

# LAST MINUTE!

LARS OBIST	5043024
THAO NGUYEN	5034297
YULIYA LITVIN	5034284
MOUMITA AHMAD	5027729

---

1. SEMESTER

MEDIENINFORMATIK

GRUPPE A

# Inhaltsverzeichnis

Grundidee .....	3
Spielidee .....	3
Spielgeschichte .....	5
Graphische Gestaltung .....	6
Playtesting .....	10
Arbeitsteilung .....	11
Technische Merkmale .....	13
Datenstruktur .....	13
Drag and Drop .....	13
Minispiele .....	14
Post Mortem .....	15
Lars Obist .....	15
Yuliya Litvin .....	15
Moumita Ahmad .....	16
Thao Nguyen .....	16
Quellen .....	17
Anlage .....	19
Ursprünglicher Spielplan .....	19
Finaler Spielplan .....	20
Spiellayout - Skizzen .....	21
Einteilung der Aufgaben mit Gantt-Diagramm .....	22
Entwicklung eines Kapitels .....	22
Beispiel home.html .....	22



# Grundidee

Beim Brainstorming für unsere Spielidee haben wir uns an unseren eigenen Vorlieben orientiert. Mehrheitlich kamen wir zu dem Entschluss, dass wir nicht daran interessiert wären, ein reines Text-Adventure Spiel zu entwickeln, da es unserer Meinung nach sehr schnell, sehr eintönig und langweilig wird.

Somit stand fest, dass wir etwas programmieren möchten, wo der Spieler interagieren muss.

Daraufhin haben wir uns zusammengesetzt und überlegt, in welche Richtung unser Spiel thematisch gehen soll. Zu Beginn kamen keine Ideen auf, von denen wir vollkommen überzeugt waren, bis zu dem Punkt, an dem wir einen Bezug auf unsere eigene Situation als Studenten gezogen haben. Deshalb haben wir die Bachelorarbeit als Ziel unseres Spiels festgelegt. Dieser Weg ist jedoch geplagt von vielen Klausuren und Projekten.

# Spielidee

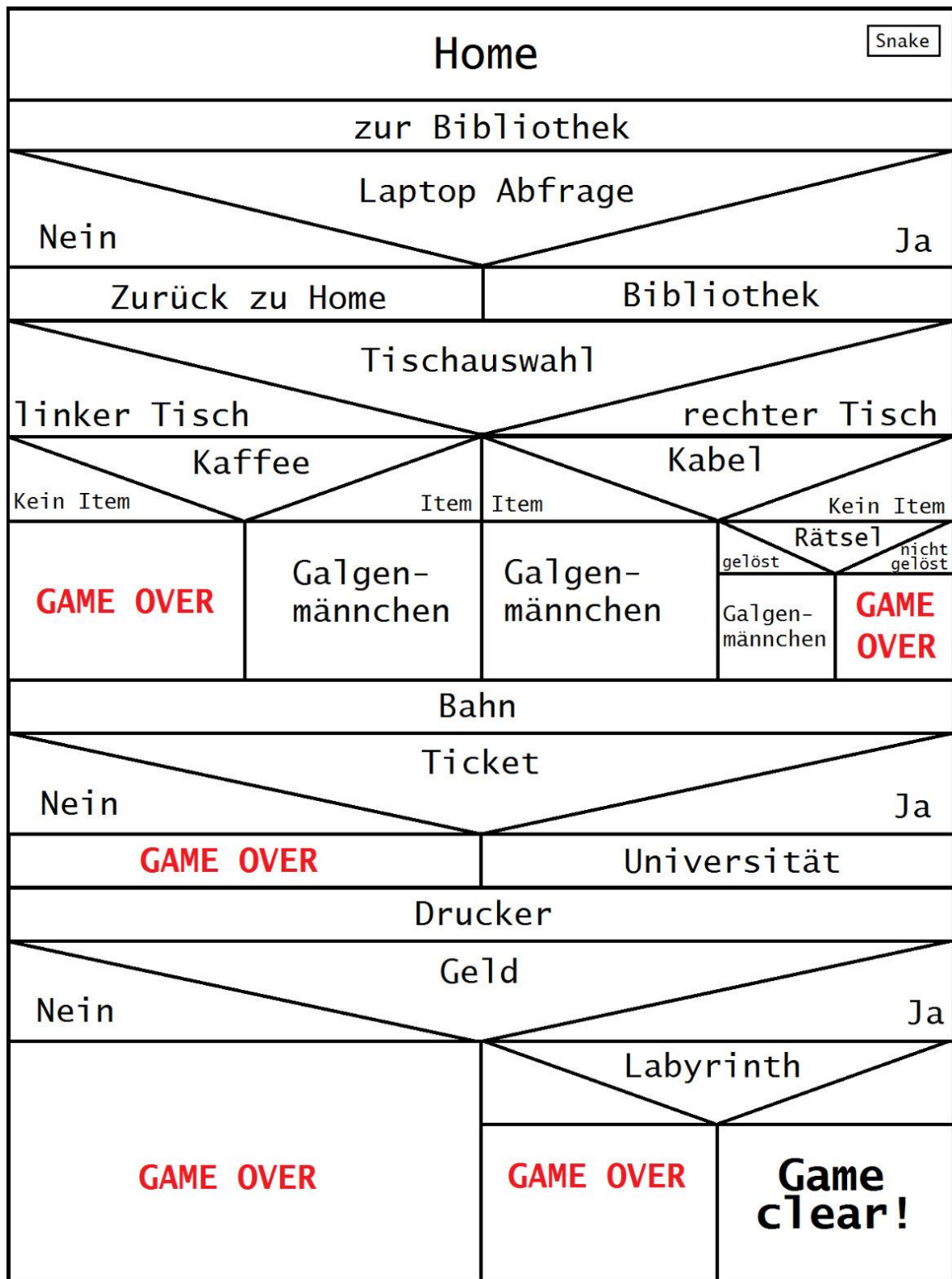
Aus diesem Gedankenzug entstand die Idee, dass man ein "Jump and Run" Spiel entwickeln kann, wo dem Spieler, in diesem Fall dem Studenten, Hindernisse in Form von Büchern, Klausuren oder Ähnlichem im Weg liegen oder entgegengeworfen werden. Diesen Hindernissen hätte er ausweichen müssen.

