

Visual Computing

Winter Semester 2020/2021, Uebung 06

Prof. Dr. Arjan Kuijper

Max von Buelow, M.Sc., Volker Knauthe, M.Sc.

Tetiana Rozenvasser, Tamer Tosun, Julian Schwind



TECHNISCHE
UNIVERSITÄT
DARMSTADT

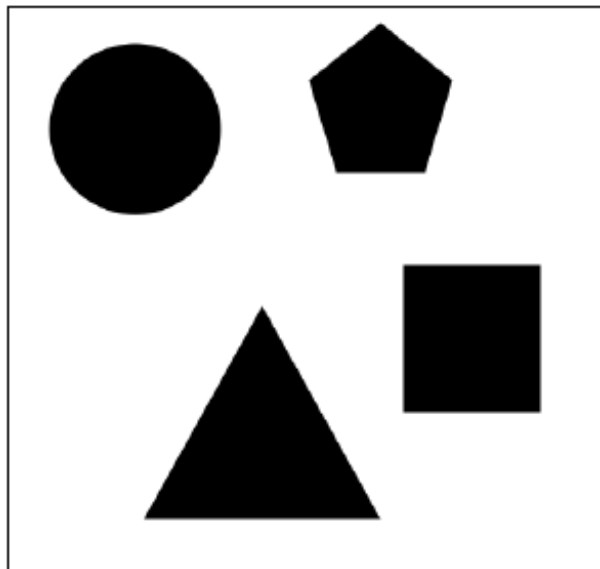
Übung 6 – Grafikpipeline Eingabemodalitäten VR+AR

Abgabe bis zum Freitag, den 18. Dez. 2020, 8 Uhr morgens, als PDF in präsentierbarer Form..

Aufgabe 1: Bäume (3,5 Punkte)

1a) (3 Punkte)

