

Sie sind hier: **FH Wedel** > **Mitarbeiter** > **Gerit Kaleck** > **Programmierpraktikum (Java)** > **Aufgaben/Termine** > **SS18: CityDomino**

## Aufgabenstellung: CityDomino

Zu implementieren ist ein Dominospiel, bei dem vier Spieler jeweils ihre eigene Stadt gestalten. Ziel des Spiels ist es, möglichst viele Stadtteile mit Prestige zu gestalten.

### Spielregeln

Jeder Spieler besitzt ein eigenes 5\*5-Zellen großes Spielfeld und legt zu Beginn sein **Stadtzentrum** mittig ab.

Ein Spielbeutel für alle Spieler enthält 48 Spielkarten in der Größe von zwei Zellen, die auf ihren zwei Hälften jeweils einen (evtl. auch den gleichen) Stadtteiltyp anzeigen. Die Stadtteiltypen unterscheiden sich durch Bild und Hintergrundfarbe voneinander. Jede Spielkarte besitzt eine definierte Wertigkeit. Auf manchen Stadtteilen sind zusätzlich ein bis drei Prestigesymbole abgebildet.

Es werden vier Karten gezogen und im **ersten Auswahlbereich** angezeigt. Dabei wird die niederwertigste Karte zuoberst, die höchstwertigste zuunterst einsortiert. Der erste Spieler markiert die Karte im Auswahlbereich, die er gerne nehmen würde, die anderen Spieler treffen ihre Auswahl der Reihe nach ebenfalls und markieren die jeweils gewünschte Karte.

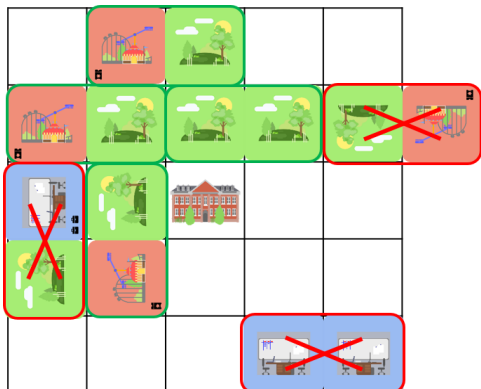


Wurden alle Karten markiert, dann werden wieder vier Karten gezogen und ebenso sortiert im **zweiten Auswahlbereich** angezeigt.



### Spielablauf

Derjenige, der die oberste Karte im ersten Auswahlbereich markiert hat, beginnt eine Runde, es folgen der Reihe nach die Spieler, die die jeweils darunterliegende Karte markiert haben. In einer Runde wird zunächst eine Karte aus dem zweiten Auswahlbereich markiert und dadurch für die kommende Runde gewählt. Je wertvoller also seine markierte Karte in dieser Runde ist, desto später ist der Spieler am Zug und desto weniger Auswahl hat er für die kommende Runde.



### Anlegeregeln

Die erste Karte muss an das Stadtzentrum angrenzen. An das Stadtzentrum darf jeder Stadtteil angrenzen. Legt man eine Karte an eine andere Karte an, so muss mindestens eine Hälfte mit einer Seite an einen identischen Stadtteiltyp einer liegenden Karte angrenzen.

Passt die abzulegende Karte weder an das Stadtzentrum noch an eine bereits ausliegende Karte, so wird sie verworfen.

Alle Spielkarten müssen in das 5\*5-Feld passen, keine Hälfte darf hinausragen. Das Stadtzentrum muss aber nicht in der Mitte liegen, sondern kann im Spielverlauf verschoben werden, wodurch sich alle bereits gelegten Karten mit verschieben.

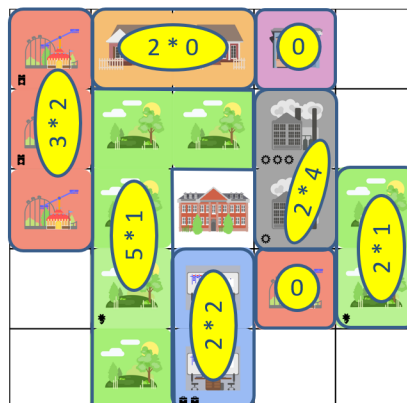
Eine abgelegte Karte kann nicht verschoben werden.

### Spielende

Wurden alle Spielkarten aus dem Beutel gezogen und von den Auswahlbereichen auf die Spielfelder platziert bzw. verworfen, werden die Punkte ermittelt.

- » Jede Stadt besteht aus mehreren Stadtteilen. Ein Stadtteil setzt sich aus waagrecht und/oder senkrecht verbundenen Zellen desselben Stadtteiltyps zusammen. Das Stadtzentrum zählt zu keinem Stadtteil dazu.
- » Die Punkte eines Stadtteils ergeben sich aus der Anzahl seiner Zellen multipliziert mit der Anzahl darin enthaltener Prestigesymbole.
- » Innerhalb einer Stadt kann es mehrere voneinander getrennte Stadtteile desselben Typs geben. Jeder Stadtteil ist einzeln auszuwerten.
- » Stadtteile ohne Prestigesymbole bringen keine Punkte.

