



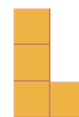
Projektdokumentation

Skriven av Johan Eckerström och Viktor Deleskog

fredag, 2008 maj 23

Kurs: TDDC74, Linköpings Universitet

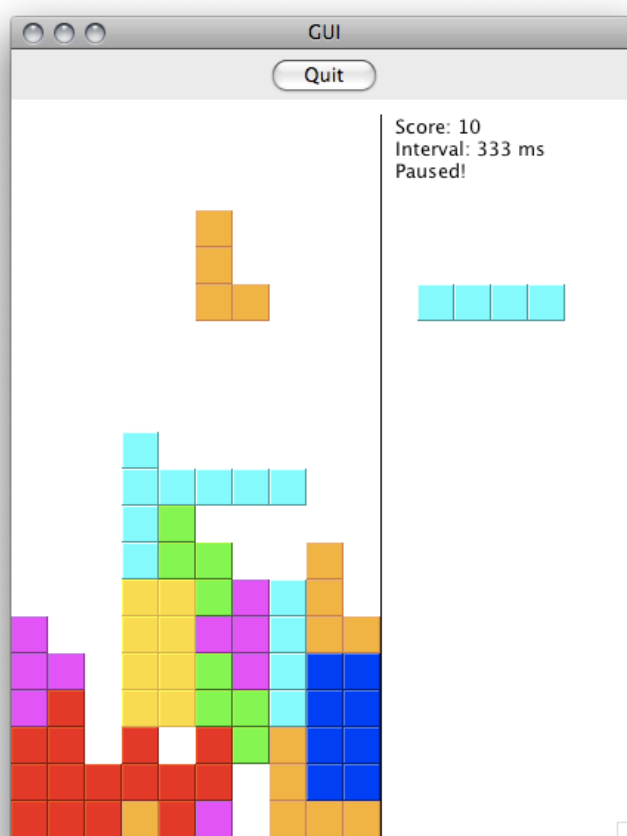
Handledare: Pierre Östlund



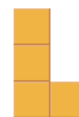
Tetris

Översikt

Detta dokument är en beskrivning över den Scheme-implementation av spelet Tetris, som utvecklades som en del av kursen TDDC74. I dokumentet beskrivs hur spelet fungerar och hur det är implementerat. Namn på filer och procedurer är skrivna med `courier`. Vi har använt en objektorienterad utvecklingsmodell.



Figur 1 - Spelplanen



Användarguide

Starta spelet genom att öppna filen `main.scm` i DrScheme (ver 362). Om du vill köra med standardinställningar kan du starta spelet genom att trycka F5 eller klicka på knappen Run i verktygsfältet. Ett nytt fönster öppnas och spelet startar.

Kontroller

Åtgärd	Tangent
Rotera	Upp-pil
Flytta vänster	Vänster-pil
Flytta höger	Höger-pil
Flytta ner	Nedåt-pil
Slå ner	Mellanslag
Öka/minska hastighet	i / d
Omstart	r
Avsluta	q

Spelet är slut då inga fler fler figurer kan placeras på spelplanen utan kollision.

Implementation

Spelet är uppbyggt av olika klasser, där varje klass representerar en typ av abstraktion i spelet.

main.scm

- Inkluderar alla filer
- Skapar den initiala spelplanen
- Definierar `update` vilken anropas för varje rendering av spelet.
- Tolkar indata från tangentbordet.

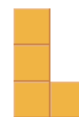
supervisor.scm

Den övergripande klassen vars ansvar det är att hålla ordning på resten.

- Referenser till de olika objekten
- Status för spelet
- Metoder för start, paus, stop och hastighetsförändring.

board.scm

Abstraktion av spelplanen, innehåller klassen `board%`.



- Innehåller metoder för att hantera listan av alla figurer på spelplanen
- Metoder för förflyttning, rotering och placering av figurer.

piece.scm

Definierar klassen `piece%` som är en representation av en figur. En figur består av flera olika block, `block%`.

- Metoder för att hantera listan av block.
- Hantera grundläggande funktioner för rotering och förflyttning.

block.scm

Innehåller klassen `block%`, vilket är en representation av ett block. Block är det mest grundläggande objektet.

- Abstraktion för de koordinater som relaterar till spelplanen
- Grundläggande metoder för block-förflyttning.

utilities.scm

Innehåller små abstraktioner och hjälpfunktioner som inte är klassbundna.

graphics.scm

Grafikfunktioner, bygger på kod som tillhandahölls av kursansvariga.

graphics_engine.scm

Grafikmotorn, hanterar uppritning av den GUI spelaren ser. Bygger på funktioner från `graphics.scm`.

- Definierar den hastighet spelet får
- Uppritning av figurer och text för status

Olika processer

En genomgång av några av de viktigaste processerna i spelet.

Uppdatering av GUI

Bygger på att huvudloopen, `loop`, anropar funktionen `draw` i `graphics_engine.scm`.

- Rensar spelplanen från den tidigare uppritningen
- Ritar upp alla figurer, där varje figur ritas som separata block.
- Skriver ut poäng, status, hastighets-interval
- Ritar designavgränsning och en skugga av den aktiva figuren

Uppdatering av spelets tillstånd

Proceduren `update` anropas vid varje uppritning, men utför bara spelförändringar med ett visst interval, beroende på interval-hastigheten.

- Kontrollerar om den får uppdatera



- Undersöker om den aktiva figuren har nått botten
- Försöker flytta den aktiva figuren ett steg ned
- Om figuren har kolliderat under flytten ned, eller om den nått botten, anropas `clean-up` hos spelplanen.
- Försöker placera en ny figur
- Om den nya figuren kolliderar är det slut.