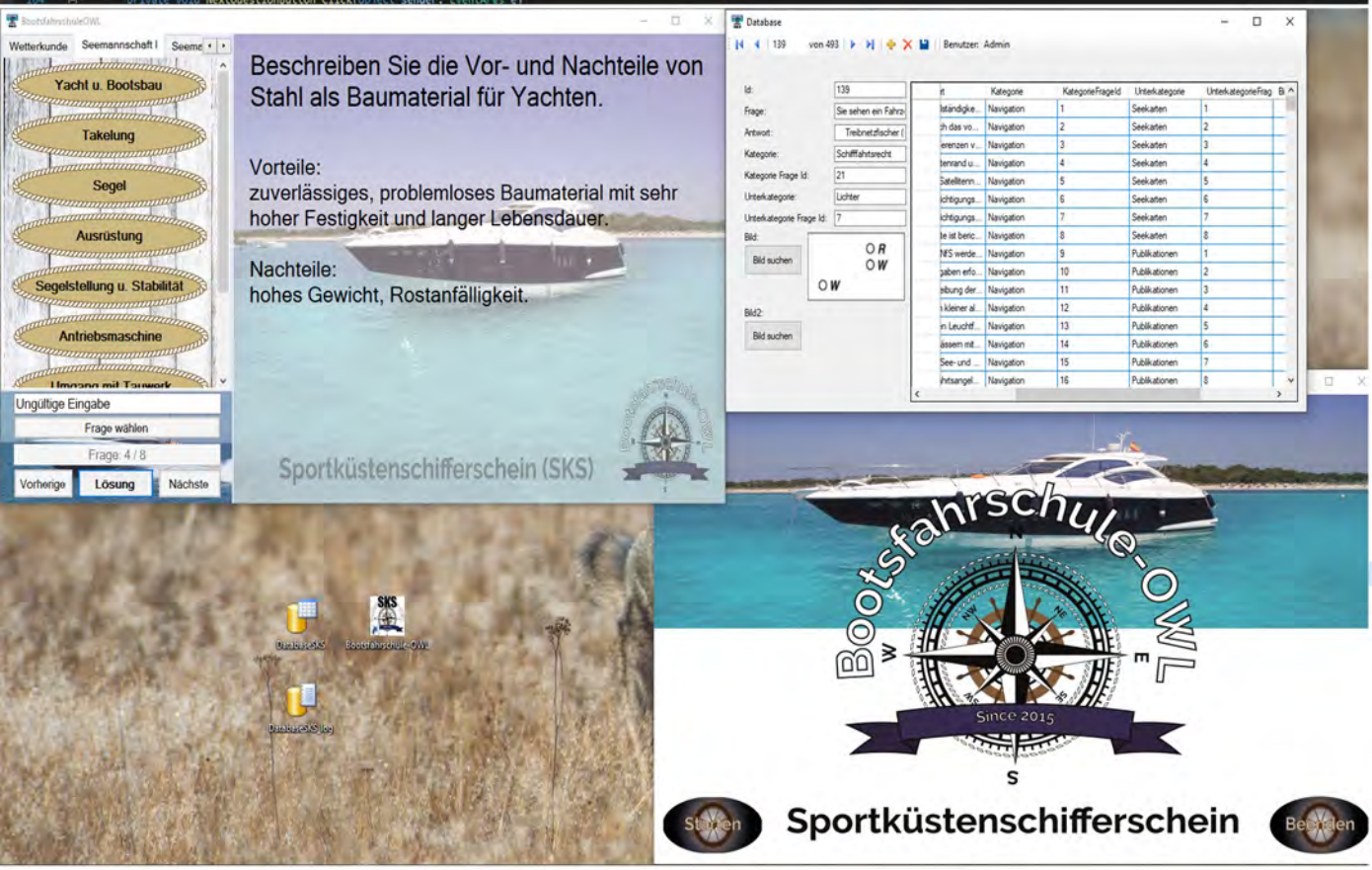
C# (Visual Studio) und SQL

Bootsfahrschule-OWL — Bootsfahrschule_OWL.SKS — SeellButton7_Click(object sender, EventArgs e)

```
124        private void SetTextLabels()
125        {
126            FragenBild.Visible = false;
127            AntwortBild.Visible = false;
128            AntwortLabel.Visible = false;
129            hasImage = false;
130            setFragenImage();
131            setAntwortImage();
132            SqlConnection connection = new SqlConnection(connectionString);
133            connection.Open();
134
135            SqlCommand cmd = new SqlCommand("Select Frage, Antwort, UnterkategorieFrageId from SksData where UnterkategorieFrageId = @Id AND Unterkategorie =@Kat", connection);
136            cmd.Parameters.AddWithValue("@Id", KatFragID);
137            cmd.Parameters.AddWithValue("@Kat", Unterkategorie);
138            SqlDataReader da = cmd.ExecuteReader();
139            while (da.Read())
140            {
141                FragenLabel.Text = da.GetValue(0).ToString();
142                AntwortLabel.Text = da.GetValue(1).ToString();
143
144                CurrentQueTxt.Text = ("Frage: " + da.GetValue(2).ToString() + " / " + AnzFragUnterkat);
145            }
146
147            connection.Close();
148
149            if (FragenBild.Visible == true)
150            {
151                FragenBild.Location = new Point(QuestionPane.Size.Width / 2 - FragenBild.Size.Width / 2, FragenLabel.Location.Y + FragenLabel.Size.Height + 10);
152                AntwortLabel.Location = new Point(FragenLabel.Location.X, FragenLabel.Location.Y + FragenBild.Size.Height + FragenLabel.Size.Height + 30);
153                AntwortBild.Location = new Point(QuestionPane.Size.Width / 2 - FragenBild.Size.Width / 2, FragenLabel.Location.Y + FragenLabel.Size.Height + FragenBild.Size.Height + AntwortLabel.Size.Height + 30)
154            }
155            else
156            {
157                AntwortLabel.Location = new Point(FragenLabel.Location.X, FragenLabel.Location.Y + FragenLabel.Size.Height + 30);
158                AntwortBild.Location = new Point(QuestionPane.Size.Width / 2 - FragenBild.Size.Width / 2, FragenLabel.Location.Y + FragenLabel.Size.Height + AntwortLabel.Size.Height + 30);
159
160            }
161        }
162
163
164        1-Verweis
           private void NextQuestionButton_Click(object sender, EventArgs e)
```



Arbeitsproben

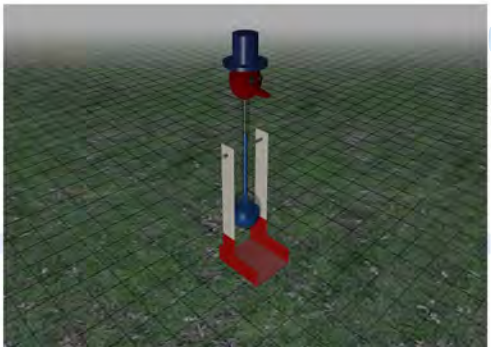
C# (Visual Studio) und SQL

Bootsfahrschule-OWL · Bootsfahrschule_OWL.SKS · SetTextLabels()

```
//Count the numbers of Questions in the Subcategories
44 Verweise
private void CountQuestions()
{
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();

    SqlCommand cmd = new SqlCommand("Select Count(*) from SksData where Unterkategorie = @Kat", connection);
    cmd.Parameters.AddWithValue("@Kat", Unterkategorie);
    SqlDataReader da = cmd.ExecuteReader();

    while (da.Read())
    {
        AnzFragUnterkat = (int)da.GetValue(0);

    }
    connection.Close();
}


1 Verweis
private void getAntwortImage()
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand cmd = new SqlCommand("Select Bild2 from SksData where UnterkategorieFrageId = @Id AND Unterkategorie =@Kat AND Bild2 Is NOT NULL", connection))
        {
            cmd.Parameters.AddWithValue("@Id", KatFragID);
            cmd.Parameters.AddWithValue("@Kat", Unterkategorie);
            connection.Open();
            byte[] bytes2 = (byte[])cmd.ExecuteScalar();
            connection.Close();
            if (bytes2 != null)
            {
                hasImage = true;
                AntwortBild.Image = Image.FromStream(new MemoryStream(bytes2));
            }
        }
    }
}
```

WebGL Drinking Bird Basic Work

```
function fillScene() {
    scene = new THREE.Scene();
    scene.fog = new THREE.Fog( 0xAAAAAA, 3000, 5000 );

    // LIGHTS
    scene.add( new THREE.AmbientLight( 0x222222 ) );
    var light = new THREE.DirectionalLight( 0xFFFFFF, 0.7 );
    light.position.set( 200, 500, 500 );
    scene.add( light );
    light = new THREE.DirectionalLight( 0xFFFFFF, 0.9 );
    light.position.set( -200, -100, -400 );
    scene.add( light );

    // GROUND
    var texture = new THREE.ImageUtils.loadTexture("media/img/cs291/textures/grass512x512.jpg");
    texture.wrapS = texture.wrapT = THREE.RepeatWrapping;
    texture.repeat.set(10,10);
    var solidGround = new THREE.Mesh(
        new THREE.PlaneGeometry( 10000, 10000, 100, 100 ),
        new THREE.MeshPhongMaterial( { map : texture} ) );
    solidGround.rotation.x = - Math.PI / 2;

    scene.add( solidGround );

    // uncomment to see grid on ground

    // put grid lines every 10000/100 = 100 units
    var ground = new THREE.Mesh(
        new THREE.PlaneGeometry( 10000, 10000, 100, 100 ),
        new THREE.MeshBasicMaterial( { color: 0x0, wireframe: true } ) );
    ground.rotation.x = - Math.PI / 2;
    // cheat: offset by a small amount so grid is on top
    ground.position.y = 0.2;
    scene.add( ground );

    // Bird
    var bird = new THREE.Object3D();
    createDrinkingBird( bird );
    scene.add( bird );
}

function createSupport( bsupport ) {
    var legMaterial = new THREE.MeshPhongMaterial( { shininess: 4 } );
    legMaterial.color.setHex( 0xb4b7b9 );
```

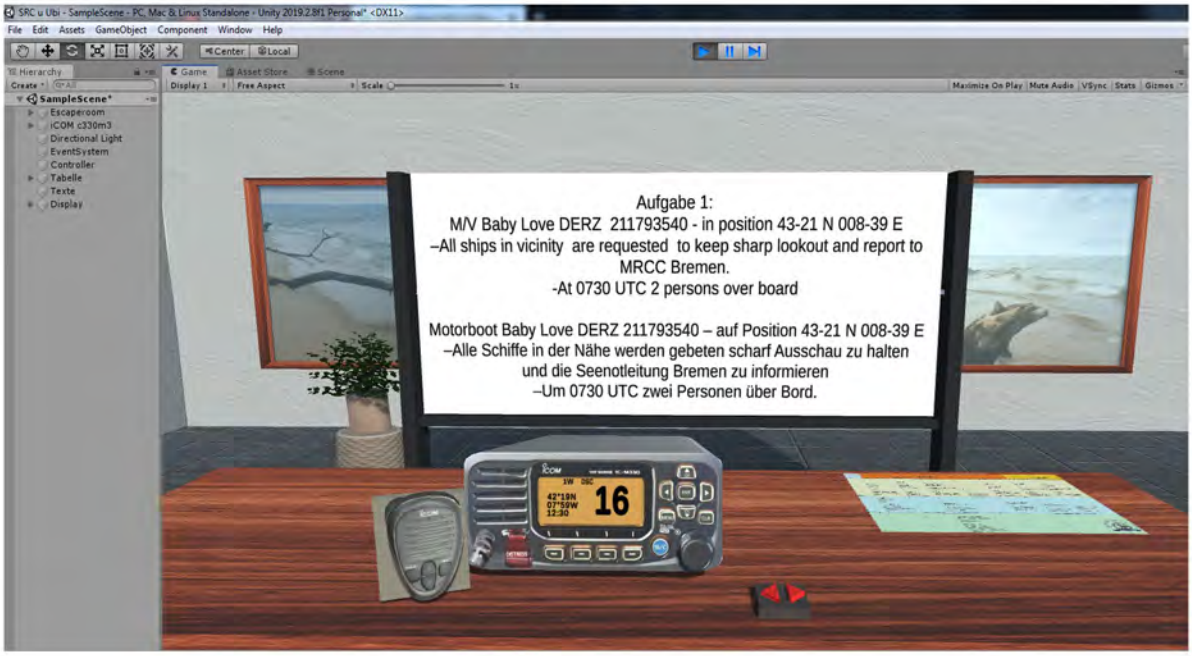


Arbeitsproben

Unity 3D

Animation von Objekten

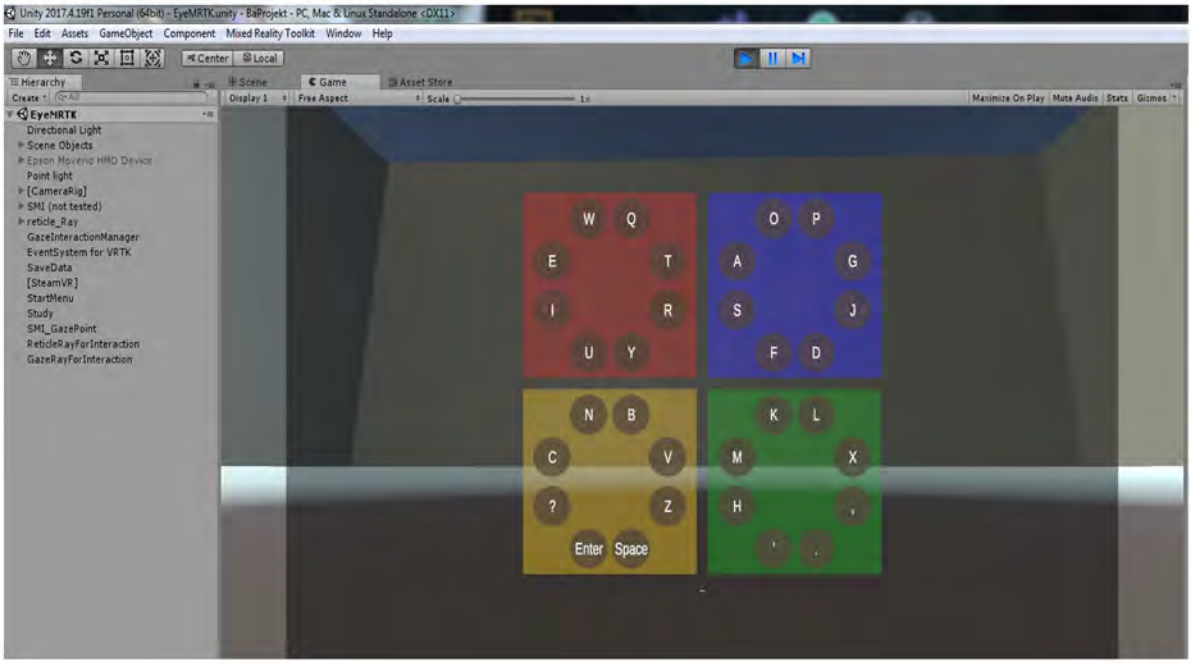
Mausklicks verändern Umgebung und Oberflächen



Arbeitsproben

C#

Tasteneingabe Layout für den SMI Eyetracker. Layout bewegt sich mit dem HMD mit und bleibt so vor der Kamera positioniert



Ordnet alle Buchtaben einem Gameobject Array zu.

```
//Write all Letters in an Array AllLetter[i], i = [0...31] by saving the children of the Layer [j] , j =[0...3]

for (int j = 0; j < keyBoardAlpha.transform.childCount; j++) {
        if (keyBoardAlpha.transform.GetChild (j).transform.tag == "Layers") {
            Layers [j] = keyBoardAlpha.transform.GetChild (j).transform.gameObject;
            keyboardKeys = new GameObject[Layers [j].transform.childCount];
            for (int i = 0; i < Layers [j].transform.childCount; i++) {
                if (Layers [j].transform.GetChild (i).transform.tag == "KeyboardKey") {

                    keyboardKeys [i] = Layers [j].transform.GetChild (i).transform.gameObject;

                AllLetter[lettercount] = keyboardKeys[i];
                lettercount++;
                }
                }

        }
    }

}
```

Arbeitsproben

C#

Allerlei Berechnungen zwischen Object und Augendaten. Der binocular POR stellt dabei eine gemittelte Berechnung zwischen den Richtungsvektoren der Augen dar.

```
        //erweitert die aktuelle Zeit um die Systemzeit
        time += Time.deltaTime;


            //Calculate Eye Vektor between actual Position and OldPosition
            EyeXPos = binocularPor.x;
            EyeYPos = binocularPor.y;

            //Debug.Log ("LetterPos [x;y] " + LetterxPosFOV + "," + LetteryPosFOV + "EyeXPos " + EyeXPos + "," + EyeYPos);

            //Richtungsvektor des Eye berechnen
            EyeDirVecX = EyeXPos - EyeOldPosX;
            EyeDirVecY = EyeYPos - EyeOldPosY;


            //Richtungsvektor des Buchstaben
            LetterDirVecX = LetterxPosFOV - LetterOldPosX;
            LetterDirVecY = LetteryPosFOV - LetterOldPosY;

            //Richtungsvektoren in Vecotrschreibweise
            //Eye
            RichtungsVecEye = new Vector2 (EyeDirVecX, EyeDirVecY);
            //Letter
            RichtungsVecLetter = new Vector2 (LetterDirVecX, LetterDirVecY);

            //Distanz zwischen alter und neuer Position (Länge des Richtugnsvektors)
            EyeDistance = Vector2.Distance (new Vector2 (EyeXPos, EyeYPos), new Vector2 (EyeOldPosX, EyeOldPosY));
            LetterDistance = Vector2.Distance (new Vector2 (LetterxPosFOV, LetteryPosFOV), new Vector2 (LetterOldPosX, LetterOldPosY));


            //normierter Richtungsvektor Letter und Eye
            NorRichtungsVecLetter = (1 / LetterDistance) * RichtungsVecLetter;
            NorRichtungsVecEye = (1 / EyeDistance) * RichtungsVecEye;


            //calculate speed of the movement of the letters with its distance and needed time
            EyeSpeed = (EyeDistance * (time-oldtime));
            LetterRealspeed = (LetterDistance * (time-oldtime));


            //Geschwindigkeitsabgleich zwischen Auge und Buchstabe
            SpeedDivision = (LetterRealspeed / EyeSpeed);

            //Winkelberechnung des Winkels zwischen Richtungsvektor Buchstabe und Richtungsvektor Auge
            DirAngelLetterToGazeVec = Vector2.Angle (RichtungsVecLetter, RichtungsVecEye);
            NorDirAngelLetterToGazeVec = Vector2.Angle (NorRichtungsVecLetter, NorRichtungsVecEye);

            //Abstand zwischen Buchstabe und Augen Anhand der POR und FOV Position
            DistanceEyeAndLetterPOR = Vector2.Distance (new Vector2 (EyeXPos, EyeYPos), new Vector2 (LetterxPosFOV, LetteryPosFOV));

        //Berechnung der Match-Werte
            calculateMatchValue(SpeedDivision, angleCameraLetter, DirAngelLetterToGazeVec);
            calculateMatchValueGewichtet (SpeedDivision, angleCameraLetter, DirAngelLetterToGazeVec);
            calculateMatchValueGewichtet2 (SpeedDivision, angleCameraLetter, DirAngelLetterToGazeVec);
```

Funktion zur Bestimmung der Oberfläche, welche am wahrscheinlichsten angeschaut wird.

```
/*function to find the Layer with the minimal match Value plus save the index of the Layer
*with minimal Match Value and increments the Matchcounter of the Layer with this index
*/
public void findLayerMatch(int Layercount){
    float minValue = 1000;
    int counter = Layercount;

    for (int k = 0; k < 4; k++) {
        float actualLayerMatchValue = (float)LayerMatchValueGewichtet.GetValue (k);
        if (minValue > actualLayerMatchValue) {
            minValue = (float)LayerMatchValueGewichtet.GetValue (k);
            minIndex = k;
        }
    }
    //increase the MatchCounter of the layer with the minimum Match Value
    Layers[minIndex].GetComponent<LayerValues>().increaseMatchCounterGewichtet();

    }
```

Arbeitsprobe

C#

Animation eines Objectes via Positions und Rotationsmanipulation.

Assembly-CSharp | AnimateChart | Update()

```
31          offsety = this.transform.localPosition.y;
32          offsetz = this.transform.localPosition.z;
33          offsetRx = this.transform.localRotation.x;
34          offsetRy = -90;
35          offsetRz = this.transform.localRotation.z;
36      }
37
38      // Update is called once per frame
39      void Update()
40      {
41          //animation of the charts
42          /*
43           * Animation direction to the camera Object
44           * Use the Cos and Sin for calculating the diretion
45           * animations stops at endposition and reverse when a button is pushed
46           */
47          if ((animate == true) && (time < maxtimer))
48          {
49              time += Time.deltaTime;
50              this.transform.localPosition = new Vector3(offsetx + (-(Mathf.Cos(1 * Mathf.PI / 4)) * time * speed * 2.8f), offsety + (Mathf.Sin(1 * Mathf.PI / 4) * time * speed * 1.25f), offsetz + (1 * tir
51              this.transform.eulerAngles = new Vector3(offsetRx + (90 * time), -90, offsetRz);
52              actPosX = offsetRx + (90 * time);
53
54          }
55
56          /*
57          * Animation direction back to the table
58          * To reverse the direction the timer runs backwards
59          * animations stops at endposition and reverse when a button is pushed
60          */
61          if ((animate == false) && (time > 0))
62          {
63              time -= Time.deltaTime;
64              this.transform.localPosition = new Vector3(offsetx + (-(Mathf.Cos(1 * Mathf.PI / 4)) * time * speed * 2.8f), offsety + (Mathf.Sin(1 * Mathf.PI / 4) * time * speed * 1.25f), offsetz + (1 * tir
65              this.transform.eulerAngles = new Vector3(0 + (90 * time), -90, offsetRz);
66
67          }
68
```

Textmanipulation des Controller Displays der Funkgeräte Simulation.

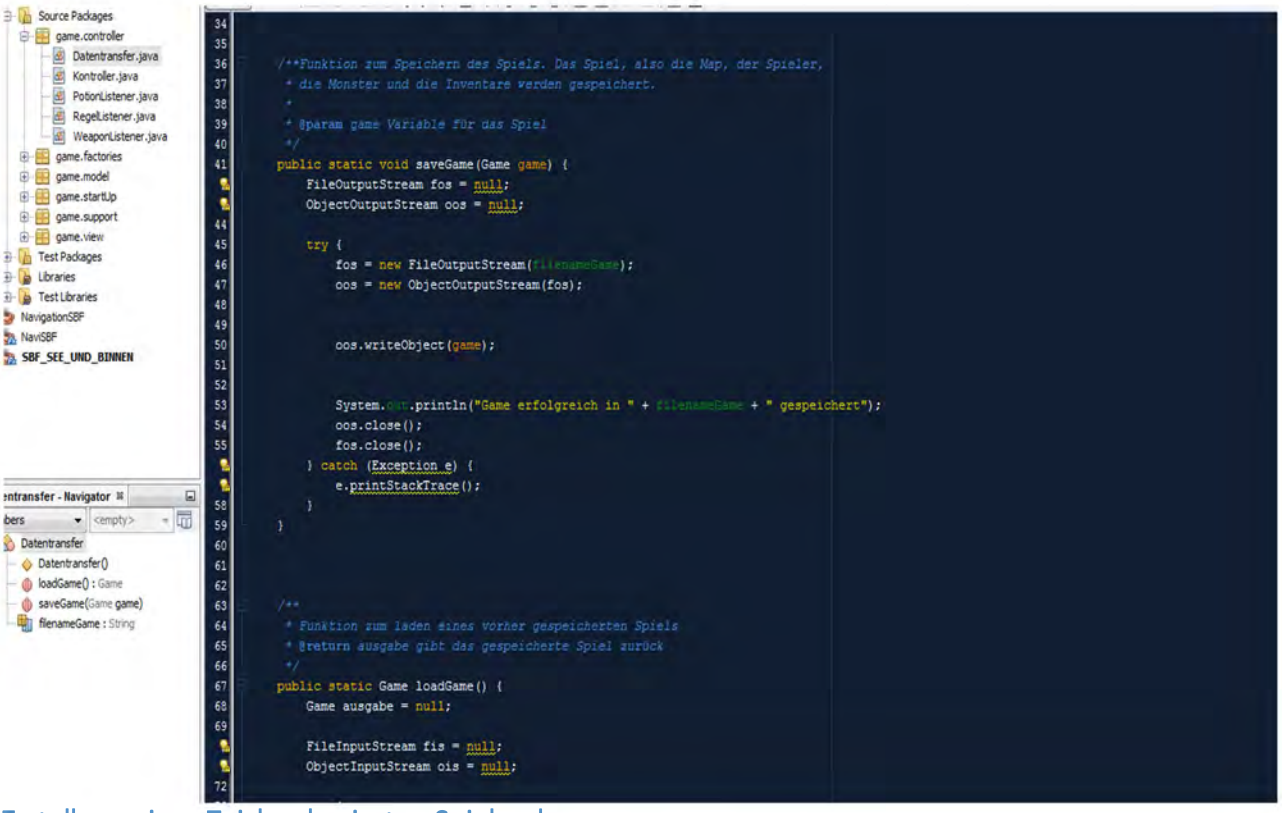
Assembly-CSharp | Controller | Update()

```
54          loesText = Solutions.transform.GetChild(0).GetComponent<TextMeshProUGUI>();
55
56          //Blackboard text became welcome message
57          textpro.text = Texte.GetComponent<texte>().getWelcome();
58
59          //Disable Solution Layout
60          Solutions.SetActive(false);
61
62
63      }
64
65      // Update is called once per frame
66      void Update()
67      {
68          if (Input.GetKeyDown(KeyCode.L) && LoesLayoutActive == false)
69          {
70              Solutions.SetActive(true);
71              LoesLayoutActive = true;
72          }
73          else if (Input.GetKeyDown(KeyCode.L) && LoesLayoutActive == true)
74          {
75              Solutions.SetActive(false);
76              LoesLayoutActive = false;
77          }
78
79          if (Input.GetKeyDown(KeyCode.LeftArrow)) {
80              PreviousText();
81          }
82          if (Input.GetKeyDown(KeyCode.RightArrow)) {
83              NextText();
84          }
85      }
86
87      //function to change the channel text to CH 16 on the Controller display
88      public void ChangeChannel(string name)
89      {
90          string currentButton = name;
91
92          if (currentButton == "16/C" || currentButton == "Micro16/C")
93          {
94              DisplayTexts[0].text = "16";
95          }
96      }
97
98      //function to change the transmition power btw. 1W and 25W and change the Controller Display
99      public void ChangeWatt(string name)
100     {
101         string currentButton = name;
102
103         if (currentButton == "MicroHiLo" || currentButton == "func4")
104         {
105             if (DisplayTexts[1].text == "25W")
106             {
107                 DisplayTexts[1].text = "1W";
108             }
109             else if (DisplayTexts[1].text == "1W")
```

Arbeitsproben

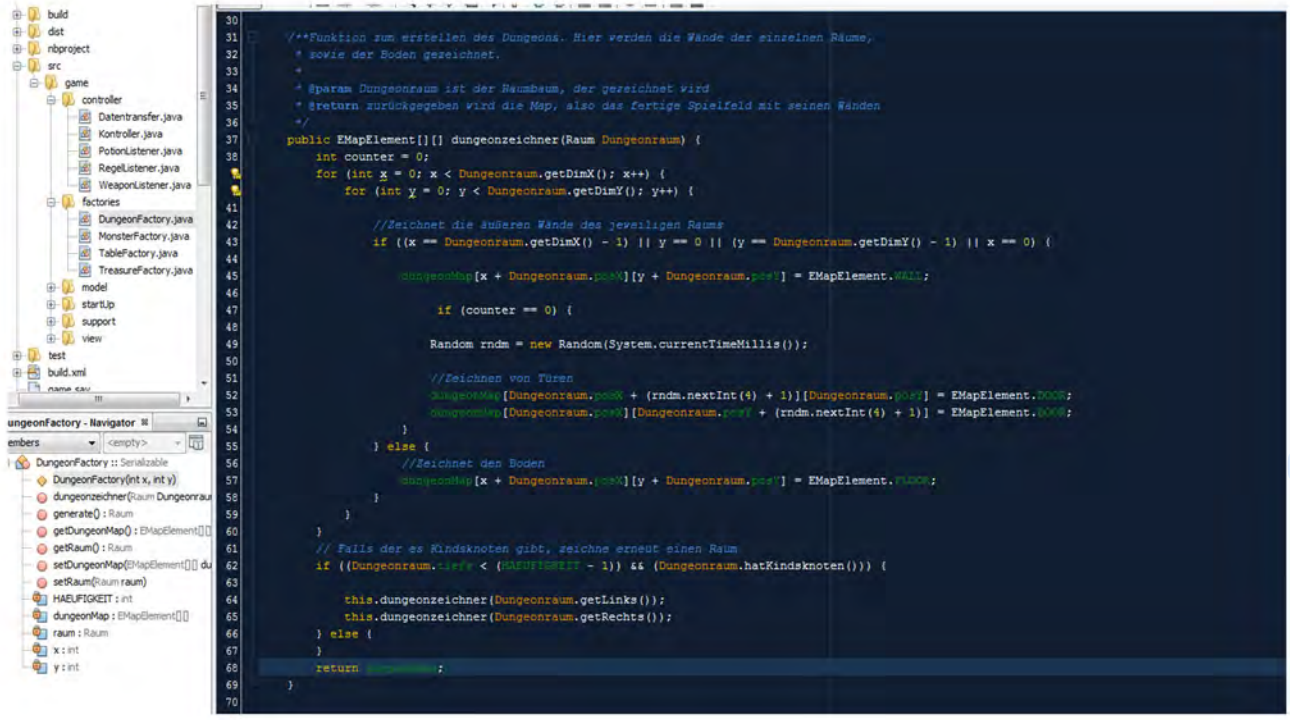
Java

Speicherung des Spielstandes



```
/**Funktion zum Speichern des Spiels. Das Spiel, also die Map, der Spieler,
 * die Monster und die Inventare werden gespeichert.
 *
 * @param game Variable für das Spiel
 */
public static void saveGame(Game game) {
    FileOutputStream fos = null;
    ObjectOutputStream oos = null;

    try {
        fos = new FileOutputStream(filenameGame);
        oos = new ObjectOutputStream(fos);


        oos.writeObject(game);


        System.out.println("Game erfolgreich in " + filenameGame + " gespeichert");
        oos.close();
        fos.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}



/**
 * Funktion zum laden eines vorher gespeicherten Spiels
 * @return ausgabe gibt das gespeicherte Spiel zurück
 */
public static Game loadGame() {
    Game ausgabe = null;

    FileInputStream fis = null;
    ObjectInputStream ois = null;
```

Erstellung eines Zeichenbasierten Spiele- dungeons.



```
/**Funktion zum erstellen des Dungeons. Hier werden die Wände der einzelnen Räume,
 * sowie der Boden gezeichnet.
 *
 * @param Dungeonraum ist der Raumbaum, der gezeichnet wird
 * @return zurückgegeben wird die Map, also das fertige Spielfeld mit seinen Wänden
 */
public EMapElement[][] dungeonzeichner(Raum Dungeonraum) {
    int counter = 0;
    for (int x = 0; x < Dungeonraum.getDimX(); x++) {
        for (int y = 0; y < Dungeonraum.getDimY(); y++) {

            //Zeichnet die äußeren Wände des jeweiligen Raums
            if ((x == Dungeonraum.getDimX() - 1) || y == 0 || (y == Dungeonraum.getDimY() - 1) || x == 0) {

                dungeonMap[x + Dungeonraum.posX][y + Dungeonraum.posY] = EMapElement.WALL;

                    if (counter == 0) {

                    Random rndm = new Random(System.currentTimeMillis());

                    //Zeichnen von Türen
                    dungeonMap[Dungeonraum.posX + (rndm.nextInt(4) + 1)][Dungeonraum.posY] = EMapElement.DOOR;
                    dungeonMap[Dungeonraum.posX][Dungeonraum.posY + (rndm.nextInt(4) + 1)] = EMapElement.DOOR;
                }
            } else {
                //Zeichnet den Boden
                dungeonMap[x + Dungeonraum.posX][y + Dungeonraum.posY] = EMapElement.FLOOR;
            }
        }
    }
    // Falls der es Kindsknoten gibt, zeichne erneut einen Raum
    if ((Dungeonraum.tiefe < (HAEUFIGKEIT - 1)) && (Dungeonraum.hatKindsknoten())) {

        this.dungeonzeichner(Dungeonraum.getLinks());
        this.dungeonzeichner(Dungeonraum.getRechts());
    } else {
    }
    return dungeonMap;
}
```

Arbeitsproben

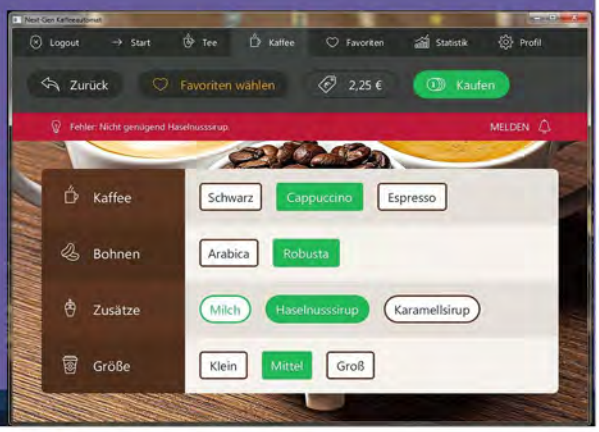
Java

Rendern des Spieldungeons mit verschiedenen Zeichen.

```java
    /**
     * Funktion zur Darstellung der Map auf dem JPanel.
     * @param map Die darzustellende Map.
     */
    public void render(EMapElement[][] map) {
        String content = "";
        EMapElement element;
        pane.setText(content);


            for(int x = 0; x < DIM_X; x++) {
            for(int y = 0; y < DIM_Y; y++) {

                element = map[x][y];
                switch(element) {

                    // Boden hinzufügen
                    case FLOOR:
                        content += " ";
                        break;

                    // Tür hinzufügen
                    case DOOR:
                        content += " ";
                        break;

                    // Kind hinzufügen
                    case CHILD:
                        content += "C";
                        break;

                    // Mann und Frau hinzufügen
                    case MAN_AND_WOMAN:
                        content += "M";
                        break;

                    // Oma und Opa hinzufügen
                    case GRANDMOTHER_AND_GRANDFATHER:
                        content += "O";
                        break;
```

Speichern einer Bestellliste. Ausschnitt aus einer HMI Schnittstelle. (Kaffeevollautomat)

```java
        // Return
        return priceString;
    }

    public final ArrayList<Integer> getOrder() {

        ArrayList<Integer> order = new ArrayList<>();

        order.add(Constants.[illegible]);
        if (this.[illegible] != 0) { order.add(this.[illegible]); }
        if (this.[illegible] != 0) { order.add(this.[illegible]); }
        if (this.[illegible] != 0) { order.add(this.[illegible]); }
        if (this.[illegible] != 0) { order.add(this.[illegible]); }

        return order;
    }
}
```

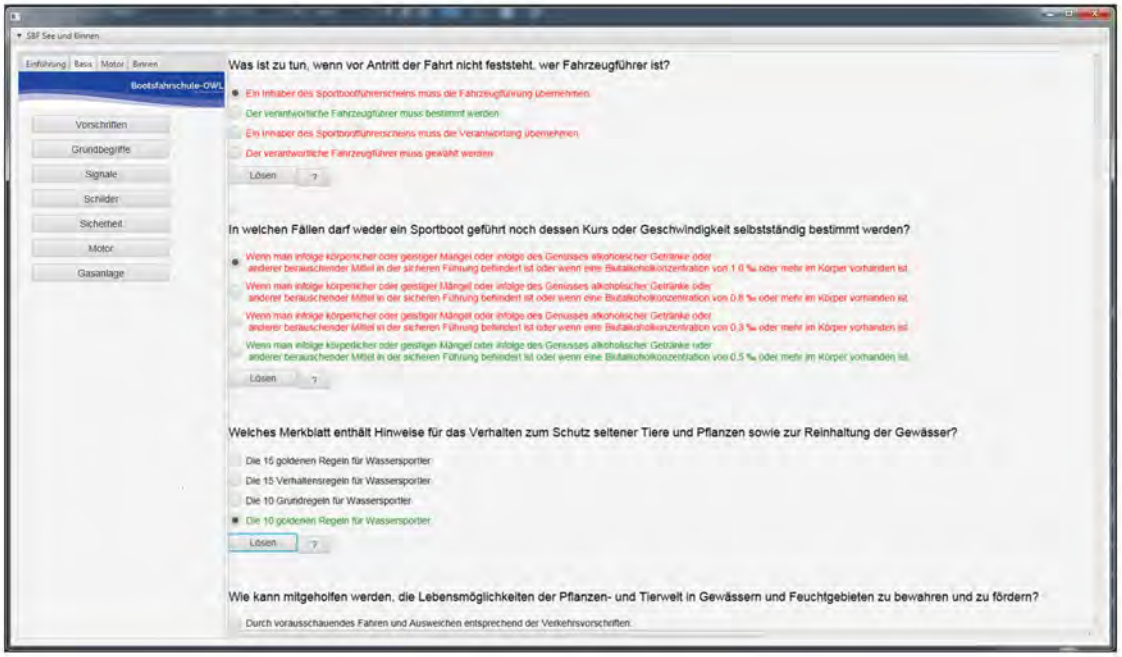


Arbeitsproben

Adobe Illustrator, Photoshop, Indesign

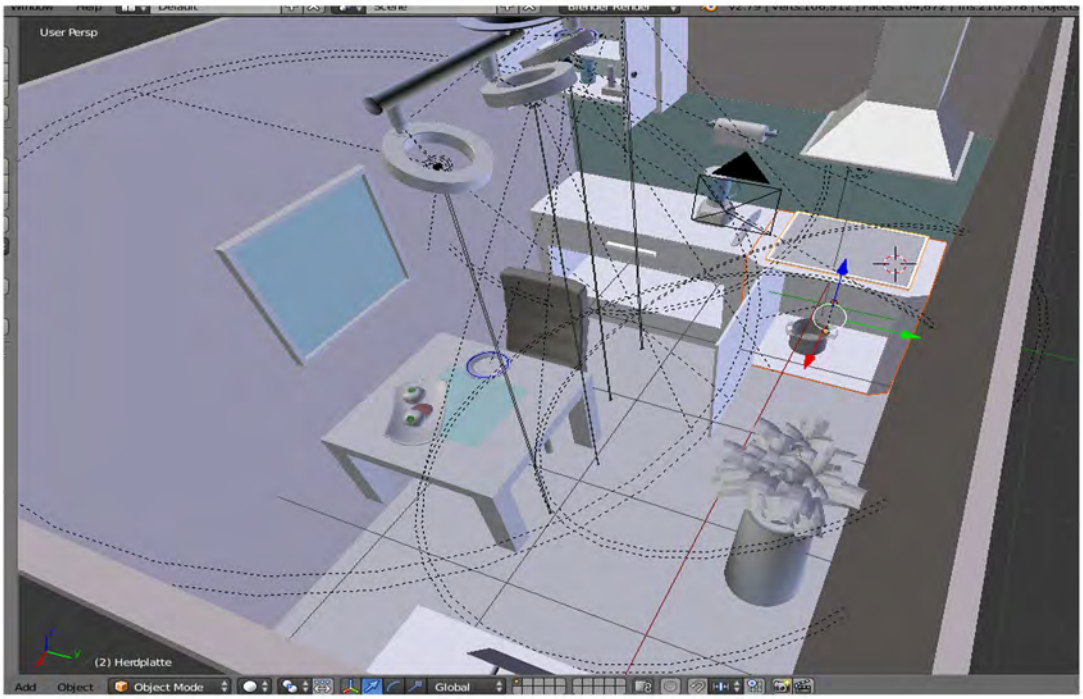
Erstellung des Briefbogens (InDesign) , Werbelogos (Illustrator) und Aktionsdesigns (Photoshop)

Ausschnitt der Lernsoftware (Java, JavaFx) mit Ausführbarer .exe und -jar Datei.



Arbeitsproben

Blender Modellierung und Rendering

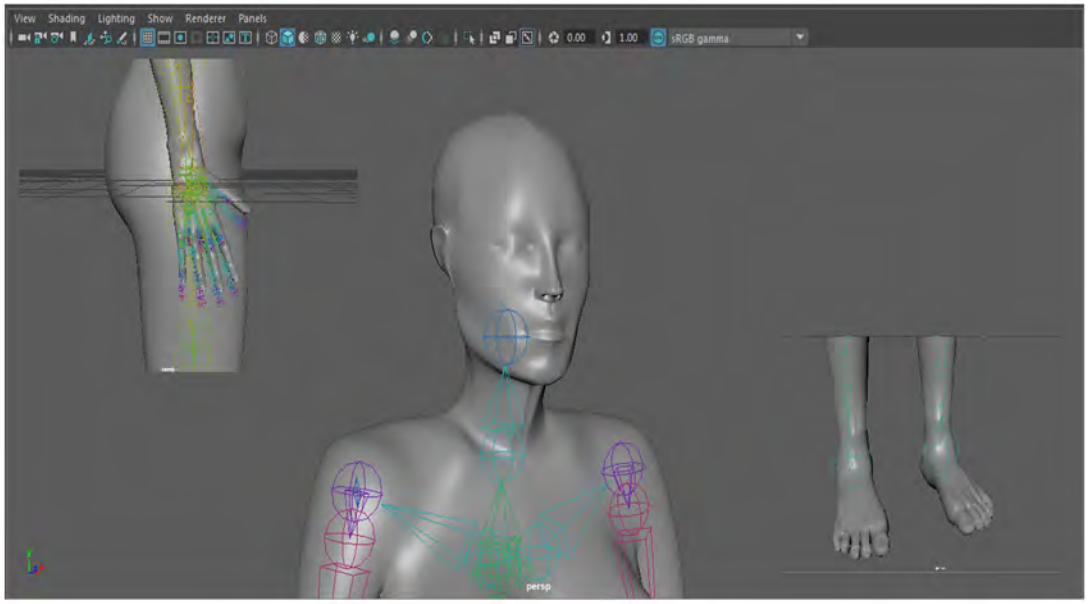


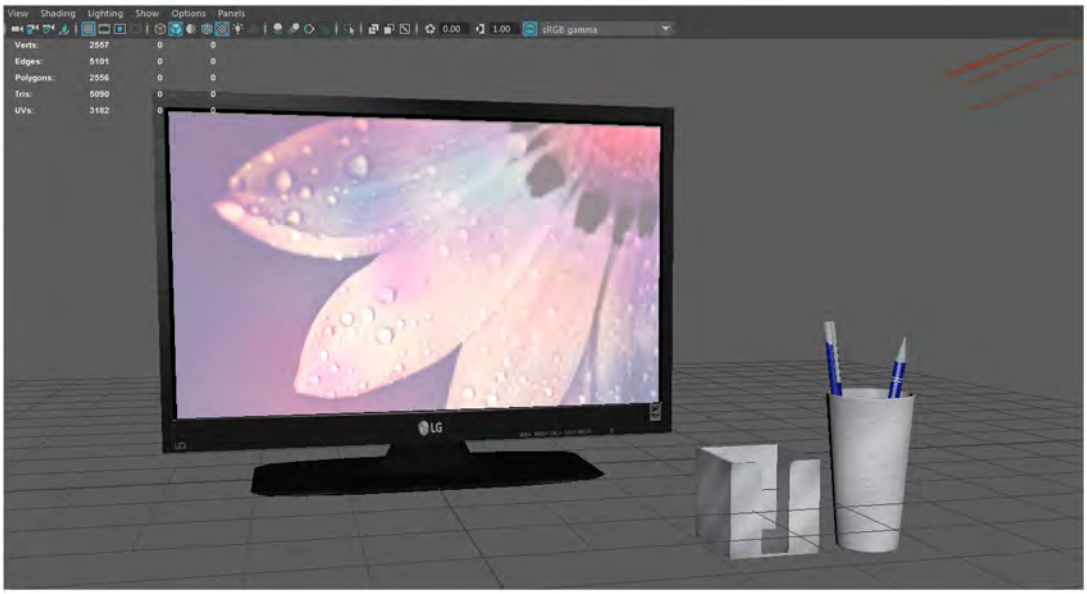
Arbeitsproben

Blender Modellierung und Rendering

Arbeitsproben

Maya Modellierung und Texturierung





Arbeitsproben