Diese Spielidee mussten wir jedoch verwerfen, da uns gesagt wurde, dass Kollisionen erst im kommenden Semester behandelt werden. Die Grundidee gefiel uns dennoch und wir überlegten, dieses Konzept beizubehalten, doch in einer anderen Art und Weise umzusetzen.

Wir kamen zurück zu dem "Text-Adventure"-Konzept, wollten dies jedoch durch andere Elemente ergänzen, die das Spielerlebnis zu einem spannenderen machen könnten. Zum einen kam uns bei den Besprechungen die Idee, dass wir ein Spiel erstellen können, welches entscheidungsabhängig ist. Der Spieler sollte zu Beginn Entscheidungen treffen, die ihm im Verlauf des Spieles entweder helfen oder ihn zurückwerfen.

Durch die, zu Beginn getroffenen Entscheidungen, stehen dem Spieler unterschiedlichste Herausforderungen entgegen. Dies hat zur Folge, dass der Spieler an verschiedensten Punkten im Spiel scheitern kann, und somit nochmal von vorne anfangen muss, um das Ende des Spiels erreichen zu können. Im besten Fall, merkt der Spieler, weshalb er gescheitert ist, und verbessert seine getroffenen Entscheidungen.

Durch das Zurückwerfen soll der Spieler animiert werden, neue Wege zu entdecken und Entscheidungen zu treffen, um mehr von dem Spiel zu entdecken. Zudem wird hierdurch ein Drang erzeugt, das Spiel zu wiederholen. Dennoch sollen alle Wege in unserem Spiel zu einem Ziel führen, den Bachelorabschluss zu bekommen.



Ein weiterer Gedanke war das Spielen von Minispielen während des eigentlichen "Text-Adventure"-Spiels zu ermöglichen. So hätten wir die Möglichkeit ein wenig Abwechslung mit reinzubringen. Die Spiele sollen den Spieler zum Nachdenken und Grübeln bringen, dies wiederum passt ideal zu dem Konzept des Studenten, da er jeden Tag Herausforderungen begegnet.

