

Es sollen die Original-Spielregeln umgesetzt werden.

Zusätzlich/Abweichend sind folgende Abläufe umzusetzen:

vor Spielbeginn:

Der Benutzer soll vor Spielbeginn für jeden der vier möglichen Spieler festlegen, ob er mitspielt, wie er heißt, ob er menschlich oder von der normalen oder erweiterten KI gesteuert wird.

Es muss mindestens ein Spieler spielen, damit das Spiel gestartet wird.

Es können auch ausschließlich KIs gegeneinander spielen.

Um die Spieldauer zu verkürzen, soll auch mit weniger Schatzkarten gespielt werden können. Abweichend von den Regeln wird dafür vor Spielstart außerdem festgelegt, wie viele Schatzkarten jeder Spieler erhält (1..24/Spieleranzahl). Per Default sei die maximal mögliche Anzahl eingestellt.

Spielfelderstellung:

Die unbeweglichen Gängekarten (siehe Spielplan in der Spielanleitung) bestehen in den Spielfeldecken aus L-Stücken in der entsprechenden Rotation, an den Rändern aus mit nach außen geschlossenen T-Stücken und an den mittleren 4 Positionen aus mit jeweils um 90 Grad im Uhrzeigersinn gedrehten T-Stücken, beginnend mit einem aufrechten T rechts oben.

Als frei bewegliche Gängekarten existieren 15 L-Stücke, 13 I-Stücke und 6 T-Stücke. Diese werden in zufälliger Rotation an zufällige freie Positionen auf dem Spielfeld verteilt, eine Gängekarte bleibt übrig.

Abweichend vom fertig gedruckten Brettspiel werden die 24 Schätze beliebigen Gängekarten zugeordnet, lediglich die 4 Eckkarten des Spielfeldes erhalten garantiert keinen Schatz. Ein Schatz muss für die Anzeige nicht wie oben dargestellt zusammen mit seiner Gängekarte gedreht werden, darf es aber.

im Spiel:

Es beginnt immer der Spieler mit der gelben Spielfigur, wenn dieser nicht mitspielt der erste im Uhrzeigersinn folgende teilnehmende Spieler.

Ist ein menschlicher Spieler am Zug, so kann er über den Tastendruck <H> erfahren, welchen Schatz er als Nächstes aufsuchen soll. Dafür wird die Gängekarte mit dem entsprechenden Schatz während des Tastendrucks hervorgehoben.

Ein menschlicher Spieler kann die freie Gängekarte drehen und dann wählen, an welcher Stelle er sie einschiebt. Gestaltet hierfür eine möglichst intuitiv nutzbare Variante.

Ist ein Mensch am Zug und bewegt den Mauszeiger über eine Gängekarte des Feldes, so erscheint diese leicht grün, wenn ein Weg vom aktuellen Spieler zu dieser Gängekarte existiert, andernfalls leicht rot.

Zum Bewegen der Figur klickt der Mensch die Gängekarte an, auf die seine Figur gesetzt werden soll.

Existiert ein Weg bis zu dieser Position, wird die Figur dort hinbewegt und der nächste Spieler ist am Zug.

Erreicht ein Spieler einen Schatz, den er einsammelt, so wird dieser Schatz von der Gängekarte entfernt.

Anzeige:

Das Spielfeld soll zu jedem Zeitpunkt komplett sichtbar sein. Diese Bilder sollen verwendet werden. Wird das Fenster vergrößert, so vergrößert sich auch das Spielfeld, allerdings bleibt es dabei quadratisch.

Zu jedem Zeitpunkt sollen sichtbar sein

- die freie Gängekarte in der aktuellen Rotation
- welcher Spieler am Zug ist
- wie viele Schätze jeder Spieler noch einsammeln muss.

Über eine Menüleiste soll ein neues Spiel begonnen, ein Spiel geladen, das Spiel gespeichert und das Programm beendet werden können.

Animation:

Wird eine Gängekarte eingeschoben, so sollen alle Gängekarten der Reihe/Spalte animiert in diese Richtung bewegt werden.

Die Figur soll sich flüssig animiert von Karte zu Karte durch die Gänge zu der gewählten Position bewegen.

