

Pacman - Dokumentation

von Leonard Tomczak

Anfangsideen:

Es sollten möglichst viele Features des Original-Pacmans implementiert werden:

- Geister-AI
- Fruits bzw. einsammelbare Objekte
- „Powerpellets“, die dem Spieler erlauben die Geister zu fressen
- mögliche Portale am Rand des Levels, die dem Spieler erlauben von einem Rand zum anderen zu springen
- Highscores
- Start-/Pausebildschirm

Zusätzlich sollte ein Level-Editor, mit dem man schnell neue Level erstellen kann

Umsetzung:

Da viele Elemente des Spiel gemeinsame Attribute, wie z.B. Koordinaten, oder Methoden, z.B. *Render()*, zum anzeigen des Objekts, haben, habe ich die Klasse *GameObject* erstellt, die die Basisklasse aller Objekte darstellt. Dies hat den Vorteil, dass wir alle Spielelemente in einer Liste von *GameObject*'s abspeichern können und dann mithilfe einer Schleife bei allen Objekten die *Update()*- und *Render()*-Methoden aufrufen können.

Es gibt verschiedene Variablen, die angeben, in was für einem Status sich das Spiel befindet:

- *inMenu* : im Startmenu
- *displayHighscores* : Highscores anzeigen
- *inPauseMenu* : Spiel pausieren
- *intro* : Spiel ist in der Countdown-Sequenz (3,2,1,Start)

Je nach ihren Werten, werden in der *draw()*-Methode verschiedene Methoden aufgerufen.

Da es sich bei dem Spielmenu, dem Pausemenu und der Highscoreansicht um nicht wesentliche Teile des Spiels handelt, werde ich im Folgenden nur den Aufbau bzw. den Ablauf des Spiels selbst dokumentieren.

Das Level:

Um die Anzeige des Levels bzw. die Abspeicherung der Level-spezifischen Daten kümmert sich die *Level*-Klasse. Sie hat ein Attribut *worldMatrix* vom Typ eines 2-dimensionalen Integer-Arrays, in dem die Information, was an einer bestimmten Position ist, also etwa Wand, Dot (die Pacman einsammeln muss) oder eine Superpille, enthalten sind. Weiter werden in dieser Klasse die Informationen über die Spielphase gespeichert, d.h. wie sich die Geister verhalten (später mehr dazu) oder ob Pacman die Geister essen kann oder nicht. Die *Level*-Klasse enthält verschiedene Methoden, die zur Umwandlung von globalen Koordinaten in Level-Koordinaten, also die Indizes in der *worldMatrix*. In der *Render()*-Methode wird die *worldMatrix* durchlaufen und die einzelnen Objekte gezeichnet, dabei werden bei den Wandteilen nur die Umrisse gezeichnet, d.h. zwischen zwei angrenzenden Wandteilen wird keine Linie gezeichnet. In der *Update()*-Methode auf Kollisionen zwischen den Geistern und dem Spieler überprüft. Je nach Spielphase wird entweder dem Spieler dann ein Leben abgezogen und die Todesanimation gestartet oder ein Geist kehrt zum Geisterhaus zurück und der Spieler erhält Punkte. Außerdem sorgt das *Level*-Objekt, dafür, dass eine Frucht gespaunt wird, wenn nur noch ein Drittel oder zwei Drittel der ursprünglichen Dots noch vorhanden sind.

Sind alle Punkte eingesammelt wird das nächste Level geladen, sofern es eins gibt. Die Level-Dateien werden mithilfe des Level-Editors erstellt und dann abgespeichert.

Der Spieler:

Die *Player*-Klasse kümmert sich um die Bewegung von Pacman. Drückt der Spieler eine Pfeiltaste (die Tastaturabfrage selbst findet nicht in der *Player*-Klasse statt, die Information wird ihr aber weitergegeben), wird dies in der *inputQueue*-Variable gespeichert. In der *Update()*-Methode wird dann jedes mal überprüft, ob Pacman die Richtung entsprechend ändern kann oder nicht und tut dies gegebenenfalls. D.h. drückt der Spieler beispielsweise die linke Pfeiltaste, wartet das Programm solange bis links neben Pacman keine Wand mehr ist und ändert dann die Richtung. Zudem wird überprüft, ob Pacman einen Punkt oder eine Superpille eingesammelt hat, wodurch der Spieler Punkte kriegt und sich die Spielphase gegebenenfalls ändert.