Die Grundidee des Spiels bauten wir mit einer Geschichte aus, die sich wie ein Roter Faden durch das Spiel zog. Somit griffen wir das Konzept eines Studenten auf und überlegten was zu ihm passen könnte. Ein Klischee, der uns allen aufkam war, dass Studenten gerne Projekte lange vor sich herschieben und selten früh genug beginnen. Dies hat oftmals zur Folge, dass sie lange Nächte vor sich haben und meist sehr unter Druck stehen. Dies war ein Gefühl, welches wir auch in unserem Spiel verbreiten wollten. Dies war ein Gefühl, welches wir auch in unserem Spiel verbreiten wollten. Der Spieler soll die Handlungen eines Studenten in den letzten Stunden bis zur Abgabe übertrieben verdeutlicht bekommen. In diesem Prozess entstand der Name unseres Spiels: Last Minute!

Die Zielgruppe des Spiels „Last Minute!“ sind vorrangig Studenten, die sich in Lage des Charakters hineinversetzen können.

## Spielgeschichte

Der Spieler spielt einen Tag aus der Sicht eines namenlosen Studenten, wodurch er sich besser in den Charakter des Spieles hineinversetzen kann.

Am Anfang des Spiels schildert der Student hysterisch seine Situation wie er die Zeit aus den Augen verloren hat und seine Bachelorarbeit heute Abend abgeben muss. Dadurch soll der Spieler unter Druck gesetzt werden. Im ersten Kapitel des Spiels angekommen, wird der Spieler informiert, dass der Student seine Tasche für den Tag packt. Der Spieler entscheidet in diesem Fall, was in den Rucksack eingepackt werden soll. Seine Entscheidungen werden Einfluss auf den weiteren Spielverlauf haben. Wählt der Spieler beispielsweise eines der drei im Zimmer verteilten Bücher, wird er erstmal aufgehalten seine Tasche weiter zu packen, da er auf einen hungrigen Bücherwurm stößt. Dieses Problem löst der Spieler aus, indem er den Bücherwurm zum Essen steuert.

Darüber hinaus kommt er im ersten Kapitel nicht weiter, wenn er seinen Schlüssel nicht eingepackt hat, weil er am Ende des Tages natürlich auch wieder in seine Wohnung wiederkehrt.

Ist der Spieler zufrieden mit dem Inhalt seiner Tasche, fährt der Charakter mit der Bahn zur Bibliothek, um dort den letzten Feinschliff an seiner Bachelorarbeit zu tätigen. In der Bahn fragt sich der Student jedoch, ob er seinen Laptop auch wirklich eingepackt hat. Sollte er ihn vergessen haben, muss er zurückfahren und seine Tasche erneut packen.

In der Bibliothek angekommen, sieht der Student, dass nur noch an zwei Tischen Platz zum Sitzen ist. Wählt er den linken Tisch aus, kippt der dort zuvor platzierte Kaffee um und läuft auf dem Laptop aus. Es könnte ein Problem auftreten, welches er nicht ohne vorher eingepackte Taschentücher lösen kann. Sollte dieser Fall eintreffen, kann ab diesem Zeitpunkt nicht weitergespielt werden. Das Spiel muss so neu gestartet oder beendet werden

Wird der rechte Tisch ausgewählt, geht der Akku des Laptops leer. Ähnlich wie am anderen Tisch muss er seine Tasche richtig gepackt haben, um dieses Problem lösen zu können. Hat er jedoch das Ladekabel vergessen, ist das Spiel hier noch nicht beendet, denn ein anderer Student bietet ihm an, ihm sein Ladekabel auszuleihen. Im

Austausch dafür muss der Student jedoch ein Rätsel lösen, welches der andere Student ihm stellt. Schafft er es nicht, tritt hier das "Game Over" ein.

Sobald der Student die Probleme an einem der beiden Tische lösen konnte, möchte er nun beginnen an seiner Bachelorarbeit weiterzuarbeiten. Dazu muss er sich an seinem Laptop anmelden. In diesem Fall muss er anhand von einem Galgenmännchen sein Passwort erraten. Im Falle von sieben inkorrekten Eingaben, scheitert er, andernfalls kann er seine Arbeit beenden.

Der Student hat nun seine Bachelorarbeit erfolgreich beendet und fährt mit der Bahn zur Universität. Dort möchte er sie ausdrucken lassen und im Anschluss bei dem Prüfungsamt abgeben.