Die Geschwindigkeit für die Bewegung der Gängekarten und davon unabhängig die Geschwindigkeit für die Bewegung einer Figur soll jeweils über einen Menüpunkt für alle kommenden Animationen frei einstellbar sein. Die Animationen können dadurch auch jeweils komplett deaktiviert werden.

KI

normale KI

Die freie Gängekarte ist so einzuschieben, dass der Spielstein möglichst dicht an das nächste Ziel gesetzt werden kann, im Idealfall direkt zum Ziel. Der Abstand zwischen zwei Gängekarten ist dabei durch horizontale plus vertikale Entfernung zu bestimmen. Eine über die Ecke benachbarte Gängekarte ist also weiter entfernt, als eine mit einer Kante anliegende Gängekarte.

Kann der Spieler (Computer) durch keine Einschubvariante einen Weg erzeugen, die ihn näher an das Ziel bringt als bisher, so schiebt er eine Spalte oder Reihe mit dem Ziel oder mit dem Spielstein. Befinden sich Schatz und Spieler auf einer nicht verschiebbaren Gängekarte, so wird ein beliebiger Schiebevorgang vorgenommen.

Beschreibt in der Dokumentation, wie Eure KI vorgeht, wenn mehrere gleichwertige Positionen oder Schiebeaktionen möglich sind.

erweiterte KI

Erstellt eine erweiterte KI, die sowohl bei Positionierung als auch bei mehreren möglichen gleichwertigen Schiebeaktionen ein von Euch definiertes schlauerer Vorgehen aufweist als Eure normale KI. Beispielsweise ist bei Nicht-Erreichbarkeit des Schatzes eine Position "über Eck" meist sinnvoller als eine an der Kante benachbarte Position, da dann der Schatz im nächsten Schiebevorgang erreichbar wird.

Beschreibt dieses Vorgehen in der Dokumentation gut und erstellt Spielstandsdateien (siehe unten), mit denen man das korrekte Vorgehen dieser KI prüfen kann.

Log

Damit der Programmierer im Falle einer unerwarteten Programmreaktion die Schritte nachvollziehen kann, wird in eine Datei ein Log geschrieben. Diese Datei wird bei jedem Spielstart überschrieben bzw. neu erstellt.

Zu Spielbeginn werden die vorgenommenen Initialisierungen dokumentiert, nach jedem Zug die vorgenommene Schiebe- und Positionierungsaktion. Der Text darf ausführlicher gestaltet werden, als folgend angegeben.

Beispiel:

player0 is human, has treasures 1, 9, 2, 7

player1 is AI 1, has treasures 6, 5, 24

player2 not involved

player3 not involved

player0 shifts row 1 right with waycard {L, rot 1, treasure 0}

player0 collects treasure 1, next treasures are 9, 2, 7

...

Der Text darf in der Logik erstellt werden, da er nicht der Kommunikation mit dem Benutzer dient.

Nach jedem Schreibvorgang muss die Datei geschlossen werden, damit im Falle eines Fehlers keine Information verloren geht.

Spielstandsdatei

Immer, wenn ein Mensch am Zug ist, soll er den aktuellen Spielstand in einem frei gewählten Verzeichnis speichern oder sich den Spielstand aus einer frei gewählten Datei laden können.

Eine Spielstandsdatei ist im json-Format abgelegt und enthält folgende Elemente:

- das Feld bestehend aus 7 Spalten mit jeweils 7 Zellen
- jede der 49 Zellen hat eine Angabe für
 - i) Typ der Gängekarte (I = 0, L = 1 und T = 2)
 - ii) Rotation der Gängekarte in 90-Grad-Schritten im Uhrzeigersinn (0..3), die nicht rotierte Gängekarte entspricht dem Schriftbild ihres Buchstabens
 - iii) Nummer des Schatzes (1..24) und 0 für keinen Schatz
- freie Gängekarte mit Typ, Rotation, Schatznummer und Position.
- Die Position ist zu Spielbeginn (-1, -1), nach einem Schiebevorgang liegt sie an der herausgeschobenen Position, also ist dann entweder die x- oder die y-Koordinate um eins kleiner bzw. größer als eine Koordinatenangabe im Spielfeld
- den aktuellen Spieler (0..3), Spieler 0 ist immer der gelbe Spieler
- für jeden der vier Spieler die Angaben:
 - i) nimmt teil
 - ii) Name
 - iii) gesteuert von Mensch = 0, AI normal = 1, AI erweitert = 2
 - iv) Position (0,0 definiert links oben)
 - v) die Nummern der einzusammelnden Schätze (1..24) in der Reihenfolge, in der sie aufgesucht werden müssen

