

eTetris

Grobdesign

- Struktur für einen herab fallenden Baustein
- Sammlung von allen verfügbaren Bausteinen + deren Ausrichtung
- Struktur für das Spielfeld
- Spiele-Timer für automatisches herab fallen der Bausteine
- Auswertung der Benutzereingaben (Menü, Links, Rechts, Rotieren, Runter)
- verschiedene Dialogboxen (Message, Input)
- Struktur für die Highscore-Verwaltung
- Splash Screen

Klassendiagramm

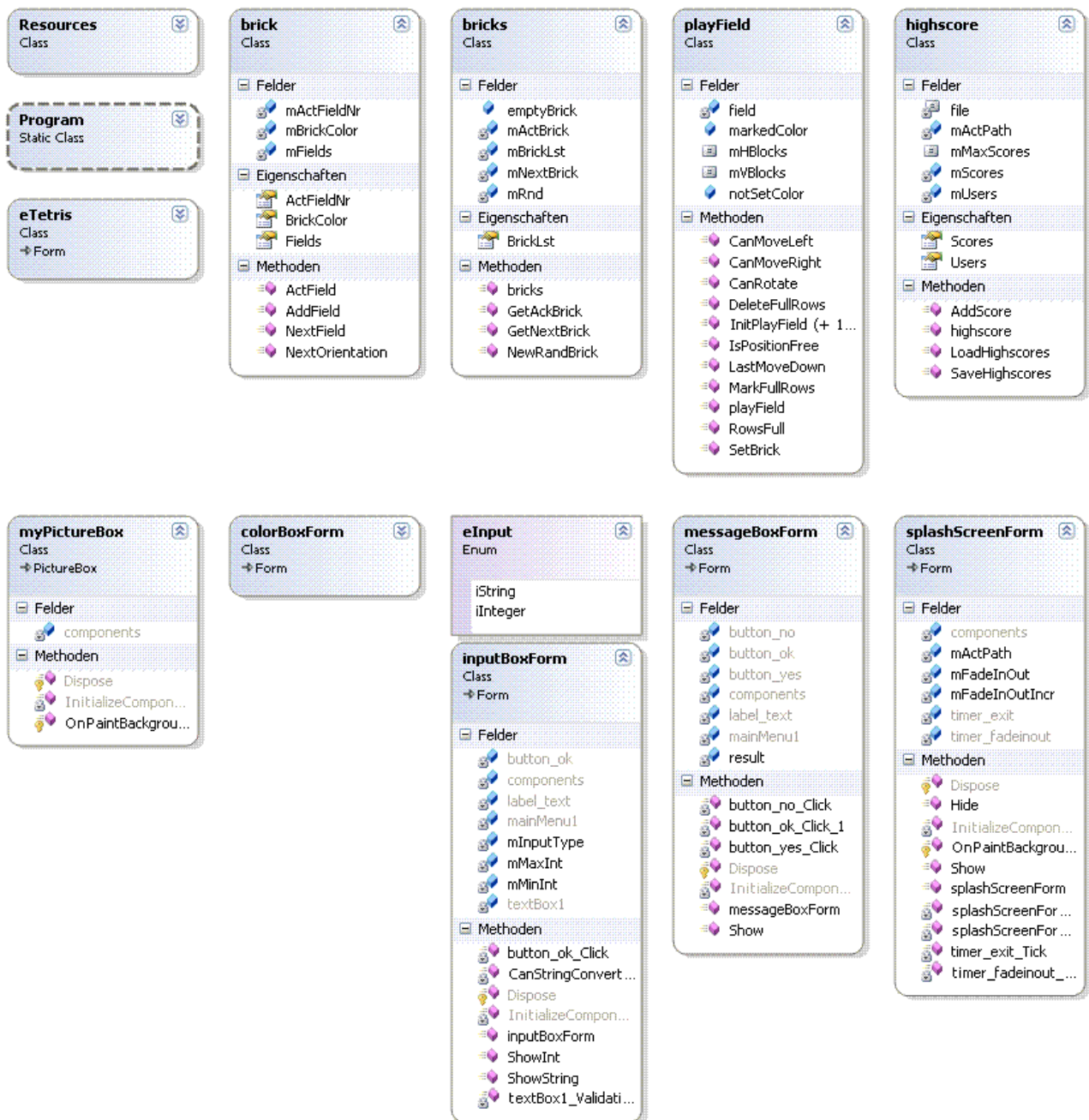


Bild 1 – eTetris Klassendiagramm

Die Klassen wurden in der Anlehnung des Grobdesigns implementiert. Die oberen Klassen (brick, bricks, playField, highscore) sind einzelne Komponenten des Tetris-Spiels, die unteren Klassen (colorBoxForm, inputBoxForm, messageBoxForm, splashScreenForm) stellen die Interaktion mit dem Spieler zur Verfügung. Der generelle Spielablauf wurde in der Haupt-Forms-Klasse (eTetris) implementiert.