In der Bahn steigen Kontrolleure ein und kontrollieren, ob jeder einen Fahrschein dabei hat. Hat der Spieler am Anfang des Spiels sein Semesterticket nicht in die Tasche gepackt, ist das Spiel an dieser Stelle vorbei. Doch wenn das Ticket eingepackt wurde, kommt der Spieler bei dem Drucker an, benötigt jedoch Geld um seine Abschlussarbeit ausdrucken zu können. Andernfalls verliert der Spieler.

Nachdem der Student nun alle Hürden überstanden hat, muss er seine fertig ausgedruckte Bachelorarbeit nur noch pünktlich um 18:00 Uhr beim Prüfungsamt abgeben. Jedoch ist es bereits 17:59 Uhr und er muss schnell zu dem Raum rennen, um seine Arbeit abzugeben. Die Zeit ist jedoch nicht sein einziges Problem, er weiß auch nicht wie der Weg zum Raum ist. Schafft der Spieler es nicht innerhalb von einer Minute seine Bachelorarbeit am richtigen Raum abzugeben, verliert er das Spiel und muss von vorne beginnen.

Schafft der Spieler es die Arbeit pünktlich abzugeben, freut sich der Charakter sehr alles abgegeben und eine Woche später die Präsentation gehalten zu haben. Er resümiert, dass er bei seinem Nächsten Projekt früher anfangen sollte, da er diesen Stress nicht noch einmal erleben wollen würde.

## Graphische Gestaltung

Unser Game Design entstand Stück für Stück im Laufe der Entwicklung unseres Spiels. Dessen Grundgerüst erstellten wir dennoch schon ganz zu Beginn, nachdem unsere Spielidee feststand.

Von Anfang an waren wir uns alle einig, dass wir die wesentlichen Inhalte unseres Games nicht über das ganze Display verteilen wollten, sondern in einem – im Verhältnis zum gesamten Bildschirm eines Laptops oder Computers – kleinerem Fenster erscheinen lassen wollten. Auf Basis dieser Idee entwickelten wir unser DIV-Box-Grundgerüst. In diesem definierten wir explizite Felder, die in einer DIV-Struktur im Programm wiedergegeben werden. Eine `#main`, ein `#field`, `#content`, `#menue`, `#home` und ein `#inventar` bildeten nun gemeinsam unsere Felder-Basis, in die wir alle unsere gewünschten Inhalte strukturiert und geordnet platzieren konnten.



Die *#main* füllt die ganze Seite des Browsers, in dem das Spiel später geöffnet werden sollte. Sie repräsentiert den Hintergrund für das Gesamtbild des Spiels und beinhaltet gleichzeitig den vollständigen Inhalt des Web-Games. Wir wählten die Farbe Weiß für dieses Feld, aus Gründen, die später genauer erläutert sein werden.

Das *#field* bildete die Grenzen, in denen das Spiel im Wesentlichen dargestellt wurde. Es wurde auch sowie die *#main* mit weißer Farbe angelegt, um mit dem voreingestellten Hintergrund zu verschmelzen. Denn keines der beiden Felder sollte sichtbar sein. Sie dienten lediglich nur dazu, um sich an unterschiedliche Displaygrößen anzupassen. In den Vordergrund sollte nämlich nur das eigentliche Spiel stechen, dass vor allem im *#content*, aber auch im *#home* und *#inventar* sichtbar gemacht wurde. Die drei soeben genannten kleineren Felder befinden sich alle im Rahmen des *#fields*.

Beim Festlegen der Platzierung dieser drei kleineren Felder gab es viel Diskussion. Insbesondere beim *#inventar* wurden wir uns nicht so schnell einig. Es gab den Vorschlag das Inventar als kleines Symbol in einer der vier Ecken unseres *#fields* zu legen. In diesem Fall wäre es solange bis zum Zeitpunkt des „Onclicks“ auf das Inventar-Symbol „geschlossen“, wonach es sich „öffnen“ und ein rechteckiger Balken zum Vorschein treten würde. Jedoch entschieden wir uns das Inventar von Anfang an „geöffnet“ – also als einen Balken – darzustellen, da es im Verlauf unseres Spiels eine fast immer aktive Rolle bekommen sollte. Unserer Meinung nach wäre es weniger sinnvoll gewesen, den User ständig aufzufordern das Inventar zu öffnen, was im Nachhinein für den Spieler vielleicht sogar lästig hätte werden können. Ein ständig „geöffnetes“ Inventar hatte somit den Vorteil, dass derjenige, der das Spiel spielt, seine Items, immer im Blick haben kann, demnach den Überblick behält und effektiver auf die Forderungen des Spiels reagieren kann. Die Überlegung den Balken des Inventars im unteren Bereich horizontal über die ganze Breite des *#fields* zu ziehen oder mit einer festgelegten Breite – auch horizontal - in dessen Mitte zu platzieren, überzeugte uns alle noch nicht. Schlussendlich, entschlossen wir uns es als senkrechten Balken mit fest definierter Höhe und Breite auf der rechten Seite des *#fields* anzubringen. Wir empfanden diese Platzierung als übersichtlich und angenehmer anzuschauen. Vor allem da *#home* nun auch direkt passend über dem *#inventar* positioniert werden konnte.