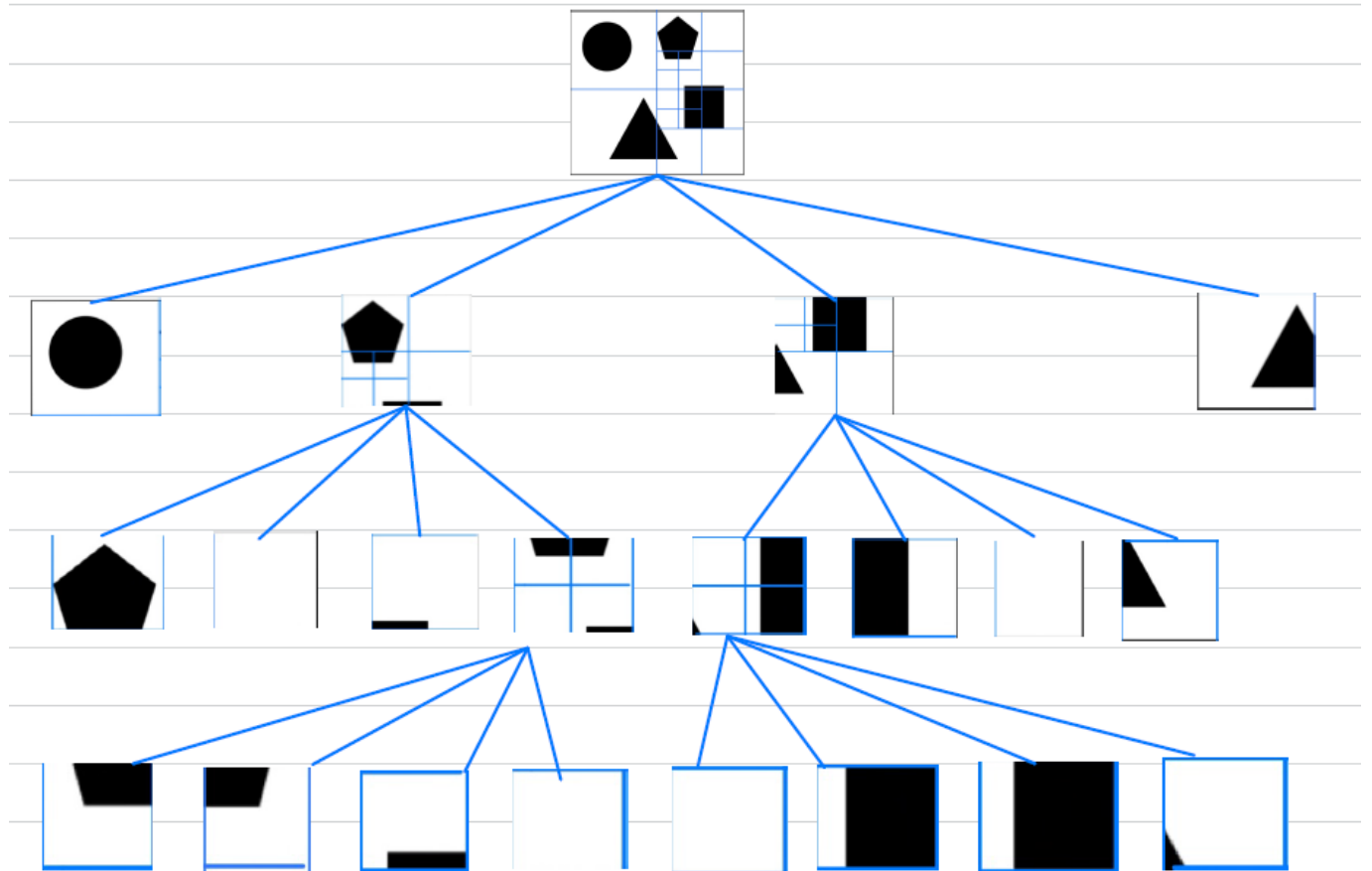
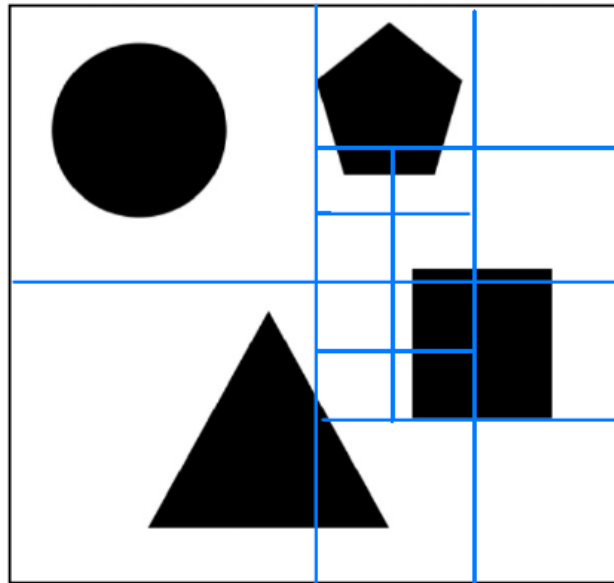
Erstellen Sie für die Abbildung einen Quadtree, einen BSP-Tree sowie einen kd-Tree. Zeichnen Sie dazu jeweils die entsprechenden Unterteilungslinien ein und stellen Sie dann jeden Baum als Node-Link-Diagramm dar. Es soll solange unterteilt werden bis jeder Blattknoten nur eines der 4 Elemente enthält. Die Reihenfolge der Knoten des Quadrees soll dem Uhrzeigersinn entsprechen (angefangen oben links) und der k-d-Tree startet mit einer horizontalen Linie.



Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

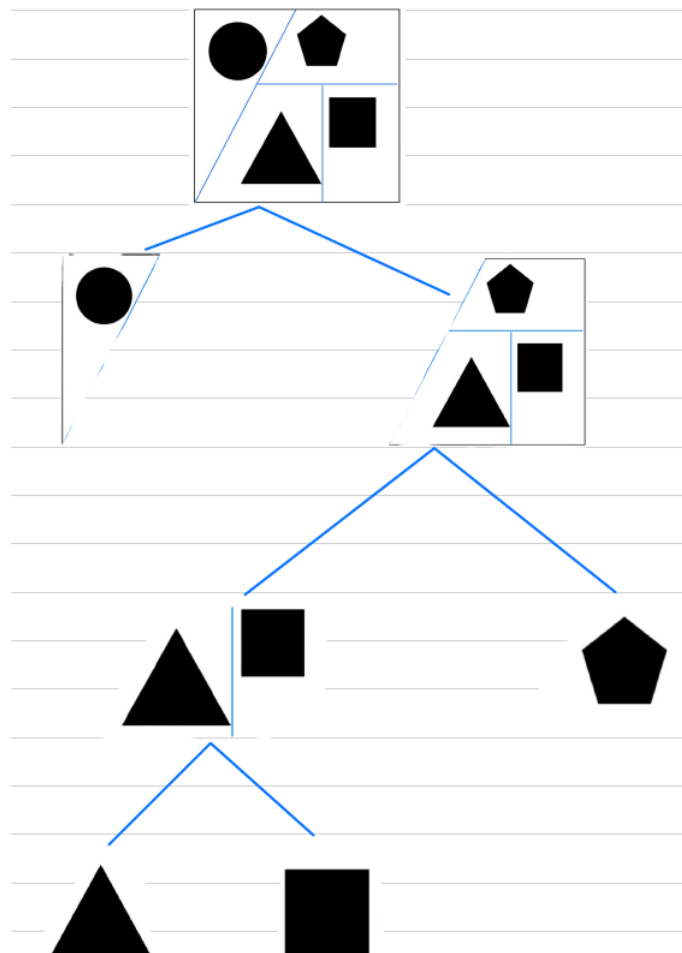
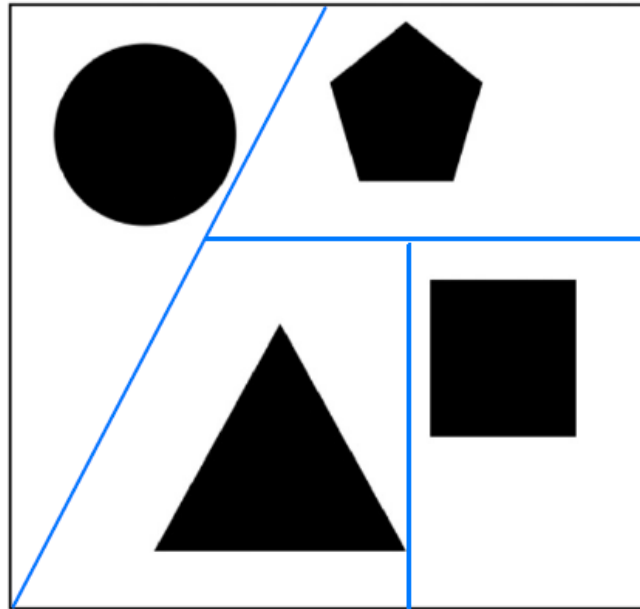
Lösungsvorschlag:

* Quadtree:



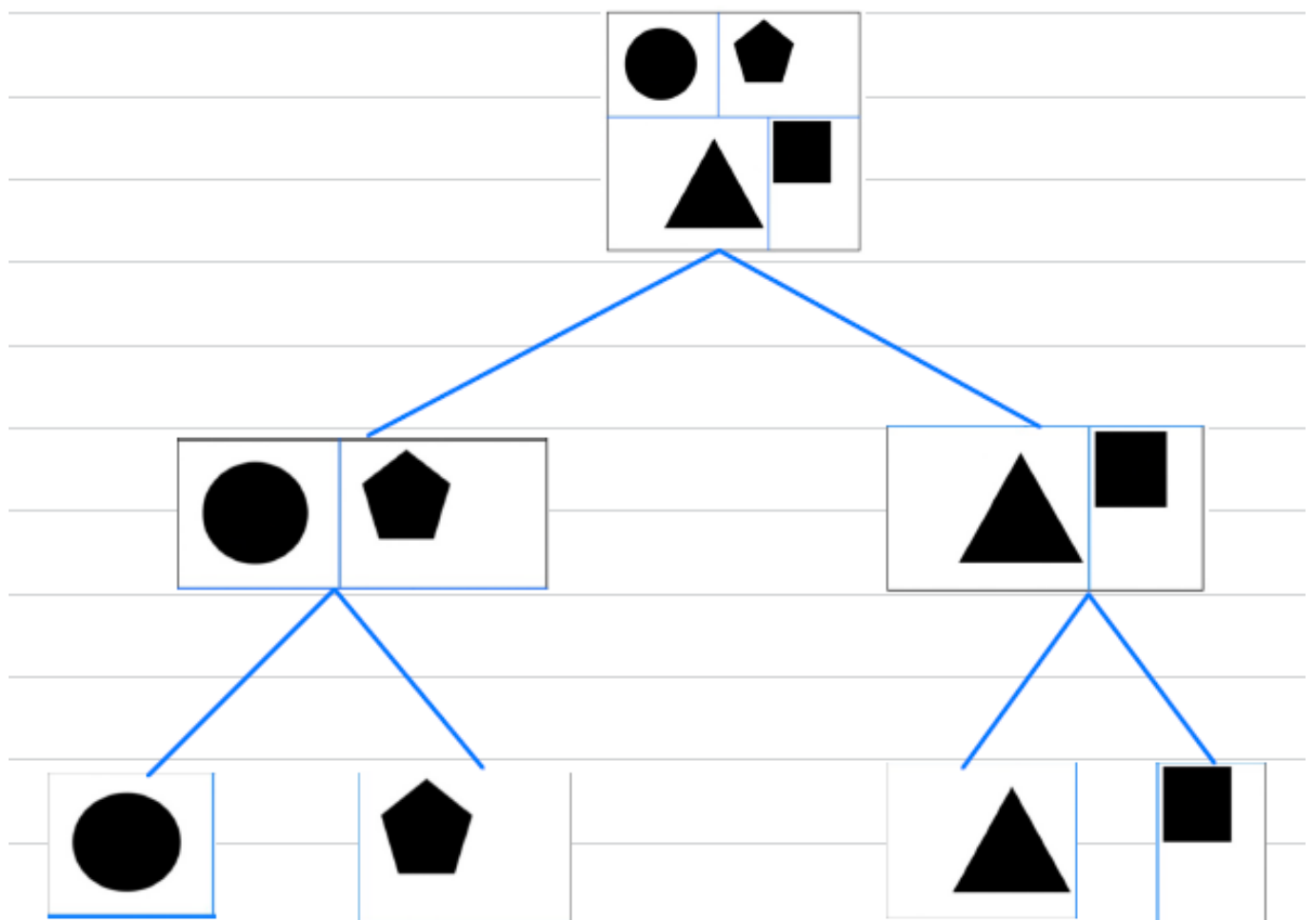
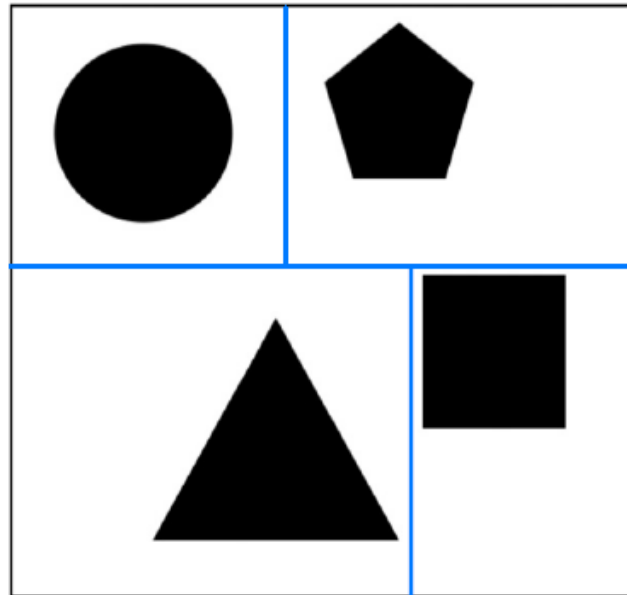
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

* BSP-Tree:



Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

* kd-Tree:



Visual Computing Uebung 6

Group 60:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

1b) (0,5 Punkte)

Ist jeder k-d-Tree ein BSP-Tree? Erläutern Sie.

Lösungsvorschlag:

Ja.
BSP Bäume ist rekursive Unterteilung des Raums in zwei Halbräume durch Verwendung beliebiger Unterteilungsebenen. Deswegen ist BSP-Tree eigentlich eines verallgemeinerte kd-tree.
Man kann auch verstehen, dass kd-tree ein "axis aligned BSP trees" bedeutet.

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

Aufgabe 2: Rasterisierung (3 Punkte)

Berechnen Sie mit dem Bresenham-Algorithmus die Rasterisierung folgender Linie:

$$\begin{array}{ll} x_{\text{start}} = 1 & x_{\text{end}} = 7 \\ y_{\text{start}} = 3 & y_{\text{end}} = 8 \end{array}$$

Geben Sie alle Zwischenschritte an und zeichnen Sie das resultierende Raster.

Lösungsvorschlag:

Iteration	x	y	Fehler vor IF	Fehler nach IF
Start	1	3	3	3
1	2	4	-2	4
2	3	5	-1	5
3	4	5	0	0
4	5	6	-5	1
5	6	7	-4	2
6	7	8	-3	3

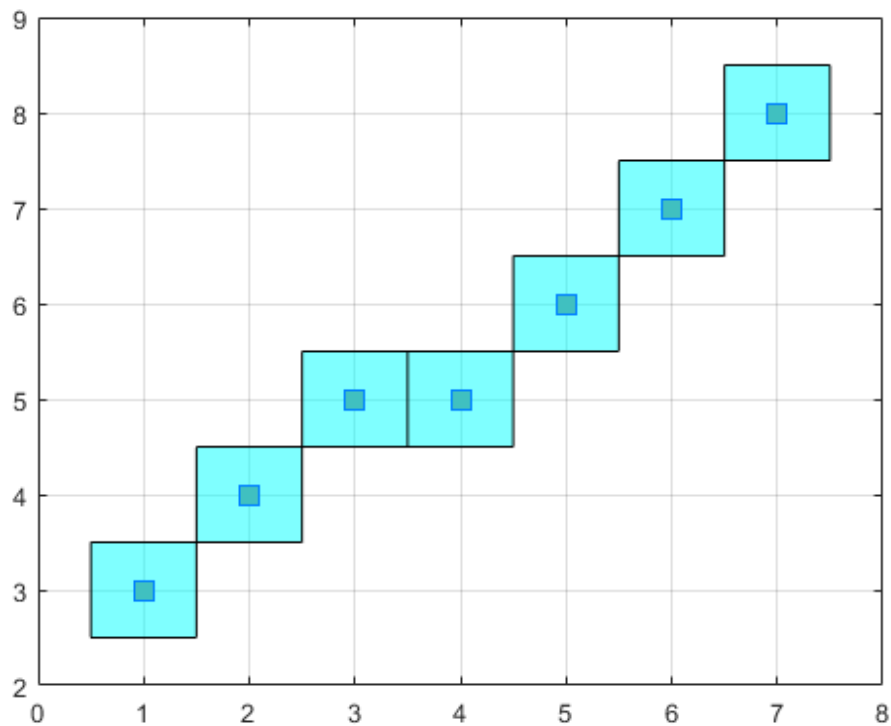


Abbildung 1: resultierende Raster

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

Aufgabe 3 (3 Punkte)**3a) (1 Punkt)**

Oftmals ist die Bewegungsfreiheit in VR-Applikationen durch die Größe des jeweiligen Aufenthaltsortes, wie beispielsweise ein Zimmer, eingeschränkt. Recherchieren Sie eine Hardwarelösung für dieses Problem und erklären Sie diese kurz.

Lösungsvorschlag:

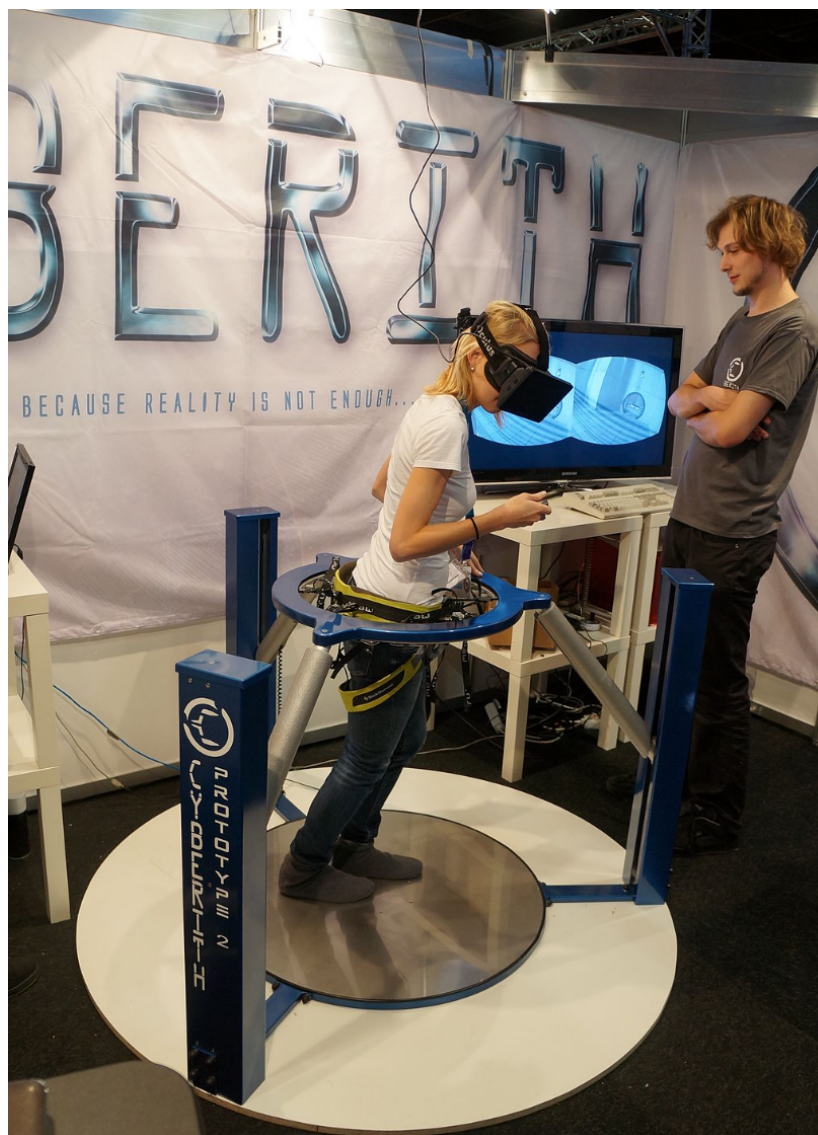


Abbildung 2: Omnidirektionale Laufband [1]

Das omnidirektionale Laufband kann die Begrenzung von Bewegungsfreiheit überwinden, mit welchem das Gehen im virtuellen Raum durch reale Gehbewegungen gesteuert wird. In diesem Fall ist eines große Zimmer zur Bewegung nicht mehr nötig.

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

3b) (1,5 Punkte)

VR-Sickness ist ein weiteres großes Hindernis in der Virtual Reality. Was ist VR-Sickness und was kann dagegen helfen? Nennen Sie mindestens zwei Maßnahmen.

Lösungsvorschlag:

* VR-Sickness:

Virtual-Reality-Krankheit tritt auf, wenn die Exposition gegenüber einer virtuellen Umgebung Symptome hervorruft, die den Symptomen der Reisekrankheit ähnlich sind. [2]

Die häufigsten Symptome sind allgemeines Unbehagen, Kopfschmerzen, Magenbewusstsein, Übelkeit, Erbrechen, Blässe, Schwitzen, Müdigkeit, Schläfrigkeit, Orientierungslosigkeit und Apathie. Andere Symptome sind Haltungsinstabilität und Würgen. [3]

Die Krankheit der virtuellen Realität unterscheidet sich von der Reisekrankheit darin, dass sie durch die visuell induzierte Wahrnehmung von Selbstbewegung verursacht werden kann. echte Selbstbewegung ist nicht erforderlich. [3]

Es unterscheidet sich auch von der Simulatorkrankheit; Nicht-Virtual-Reality-Simulator-Krankheit ist tendenziell durch okulomotorische Störungen gekennzeichnet, während Virtual-Reality-Krankheit tendenziell durch Orientierungslosigkeit gekennzeichnet ist. [4]

* Maßnahmen:

- 1 Nach mehreren Studien kann die Einführung eines statischen Referenzrahmens (unabhängiger visueller Hintergrund) die Simulationskrankheit verringern. [5, 6, 7]
- 2 Andere Techniken zur Verringerung von Übelkeit umfassen die Simulation von Verschiebungsarten, die keine Diskrepanzen zwischen den visuellen Aspekten und der Körperbewegung erzeugen oder verringern, wie z. B. das Reduzieren von Rotationsbewegungen während der Navigation [8], das dynamische Reduzieren des Sichtfelds [9], der Teleportation [10] und der Bewegung in der Schwerelosigkeit. [11]

Visual Computing Uebung 6	Group 60:	Vorname Name Matrikel-Nr.
		Yi Cui 2758172
		Yuting Li 2547040
		Xiaoyu Wang 2661201
		Ruiyong Pi 2309738

3c) (1 Punkt)

Nennen Sie zwei Ansätze, wie aus realen Objekten 3D-Modelle generiert werden können.

Lösungsvorschlag:

Ansätze:

- 1 Community basierter Ansatz : Bilddaten und das soziale Web
- 2 Multi View Stereo for Community Photo Collections

Literatur

- [1] Wikipedia, [Virtuelle Realität](#)
- [2] LaViola, J. J. Jr (2000), "A discussion of cybersickness in virtual environments". ACM SIGCHI Bulletin. 32: 47–56.
- [3] Kolasinski, E. M. "SSimulator sickness in virtual environments (ARI 1027)", U.S. Army Research Institute for the Behavioral and Social Sciences. Retrieved 22 July 2014.
- [4] Stanney, K. M.; Kennedy, R. S.; Drexler, J. M. (1997). "Cybersickness is not simulator sickness". Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 41 (2): 1138–1142.
- [5] Lin, James Jeng-Weei; Abi-Rached, Habib; Kim, Do-Hoe; Parker, Donald E.; Furness, Thomas A. (2002-09-01). "Natural Independent Visual Background Reduced Simulator Sickness". Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 46 (26): 2124–2128.
- [6] Prothero, J. D.; Draper, M. H.; Furness, T. A.; Parker, D. E.; Wells, M. J. (March 1999). "The use of an independent visual background to reduce simulator side-effects". Aviation, Space, and Environmental Medicine. 70 (3 Pt 1): 277–283.
- [7] Duh, Henry Been-Lirn; Parker, Donald E.; Furness, Thomas A.; A, Henry B. L. Duh; B, Donald E. Parker; A, Thomas A. Furness (2001). Does a Peripheral Independent Visual Background Reduce Scene-Motion-Induced Balance Disturbance in an Immersive Environment?.
- [8] Kemeny, Andras; George, Paul; Mérienne, Frédéric; Colombet, Florent (2017-01-29). "New VR Navigation Techniques to Reduce Cybersickness". Electronic Imaging. 2017 (3): 48–53.
- [9] Fernandes, A. S.; Feiner, S. K. (March 2016). "Combating VR sickness through subtle dynamic field-of-view modification". 2016 IEEE Symposium on 3D User Interfaces (3DUI). pp. 201–210.
- [10] ARVI Games. "Combating VR Sickness: Debunking Myths And Learning What Really Works"
- [11] "How game designers find ways around VR motion sickness". The Verge. Retrieved 2017-10-11.

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

Folgende sind genutzte Code:

```

1  x_start = 1;
2  y_start = 3;
3  x_end = 7;
4  y_end = 8;
5
6  % rendering with Bresenham-Algorithmus
7  [rendering_x, rendering_y, error_vor_IF, error_nach_IF] = ...
8      Bresenham_line(x_start, y_start, x_end, y_end)
9
10 % draw pixel diagramm
11 draw_pixel(rendering_x, rendering_y)

1  function [x_array, y_array, error_before, error_after] = Bresenham_line(x_start, y_start, ...
2  x_end, y_end)
3  %Bresenham-Algorithmus
4  %   x_start start position in x direction
5  %   y_start start position in y direction
6  %   x_end end position in x direction
7  %   y_end end position in y direction
8  steep = abs(y_end - y_start)/abs(x_end - x_start) > 1;
9
10 % convert special case in general case
11 if steep
12     [x_start, y_start] = swap(x_start, y_start);
13     [x_end, y_end] = swap(x_end, y_end);
14 end
15 if x_start > x_end
16     [x_start, x_end] = swap(x_start, x_end);
17     [y_start, y_end] = swap(y_start, y_end);
18 end
19
20 % compute step length
21 delta_x = x_end - x_start;
22 delta_y = y_end - y_start;
23
24 % initial output
25 error_before = zeros(delta_x+1, 1);
26 error_after = zeros(delta_x+1, 1);
27 x_array = zeros(delta_x+1, 1);
28 y_array = zeros(delta_x+1, 1);
29
30 if y_end > y_start
31     y_steep = 1;
32 else
33     y_steep = -1;
34 end
35
36 % initial intermediate variables
37 error = delta_x / 2;
38 delta_error = - delta_y;
39 y_intermediate = y_start;
40 index = 1;
41
42 % assign first element in output array
43 x_array(index, 1) = x_start;
44 y_array(index, 1) = y_intermediate;
45 error_before(index, 1) = error;
46 error_after(index, 1) = error;
47

```

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Xiaoyu	Wang	2661201
Ruiyong	Pi	2309738

```
48 % iteration
49 for x = x_start+1 : 1 : x_end
50     % fast orientation
51     index = index + 1;
52     error = error + delta_error;
53     % assignment
54     x_array(index, 1) = x;
55     error_before(index, 1) = error;
56
57     if error < 0
58         % slow orientation
59         y_intermediate = y_intermediate + y_steep;
60         error = error + delta_x;
61     end
62     % assignment
63     y_array(index, 1) = y_intermediate;
64     error_after(index, 1) = error;
65 end
66
67 end

1 function [] = draw_pixel(randerings_x, randerings_y, size)
2 %DRAW_PIXEL draw randerings line in pixel form
3 % randerings_x, [n, 1] randerings positions in x orientation
4 % randerings_y, [n, 1] randerings positions in y orientation
5 % size: int. pixel size (default as 1)
6 % set default pixel
7 if nargin == 2
8     size = 1;
9 end
10 figure()
11 boundary_west = randerings_x - size/2;
12 boundary_east = randerings_x + size/2;
13 boundary_north = randerings_y + size/2;
14 boundary_south = randerings_y - size/2;
15
16 plot(randerings_x, randerings_y, 's', 'MarkerSize',10 , 'MarkerEdgeColor','b',...
17     'MarkerFaceColor',[0.5,0.5,0.5])
18 hold on
19 grid on
20 for i = 1:length(randerings_x)
21     x_boundary = [boundary_west(i, 1), boundary_west(i, 1),...
22         boundary_east(i, 1), boundary_east(i, 1)];
23     y_boundary = [boundary_south(i, 1), boundary_north(i, 1),...
24         boundary_north(i, 1), boundary_south(i, 1)];
25     fill(x_boundary, ...
26         y_boundary,...
27         'cyan', 'FaceAlpha', 0.5)
28 end
29 end
```