Funktionsbeschreibung

Für die Einstellungen eines Spiels steht ein Menü zur Verfügung:

- Game
 - New ... neues Spiel beginnen
 - Pause ... Spiel pausieren
 - Quit ... Spiel / Anwendung beenden
- Level ... Levelauswahl
 - 1
 - ...
 - 10
- Fill ... zufälliges Befüllen des Spielfelds (Angabe der Reihen)
- Score ... Anzeige des Highscores
- Colors ... Einstellung der Bausteinfarben
- About

Weitere Eingabemöglichkeiten stellen die Buttons und die Tasten für die Spielsteuerung (links, rechts, rotieren, nach unten) dar (Bild 2). Die Buttons wurden so gewählt, damit die Spielsteuerung auch per Touchscreen erfolgen kann, für ein schnelles Spiel am Computer sind dennoch die Tasten am besten geeignet.

Das Fenster ist in drei Bereiche unterteilt: Spielfeld, Informationen, Eingabe-Buttons. Der Spielfeld-Bereich wird durch eine Paint-Methode befüllt. Der Informations-Bereich enthält die Darstellung des nächsten Bausteins, das aktuelle Level und die aktuell erreichten Spielpunkte.

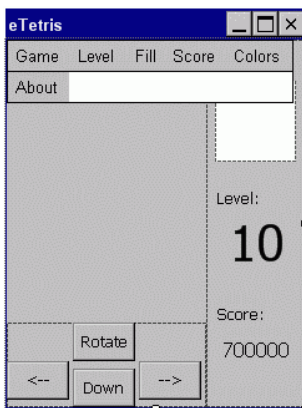


Bild 2 – eTetris Form

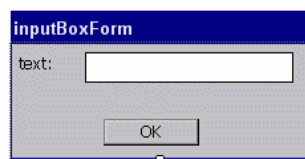


Bild 3 – Input Box

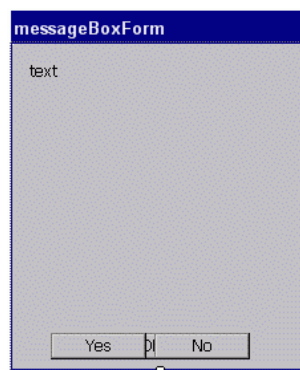


Bild 4 – Message Box

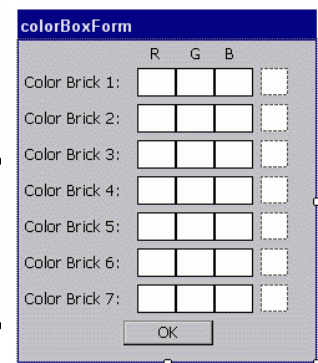


Bild 5 – Color Box

Nach dem Start eines neuen Spiels, wird der Timer aktiviert, welcher nach einem Zeitintervall die Position des Bausteins um Eins nach unten setzt. In der Zwischenzeit kann das Spiel durch Benutzereingaben beeinflusst werden. Bei jeder Veränderung der Position bzw. Ausrichtung des Bausteins wird die neue Position überprüft (Einhaltung der Grenzen, unterste Position erreicht, Rotieren erlaubt, ...). Falls die Überprüfung erfolgreich, so wird die Aktion durchgeführt.

Damit die Bausteine, welche noch nicht fest unten eingetragen sind, weiterhin im Spielfeld nicht fest eingetragen werden, werden sie nur eingetragen wenn sie fest erhalten bleiben sollen. Dennoch liefert die SetBrick-Methode das Feld mit dem Baustein zurück.

eintragen, welcher in die Highscore-Liste, falls gut genug, eingetragen wird. Weiters wird er gefragt ob ein neues Spiel begonnen werden soll.

Das aktuell laufende Spiel kann durch „Pause“ angehalten und später mittels „Resume“ fortgesetzt werden. Durch Auswahl eines Levels im Menü, wird das gewünschte Level eingestellt. Der Fill-Button ermöglicht das zufällige Befüllen von Reihen, somit wird das Spiel interessanter. Der Score-Button öffnet eine neue Oberfläche, welche die aktuelle Highscore beinhaltet. Um die Farben der Bausteine eigenen Wünschen anzupassen, wird mit dem Color-Button eine Oberfläche für die Farbeinstellung geöffnet. „About“ zeigt generelle Informationen über eTetris in einer eigenen Oberfläche an.

Sequenzdiagramm

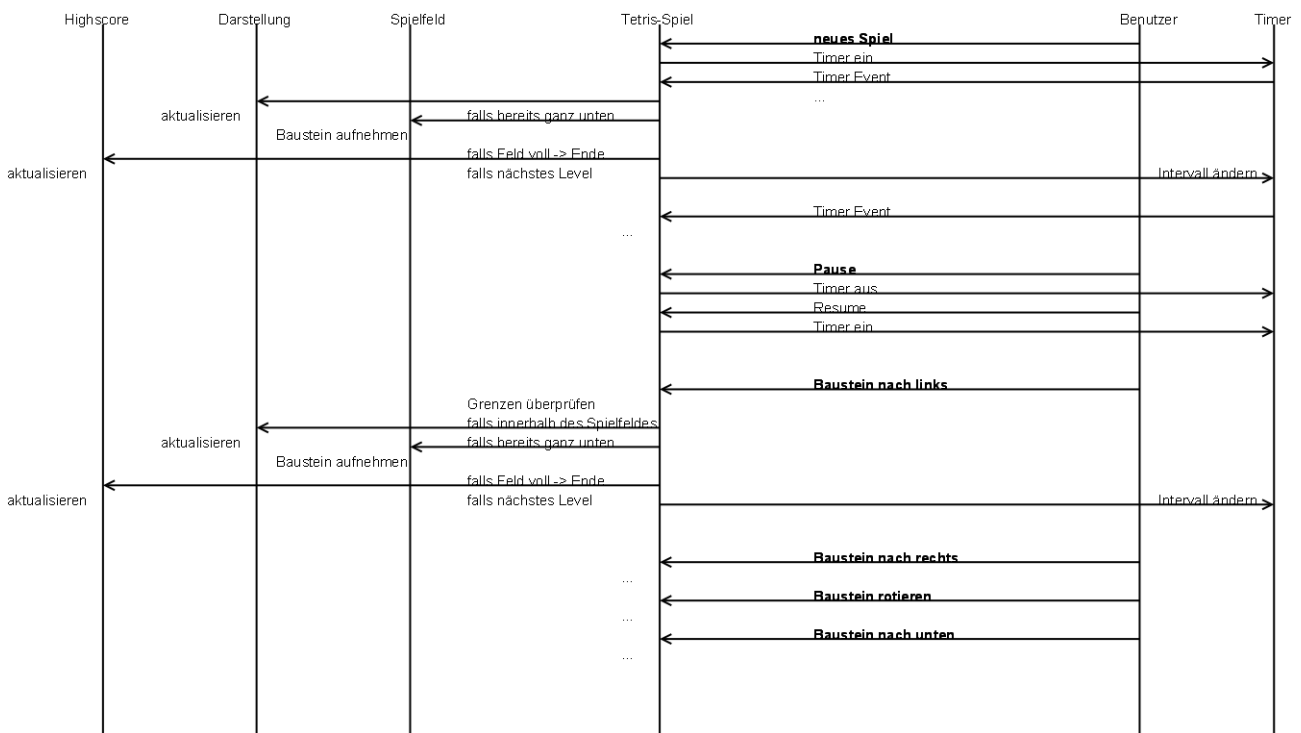


Bild 6 – generelle Spielablauf

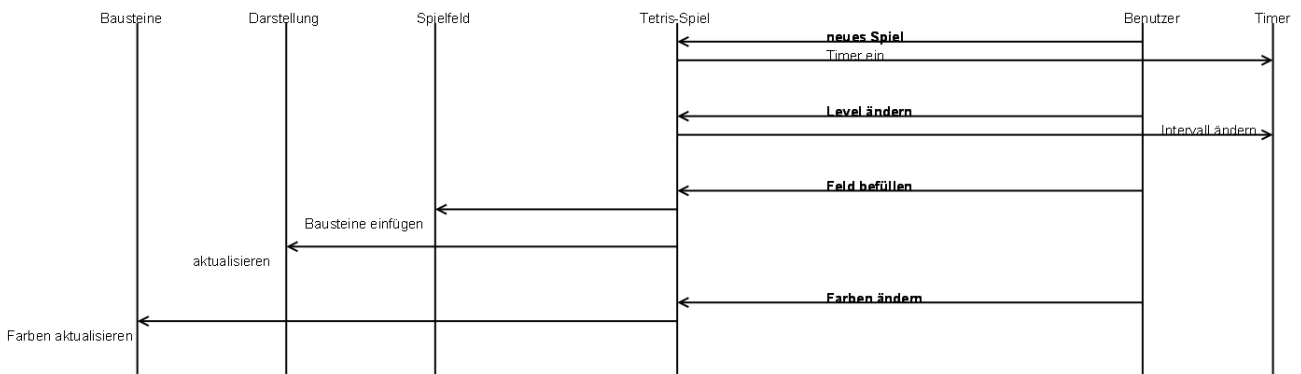


Bild 7 – Einstellungen des Spiels

Feindesign

Klasse brick

- Speicherung der Baustein-Strukturen (sämtliche Orientierungen)
- Speicherung der Baustein-Farbe

Klasse bricks

- Anlegen aller verfügbaren Bausteine (Liste von Brick's)
- Auswahl zufälliger Bausteine (aktuelle und nächsten) durch Zufallszahlen-Generator

Klasse playField

- Speicherung einer Matrix für das Spielfeld (Farbe pro Feld)
- Spielfeld initialisieren (Farbe notSetColor)
- Spielfeld mit befüllten Reihen initialisieren
- Abfrage ob bestimmte Reihe bereits ein Baustein eingetragen ist
- Eintragen eines Bausteins (falls bereits unten wird Baustein eingetragen, andernfalls nur in ein temporäres Feld, welches zur Darstellung zurück geliefert wird)
- Markieren voller Reihen (Farbe markedColor)
- Löschen von markierten Reihen
- Abfrage ob Position für Baustein noch frei ist (0 → Position nicht frei, >0 → Anzahl Steine des Bausteins welches eingefügt werden soll)
- Abfrage ob Baustein nach rechts/links bewegt werden kann
