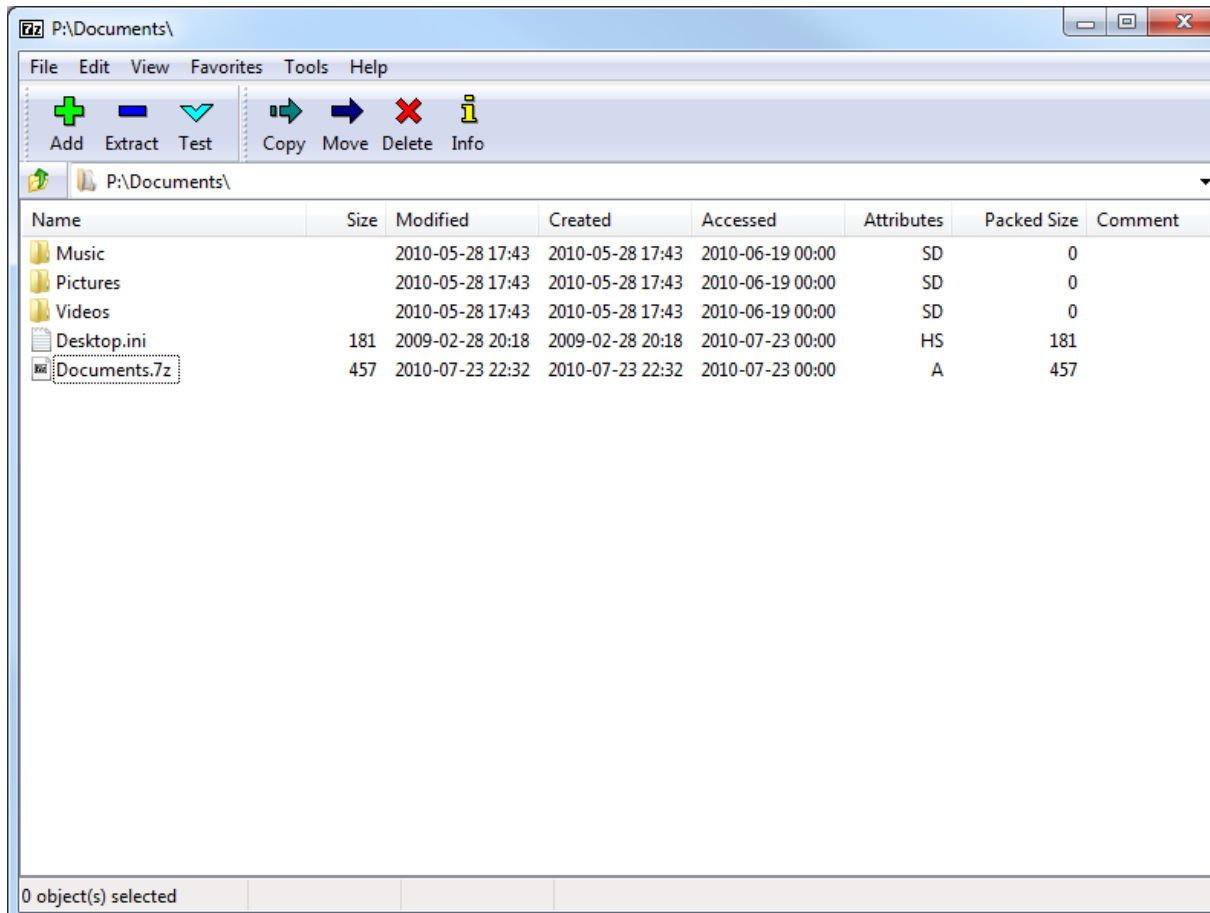


The screenshot shows the SmartTD 0.1 game interface. On the left is a menu with a grid of numbers (0, 5, 8) representing resources or costs. The main area is a green map with a grey path and several towers: a yellow tower, an orange tower, a blue tower, and a red tower. A grey path leads from the bottom left towards the top right. The right side of the interface displays 'Information about current tower' with the following details:

Type	Placeholder
Level	1
Exp	35
MultiShoot?	1
NrOfTargets	0
Rate of Fire	1.0
Buffers	5 0.0
DPS	10.0 Dmg 10 (+ 5)
Current WaveNr	2 Time to next Wave 0
mainTowers	20, 10, 10,
Gold	30
Lives	98

Below the map are three buttons: 'Build tower A', 'Build tower B', and 'Sell'.

- Zip-verktyg

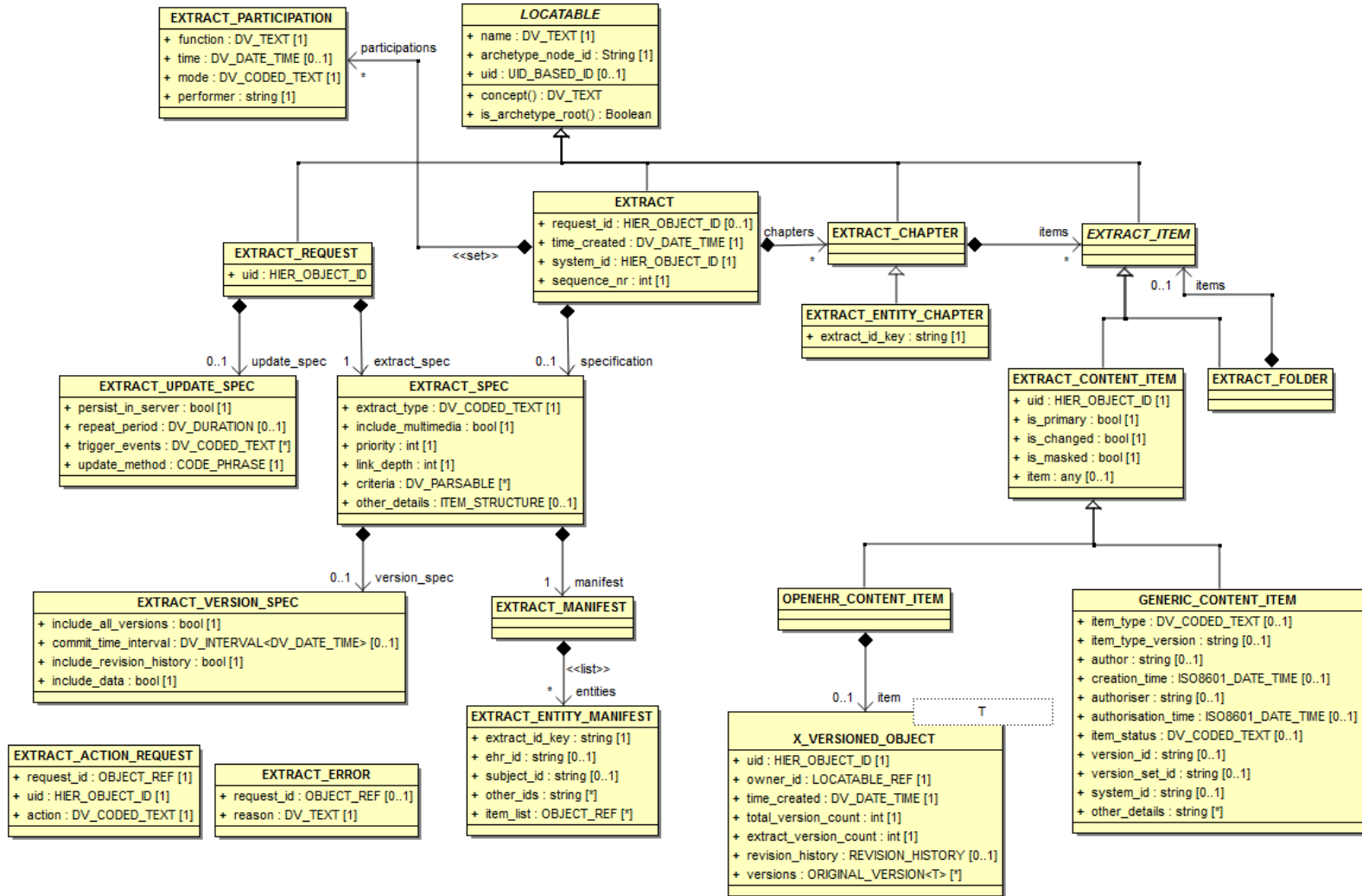


(Bilden är inte från kursen)

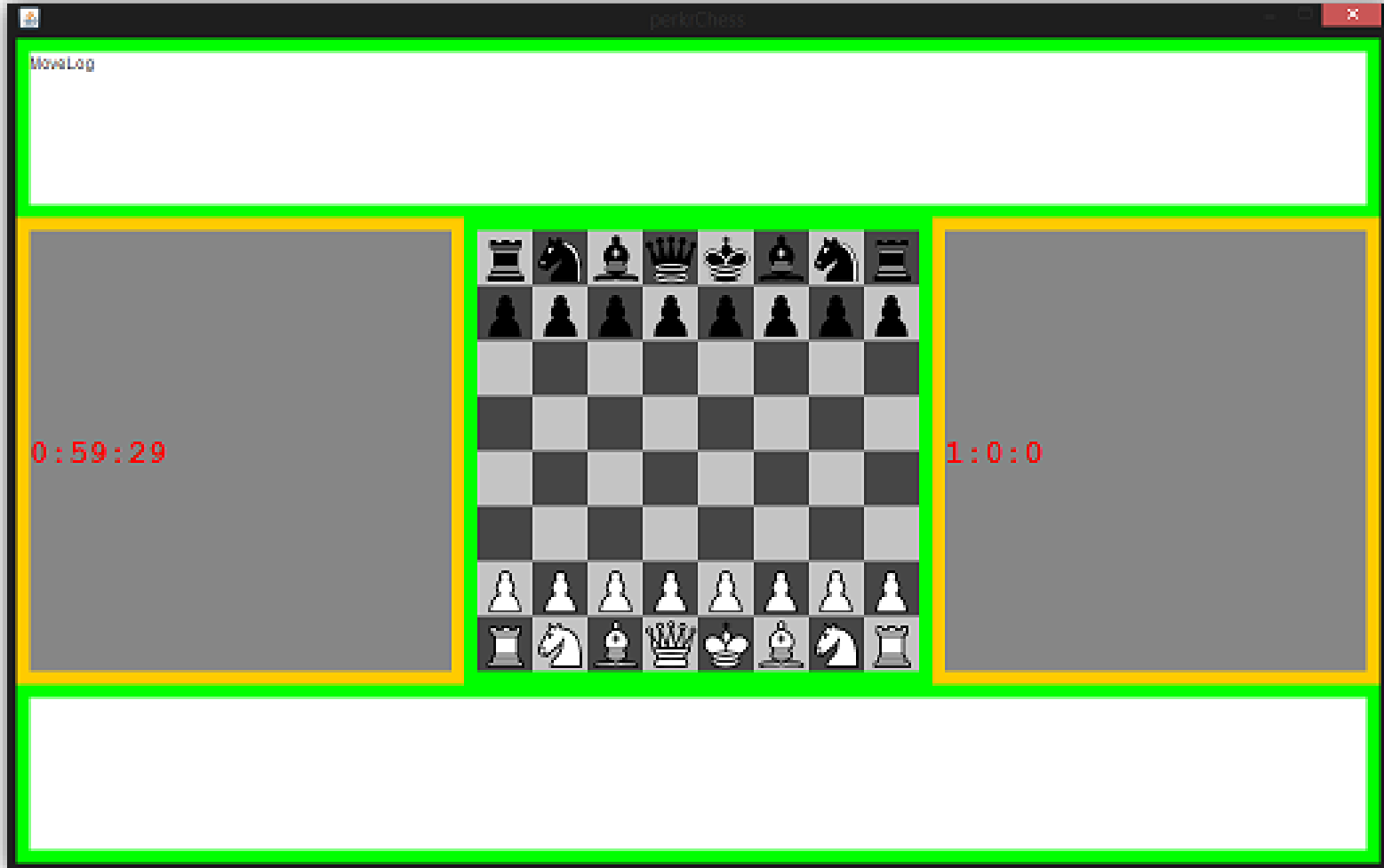
- Treasure Hunter



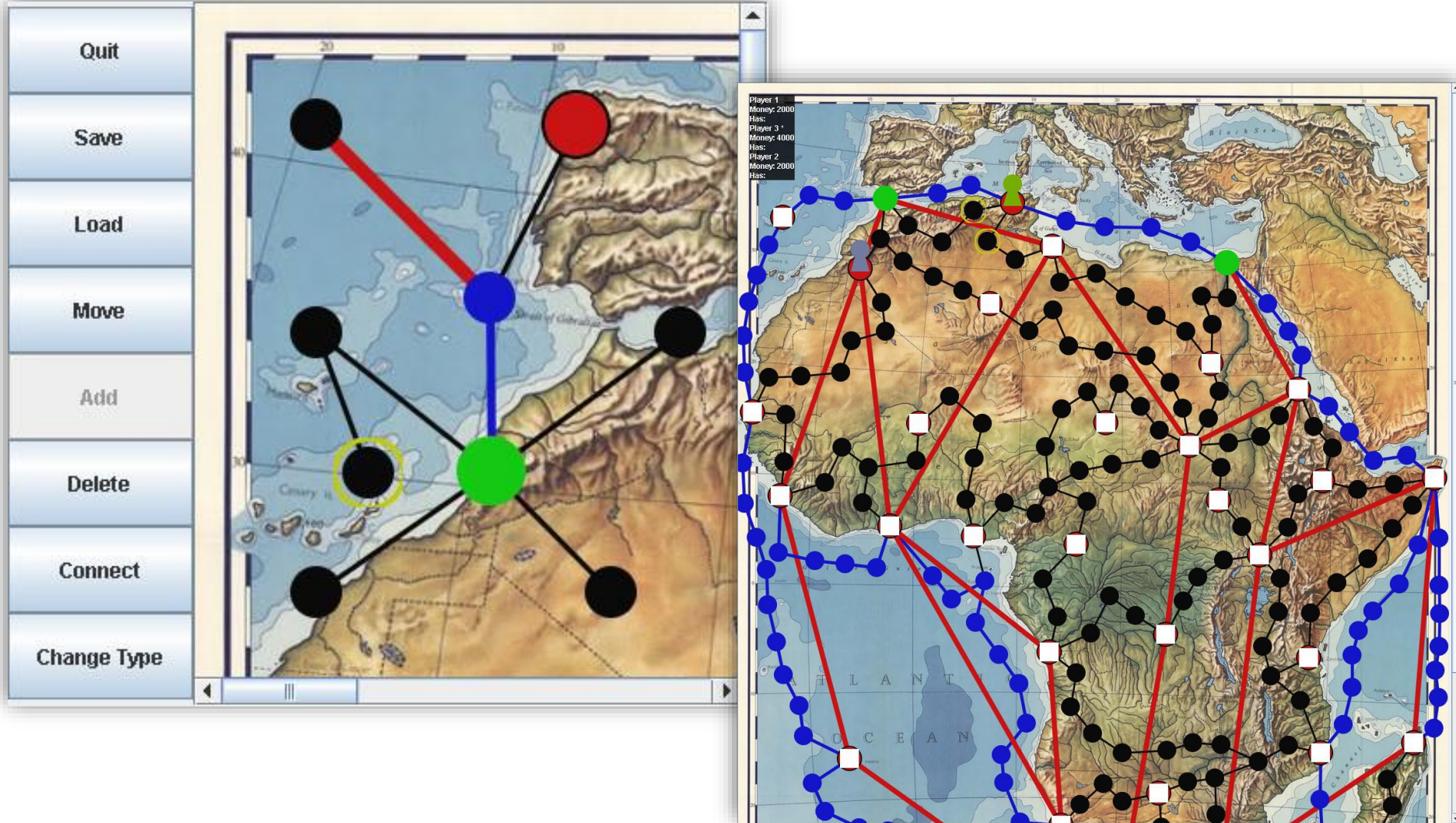
UML-visare



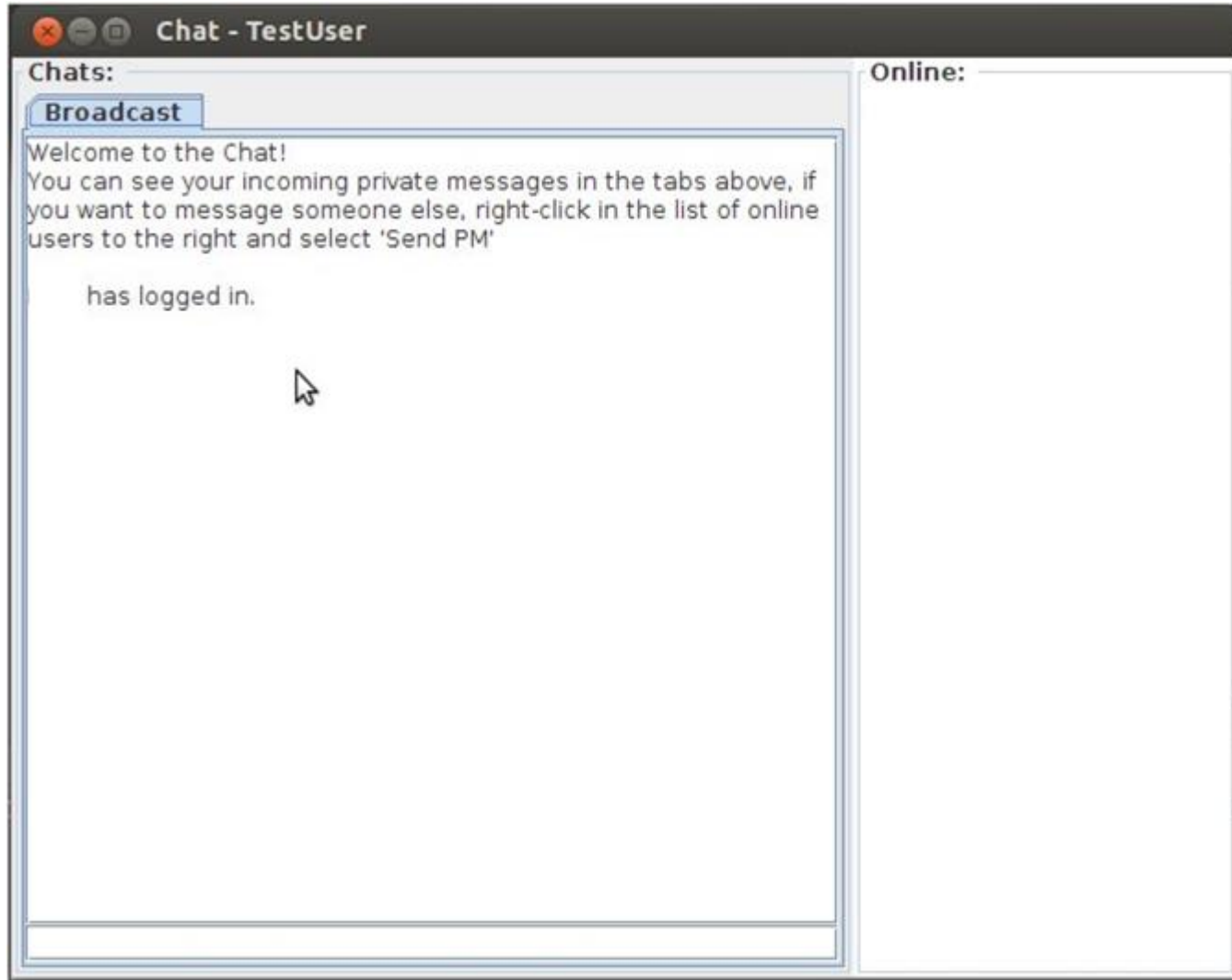
- Schack



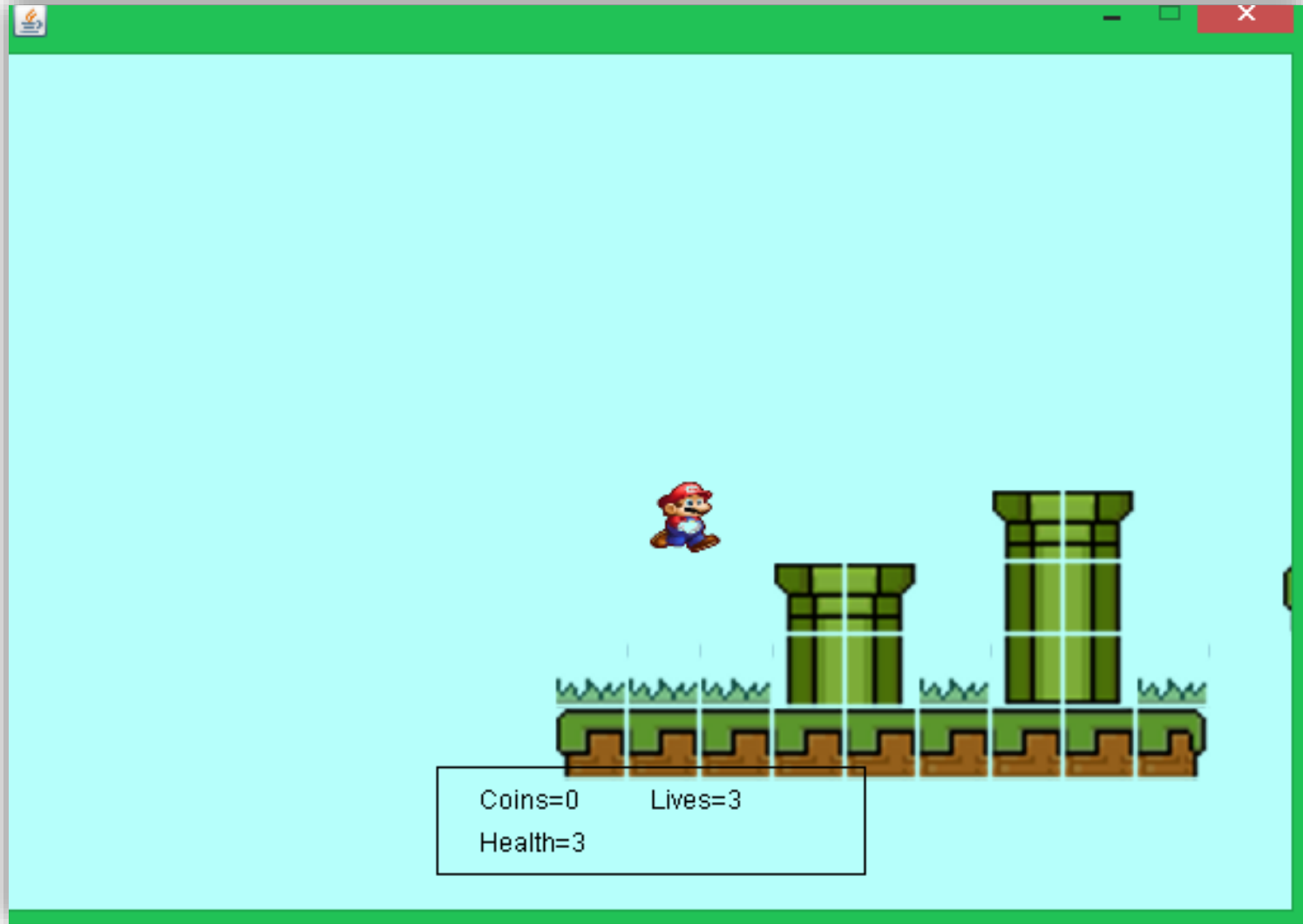
■ Brädspel