Die Datei soll auch geladen und gespielt werden können, wenn eine von der Vorgabe abweichende Anzahl von I-, L- oder T-Gängekarten gespeichert wurden (z.B. gar keine I-Gängekarten) oder die unbeweglichen Gängekarten anders belegt sind als vorgegeben.

Bei der Bewertung werden Spielstandsdateien genau in diesem Format eingesetzt und müssen gelesen werden können. Enthält eine Spielstandsdatei Fehler, so muss dieser möglichst präzise an den Benutzer gemeldet werden.

GUI

Alle für die Bedienung des Spiels notwendigen Komponenten müssen in einer FXML-Datei vorhanden, aber noch nicht nach Programmstart bedienbar sein. Komponenten, die dynamisch erstellt werden sollen (z. B. das Spielfeld), müssen nach Programmstart bereits sichtbar sein.

Ist beispielsweise ein zusätzliches Fenster notwendig, so muss dies mit seinen Komponenten vorhanden sein, aber noch nicht in einem Programmfluss geöffnet werden.

Der Spielplan muss erkennbar sein und sich in seiner Größe anpassen, wenn die Fenstergröße verändert wird. Es darf in allen Zellen die gleiche Gängekarte angezeigt werden.

Alle Figuren müssen an ihren Startpositionen stehen und sich bei Ändern der Fenstergröße mit bewegen.

In der Logik müssen Datenstrukturen erkennbar sein

Über einen Konstruktor muss eine Spielsituation erstellbar sein.

Die Datenstruktur des Feldes muss abgelegt sein.

Die Methodensignatur zum Ermitteln eines KI-Zuges muss vorhanden sein.

Tests dürfen noch fehlschlagen, müssen aber vorhanden und kompilierbar sein und natürlich die korrekten Erwartungswerte enthalten. Mehrere Konstruktoren sind zu erstellen, um eine Spielsituation für einen Test erstellen zu können. Mindestens folgende Testfälle müssen abgedeckt werden:

- Ist nach einem Spielzug der nächste Spieler am Zug?
- Ist ein Schatz an der erwarteten Position?
- Führt ein Weg von einer Position zu einer anderen?
- KI: Wird die korrekte Reihe geschoben, damit das Ziel erreichbar wird?
- KI: Wird die korrekte Spalte geschoben, damit das Ziel erreichbar wird?
- KI: Wird eine Verschiebung gemacht, ohne dass ein bestehender Weg zum Ziel zerstört wird?

Tipp:

Zum einfachen Testen könnte ein Konstruktor erstellt werden, der ein Spielfeld in Form eines Strings entgegennimmt, z.B.

```
"L100,I101,T000,I102,L200\n" +
```

```
"I003,I104,I005,I106,I007\n" +
```

```
"T300,I108,T000,I109,T100\n" +
```

```
"I010,I111,I012,I113,I014\n" +
```

```
"L000,I115,T200,I116,L300"
```

So kann sehr einfach ein kleineres Spielfeld definiert werden. Hier ist ein 5*5-Spielfeld definiert. Der Buchstabe gibt den Wegtyp an, die erste Ziffer die Rotation und die folgende Zahl die Schatznummer. In diesem Beispiel bestehen alle beweglichen Gängekarten aus I-Stücken und beherbergen einen Schatz, alle unbeweglichen Gängekarten enthalten keinen Schatz.

An so einem kleinen (oder sogar noch kleineren) Spielfeld können alle Tests durchgeführt werden.

Für die Überprüfung der Wegfindung sollte auch ein Feld mit einem verschlungenen Weg definiert werden.