- Abfrage ob Baustein rotiert werden kann
- Abfrage ob Baustein bereits unten angekommen ist

Klasse highscore

- Einlesen vorhandener Punkteliste
- Eintragen neuer Punkte mit Spieler-Namen
- Speichern der Punkteliste (Textformat)

Klasse colorBoxForm

- statische Methode, welche die Farbauswahl-Form initialisiert und darstellt
- Überprüfung der Eingabewerte (Werte für Rot/Grün/Blau zwischen 0-255)
- Aktualisierung des Farbpanels, falls Farbe geändert wurde (zur Vorschau)
- nach Schließen der Form → Rückgabe der eingestellten Farben (per Referenz-Übergebeparameter)

Klasse inputBoxForm

- statische Methode, welche die Eingabe-Form initialisiert und darstellt
- Überprüfung der Eingabewerte (string bzw. integer mit min.-max. Wert)
- nach Schließen der Form → Rückgabe des Eingabewertes (per Return-Wert)

Klasse messageBoxForm

- statische Methode, welche die Nachrichten-Form initialisiert und darstellt
- nach Schließen der Form → Rückgabe welcher Button zum Schließen verwendet wurde

Klasse splashScreenForm

- statische Methode, welche den Splash-Screen darstellt
- Darstellung per Timer gesteuert (fad in, fad out)
- falls Eingabe → Splash-Screen abbrechen und schließen

Klasse eTetris

- eigentliche Oberfläche des Spiels
- Paint-Methode für das Spielfeld („double buffered“ damit flickerfrei)
- Paint-Methode für den nächsten Baustein
- Resize-Methode für das Anpassen des Spielfeldes nach Änderung der Größe
- Timer-Tick-Event für automatisches Herabfallen der Bausteine
- Methode für volle-Reihen-Überprüfung
- Methode für Generierung des nächsten Bausteins
- Methode zur Überprüfung ob Spiel läuft
- Methode zur Änderung des aktuellen Spiel-Levels
- Behandlung der Button-Events
- Behandlung der Tasten-Events