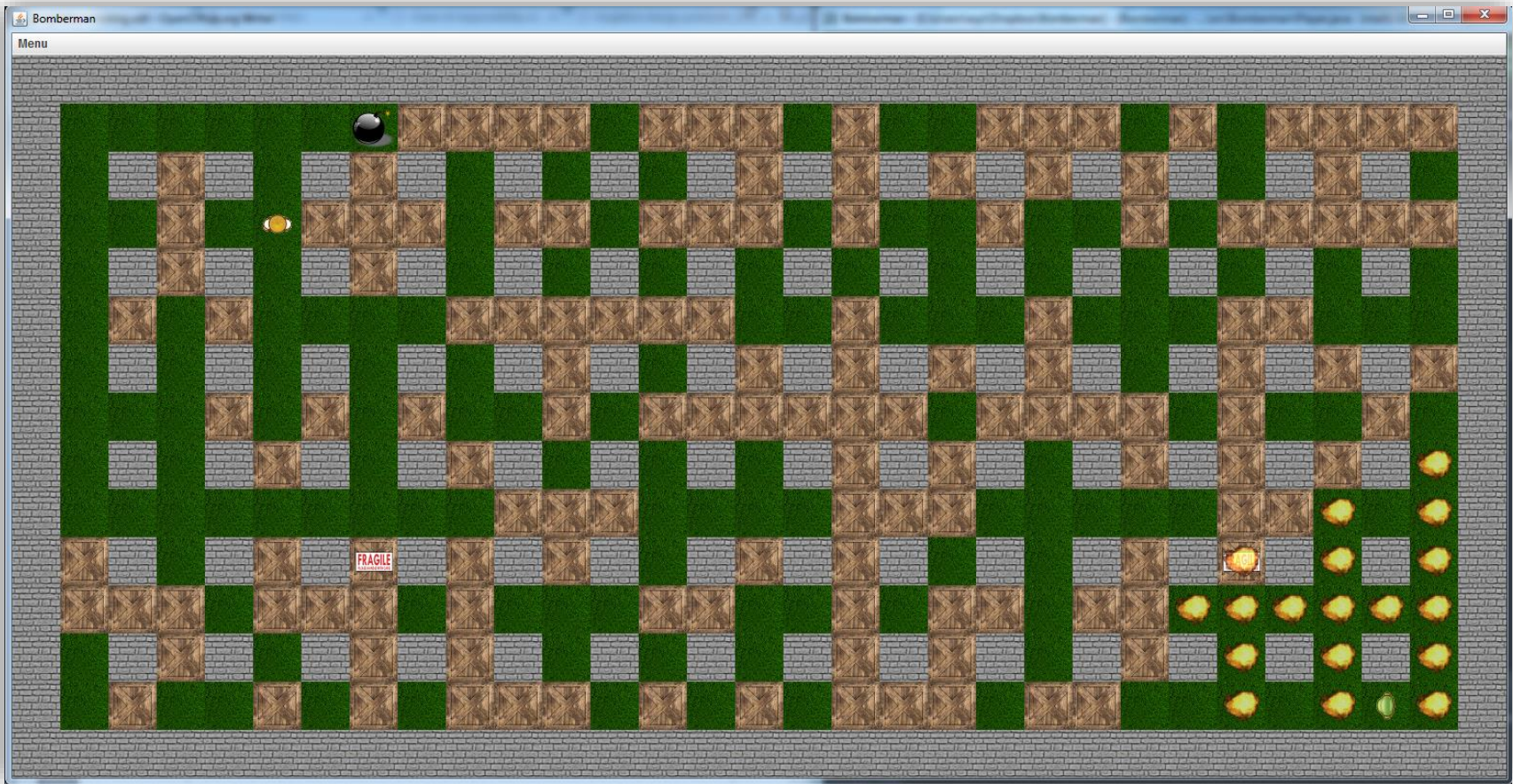
- Chat-system



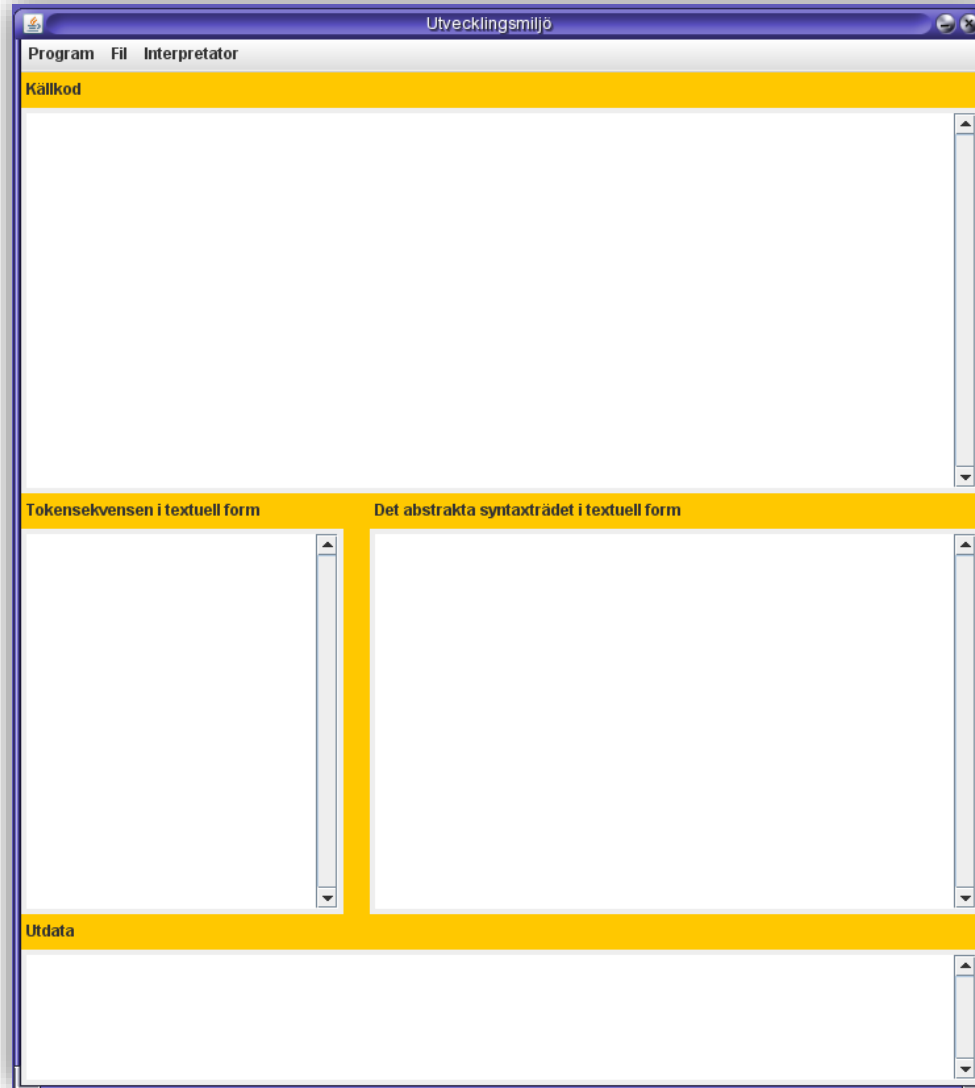
- Spel



- Spel



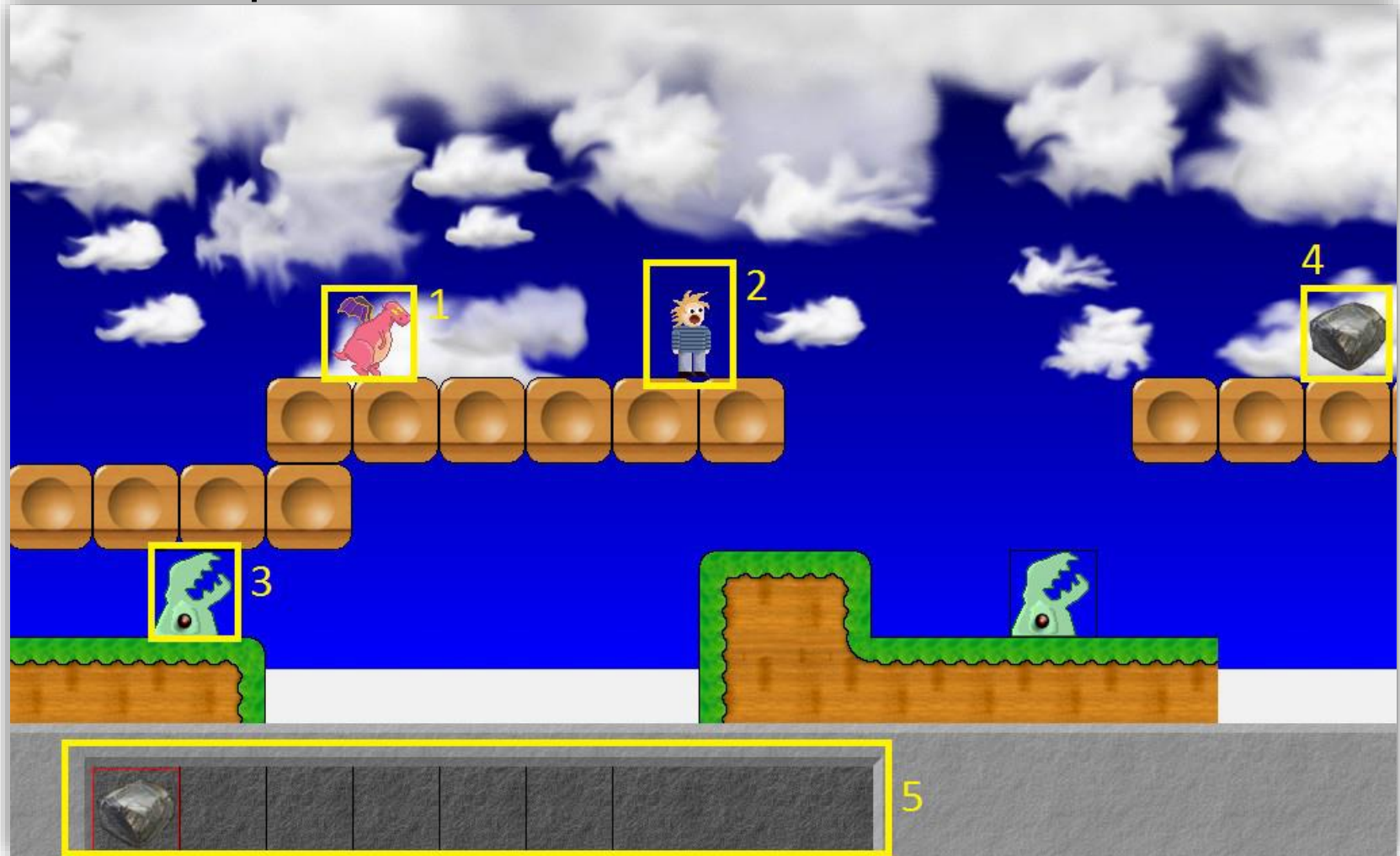
- Utvecklingsmiljö för ett litet programspråk



