

Erläuterungen:

Auf einem Spielfeld (z. B. 4 Zeilen à 8 Spalten) erscheint jeweils am Rand (z. B. rechts) in einer zufällig ermittelten Zeile eine Figur und wandert in jedem Zyklus solange um ein Feld in Richtung des gegenüberliegenden Randes (z. B. nach links), bis es am Rand angekommen ist oder an einem Feld ankommt, dessen nächstes Nachbarfeld bereits besetzt ist.

Ist die Figur dieses Nachbarfeldes gleichartig, verschwinden beide. Das Spielende ist erreicht, wenn in einer Zeile alle Felder belegt sind.

Der Spieler kann eingreifen, indem er durch Anklicken einer Zelle markiert, dass die wandernde Figur ihren Weg auf der entsprechend angeklickten Zeile fortsetzt.

Vorüberlegungen:

- Das Spielfeld könnte mittels StringGrid sg_feld (anfangs in Testphase 4x4) realisiert werden.
- Der nächste Zyklus sollte in der Testphase mittels ButtonClick angestoßen werden, später durch Implementation eines Timers (Delphi-Hilfe dazu lesen)
- Im Verlauf des Spieles werden einige Ereignisse eintreten, zu denen jeweils geeignete Methoden entworfen werden müssen; z. B. sind dies:
 - Ereignisse 'Spielende erreicht', 'Neue Figur erforderlich'
 - Methoden 'Nächster Zyklus (Schritt)', 'Figur soll wandern', ...
- Die Anzahl verschiedenartiger Figuren, die Größe des Spielfeldes sollten später benutzerdefiniert gestaltet werden; anfangs in der Testphase globale Konstanten verwenden.
- Die Figuren (alle?; nur die aktiv wandernde?) bzw. genauer ihre Eigenschaften werden in einem Verbund (RECORD) gespeichert.

Beispiel für Deklaration und Aufruf:

```

VAR Figur : RECORD
    Art           : Char;
    Spalte, Zeile : Byte;
    wandert       : Boolean;
END;

begin
    Figur.Art := 'A';
    // oder bequemer:
    WITH Figur DO
        begin
            Art := 'A';
            Spalte := 4;
            Zeile := Random (5) + 1;
            // ... u.s.w.
        end; // von WITH
    end;

```

Aufgaben:

- a) Entwerfen Sie ein Formular mit einem StringGrid, einem NeuesSpiel-, Abbruch- und einem Ende-Button nebst Methoden. Codieren Sie dazu eine separate Prozedur InitFeld, die das StringGrid dimensioniert (Anzahl Zeilen und Spalten) und alle Zellen leert. Sie wird beim Abbrechen und für ein neues Spiel aufgerufen. Für die weitere Testphase implementieren Sie einen Button 'Neue Figur' und die zugehörige Methode, mit der eine neue Figur entworfen und angezeigt wird. Die Art und die Zeile sollen hier zufällig gewählt werden. Diese Methode muss anfangs ermitteln, ob überhaupt noch eine freie Zelle am Anfangsrand existiert, sonst eine Info ausgegeben.
- b) Formulieren Sie AUF PAPIER in Form eines Flussdiagrammes alle notwendigen Abfragen und Anweisungen für die Methode, die zum Ereignis 'Figur soll wandern' und 'Nächster Zyklus' gehören.
Hinweis: Die Methode 'Figur soll wandern' kennt drei verschiedene, abschließend abzuarbeitende Aktionen (welche?).
- c) Legen Sie die Bewegungsrichtung der Figuren fest. Platzieren Sie für die Testphase einen weiteren Button 'Nächster Schritt' und eine Methode, mit der eine Figur von einem zum anderen Bildrand wandert. Setzen Sie dazu ihre Erkenntnisse aus dem obigen Flussdiagramm um. Testen Sie diese Methode ausgiebig für jeweils eine Figur.
- d) Erweitern Sie diese Click-Methode um die notwendigen Anweisungen für das Ereignis 'Neue Figur erforderlich'. Testen Sie diese erweiterte Methode ausführlich.
Hinweis: Die ergänzende Methode dazu haben Sie bereits (für den Testbutton 'Neue Figur') entworfen.
- e) Erweitern Sie diese Click-Methode nun noch um die Überprüfung, ob das Ereignis 'Spielende erreicht' eingetreten ist. Dann soll der Button inaktiv (aber sichtbar) werden.
Hinweis: Die zugehörige Funktion haben Sie bereits anfangs entworfen und in der Methode zur Erzeugung einer neuen Figur benutzt.
- f) Vervollständigen Sie nun den Algorithmus um den oben beschriebenen Eingriff des Spielenden.
Implementieren Sie nun auch den Timer, der jeden neuen Zyklus anstößt, und ein Editfeld, das die Zyklen zählt.
Benennen Sie die Click-Methode um in die Prozedur Zyklus, die von dieser Click-Methode und jetzt auch von der Timermethode aufgerufen wird.
Testen Sie ausgiebig den korrekten Ablauf der Zyklen. Gibt es 'leere Zyklen', in denen nichts geschieht?
- g) Fertigen Sie nun eine Finalversion MiniTetris2, in der alle für die Testphase implementierten Objekte entnommen sind, und in der die Spielfeldgröße und Figurenart-Anzahl vom Benutzer bestimmt werden kann. Auch eine kurze Info über das Spiel, vielleicht auch eine TopTen-Liste der größten erreichten Zyklenanzahlen wäre denkbar.