

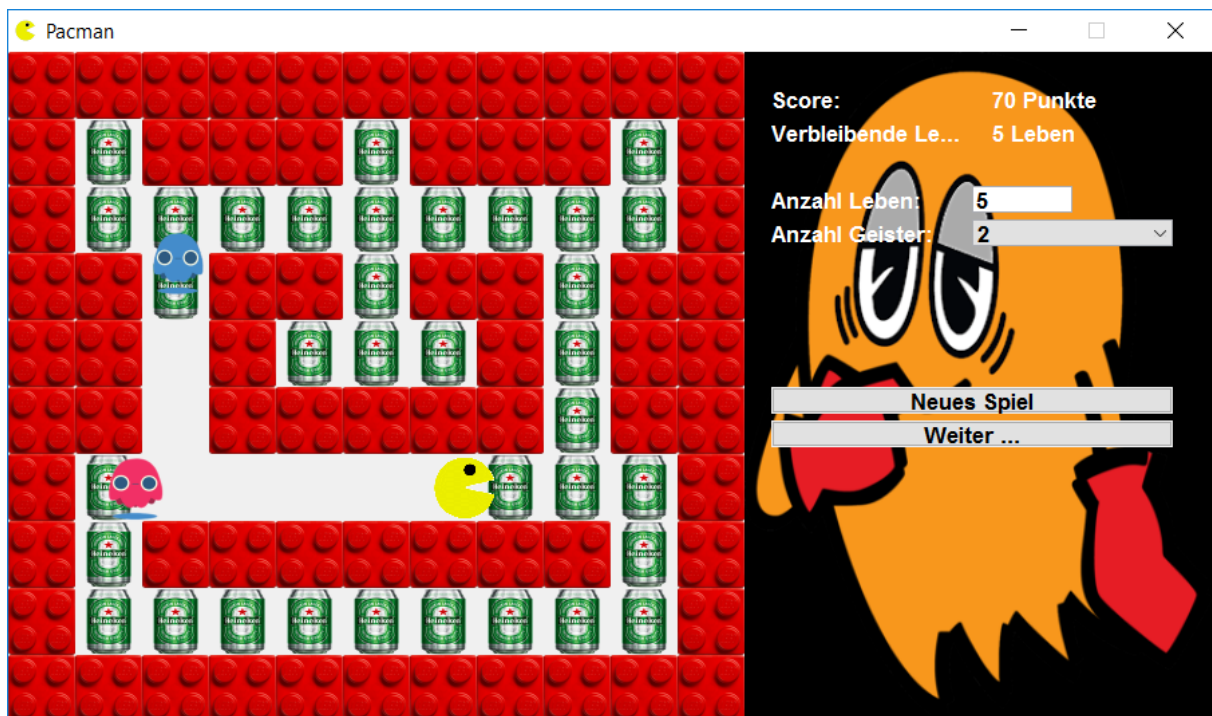
JAVA PRÜFUNG

Bedingungen:

- Erlaubte Hilfsmittel: Unterrichtsunterlagen, Java Buch und Übungen.
- Die Prüfung ist schrittweise, gemäss Aufgabenstellung lokal auf Ihrem Computer zu lösen. Kopieren sie zu diesem Zwecke den gesamten Ordner *P_HS17_18_Pacman_Vorlage* auf Ihre lokale Harddisk und importieren sie das Projekt in Eclipse. Am Ende der Prüfung ist der Ordner *src* umbenannt in *NameVorname* abzugeben.
- Setzen sie als erstes Ihren Namen und Vornamen in die Dateien.
- Gegenseitiges Abschreiben in irgendeiner Form führt zur Note 1!
- Folgend sie bei der Wahl von Variablen exakt den Angaben in der Aufgabenstellung.
- Die Beilage mit dem Layout muss unterschrieben zurückgegeben werden!

Beschreibung:

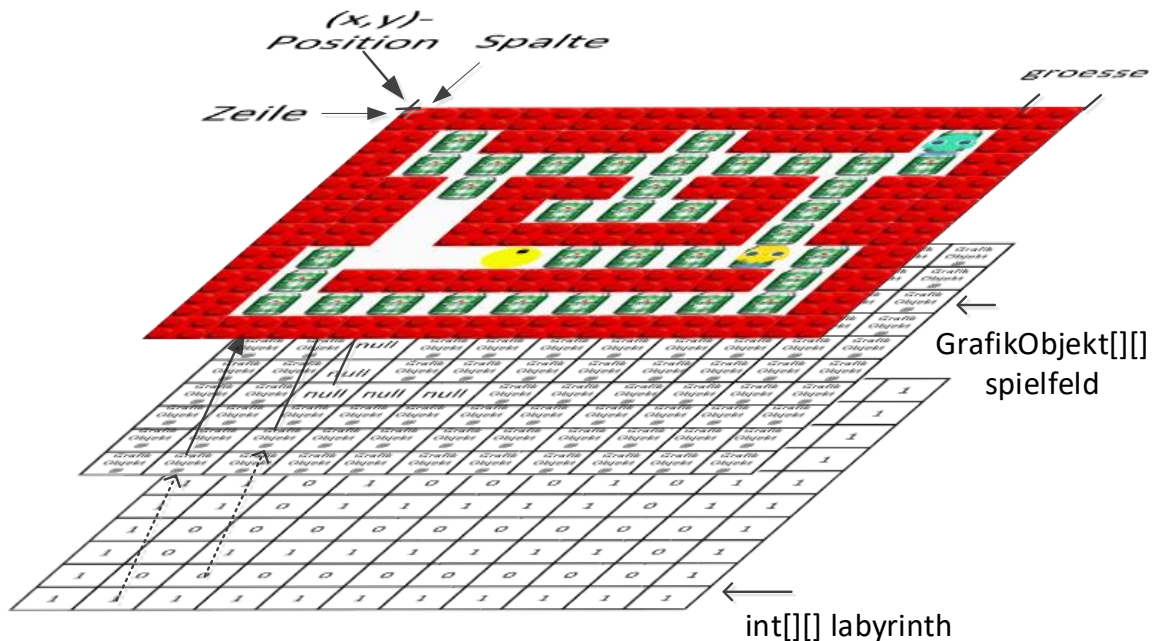
Ziel dieser Prüfung ist es, die Spielanwendung Pacman zu programmieren. Pacman kann mit den Pfeiltasten gesteuert werden und trinkt in unserem Fall Bier¹. Er wird von Geistern heimgesucht, die sich zufällig durch das Labyrinth bewegen. Wenn es zu einer Kollision mit einem Geist kommt, verliert Pacman ein Leben und das Spiel kann mit dem *Weiter-Button* fortgesetzt werden. Sobald alle Leben verbraucht oder alle Dosen getrunken wurden, kann mit dem Button *Neues Spiel* ein anderes Spiel mit der gewählten Anzahl von Leben und Geistern gespielt werden. Das Spiel hat zum Ziel, in wenigen Leben viele Biere zu trinken, jedes Bier zählt den Score um 10 nach oben.



Die Anwendung ist im Null-Layout gestaltet und das Layout ist entsprechend den Angaben im Beiblatt organisiert. Die Elemente der Benutzeroberfläche und das Pacman Spielfeld befinden

¹ Für abstinente Pac-Men oder schwangere Pac-Women ist auch Cola vorhanden ...

sich auf der *TopView*, die von *JPanel* erbt. Die *TopView* verfügt über einen *SimpleTimer*, um die notwendigen Updates auszulösen. Das Pacman Spielfeld ist ein *JPanel* und hat ein zweidimensionales Ganzzahl-Array labyrinth, das das Labyrinth definiert:



Eine 1 im Ganzzahl-Array *labyrinth* repräsentiert ein grafisches Objekt mit einem Lego-Block, eine 0 steht für ein grafisches Objekt mit einer Bierdose. Die entsprechenden Objekte werden im zweidimensionalen *GrafikObjekt*-Array *spielfeld* gespeichert. Wenn ein Bier von Pacman konsumiert wird, wird der entsprechende Eintrag auf null gesetzt.

Das Pacman-Spielpanel verfügt weiter über einen Array mit Geistern und über den Pacman. Der *KeyListener* des Pacman-Spielpanels wird automatisch bei sämtlichen Komponenten registriert. Die Klasse *Geist* ist bereits vorgegeben. Die Klassen *Pacman* und *Geist* erben je von *AnimObjekt*. *AnimObjekt* ist die um Animation und Labyrinth erweiterte Version von *GrafikObjekt*. Die Klasse *GrafikObjekt* dient als Basis und hat neben der (x,y)-Position des Objektes noch deren Gösse und das zugehörige Bild.