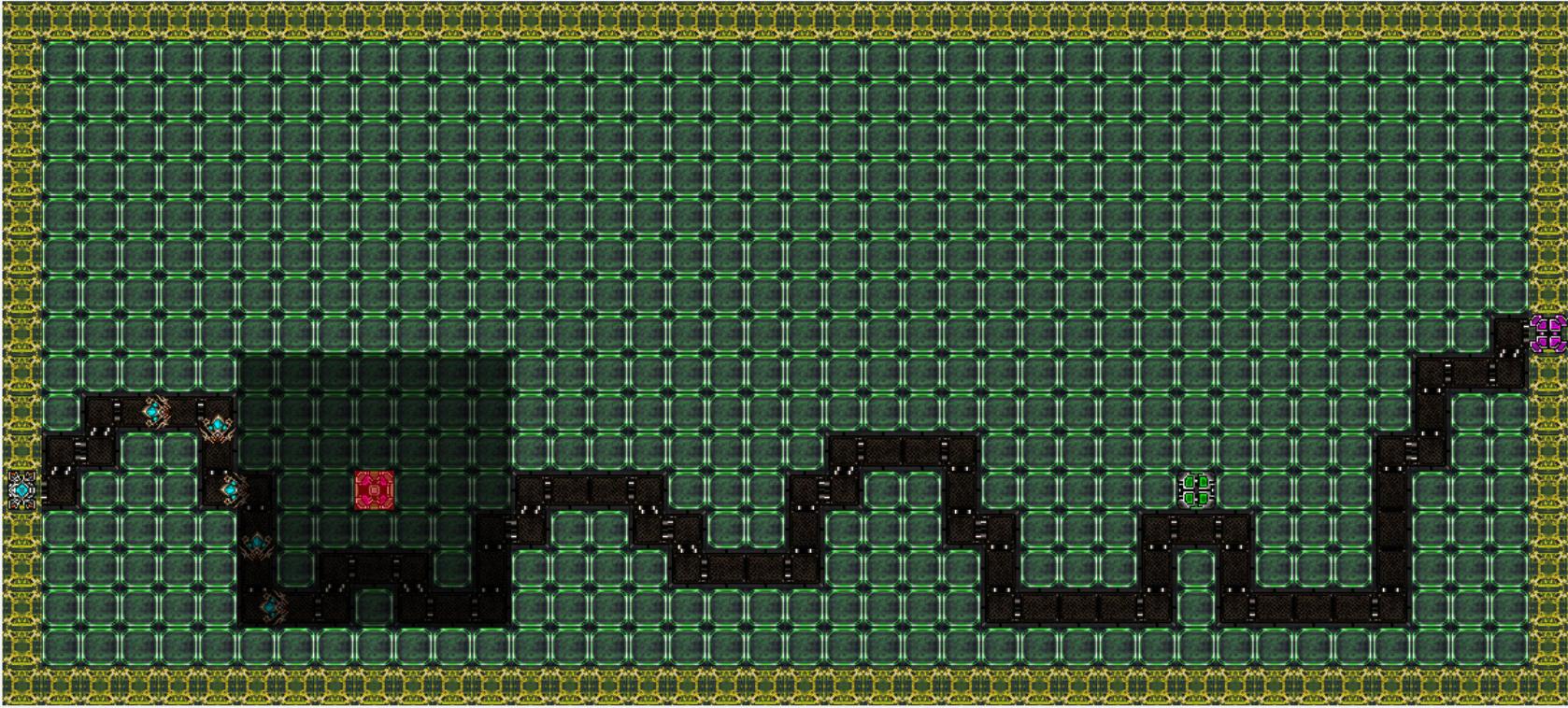
- Bilspel



- Plattformsspel



■ Tower Defense



The screenshot shows a tower defense game window titled "Boom TD". The main area is a grid of green tiles with a path of black tiles. Several towers are placed on the grid, and a few enemies are visible. The interface includes a "Menu" and "Controls" bar at the top left. At the bottom, there are statistics for the current wave and a specific tower.