Spieldarstellungen – Windows CE Derivat am PDA

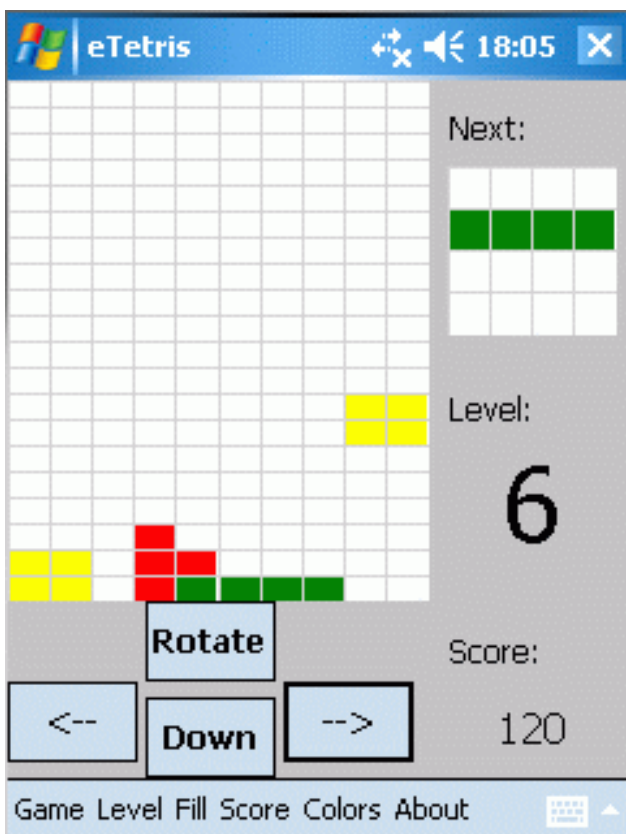


Bild 8 – laufendes Spiel

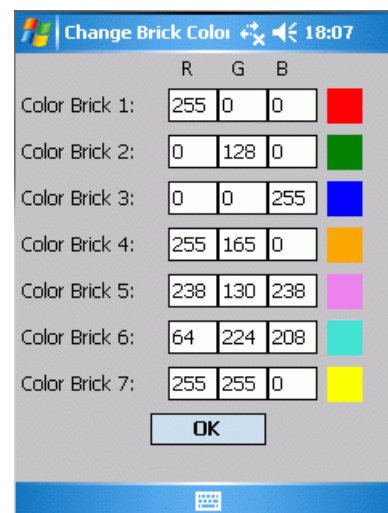


Bild 9 - Farbauswahl

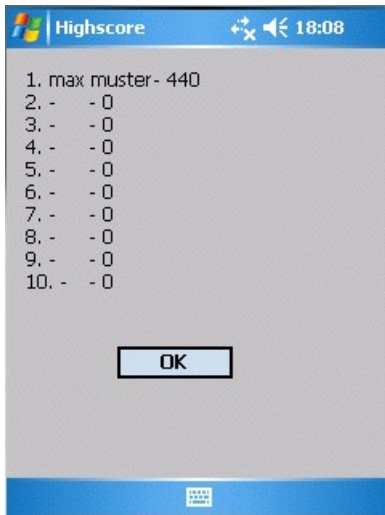


Bild 10 – Highscore Liste

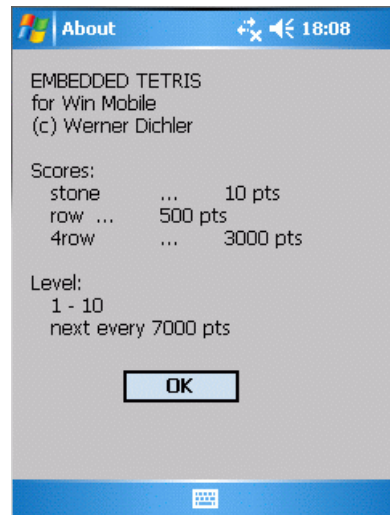


Bild 11 – About Box

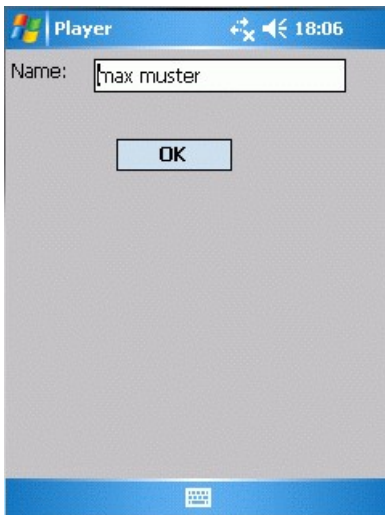


Bild 12 - Spielende, Eingabe des Namens

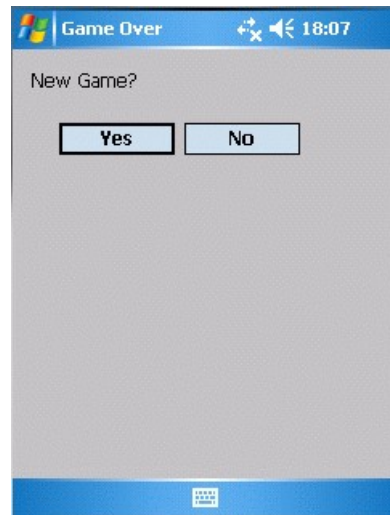


Bild 13 - Spielende, neues Spiel starten?

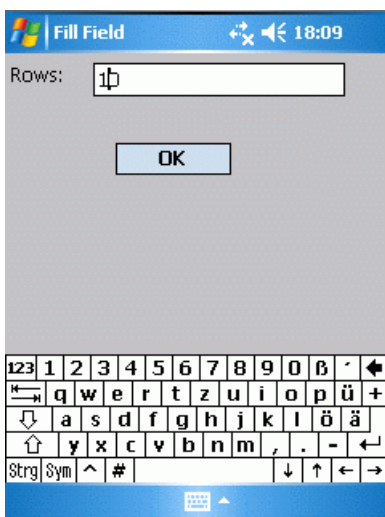


Bild 14 – Eingabe Felder befüllen

Spieldarstellungen – Windows XP

Damit das Spiel auch an einem Desktop-PC läuft wird das normale .NET Framework benötigt und weiters muss die Pfadermittlung deaktiviert werden (bei der Initialisierung von eTetris).