Die `#home`-DIV sollte nur die Funktion der Pause im Spiel beinhalten und zu einem Menü mit weiteren Auswahlmöglichkeiten weiterleiten. Deshalb füllten wir `#home` sichtbar nur mit einem schlichten, allseits bekannten Pause-Button, dessen von uns gewählte Positionierung oben rechts im Spielfeld im Übrigen wie in den meisten Fällen als gewöhnlich bezeichnet werden kann. Gestaltet haben wir das Symbol für ‚Pause‘ auch möglichst gewöhnlich und schlicht, zudem auch in einem Grauton, der zu unserer allgemeinen Thematik passt. Wir haben ihn mit einem weißen schlichten Hintergrund hinterlegt und mit einer passenden dünnen grauen Kante mit leicht abgerundeten Ecken umgeben, um die allgemeine Wirkung etwas weniger streng ausfallen zu lassen. Eine leichte Schattierung fügten wir dem Button auch noch hinzu, um einen kleinen 3D-Effekt zu erzeugen, den sowohl das `#home`-Feld als auch das `#inventar` - im gleichen Muster und aus dem gleichen Grund wie der Button -, zusätzlich erhielten, um eine kleine Hervorhebung zu erzeugen. Zudem packten wir `#home` und `#inventar` zusammen in eine weitere DIV-Box namens `#menue`, um diese beiden kleineren Feldern etwas besser und „sauberer“ anordnen zu können.

Die Größe des `#content`-Felds haben wir so festgelegt, dass es die gesamte Höhe, aber nicht die ganze Breite, des `#fields` einnimmt. Das `#content` sollte nämlich immer ein entsprechendes Hintergrundbild kriegen, dass die gesamte Fläche links bis hin zum `#menue` einfärbt, jedoch nicht rechts über das `#menue` herausragt.

So sollte eine saubere Grenze rechts neben dem Pause-Button und dem Inventar zu schaffen. In der `#content`-DIV befand sich das Wesentliche unseres Spiels.

Die Hintergrundbilder und GIFs, die wir auf unseren verschiedenen HTML-Seiten benutzt haben, sind alle vollständig von uns mit Adobe Illustrator gezeichnet und mit Adobe Photoshop animiert. Dabei hielten wir uns an einen vorher bestimmten Stil, den wir uns ausgedacht hatten. Da unser Spielthema das Studentenleben ist, wollten wir ein „skizzenartiges“ Aussehen unserer Seiten. Denn das häufigste, was ein Student macht, ist wohlmöglich Notizen zu schreiben und ab und zu Skizzen anzufertigen. Aus dieser Gedankenfolge entstand unsere allgemeine Designidee.

Demnach zeichneten wir alle unsere Bilder so, dass es klar zu erkennen ist, dass es sich eher um ausgearbeitete „Skizzen“ handelt als wirklich vollendete Zeichnungen.

Deshalb wählten wir die Farben Schwarz und Weiß als unsere Grundfarben für das gesamte Spiel und variierten zusätzlich mit verschiedenen Graustufen. Dazu wirken auch die unscharfen Kanten der – insbesondere – Hintergrundbilder mit, was das Gefühl von Schnelligkeit, Unsauberkeit und ein wenig Zerstreuung gibt. Diese Gefühle spiegeln sich nämlich in der Regel sehr oft in studentischen Notizen wieder.