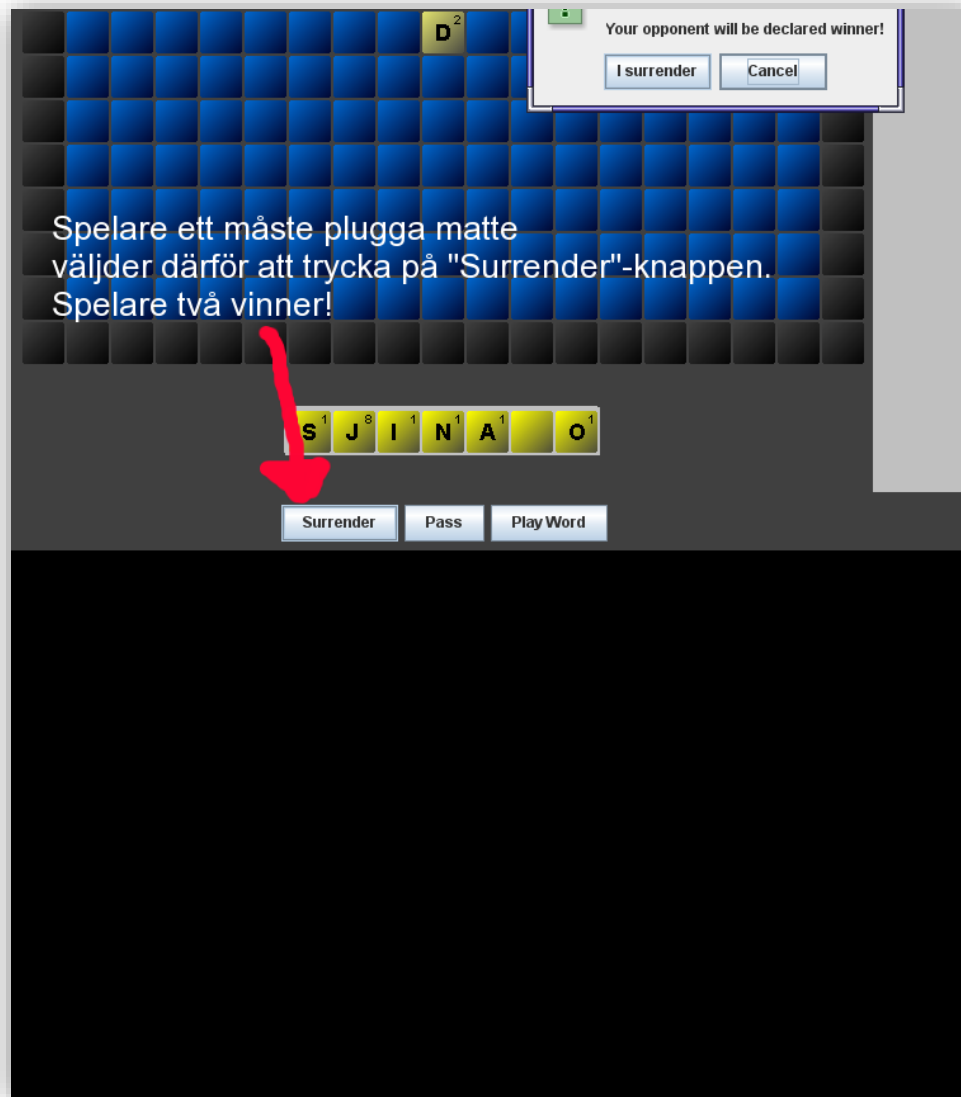
Menu Controls

WAVE 2
6 balanced enemies
408 health points
20 Speed

Tower 2 Level: 2
Attack: 18
Rate of Fire: 90ms
Range: 3 tiles
Upgrade cost: 132 coins
Sell value: 165

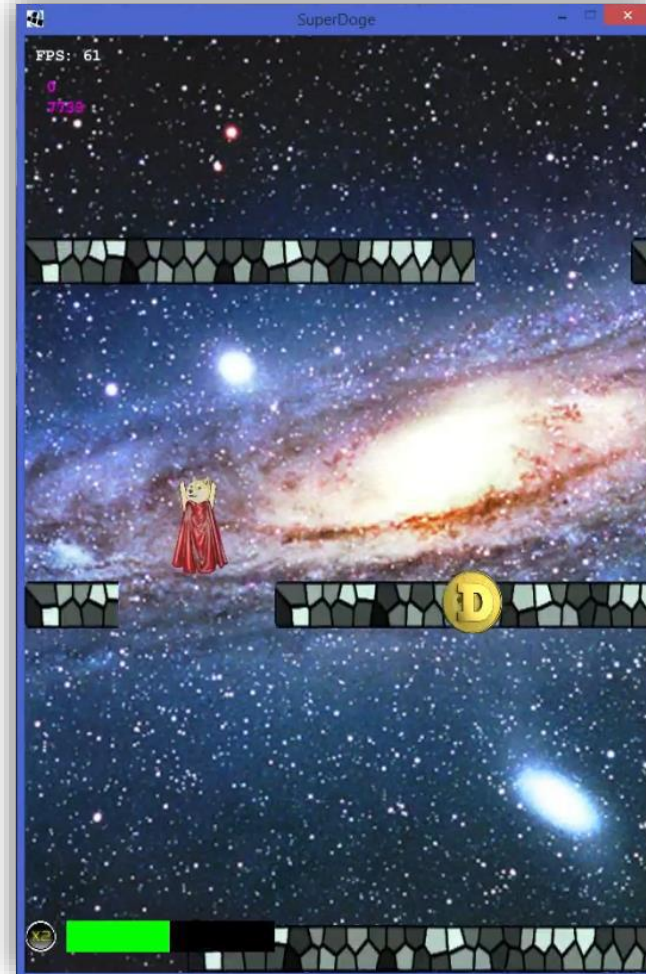
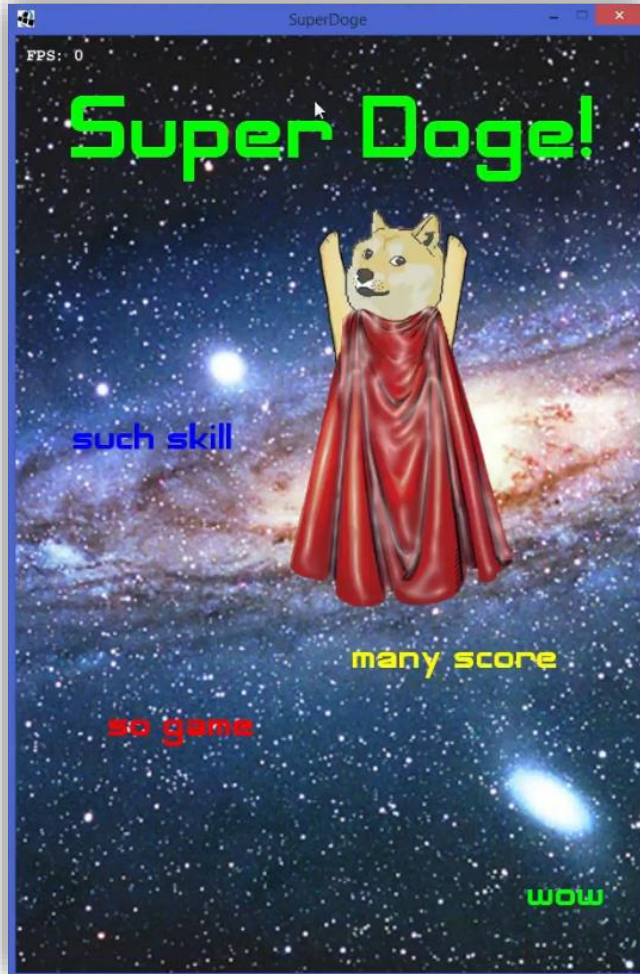
22
20