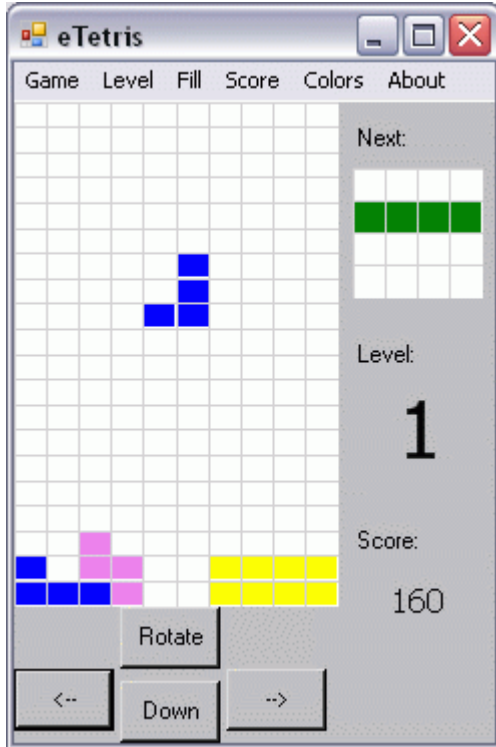


Bild 15 – laufendes Spiel

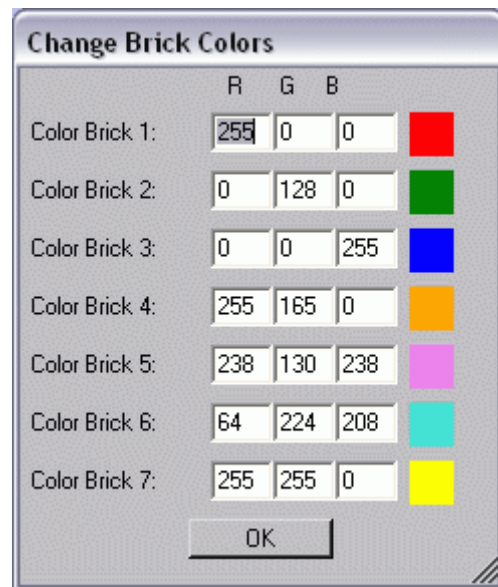


Bild 16 - Farbauswahl

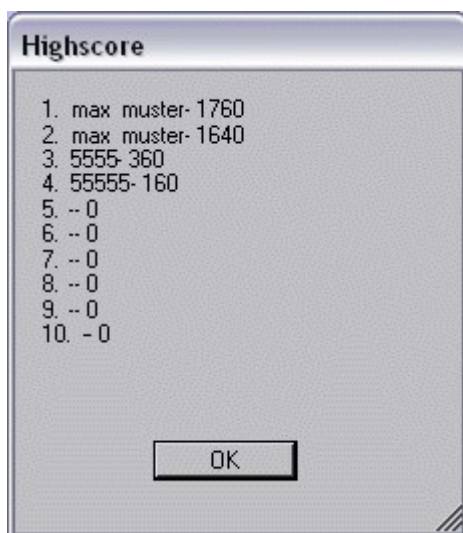


Bild 17 – Highscore Liste