So sind zum Beispiel auch die Elemente beziehungsweise Items, die im ersten „Kapitel“ vorhanden sind, im ganzen Raum verstreut.

Das Wort „Kapitel“ benutzen wir bewusst anstatt des Wortes „Level“ - auch wenn das Vokabular eines Spiels „Level“ bevorzugen würde-, weil wir unser Spiel etwas an eine Art Notizbuch anlehnen wollten. Deswegen ist das Intro beziehungsweise die Titelseite unseres Games schlicht schwarz gehalten, mit nur dem weißen Titel des Spiels beschriftet, was Ähnlichkeiten mit einem Buch wie auch einem Notizbuch hat. Diese Vorderseite schlossen wir auch mit einer Rückseite ab, die durch die „Credits“-Seite unseres Spiels dargestellt wurde. Diese ist auch schlicht schwarz gehalten, nur mit wenigen weißen Worten gefüllt. Bei beiden Seiten wird die Schrift mit einer Animation im Bildschirm eingeblendet, um einen dynamischeren Eindruck zu erzeugen.

Die Seiten nach dem „Titelblatt“ und vor der „Rückseite“, erfüllen eine ähnliche Rolle wie die des Prologs und Epilogs. Um diese möglichst klassisch und realistisch zu



halten, färbten wir beide Seiten in weiße Hintergrundfarbe ein und ließen schwarzen Text darauf „eintippen“.

Das „Eintippen“ des literarischen Inhaltes macht die beiden Seiten trotz ihres eher simplen Auftretens lebendiger und etwas ungewöhnlicher, aber auch technischer wie ein digitales Notizbuch. Unsere Thematik wird auf diesen beiden Seiten zudem von einer von uns skizzierten männlichen studentischen Figur unterstützt, die vom ‚Prolog‘ bis zum ‚Epilog‘ ihre Gestik und Mimik von gestresst zu glücklich ändert, um den Spieler auf etwas emotionaler Basis zu bewegen und durch die Geschichte mitfühlen zu lassen.

Das ‚Pause-Menü‘, das im Pause-Button oben rechts verlinkt wurde, zieht sich mit einem transparenten schwarzen Hintergrund über den gesamten Bildschirm, in gewissermaßen also über unsere gesamte *#main-DIV*. Wir haben uns bewusst dafür entschieden, es nur über das *#field* zu legen, damit der Spieler auch wirklich ein Gefühl von einem richtig pausierten Spiel bekommt. In dessen Zustand soll er nichts Anderes anklicken und bedienen können außer dem Menü. Das ‚Pause-Menü‘ selbst, ist mit nur drei verschiedenen Anklickmöglichkeiten - „Zurück zum Spiel“, „Neues Spiel“ und „Spiel beenden“ - einfach und schlicht gehalten, die in der Regel sehr typisch für ein Web-Game sind.

Zu Beginn jedes ‚Kapitels‘ haben wir Pop-ups eingestellt, die unsere Aufgaben zu den gewissen Zeitpunkten erklären. Diese sind durch schwarzen transparenten Hintergrund der Pop-Ups, gefüllt nur mit ein wenig weißem Text und einem Button auch schlicht gestaltet. Um diese auch zumindest etwas lebendig wirken zu lassen, haben wir die Buttons je mit einem grauen Hover versehen.

Darüber hinaus, haben wir die Methode eines Pop-ups durchgehend in unserem Spiel genutzt, um gewisse Rätsel und Abfragen im Laufe des Games erscheinen zu lassen. Alle diese Pop-ups sind sehr ähnlich aufgebaut, auch wenn sie unterschiedliche Größen haben, die dann zum jeweiligen Thema angepasst wurden, wobei diese Pop-ups immer innerhalb des Spielbereichs blieben und nie wie zum Beispiel das ‚Pause-Menü‘ über den gesamten Bildschirm reichten, denn der Spieler sollte sich währenddessen immer noch mitten im Spiel wahrnehmen. Diese Pop-ups enthalten zum Teil auch Bilder oder Animationen und nicht nur Text, die von uns selbst erstellt wurden, und die Aufgaben oder Fragen zusätzlich nochmal illustrativ unterstützen. Zum Teil hat eines solcher Bilder in einer bestimmten Situation aber auch die Funktion eines Rätsels.