- Word Wars



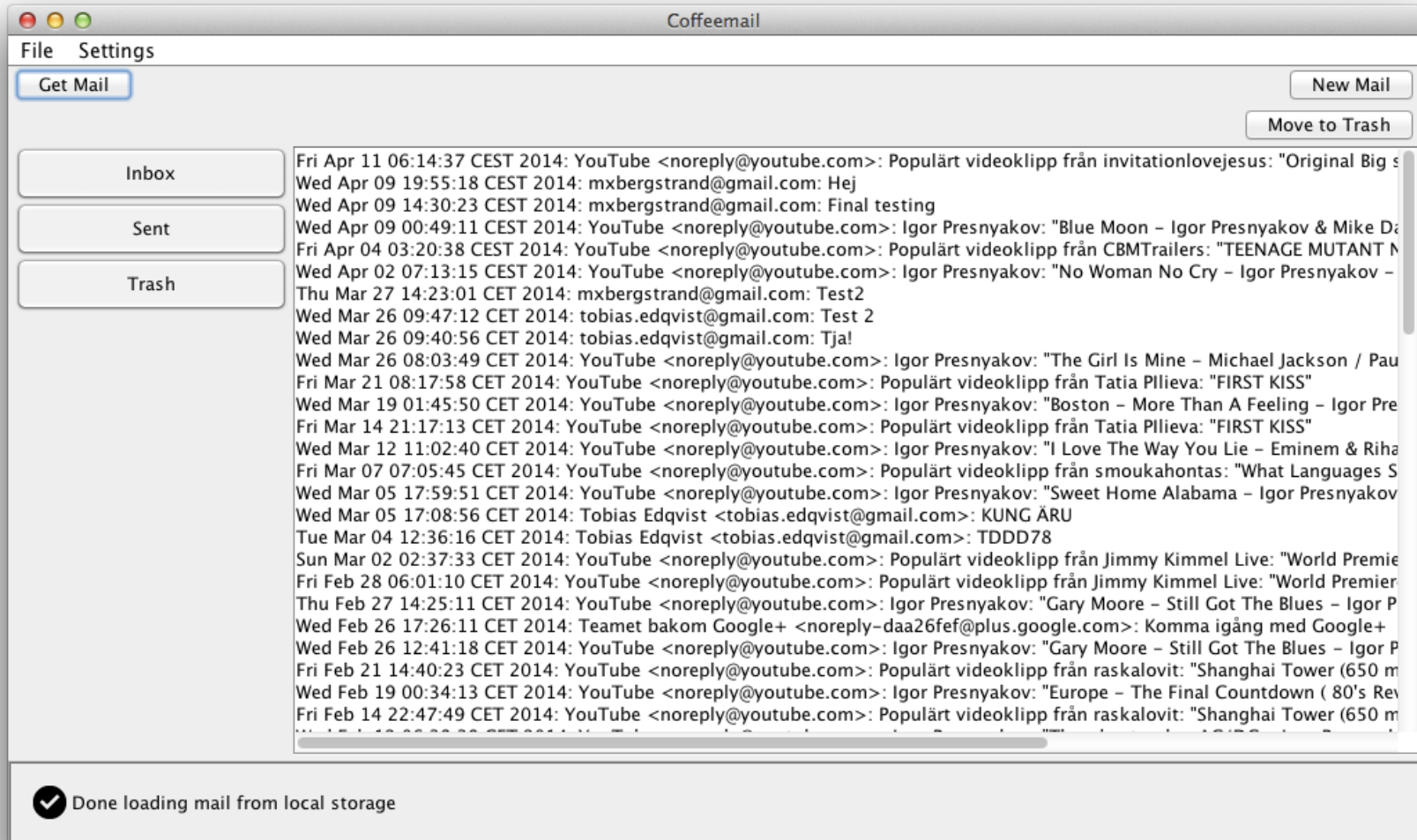
Exempel

- Sidoscrollande spel

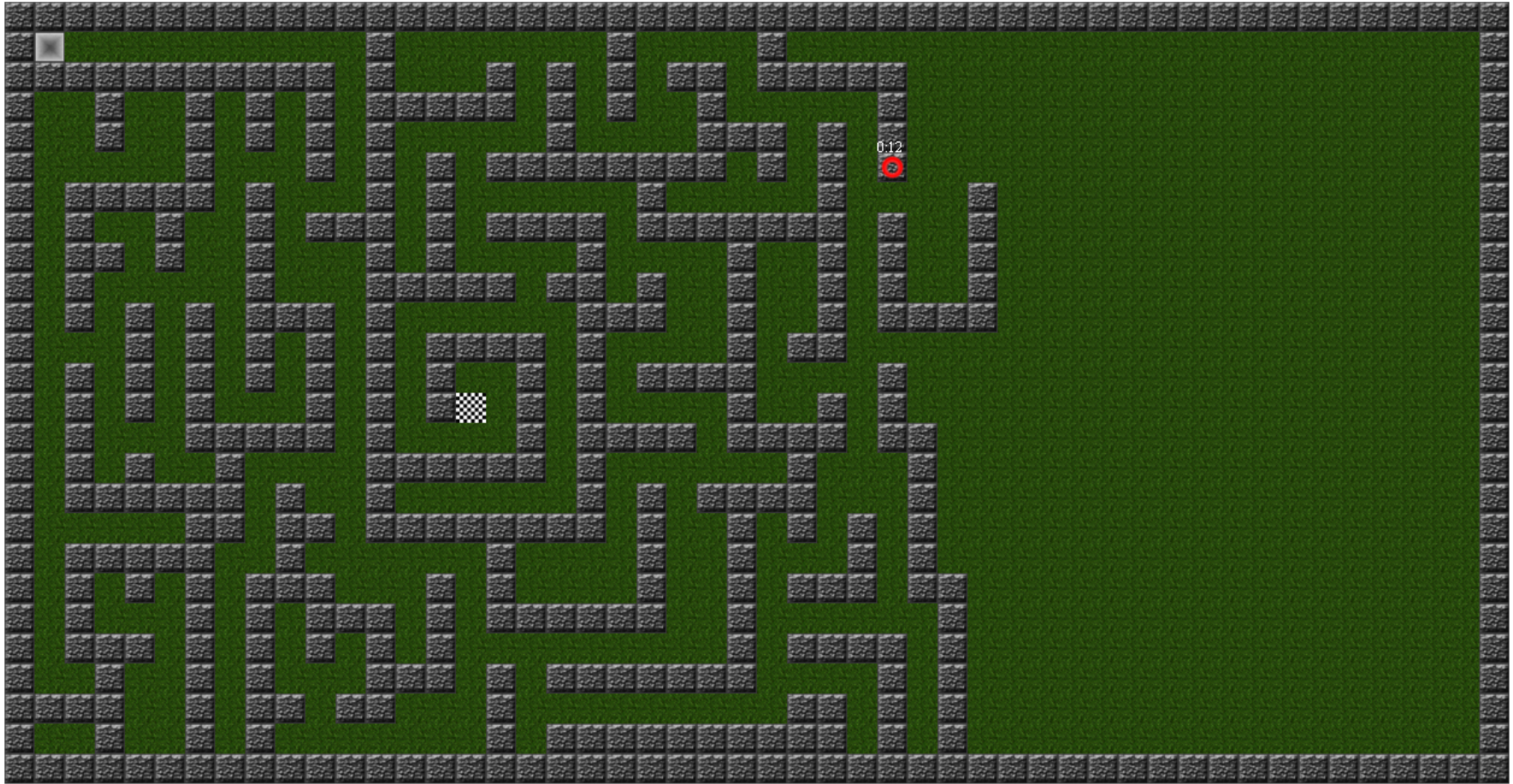


Exempel





Exempel



Exempel

All shows

The screenshot shows a web application interface for TV shows. At the top, there are navigation options: 'File', 'Air date', 'Unwatched', 'Delete show', 'Mark as: Watched Unwatched', 'Update', and 'Add show'. The main content is a table of shows and episodes. The 'Continuum' show is highlighted in blue. A pink box highlights the 'Continuum' row and the 'Minute To Win It' episode row. A green box highlights the details for the 'Minute To Win It' episode, including its synopsis and a poster image. A black arrow points from the text 'All shows' to the 'Continuum' row. A pink arrow points from the text 'Episodes of selected show' to the 'Minute To Win It' episode row. A green arrow points from the text 'Details of selected episode' to the episode details panel.

Show	Air date	Episode
Arrow	2014-04-02	Minute By Minute S03E01
The Big Bang Theory	2014-04-03	Minute Man S03E02
Hannibal	2014-04-04	Minute To Win It S03E03
Hit of Dixie	2014-04-04	A Minute Changes Everything S03E04
Continuum	2014-04-06	30 Minutes To Air S03E05
The Blacklist	2014-04-07	tba S03E06
The Following	2014-04-07	tba S03E07
The Mentalist	2014-04-13	So Do Our Minutes Hasten S03E08
Person of Interest	2014-04-15	tba S03E09
House of Cards (US)	Unknown	tba S03E10
True Detective	Unknown	tba S03E11

Minute To Win It
S03E05
Mark as unwatched
2014-03-30

Kiera has to solve a series of Liber8 connected bank heists and discovers the robbers are controlled by a recently escaped Lucas. Carlos' new and unwanted knowledge about time travel begins to affect his work.