Für die Auswertung wird für jeden Spieler die Summe der Punkte seiner Gebiete ermittelt. Gewonnen hat der Spieler mit den meisten Punkten. Bei einem Gleichstand gewinnt der Spieler mit dem größten einzelnen Gebiet. Besteht auch hier Gleichstand, so siegen beide Spieler gleichermaßen.



### KI

Außer dem menschlichen Spieler, der im Spiel stets beginnt, existieren 3 computergesteuerte Spieler. Diese sollen einer sehr primitiven Logik folgen:

- » bei der Auswahl wird die Karte markiert, mit der bei Auslage im eigenen Feld aktuell am meisten Punkte erzielt werden könnten
- » dafür wird für jede freie Karte der Auswahlbank auf jeder freien Position des Spielfeldes und in jeder Rotation der Punktgewinn ermittelt
- » bei Punktgleichheit mehrerer Positionen wird darauf geachtet, dass keine leeren Einzelzellen erzeugt werden
- » bei der Ablage wird die so ermittelte Position genutzt
- » die mögliche Verschiebung des Stadtzentrums wird nicht durchgeführt

Wer möchte, kann zusätzlich intelligentere KIs implementieren, die z.B. das Stadtzentrum verschieben, die Kartenwahl der kommenden Runde einbeziehen, verhindern, dass andere Spieler viele Punkte erhalten, oder Schlüsse aus den bereits abgelegten Karten ziehen.

## Oberfläche

Existieren müssen folgende Elemente:

- » ein Spielfeld für den menschlichen Spieler
- » ein erster Auswahlbereich für die aktuelle Runde
- » ein zweiter Auswahlbereich für die kommende Runde
- » das Legen einer Karte auf das Spielfeld per Drag & Drop, gültige Ablegepositionen werden beim DragOver sichtbar markiert
- » eine Möglichkeit, um die Karte zu verwerfen. Das kann z.B. ein Button sein, der zum Verwerfen der aktuellen Karte betätigt wird, oder auf den die aktuelle Karte gezogen wird. Verworfen Karten müssen nicht angezeigt werden.
- » die Spielfelder der drei KI-Spieler, so dass die dort abgelegten Karten jederzeit erkennbar sind.

Mögliche Lösungen für das Legen einer Karte:

- » Die aktuell zu legende Karte kann in einer Drehbox erscheinen, in der die Karte durch einfaches Anklicken gedreht werden kann. Hat sie die gewünschte Orientierung erreicht, so kann die Karte per Drag & Drop auf das eigene Spielfeld gelegt oder verworfen werden.
- » Die aktuell gedragte Karte wird durch Tastendruck unter dem Mauscursor gedreht.

Die Reihenfolge der Spieler muss erkennbar sein. Es muss also zugeordnet werden können, welches der angezeigten Spielfelder zu welcher Kartenauswahl im Auswahlbereich gehört (z.B. durch gleiche Symbole oder farbliche Markierungen an beiden Stellen).

Das Stadtzentrum eines Spielfeldes muss zusammen mit allen bereits gelegten Karten verschoben werden können, entweder per Drag & Drop oder beispielsweise durch Buttons am Spielfeldrand. Dabei dürfen keine Stadtteile aus dem Spielfeld geschoben werden.

Die Auswertung eines Spiels muss für jeden Spieler die erreichten Punkte pro Stadtteiltyp darstellen.

Die Bedienung des Spiels muss intuitiv möglich sein für jemanden, der die Spielregeln kennt.

Die Größe des Fensters darf zu Spielbeginn höchstens 1600 \* 900 Pixel betragen.

## Karten

Die für ein Spiel vorhandenen Karten sind in [dieser](#) Datei definiert. Die zu den Stadtteiltypen gehörigen Bilder findet man [hier](#).

Pro Zeile wird eine Karte mit ihren beiden Hälften und ihrem Wert festgelegt:

```
<Art><Symbolanzahl>,<Art><Symbolanzahl>,<Wert>
```

Art ist dabei der Anfangsbuchstabe eines Stadtteiltyps (**A**musement, **I**ndustry, **O**ffice, **P**ark, **S**hopping, **H**ome), die Symbolanzahl eine Ziffer von 0 bis 3. Eine mögliche Zeile wäre also

```
H1,P0,24
```

für eine Karte mit einem Symbol auf einem Haus und ohne Symbol in einem Park und einem Wert von 24 Punkten.