Die Geister:

Die Geister haben jeweils eine *targetPoint*-Variable, also einen Punkt im Level, den sie anstreben. Je nach Spielphase, Geist und Spielerposition ist dieser unterschiedlich. Im Allgemeinen sind die Geister entweder in einer *Chase*- oder in einer *Scatter*-Phase. In der *Chase*-Phase verfolgen sie *Pacman*, während sie in der *Scatter*-Phase in einer Ecke des Levels patrouillieren. Diese Phasen wechseln in bestimmten Zeitintervallen

Die Bewegung lässt sich in zwei Teile einteilen:

- 1) Das Bestimmen der Bewegungsrichtung, um zu dem Zielpunkt zu gelangen
- 2) Das Bestimmen des Zielpunktes

Zu 1):

Sofern sich der Spieler nicht im *energized*-Modus befindet, d.h. er nicht die Geister essen kann, überprüft ein Geist, in wieviele mögliche Richtungen er gehen kann. Sind es nur zwei, also z.B. bei einer geraden Strecke zwischen zwei Wänden, bewegt er sich in diejenige, von der aus er nicht gekommen ist, also bewegt er sich weiter. Sind es drei oder vier mögliche Richtungen, also bei einer Kreuzung, wird das Feld bestimmt, das den geringsten Abstand (Luftlinie!) zum Zielpunkt hat. Auch hier wird nicht das Feld gewählt, von dem der Geist herkommt. Bei einer möglichen Richtung muss er umkehren. Sonst dreht sich der Geist nur um, wenn sich die Spielphase ändert, also entweder der Spieler eine Superpille gegessen hat oder sich der Geistermodus von *Scatter* zu *Chase* oder umgekehrt ändert.

Es ist anzumerken, dass diese Strategie nicht immer den schnellsten Weg zum Zielpunkt angibt und bei bestimmten Level auch dazu führen kann, dass sich ein Geist (in der *Chase*-Phase) im Kreis bewegt.

Sollte sich der Spieler im *energized*-Modus befinden, sucht sich der Geist immer eine zufällige Richtung aus, die nicht der entspricht, aus der er gekommen ist.

Zu 2):

In der *Scatter*-Phase ist jedem Geist eine Ecke zugeordnet, die sein Zielpunkt darstellt.

In der *Chase*-Phase wird der Zielpunkt je nach Geist unterschiedlich bestimmt:

Der rote Geist wählt als seinen Zielpunkt die Position des Spielers.

Der Zielpunkt des blauen Geistes wird bestimmt, indem als erstes das Feld zwei Felder vor Pacman berechnet wird. Der Zielpunkt ist dann das Feld, das zwei mal so weit von diesem vom roten Geist aus betrachtet entfernt liegt.

Das Ziel des pinken Geistes ist das Feld, das vier Felder vor Pacman liegt.

Beim orangenen Geist ist es unterschiedlich, wie weit dieser von Pacman entfernt ist. Ist er mehr als 8 Felder entfernt, ist der Zielpunkt Pacman selbst, ansonsten die Ecke aus der *Scatter*-Phase.

Indem man die *debugView*-Variable im Hauptprogramm auf *true* setzt, werden die Zielpunkte der Geister jeweils angezeigt.

Die Informationen zu der Geister-Bewegungen stammen von:

<http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior>

Ist das Spiel beendet, wird überprüft, ob es sich um einen neuen Highscore handelt und ggf. abgespeichert.

Das Klassendiagramm befindet sich als png-Datei im selben Verzeichnis wie dieses Dokument, da es unübersichtlich werden würde, jenes in dieses zu tun.

Bemerkung: Die Credits für das letzte Level (level7.txt) gehen an Dominik Duda.

Anleitung:

Das Projekt mithilfe einer der .pde-Dateien in der obersten Ebene des Pacman-Ordners in Processing öffnen und dann starten.

Pfeiltasten - Bewegung

p - Pause/Resume

Um verschiedene Level auszuprobieren, ohne alle die, die davor kommen durchzuspielen, kann einfach die *currentLevelIndex*-Variable im Hauptprogramm geändert werden.

Bekannte Probleme (Natürlich sind dies bloß Features und nicht Bugs oder ähnliches):

- Ab und zu ist es vorgekommen, dass ein Geist in einer Wand hängenbleibt, was nur durch das Abschließen des Levels behoben werden konnte