CONTINUUM

Episodes of selected show

Details of selected episode

Exempel



Exempel



Förberedelser

- **Hitta projektpartner**, om du inte vill jobba ensam
 - Som hjälp: Samling **här** under pauserna / efter föreläsningarna
 - Kom överens om projekt och hur ni ska arbeta

- **Anmäl er i WebReg**
 - **U1**: Öppnas snart
 - Grupp U1P1, U1P2
 - **D1 / kogvet**: Öppnas senare
 - Grupp D1P1, D1P2, D1P3

För TDDE30 (D, kogvet)

- Skriv en första **projektbeskrivning** enligt mall på nätet
 - Projekttyp
 - Milstolpar
 - Diskussion av hur ni har tänkt arbeta

- Lämna in till handledare
 - Lämna senast **180306** → får ev. kommentarer till projektstart **180319**
 - Lämna senast **180331** → kan redovisa före sommaren

För TDDD78 (U)

- Projektbeskrivning
 - Välj bland inspirationsprojekten, eller välj helt eget
 - Justera och anpassa
 - Diskutera hur ni har tänkt arbeta
- Inspirationsprojekt → inlämning frivillig
 - Be om kommentarer om ni vill
- Eget projekt → lämna in så snart ni kan
 - Tidigare inlämning → tidigare kommentarer

Projektbeskrivning: Studenternas åsikter 1



- Projektbeskrivningen var annars ett **bra hjälpmedel** med tanke på planeringen som skrevs innan programmerandet började. Jag hade skrivit **bra milstolpar** som jag följde till stor del och som var till stor hjälp.
- Vi har **använt projektbeskrivningen ganska mycket** för att pricka av mot milstolparna hur långt vi hade kommit med projektet. Detta har varit till **stor hjälp för planering av tiden**.
- Det är viktigt att börja i god tid och att **ha en plan för arbetet klar** eftersom att det underlättar arbetet väldigt mycket.
- Vi har lärt oss att man ska lägga ner mycket tid på **tänka ut hur man ska bygga upp projektet** innan man börjar koda. Om man tänkte igenom allt ner till minsta detalj sparar man mycket tid när man sedan ska koda det.
- The **project description [was] great** at the beginning of the project because you had to really think about how you would structure the code. **The best thing was milestones** since it helped a lot later in the project. You always knew what to do next and it helped to split the project into smaller parts.

- Vi har inte haft någon nytta av projektbeskrivningen, eftersom man började med det man tyckte behövdes mest. Hade vi vetat i förväg exakt hur vi skulle göra så hade vi följt den men nu hade vi inte det.

Att genomföra ett projekt

- Mål:
 - Visa upp vad ni kan inom objektorientering och Java
- Medel:
 - Skriv ett eget projekt
 - Välskriven kod av hög kvalitet
 - Använd objektorienteringen och Javas finesser
 - Demonstrera vad ni kan inom kursens mål
 - Tänk på vad man ser när man läser koden
 - Inte *vad* programmet utför, utan *hur*
 - Dokumentera
 - Mer att skriva i projektbeskrivningen – visar hur ni *tänker*, som på en tenta där man inte bara skriver slutliga svaret

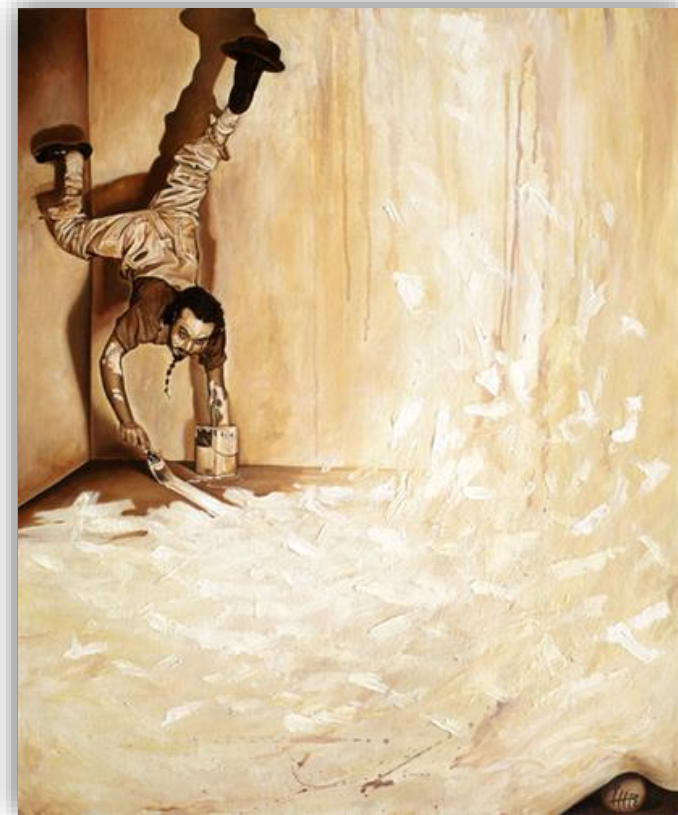
- Det tar **längre tid** än man tror
- Räkna med att allt tar **längre tid** än vad man tror
- Start on the project early. It will take **a lot of time** so you can't only work the last days before deadline, especially not if you aim for a high grade.
- Gör projektet. **Vänta inte**, var helst klar till första redovisning. Undvik vänta till omtenta-p.

**Fixa ändå helst Tetris före projektet,
annars missar ni nödvändig kunskap!**

Det kan också vara en bra idé att **fråga assistenterna om hjälp** om man kör fast, istället för att sitta och **slå huvudet i bordet** tills man tappar motivationen att fortsätta.

■ **Måla inte in er i ett hörn!**

- Ständig uppföljning med handledare
 - Diskutera problem och lösningar
- Be handledaren titta på koden
 - Fånga upp problem tidigt
- Jobba inte enbart hemma!



- Gör projektet **steg för steg**
 - Lägg till lite mer funktionalitet
 - Se till att det fungerar
 - Dokumentera
 - Putsa och förbättra
 - Kontrollera (inspektioner, ...)
 - Checka in resultatet som en fast punkt (versionshantering)
 - Repetera...

Tips till nästa års studenter skulle nog vara just att skriva på **projektbeskrivningen** under tiden som ni programmerar. Det tror jag ökar chansen till ett högre betyg.

Det var inte förrän slutet som vi gjorde kodinspektioner, som till vår fasa visade
1,661 varningar
(ungefär en tredjedel magiska konstanter).
De flesta av dessa är nu fixade, men vi borde absolut kollat på detta under tiden vi skrev programmet istället.

Skriv ner alla **designbeslut** ni tar.
Ni kommer glömma bort dem.

Däremot borde jag ha **skrivit** på resten av projektbeskrivningen
medan jag skrev koden.
Inte minst för att då lättare få med fler av kriterierna för att få ett högre betyg.

- Steg för steg → kan införa *feature freeze* innan slutet
 - Ger tid till **slutlig finputsning** – kvaliteten räknas...
 - Gå genom kodinspektioner
 - Se över *hela* strukturen
 - Uppdatera gammal kod, nu när ni vet bättre
 - Ta bort "dubletter", hitta ineffektiv kod
 - Uppdatera kommentarer och dokumentation
 - Slutför projektbeskrivningen
 - ...

Redovisning, inlämning, deadlines



TDDD78 / U

**Slutet av läsperioden:
Demo 180306, inlämning 180311**

**Fortsätt i tenta-p?
Demo 18031x, inlämning 180318**

**Jobba vidare?
1808xx, 1810xx, 1901xx**

TDDE30 / D, kogvet

**Slutet av läsperioden:
Demo 180503(?), inlämning 180506**

**(Sedan rapportskrivning,
språkgranskning)**

**Jobba vidare?
1808xx, 1810xx, 1901xx**

- Demonstration
 - Visa vad ni gjort
 - Visa att ni förstår vad ni gjort
- Vid inlämning:
 - Följ instruktionerna – annars får ni en retur!

Betygskriterier

- Flera kvalitativa kriterier för betygsättning (3/4/5, G/VG):
 - Objektorientering och Java
 - Använd objektorienterade lösningar
 - Visa att ni har förstått OO-principer
 - Följ rekommendationer från föreläsningar:
Hur man skriver *bra* kod, som gör på *rätt* sätt
 - Läsbarhet och struktur
 - Skriv tydlig, lättläst, rimligt dokumenterad kod
 - Omfattning motsvarande 80 timmar/person
 - Sekundärt kriterium – ändå viktigt
 - Ju mindre projekt, desto mer tid för finputsning → högre kvalitet krävs

Kvalitativa → svåra att mäta, bedömningsfråga
Se websidorna för mer vägledning!

- Även några mer **mätbara** kriterier för specifika betyg
 - Krav på **koden**
 - Paketstruktur, namngivning, felhantering, upprepning, modellering av uppräkningsbara värden, resurshantering, ...
 - Visa **förståelse** i projektbeskrivningen: *Förklara kod, motivera lösningar*
 - n objektorienterade egenskaper – hur används de, vad vinner man på det, ...
 - n designbeslut – hur gjorde ni, vad var alternativen, ...
 - n UML-diagram – använd dem för att förklara koden
 - Reflektera → lär er ännu mer!

Ignorera inte – vi ger komplettering

Se websidorna för detaljer – kommer snart!