Unser Labyrinth, dessen Timer, wie auch das ‚Snake‘-Minispiel haben wir wie alles andere auch in passenden Graustufen gestaltet. Im Labyrinth ist der Kreis, der den Aufenthalt des Spielers im aktuellen Feld des Labyrinths anzeigt und das Rechteck, das das Ziel repräsentiert, bewusst in einem dunklen Grauton eingefärbt, um diese für den Spieler zu jedem Zeitpunkt gut sichtbar zu machen. Dabei ist die ablaufende Zeit im Timer auch mit einem dunkelgrauen, runter laufenden Balken repräsentiert. Das diene dem Zweck, dass der Spieler, sich in der Zeit und nicht nur im Labyrinth gut orientieren kann, was auch noch die zusätzliche Ziffernanzeige der verbliebenen Zeit unterstützt. Des Weiteren entschieden wir uns das „Pause-Menü“ aus dem Labyrinth-Kapitel zu entfernen, da es inhaltlich keinen Sinn macht an dieser Stelle eine „Neustart“-Funktion einzubauen.

Beim Verlieren eines Levels tritt das Pop-up das Game Overs ein. Dieses ist absichtlich ähnlich wie das ‚Pause-Menü‘ gestaltet, um das Spiel einheitlich aussehen zu lassen und den gewissen Stil beizubehalten. Allerdings gibt es hier nur zwei

Auswahlmöglichkeiten zum Anklicken – „Neues Spiel“ und „Spiel beenden“ -, denn ein „Zurück zum Spiel“-Button würde hier absolut keinen sinnvollen Zweck erfüllen. Das Gefühl des Spielers verloren zu haben, unterstützt zusätzlich nochmal die Illustration des gestressten Studenten.

Unser Text, der durch den Verlauf des Spiels immer wieder auftritt, hat durchgehend die Schriftart „Lucida Sans Typewriter“, die zum Teil an eine typische technische Computerschrift erinnern soll. Diese haben wir gewählt Studenten oftmals eher an Laptops arbeiten und das im Spiel widergespiegelt werden sollte. Des Weiteren haben wir mit verschiedenen Schriftgrößen gearbeitet, die immer situationsbedingt angepasst wurden. Die Schriftfarbe war in den meisten Fällen weiß, da wir auf dunklen Hintergründen gearbeitet haben, jedoch haben wir auch die typische Schriftfarbe Schwarz genutzt, wenn wir uns auf weißem Hintergrund befanden, da es uns wichtig war, dass der Text in allen Fällen gut lesbar war.

Bemerkenswert ist auch noch der ‚Arrow‘-Button, der sich wie ein roter Faden durch das gesamte Spiel zieht. Dieser ist mit einer dynamischen Bewegung versehen, um ihn auffallender darzustellen und als „Weiterklick“-Symbol für den Spieler verständlich zu machen; deshalb auch die Form eines Pfeiles. Je nachdem, auf welchem Hintergrund man sich gerade befindet, wird dieser Button entweder weiß oder schwarz mit einem Effekt eines Verlaufs von verschiedenen Graustufen angezeigt.

Des Weiteren haben wir auf ein weiteres kleines Detail geachtet, was beim Spielen nicht auffallen mag. In dem Tab des Spiels befindet sich neben dem Spielnamen auch ein kleines Logo. Das Logo ist eine stärkere Zeichnung des Männchens, welches sich ebenfalls am Anfang sowie dem Ende des Spiels befindet.

## Playtesting

Unser Projekt haben wir immer wieder mal durch Freunde und Bekannte testen lassen, was uns bei der Entwicklung unseres Spieles oft weitergeholfen hat.

Dabei war das Bedeutendste wohl der Playtest in der Hochschule.

Zu diesem Zeitpunkt war das Home-Level zwar noch nicht ganz fertig, jedoch konnte der Rest schon komplett durchgespielt werden. Nichtsdestotrotz war es für den Tester bereits möglich, ein Gefühl für unser Spiel zu entwickeln und ein Feedback abgeben zu können.



Besonders hilfreich war dabei auf fehlende Informationen hingewiesen zu werden, die wir ergänzen sollten. Zum Beispiel wurde das Galgenmännchen als solches nicht von jedem erkannt. Auf die Tasten, die im Labyrinth genutzt werden sollten, wurde wohl auch nicht genügend hingewiesen, da dies für die Tester nicht eindeutig genug wurde. Auch eine erfolgreich beendete Aufgabe, wurde nicht immer als solche interpretiert.