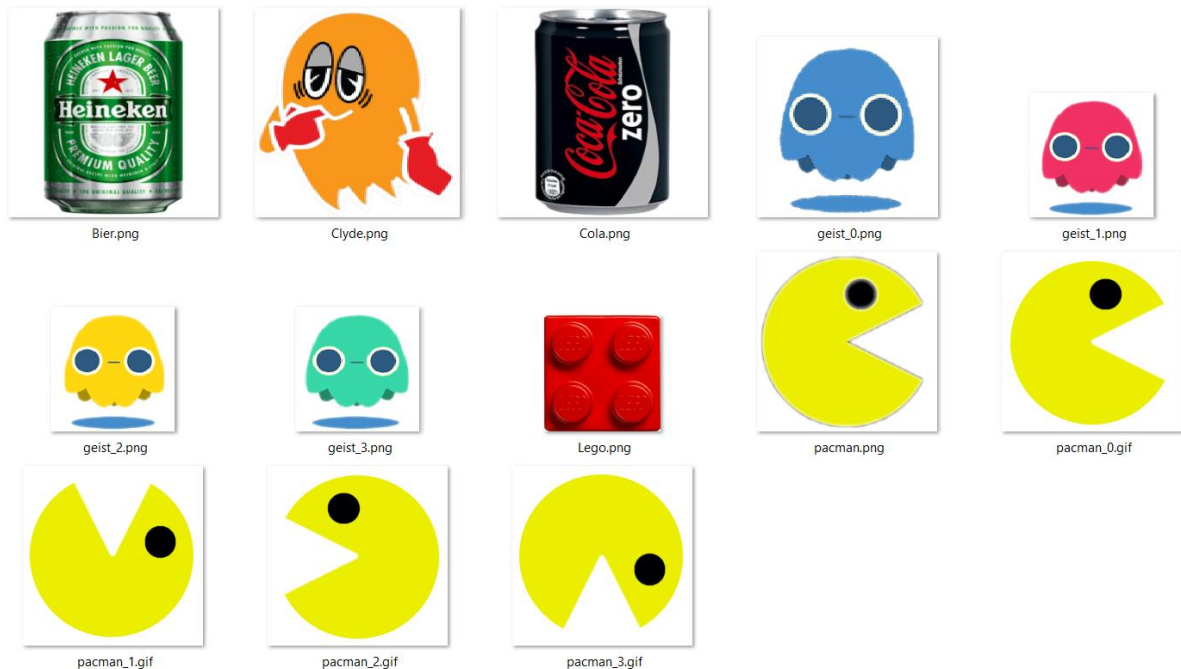
Die Aufgabenstellung führt Sie schrittweise zum Ziel. Folgen Sie daher beim Lösen der Prüfung der Aufgabenstellung. Sämtliche Bilder sind im Ordner *src/bilder* zu finden und selbst sprechend bezeichnet.

Achten Sie darauf, dass Ihr Code kompilierbar bleibt!

Aufgabe 1: Erstellen des Grundgerüsts (~5 Pte.)

Als Erstes wollen wir das Grundgerüst des Programms anhand des Klassendiagramms kontrollieren und allenfalls ergänzen. Die meisten Elemente der Klassen sind dabei bereits gegeben.

- a) Ergänzen sie sämtliche *Attribut-* und *Methodendeklarationen* gemäss Klassendiagramm. Machen Sie sich mit den vorhandenen Attributen und Methoden-Deklarationen sowie mit den im Ordner `src/bilder` abgelegten Bildern bekannt:



Hinweis: Die .gif – Bilder sind animiert!

Aufgabe 2: Klasse *TopView*: User - Interface (~30 Pte.)

Als Nächstes wollen wir das User – Interface sowie die zum *ActionListener* und zum *SimpleTimerListener* zugehörigen Methoden sowie *paintComponent()* implementieren. Die Anordnung der Elemente geschieht im *null – Layout* gemäss Beilage.

- a) Kompletieren Sie die Klasse *TopView* gemäss Dokumentation und Beilage.

Hinweis: Der gewählte Index der *JComboBox* kann, durch Aufruf von *getSelectedIndex()* des entsprechenden Objektes, ausgelesen werden.

Aufgabe 3: Klasse *GrafikObjekt* (~5 Pte.)

Nun wollen wir uns der Klasse *GrafikObjekt* zuwenden. Achten Sie beim Laden des Bildes darauf, dass Sie *groesse* als Breite und Höhe des Bildes verwenden!

- a) Implementieren Sie die Klasse *GrafikObjekt* gemäss Dokumentation.

Aufgabe 4: Klasse *AnimObjekt* (~10 Pte.)

Objekte der Klasse *AnimObjekt* werden durch Angabe von (spalte, zeile) im Spielfeld positioniert. Die entsprechenden Attribute müssen im Konstruktor aber nicht gesetzt werden.

- a) Implementieren Sie den Konstruktor und die Methoden der Klasse *AnimObjekt* gemäss Dokumentation.

Aufgabe 5: Klasse *Pacman* (~20 Pte.)

Die Klasse *Pacman* erweitert *AnimObjekt* um die Pacman-Spezifischen Eigenschaften. Der Pacman bekommt via Pfeiltasten die neu einzuschlagende Richtung und hat zu deren Identifikation entsprechende static finals.

- a) Implementieren Sie den Konstruktor und die Methoden der Klasse *Pacman* gemäss Dokumentation.

Hinweis: Der Image-Array der Grösse 4 wird lediglich für die letzte Aufgabe benötigt!

Aufgabe 6: Klasse *PacmanSpielPanel* (~40 Pte.)

Das *PacmanSpielPanel* beheimatet das Spielfeld und hat gemäss Klassendiagramm die entsprechenden Elemente. Der zugehörige *KeyListener* wird automatisch bei allen Komponenten registriert.

- a) Implementieren Sie den Konstruktor und die Methoden der Klasse *PacmanSpielPanel* gemäss Dokumentation.
- b) Ergänzen Sie den Code um noch allfällig fehlenden Code.

Aufgabe 7: *Challenge League* (~10 Pte.)

- a) The playground is yours: Ergänzen Sie den Code so, dass Pacman beim Laufen jeweils in die richtige Richtung schaut!