## Log

In einer Datei (gleichzeitig auch auf System.out) sind durchgeführte Aktionen zu protokollieren. Der zuerst angegebene Stadtteil einer Karte ist dabei immer der an der angegebenen Position, bei horizontaler Ausrichtung liegt der zweite Stadtteil rechts davon, bei vertikaler darunter. Beispiel:

```
BOT1 chose [H1, P0] at index 1 for next round
BOT1 put [A0, P2] horizontally to (1, 2)
```

```
HUMAN chose [P0, S0] at index 0 for next round
HUMAN dragged center to (2, 3)
HUMAN put [A0, A0] vertically to (0, 0)
```

```
BOT3 chose [00,I2] at index 3 for next round
BOT3 did not use [00, A1]
```

## Spielstand

Der aktuelle Spielstand soll gespeichert und geladen werden können. Laden/Speichern soll nur möglich sein, wenn der menschliche Spieler am Zug ist. Eine Spielstandsdatei enthält die 4 Spielfelder der Spieler in ihrer Reihenfolge (das erste Feld gehört immer dem menschlichen Spieler 0), die zwei Auswahlbereiche und die im Beutel verbliebenen Karten mit folgenden Bereichen:

```
<Spielfeld>
<Spielfeld>
<Spielfeld>
<Spielfeld>
<Bänke>
<Beutel>
```

Die einzelnen Bereiche enthalten jeweils eine Einführungszeile (einen Kommentar) gefolgt von Inhaltsangaben:

- » Ein Spielfeld enthält einen Kommentar, zu wem es gehört, und in Folge für jede Zelle eine Inhaltsangabe:
  - » '-' für eine nicht belegte Zelle,
  - » 'CC' für das Stadtzentrum und
  - » zwei Buchstaben für eine Hälftenbeschreibung.

Die Zellen sind durch Leerzeichen separiert. Beispiel:

```
<Spielfeld>
-- -- -- -- --
-- -- H1 P0 --
-- -- CC -- --
-- -- -- -- --
-- -- -- -- --
```

- » Die Bänke enthalten Angaben für die aufliegenden Karten und von wem diese bereits gewählt wurde. Die erste Bank kann weniger als vier Karten enthalten, in entsprechender Anzahl enthält die zweite Bank dann bereits Markierungen (sonst '-' für fehlende Markierung). Die erste Bank wird als erste Markierung immer Spieler 0 (den menschlichen Spieler) aufweisen. Beispiel:

```
<Bänke>
0 H1P0,2 P001,3 I1P0
- P0P0,- A0A0,1 H0A0,- P1H0
```

- » Der Beutel enthält alle im Beutel befindlichen Karten kommasepariert. Im folgenden Beispiel befinden sich nur noch 4 Karten im Beutel:

```
<Beutel>
P0P0,P0P0,A1H0,I2P0
```

## Zwischenstand

Bei der Abnahme des Zwischenstands müssen die Vorgaben erfüllt werden.

- » Datenstrukturen müssen erkennen lassen
  - » wie und wo die Spieler definiert werden.
  - » wie und wo die Spielfelder der Teilnehmer definiert werden.
  - » wie und wo die Auswahlbereiche und die getroffenen Wahlen definiert werden.
  - » in welcher Struktur die erreichten Punkte zusammengefasst werden.
- » Tests dürfen noch fehlschlagen, müssen aber schon vorhanden und kompilierbar sein. Mindestens folgende Testfälle müssen abgedeckt werden:
  - » Erzeugen eines Spielfeldes mit definierter Belegung der Zellen über einen Konstruktor.
  - » Überprüfen, ob eine Karte auf einem definierten Spielfeld an eine Position passt.
  - » Überprüfen der erreichten Punktzahl eines Spielfeldes in verschiedenen Belegungssituationen
    - » mit verbundenen Stadtteilen ohne Symbole
    - » mit verbundenen Stadtteilen mit Symbolen
    - » mit nicht verbundenen Stadtteilen ohne/mit Symbolen
  - » Erzeugen einer Spielsituation mit definierter Belegung der Spielfelder und der Ablagebereiche über einen Konstruktor.
  - » Überprüfen, ob die ermittelte Ablageposition für eine Spielkarte den KI-Regeln entspricht.
- » GUI: Es muss außer dem Drag&Drop noch keine Interaktion mit dem Benutzer möglich sein.
  - » In einer FXML-Datei müssen alle Bedienelemente enthalten sein, die für die Bedienung des Programms notwendig sind.
  - » Nach Programmstart muss ein Spielfeld mit zunächst per Konstante vorgegebener Kartenverteilung zu sehen sein. Eine Karte muss auf das Spielfeld gezogen und platziert werden können, ohne dass hierbei auf Anlageregeln geachtet werden muss.
  - » Das Spielfeld des menschlichen Spielers muss sich bei Vergrößerung der Fenstergröße ebenfalls vergrößern, die anderen Anzeigeelemente entsprechend an eine Seite angeschmiegt bleiben. Die Bilder dürfen dabei verzerrt werden, das Grid muss also nicht quadratisch sein.

## Hilfe

Im herausgegebenen Code für die Bonusaufgabe ist Drag&Drop einer Spielkarte bereits umgesetzt und kann so übernommen werden.

**Viel Erfolg!**

### Kontakt FH Wedel

Staatlich anerkannte  
Fachhochschule Wedel  
Gemeinnützige Schul-  
gesellschaft mbH  
Feldstraße 143  
D - 22880 Wedel  
Tel.: +49 (0)4103 - 8048-0  
Fax: +49 (0)4103 - 8048-39  
E-Mail: sekretariat@fh-wedel.de