

Spielidee

Snake ist ein klassisches Spiel, das sich mit minimalen Grafiken darstellen lässt. Das Spiel passt daher zu den eingeschränkten 24x48 Display.

Der Spielverlauf ist simpel und einfach zu verstehen. Die Spielelemente sind als Punkte auf dem Bild dargestellt.

Der Spieler steuert eine „Schlange“, die nach „Nahrung“ sucht. Wenn die Schlange die Nahrung berührt, wird die Schlange länger. Die Bewegung der Schlange wird mit den Pfeil-Tasten (up, down, left, right) gesteuert.

Das Spiel endet, wenn die Schlange die Wände oder ihren eigenen Körper berührt. Eine typische Spielrunde von Snake variiert nach dem gewählten Schwierigkeitsgrad.

Der Schwierigkeitsgrad des Spiels lässt sich konfigurieren, indem man die Geschwindigkeit der Bewegung von der Schlange auf dem Bild verändert. Um die Schlange auf hohen Geschwindigkeiten präzise steuern zu können, braucht der Spieler eine gute Hand-Augen-Koordination.

Aus diesen Gründen haben wir entschieden, Snake zu implementieren.

Spielbeschreibung

Bei unserer Implementation von Snake gilt folgendes:

- Die Grenzen sind mit roten Punkten markiert.
- Der Hintergrund (valide Punkte für Bewegung) sind mit grauen Punkten markiert.
- Die Schlange wird mit grünen Punkten dargestellt.
- Die Nahrung ist ein gelber Punkt.

Am Beginn des Spiels kann der Spieler/in seinen gewünschten Schwierigkeitsgrad auswählen:

- Easy
- Medium
- Hard
- Hardcore

Nach der Wahl des Schwierigkeitsgrads wird ein Countdown auf dem Bild gezeigt.

Nach dem Countdown beginnt sich die Schlange automatisch zu bewegen.

Die Richtung der Bewegung muss der Spieler mit den Pfeil-Tasten steuern.

Wenn die Schlange die Nahrung berührt, verschwindet die Nahrung und taucht wieder auf einen zufälligen validen Punkt auf.

Wenn das Spiel endet, ein Game Over Pop-Up wird auf dem Bildschirm gezeigt, gefolgt von einer High-Score-Tabelle.

Die High-Score Tabelle zeigt die High-Scores für den gewählten Schwierigkeitsgrads.

Der Spieler/in wird dann aufgefordert, entweder das Spiel neu zu starten oder das Programm zu beenden.

Kurze Beschreibung des Programmablaufs

SnakeMain.java ist die **main** Klasse des Programms.

InternalLedGameThread.run() wird erstens aufgerufen, gefolgt von der Erstellung eines neuen Objektes von der **GameBoard** Klasse.

Der Konstruktor von **GameBoard** ruft danach die Methoden **initializeField()** und **initializeGame()** sowie erstellt ein neues Objekt von **Timer**.

initializeGame() ist für folgendes zuständig:

- die Darstellung der Schwierigkeitsgradwahl
- die Initialisierung der Spielobjekte [durch Aufruf von **initializeGameObjects()**]
- den Countdown

initializeField() erstellt das Spielfeld.

Snake.java beinhaltet den Programmcode für die automatische und spielergesteuerte Bewegung der Schlange.

Die Schlange wird durch einen ArrayList implementiert. Ein Tile wird immer am vorne hinzugefügt und hinten wird eines entfernt.

FieldTile ist eine abstrakte Klasse mit den Unterklassen **BackgroundTile**, **BorderTile** und **EatTile**.

Jeder Punkt auf dem Bild wird einem enum von CollisionType zugeordnet. Die enums sind:

- | | |
|-----------------------|-------------------------------|
| - death (rote Punkte) | - nothing (graue Punkte) |
| - food (gelbe Punkte) | - countdown (schwarze Punkte) |

Die Methode **calculateNextTile()** in **GameBoard.java** ist für die Moment-zu-Moment-Gameplay zuständig. Sie bestimmt, welche Aktionen durchzuführen als Reaktion auf die Handlung des Spielers.

UiFieldTranslator.java wandelt die FieldTiles in einem short Array um und verwendet die Methode **InternalLedGameThread.showImage()**, um das finale Bild darzustellen.

Persönliche Kommentare zum Projekt und Zusammenfassung

Miguel Meindl:

Muhammad Daniel:

Wie organisiere ich meinen Code? Wie verwende ich Klassen, sodass sie die Übersichtlichkeit dienen? Auf welche Weise kann das Spiel neu gestartet werden, ohne dass das ganze Programm beendet werden muss?

All diese Fragen (und andere) wurden mehr oder weniger während des Ablaufs dieses Projekts beantwortet.

Dass das Projekt eine Gruppenarbeit war, ist übrigens toll. Miguel hat mir sehr geholfen, wenn ich Probleme bei der Implementierung hatte. Seine gegebenen Feedbacks und Verbesserungsvorschläge habe ich sehr wertvoll gefunden.

Im Allgemeinen war dieses Projekt eine echt gute Lernerfahrung.

Zusammenfassung: