

Das Spiel: „Der Physikus wird superschlau“ wird im Text immer als Pws abgekürzt genannt.

1. Analyse zur Umsetzung eines digitalen Spiels mit Physikthemen der 8. Jahrgangsstufe

1.1. Einsatzbereich und Zielsetzung des Spiels „Der Physikus – wird superschlau Pws“

Das Spiel Pws ist zur freien Nutzung von Schülern der 8. Klassen gemacht. Es wäre für uns Macher sehr erfreulich, wenn die Fachschaftlehrkräfte das Spiel während dem Unterricht als Unterstützung oder bei Vertretungsstunden einsetzen. Da Pws auf einer freizugänglichen Webseite abgelegt wurde, kann der Zugang dazu eben unkompliziert erfolgen und am Schultablet oder Schul-PC genutzt werden.

1.2. Entscheidungsfindung über die Art des Spiels

1.2.1. Pro und Kontra der unterschiedlichen Spielformen

1.2.1.1. Möglichkeiten mit Spielform Quiz

Quiz kann zum einen so aufgebaut sein, dass es nur Ja- oder Nein-Fragestellungen gibt. Es kann auch mit verschiedenen Antwortmöglichkeiten gearbeitet werden, z. B. Multiple-Choice-Fragen oder Fifty-Fifty. Anwendungstechnisch könnte geklickt werden oder vom Schüler Text- bzw. Zahlenangaben gemacht werden. Von der Programmierung aus, war das die am einfachsten umsetzbare Spielweise. Hierfür bieten sich mehrere Programmiersprachen an (HTML, css, Java-Script, C++ , C#, ...) und auch die Datenmenge hält sich in einem überschaubaren Rahmen, d. h. das komplette Spiel könnte auf einen Datenträger kopiert werden und schnell auch am Endgerät geladen werden.

1.2.1.2. Möglichkeiten mit Spielform Rollenspiel

Es gibt auch die Möglichkeit einer fiktiven Person/Wesen, die physikalische Prozesse selbst durchlebt und daraus Erfahrungen oder neue Aufgaben sammelt. Gibt es dazu eine lineare Geschichte oder bereist der Avatar einen fiktiven Ort mit den physikalischen Themen?

Da keinem von uns und auch mir keine richtige Charakterentwicklung anhand einer Geschichte oder einem Avatar eingefallen ist, schied ein komplettes Rollenspiel auch an meiner persönlichen Programmierfähigkeit und Zeitaufwand aus. Hierfür müssten alle sehr viel Zeit und große Datenmengen aufwenden. Außerdem ist hierfür keiner von uns in einer Programmiersprache ausreichend gut.

1.2.1.3. Möglichkeiten mit Spielformen Strategiespiel oder Jump-and-Run

Ein Spiel zu entwickeln, dem Schüler gegen Schüler antreten können oder ganze Klassen gegeneinander, mit einem Auftrag oder Avatar (als Physik-Kämpfer sozusagen) klingt sehr verlockend, doch vermutlich eben nicht für das Ziel, das es für Schüler der 8. Klasse zum näherbringen der Physikthemen hilfreich.

Auch ist Jump-and-Run sehr beliebt, allerdings fiel mir dazu kaum etwas ein, was wir in Verbindung mit den Physikthemen bringen konnten noch ist die Programmierung hierfür einfach. Somit fielen beide Spielformen aufgrund der Programmierbarkeit und Ziel Wissen über Physik zu vermitteln aus.

1.2.1.4. verwendete Spielform bei Pws

Das Spiel Pws nutzt letztendlich ein Frage-Antwort-Klick-System, das aufgrund der verschiedenen Frageformen (Lückentext, Multiple-Choice, Fifty-Fifty,...) und kurzen Klickaufgaben als erweitertes Quiz angesehen werden kann. Inhaltlich bietet es auch in Textform Definitionen und Erklärungen zu den einzelnen Physikthemen und stellt kleine Physikaufgaben.

1.2.2. Argumentation für das Punktesystem im Spiel Pws

Ausgehend von der Frage, was zum zocken motiviert, wenn das Spiel schulische Themen hat und innerhalb der Schule angeboten wird, fanden folgende Überlegungen zum Anreiz oder Belohnungssystem des Spiels statt:

Erhält der Schüler Punkte, die zusammengezählt werden und am Ende einzelne Schüler oder ganze Klassen sich vergleichen können, also eine Art Ranking möglich ist? Werden die Punkte durch die Darstellung allein schon Anreiz genug, z. B. Weil sie Münzen, Juwelen, sind oder weil der Avatar sich weiter entwickeln kann z. B. Mit Erfahrungspunkten oder Ausstattung?

Kann in das Spiel sogenannte „Strafen“ einprogrammiert werden, damit verhindert wird, dass Schüler sich nur durch „alles anklicken“ das Spielende erreichen? Gibt es für falsches Klicken/falsche Antworten Punktabzüge oder sogar einen Spielstart von Beginn? Aufgrund der Programmierung und des Zeitaufwands, entschieden wir im Pws für eine Punktevergabe bei richtigen Antworten und Punktabzug bei falschen Antworten. Somit ist am Ende des Spiels ein Ranking möglich, an dem sich Schüler untereinander vergleichen können, aber das Spiel unabhängig von der Punkteanzahl zum Ende kommt.

1.2.3. Optik und grafische Darstellung des Spiels

Wir stellten uns folgenden Fragen: Hängt das Aussehen des Spiels mit dem physikalischen Inhalten zusammen bzw. können diese miteinander verbunden werden? Inwieweit hängt die Optik des Spiels mit der Art des Spiels zusammen und muss das aufeinander Bezug nehmen?

1.2.4. Namensgebung des Spiels Pws

Grundsätzliche Überlegungen waren, ob der Name des Spiels mit der Schule/Schulnamen einen Bezug haben könnte, z. B. Paul für den Avatar, PWS als Abkürzung für Physics with school usw. Soll der Spieltitel schon etwas über den Inhalt (Physik) aussagen oder die 8. Klasse erwähnen, z. B. Physik Okta für Acht?

Das Ergebnis war: „Der Physikus“ als Haupttitel, Untertitel „wird superschlau“ und als Abkürzung ergibt sich daraus, an die Abkürzung unserer Schule angelehnt: Pws.

Wir haben im Internet noch nachgesehen, ob der Name unseres Spiels schon vergeben ist und haben mehrere PC-Spiele mit dem Titel „Physikus“ oder „Physicus“ gefunden, auch mit Untertiteln aber keines mit – wird superschlau und in der Abkürzung Pws.

1.3. Analyse zur Einarbeitung der Themenbereiche aus Physik in ein digitales Spiel

1.3.1. Allgemeiner Aufbau des Spiels Pws

Als Vorgabe hatten wir aus ISB 2021 Grundwissen, Mechanik, Wärmelehre und Elektrizitätslehre erhalten. Da wir 3 Gruppenmitglieder sind, haben wir uns auf die Themenschwerpunkte Mechanik, Wärmelehre und Elektrizitätslehre beschränkt.

Das Spiel „Der Physikus wird superschlau“ startet nach der Titelseite mit einem Menü, auf dem die Auswahl der Themengebiete getroffen werden kann. So kann der Lehrer oder der Schüler das Spiel auch während des Schuljahres nutzen, nicht erst, wenn alle Themen bearbeitet wurden. Der Spieler wählt also am Menü eines der Themengebiete aus. Aus Sicht der Programmierung hieß es hier lediglich eine Funktion einzubauen, die nach dem Klicken den jeweiligen Datensatz für das entsprechende Physikthema öffnet.

Nun kann der Spieler das gesamte Themengebiet machen, erhält am Ende seine Punktzahl dafür und könnte ein weiteres Themengebiet beginnen und addieren.

Innerhalb jedem Themengebiets stehen verschiedene Lückentexte, Fragen, Berechnungen oder Klickmanagement-Aufgaben zur Verfügung, die linear abgearbeitet werden. Am Ende kommt der Spieler bei seiner Punktzahl an, welche innerhalb der Klasse oder im Klassenverbund verglichen werden kann.

1.3.2. Zusammenführung von Physikthema mit Spieleinheit

Anhand der verschiedenen Quellen, die an den Beispielen genau benannt werden, fiel die Entscheidung, ob es bildliche Darstellungen, Experimente, Videos oder Erklärungstexte gibt und ob diese für das Spiel im Sinne des aktiven anklicken eingesetzt werden können.

Begrenzt wurden alle Umsetzungen innerhalb des Spiels stets vom Programmierungsverständnis, Datenmengen und Zeitaufwand. Insgesamt wurde bei der Reihenfolge des Spielverlaufs zumeist die Reihenfolge der Inhalte von ISB 2021 übernommen. Das heißt, beim Themengebiet Mechanik, die Struktur Arbeit, Energie, Energieerhaltungssatz, Leistung und dann Wirkungsgrad (vgl. ISB 2021 Seite 4). Beim Themengebiet Wärmelehre, die Struktur Innere Energie, Temperatur, Wärme und Energietransport/Wärmetransport (vgl. ISB 2021 Seite 5)

Beim Themengebiet Elektrizitätslehre, die Struktur Ladung, Elektrisches Feld, Elektrischer Strom und Stromstärke (vgl. ISB 2021 Seite 6)

1.3.2.1. Am Beispiel Mechanik

Die Definition von Arbeit habe ich nach (leifiphysik), (frustfrei-lernen) und der Angaben von ISB 2021 angewandt und habe im Spiel aus den verschiedenen Formulierungen diesen Text gemacht: „Der Betrag von verrichteter Arbeit W ist auch der Betrag, um den sich Energie eines System bei einem Vorgang ändert.“ (vgl. ISB 2021, leifiphysik, frustfrei-lernen)

Auf der 2. Seite im Bereich Mechanik, leitet die Definition von Arbeit (vgl. ISB 2021, leifiphysik...) das erste reine Klickevent (ohne Vorwissen bzw. Aufgabe) zu den verschiedenen Unterbereichen: Hubarbeit, Beschleunigungsarbeit, Spannarbeit und Reibungsarbeit (vgl. ISB 2021, leifiphysik...). In der Programmiersprache JavaScript mit der Library PIXIJS bedeutet allein für diesen Klickevent auf dieser Seite folgender Code:

```
<Object>.on("pointerdown", (event) => {<OnClick>}); <Object>.eventMode = "static";
```

Ein anderes Klicksystem wird auf Seite 4 bei Mechnik angewandt: Die Aufgabe wurde aus bds Physik aktuell 8I Seite 8 abgeändert. Hierfür nutzten wir das erste Mal die Vergabe von Punkten und die optische Unterstützung eines roten Feldes bei Fehler, grünes Feld bei Erfolg. Der Programmierungscode für alldies lautet:

```
let questionTxt = new PIXI.Text(question,{fontFamily : 'Arial', fontSize: 20, fill : 0xffffffff, align : 'center'});
```

```
questionTxt.x = 10;  
questionTxt.y = 200;  
app.stage.addChild(questionTxt);
```

```

let answer1Btn = new Button(answer1, 300, 100, 100, 400, 0xffffffff, 0x313131, 20);
answer1Btn.Draw();

let answer2Btn = new Button(answer2, 300, 100, 100, 550, 0xffffffff, 0x313131, 20);
answer2Btn.Draw();

let answer3Btn = new Button(answer3, 300, 100, 700, 400, 0xffffffff, 0x313131, 20);
answer3Btn.Draw();

if(rightAnswer == 1) answer1Btn.OnClick = onclick;
if(rightAnswer == 2) answer2Btn.OnClick = onclick;
if(rightAnswer == 3) answer3Btn.OnClick = onclick;
if(rightAnswer != 1){
    answer1Btn.OnClick = () => {
        let answer1Btn = new Button(answer1, 300, 100, 100, 400, 0xffffffff, 0xff0000, 20);
        answer1Btn.Draw();
        G_Points--;
        E_header();
    }
}
if(rightAnswer != 2){
    answer2Btn.OnClick = () => {
        let answer2Btn = new Button(answer2, 300, 100, 100, 550, 0xffffffff, 0xff0000, 20);
        answer2Btn.Draw();
        G_Points--;
        E_header();
    }
}
if(rightAnswer != 3){
    answer3Btn.OnClick = () => {
        let answer3Btn = new Button(answer3, 300, 100, 700, 400, 0xffffffff, 0xff0000, 20);
        answer3Btn.Draw();
        G_Points--;
        E_header();
    }
}
}

```

1.3.2.2. Am Beispiel Wärmelehre

Zu Beginn des Themas Wärmelehre wird ein Lückentext „Wärme ist die Energie, die in Folge von Temperaturunterschieden übertragen wird. Dies kann innerhalb eines Körpers erfolgen oder zwischen zwei Körpern mit unterschiedlicher Temperatur“ (siehe ISB 2021 Seite 5) gezeigt. Dieser muss durch Anklicken des Textbutton und der dazu richtigen Textlücke im oberen Text vervollständigt werden. Die Programmierung beispielhaft für die Lückentexte ist folgende:

```

function E_ltext(Text, answer1, answer2, answer3, answer4, rightAnswer1, rightAnswer2, l1X, l1Y, l2X, l2Y, onclick){
    let btn1X = 100;
    let btn1Y = 500;

```

```

let btn2X = 600;
let btn2Y = 440;
let btn3X = 200;
let btn3Y = 650;
let btn4X = 700;
let btn4Y = 600;
let lTextObj = new PIXI.Text(Text,{fontFamily : 'Arial', fontSize: 20, fill : 0xffffffff, align :
'center'});
lTextObj.x = 10;
lTextObj.y = 300;
app.stage.addChild(lTextObj);

var selectedT = 0;
var dones = 0;

let flBtn = new Button("_____ ", 250, 20, l1X, l1Y, 0xffffffff, 0x212121,
20, 0);
flBtn.Draw();

let slBtn = new Button("_____ ", 250, 20, l2X, l2Y, 0xffffffff, 0x212121,
20, 0);
slBtn.Draw();

let fTextBtn = new Button(answer1, 300, 100, btn1X, btn1Y, 0xffffffff, 0x313131, 20);
fTextBtn.Draw();

let dTextBtn = new Button(answer3, 300, 100, btn4X, btn4Y, 0xffffffff, 0x313131, 20);
dTextBtn.Draw();

let aTextBtn = new Button(answer4, 300, 100, btn3X, btn3Y, 0xffffffff, 0x313131, 20);
aTextBtn.Draw();

let sTextBtn = new Button(answer2, 300, 100, btn2X, btn2Y, 0xffffffff, 0x313131, 20);
sTextBtn.Draw();

let checkBtn = new Button("Check", 100, 50, 900, 750, 0xffffffff, 0x313131, 20);
checkBtn.Draw();

fTextBtn.OnClick = () => {
    selectedT = 1;
    if(rightAnswer1 != 1 && rightAnswer2 != 1){
        let tmp = new Button(answer1, 300, 100, btn1X, btn1Y, 0xffffffff, 0xff0000, 20);
        tmp.Draw();
        G_Points--;
        E_header();
    }
};

sTextBtn.OnClick = () => {
    selectedT = 2;
    if(rightAnswer1 != 2 && rightAnswer2 != 2){
        let tmp = new Button(answer2, 300, 100, btn2X, btn2Y, 0xffffffff, 0xff0000, 20);
        tmp.Draw();
    }
};

```

```

        G_Points--;
        E_header();
    }
};
dTextBtn.OnClick = () => {
    selectedT = 4;
    if(rightAnswer1 != 4 && rightAnswer2 != 4){
        let tmp = new Button(answer3, 300, 100, btn4X, btn4Y, 0xffffffff, 0xff0000, 20);
        tmp.Draw();
        G_Points--;
        E_header();
    }
};
aTextBtn.OnClick = () => {
    selectedT = 4;
    if(rightAnswer1 != 3 && rightAnswer2 != 3){
        let tmp = new Button(answer4, 300, 100, btn3X, btn3Y, 0xffffffff, 0xff0000, 20);
        tmp.Draw();
        G_Points--;
        E_header();
    }
};
flBtn.OnClick = () => {
    if(selectedT == rightAnswer1){
        let ranswer;
        let bttnX;
        let bttnY;
        if(rightAnswer1 == 1){
            ranswer = answer1;
            bttnX = btn1X;
            bttnY = btn1Y;
        }
        if(rightAnswer1 == 2){
            ranswer = answer2;
            bttnX = btn2X;
            bttnY = btn2Y;
        }
        if(rightAnswer1 == 3){
            ranswer = answer3;
            bttnX = btn3X;
            bttnY = btn3Y;
        }
        if(rightAnswer1 == 4){
            ranswer = answer4;
            bttnX = btn4X;
            bttnY = btn4Y;
        }
        let tmp = new Button(ranswer, 250, 20, l1X, l1Y, 0xffffffff, 0x212121, 20, 0);
        tmp.Draw();
        selectedT = 0;
        let tmp2 = new Button("a", 300, 100, bttnX, bttnY, 0x212121, 0x212121, 20);
        tmp2.Draw();
    }
};

```

```

        dones++;
        G_Points++;
        E_header();
    }
};
slBtn.OnClick = () => {
    if(selectedT == rightAnswer2){
        let ranswer;
        let bttnX;
        let bttnY;
        if(rightAnswer2 == 1){
            ranswer = answer1;
            bttnX = btn1X;
            bttnY = btn1Y;
        }
        if(rightAnswer2 == 2){
            ranswer = answer2;
            bttnX = btn2X;
            bttnY = btn2Y;
        }
        if(rightAnswer2 == 3){
            ranswer = answer3;
            bttnX = btn3X;
            bttnY = btn3Y;
        }
        if(rightAnswer2 == 4){
            ranswer = answer4;
            bttnX = btn4X;
            bttnY = btn4Y;
        }
        let tmp = new Button(ranswer, 250, 20, l2X, l2Y, 0xffffffff, 0x212121, 20, 0);
        tmp.Draw();
        selectedT = 0;
        let tmp2 = new Button("a", 300, 100, bttnX, bttnY, 0x212121, 0x212121, 20);
        tmp2.Draw();
        dones++;
        G_Points++;
        E_header();
    }
};
checkBtn.OnClick = () => {
    if(dones >= 2){
        onclick();
    }
}
}

```

Bei dem Thema Reibungsarbeit, wurde eine interaktive Animation eingebaut, die durch permanentes Klicken beschleunigt und zu einem Ziel, hier 60 ° Celsius, führt. Solche

animierten Einheiten sind sehr aufwendig zu konstruieren und müssen jedesmal für eine neue interaktive Animation entwickelt werden.

```
function e2m10(){
  E_clear(G_BACKGROUND);
  //E_background('https://raw.githubusercontent.com/Mad-Mushroom/DerPhysikus/main/game/js/
res/e2m10a_background.png');
  E_topText("Wärmeleitung erfolgt stets von Stellen höherer Temperatur\nzu stellen niedrigerer
Temperatur.");
  E_header();
  let nextButton = new Button("->", 100, 50, 900, 750, 0xffffffff, 0x313131, 20);
  nextButton.Draw();
  nextButton.OnClick = () => {
    E_clear(G_BACKGROUND);
    //E_background('https://raw.githubusercontent.com/Mad-Mushroom/DerPhysikus/main/game/
js/res/e2m10b_background.png');
    E_topText("Schaffe es durch Reibung insgesamt 60°C zu erreichen.");
    E_header();

    let outerCirc = new PIXI.Graphics(); outerCirc.beginFill(0xffffffff);
    outerCirc.drawCircle((window.innerWidth)/2, (window.innerHeight)/2, 100);
    app.stage.addChild(outerCirc);
    let outerInnerCirc = new PIXI.Graphics(); outerInnerCirc.beginFill(G_BACKGROUND);
    outerInnerCirc.drawCircle((window.innerWidth)/2, (window.innerHeight)/2, 90);
    app.stage.addChild(outerInnerCirc);

    L210_ball = new PIXI.Graphics(); L210_ball.beginFill(0xffffffff);
    L210_ball.drawCircle(((window.innerWidth)/2)-42, ((window.innerHeight)/2)-42, 32);
    L210_ball.position.set(window.innerWidth/2, window.innerHeight/2);
    L210_ball.pivot.x = window.innerWidth/2;
    L210_ball.pivot.y = window.innerHeight/2;
    app.stage.addChild(L210_ball);

    let innerCirc = new PIXI.Graphics(); innerCirc.beginFill(0xffffffff);
    innerCirc.drawCircle((window.innerWidth)/2, (window.innerHeight)/2, 30);
    app.stage.addChild(innerCirc);
    let innerInnerCirc = new PIXI.Graphics(); innerInnerCirc.beginFill(G_BACKGROUND);
    innerInnerCirc.drawCircle((window.innerWidth)/2, (window.innerHeight)/2, 20);
    app.stage.addChild(innerInnerCirc);

    eventBtn = new PIXI.Graphics();
    eventBtn.beginFill(0xffffffff);
    eventBtn.drawRect(((window.innerWidth)/2)-100, ((window.innerHeight)/2)-100, 200, 200);
    eventBtn.alpha = 0;
    app.stage.addChild(eventBtn);

    eventBtn.on('pointerdown', () => {
      L210_rotSpeed += 0.01;
    });
    eventBtn.eventMode = 'static';
    G_LEVEL = 210;
  }
}
```

```

}

if(G_LEVEL == 210){
    L210_ball.rotation += L210_rotSpeed;
    let temp = (L210_rotSpeed * 2 * 20) + 20;
    let temptxt = temp + "°C";
    //document.getElementById("htmltxt").innerHTML = temptxt;
    if(temp >= 60){
        let checkBtn = new Button("->", 100, 50, 900, 550, 0xffffffff, 0x313131, 20);
        checkBtn.Draw();
        checkBtn.OnClick = () => {
            G_Points++;
            e2m11();
        }
    }
}
}

```

1.3.2.3. Am Beispiel Elektrizitätslehre

Hier verwenden wir auf Seite 2 eine der Zuordnungsaufgaben, bei denen durch das erste Anklicken eines Textfeldes/einer Grafik die Zuordnung am richtigen Textfeld/Grafik mit dem zweiten Klicken gelingt. Es taucht nur bei der richtigen Kombination der Verbindungsstrick auf. Code hierfür:

```

function e3m2(){
    G_LEVEL = 302;
    E_clear(G_BACKGROUND);
    E_header();
    E_topText("Ordne zu:");

    let selection = 0;
    let dones = 0;

    let e1Btn = new Button("neutrales Atom", 200, 150, 50, 200, 0xffffffff, 0x212121, 20);
    e1Btn.Draw();
    e1Btn.OnClick = () => {selection = 1;};
    let e2Btn = new Button("Elektronenmangel", 200, 150, 50, 300, 0xffffffff, 0x212121, 20);
    e2Btn.Draw();
    e2Btn.OnClick = () => {selection = 2;};
    let e3Btn = new Button("Elektronenüberschuss", 200, 200, 50, 500, 0xffffffff, 0x212121, 20);
    e3Btn.Draw();
    e3Btn.OnClick = () => {selection = 3;};

    let eg1Btn = new SpriteButton("https://raw.githubusercontent.com/Mad-Mushroom/Archive/main/Apocalypse/res/game/text/placeholder.png", 200, 200, 600, 100, 0xffffffff, 0x212121, 20);
    eg1Btn.Draw();
    eg1Btn.OnClick = () => {
        if(selection == 3){
            dones++;
        }
    }
}

```

```

        G_Points++; E_header();
        let line = new PIXI.Graphics();
        line.lineStyle(2,0xffffff,1);
        line.moveTo(250,600);
        line.lineTo(700,200);
        app.stage.addChild(line);
    }
    if(dones >= 3) E_continue(() => {e3m3();});
};
let eg2Btn = new SpriteButton("https://raw.githubusercontent.com/Mad-Mushroom/Archive/main/Apocalypse/res/game/text/placeholder.png", 200, 200, 600, 350, 0xffffff, 0x212121, 20);
eg2Btn.Draw();
eg2Btn.OnClick = () => {
    if(selection == 1){
        dones++;
        G_Points++; E_header();
        let line = new PIXI.Graphics();
        line.lineStyle(2,0xffffff,1);
        line.moveTo(150,300);
        line.lineTo(700,400);
        app.stage.addChild(line);
    }
    if(dones >= 3) E_continue(() => {e3m3();});
};
let eg3Btn = new SpriteButton("https://raw.githubusercontent.com/Mad-Mushroom/Archive/main/Apocalypse/res/game/text/placeholder.png", 200, 200, 600, 600, 0xffffff, 0x212121, 20);
eg3Btn.Draw();
eg3Btn.OnClick = () => {
    if(selection == 2){
        dones++;
        G_Points++; E_header();
        let line = new PIXI.Graphics();
        line.lineStyle(2,0xffffff,1);
        line.moveTo(150,350);
        line.lineTo(700,700);
        app.stage.addChild(line);
    }
    if(dones >= 3) E_continue(() => {e3m3();});
};
}

```

TEILCHEMODELL Gitterionen und Elektronen kommt noch :)