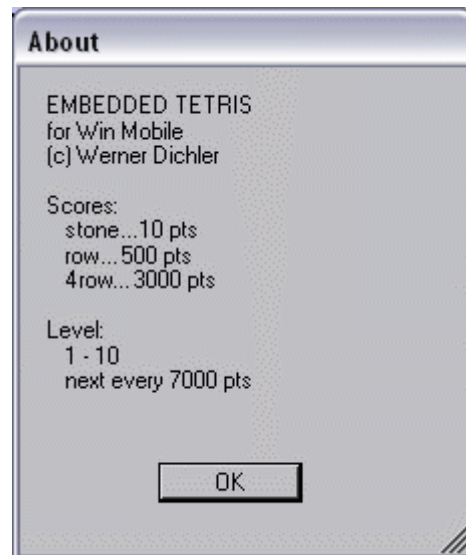


Bild 18 – About Box

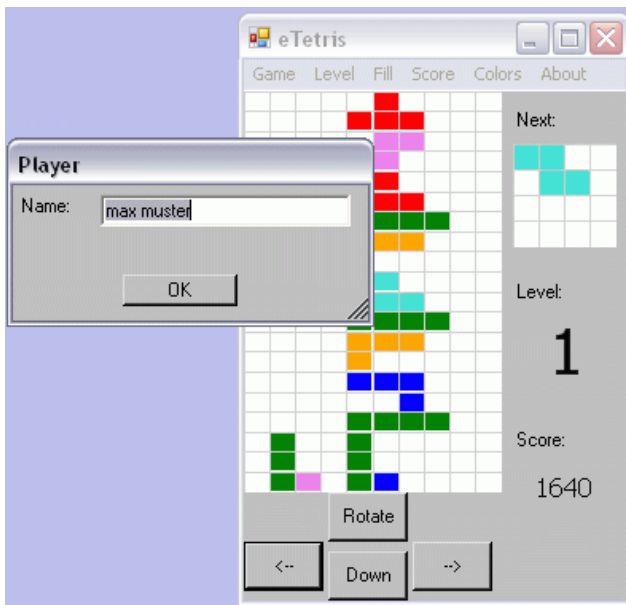


Bild 19 – Spielende, Eingabe des Namens

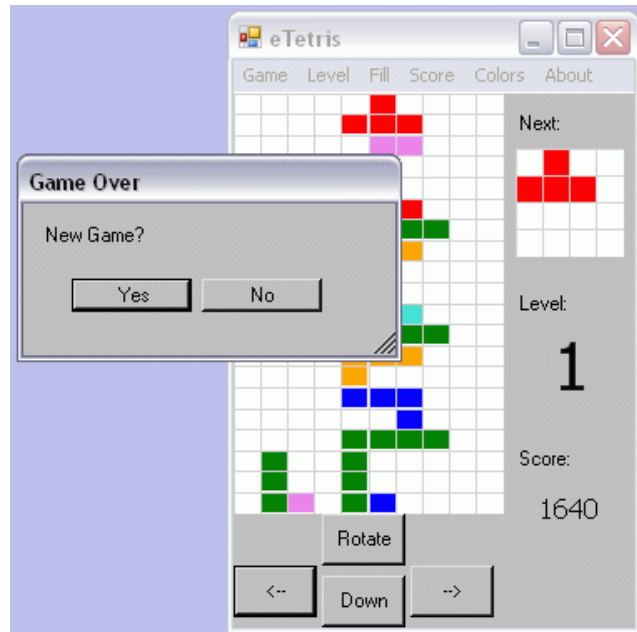


Bild 20 – Spielende, neues Spiel starten?