Interessant war auch, dass viele die gleiche Entscheidung bei der Tischauswahl trafen, also den linken Tisch wählten, sodass wir schon überlegten, das Rätsel auch auf diese

Seite zu packen. Allerdings haben wir uns letztendlich dagegen entschieden da dies nicht gut zu unserer Story gepasst hätte und wir die Wege weiterhin unterschiedlich gestalten wollten.

Des Weiteren traten Probleme durch unerwartete Ausführungen des Nutzers, die nur durch mehrmaliges Verlieren oder sinnloses Klicken entstanden.

Ein weiteres Thema war die verfügbare Zeit im Labyrinth, da verschiedene Tester zum Teil oft mit eher unterschiedlichen Zeiten zurechtkamen. Wir haben uns letztendlich für eine Minute entschieden, da dies den Mittelwert bildete.

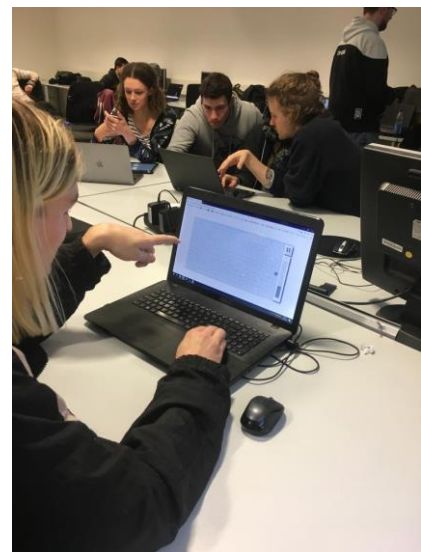
Bei diesem Playtest fiel dann auch die Entscheidung das ursprüngliche Matherätsel durch einen Logiktest, mit einem Hinweis, zu ersetzen, da vielen Testern die Lust an unserem Spiel sonst schnell verging.

Bei einigen Designstreitigkeiten wurde uns bei diesem Spieltest sehr geholfen.

Insbesondere das Design der Modal Dialogs haben wir danach ein wenig umgestellt, um dem Blick des Spielers in der gewollten Reihenfolge auf unsere Texte zu lenken.

Des Weiteren waren für manche die "Weiter-Pfeile" nicht klar erkennbar, so dass wir diese etwas dicker gestalteten.

Letztendlich gab es grundsätzlich ein positives Feedback für unseren Spiel-Prototypen, was unserer Gruppe einen positiven und motivierenden Anreiz gegeben hat.



## Arbeitsteilung

Die Arbeitsteilung funktionierte bei uns sehr ganz gut, da wir alle entschlossen waren ein sehr gutes Projekt abzuliefern und uns dementsprechend auch beteiligt haben.

Dabei war es uns wichtig, dass jeder einen Teil des Spiels selber programmiert.

Nachdem unser Spielkonzept stand, haben wir uns auf den wesentlichen Handlungsstrang konzentriert. So hat Thao sich ans Home-Kapitel gesetzt, Yuliya und Moumita an die Bibliothek und Lars an das Labyrinth. Es haben sich zwei an die Bibliothek gesetzt, da wir dort zwei Handlungsstränge aufbauen wollten.

Die Zeichnungen wurden währenddessen gemacht.

Den Aufwand des Home-Levels haben wir unterschätzt, sodass sich die Arbeit an diesem Kapitel bis zum Ende zog.

Da wir schneller als erwartet fertig wurden, nahmen wir noch Erweiterungen vor.

So entstanden dann nach und nach die Startseite, die Bahn, der Drucker, das Pause-Menü und die Endseiten.

Die Texte von allen Seiten haben wir im Nachhinein alle gemeinsam überarbeitet.

Jedoch verschwamm diese klare Aufgabenverteilung zu Ende hin, da wir gerne auch die Gruppe um Hilfe gefragt mm und uns alle gegenseitig geholfen haben. Ein Umstand, der uns sehr viel gebracht hat, da sich jeder überall eingebracht hat.

AUFGABEN	BEARBEITET VON
Spielkonzept	gemeinsam
Zeichnungen	Yuliya (Home + Rätselbild) Thao (Bibliothek + Drucker) Moumita (Icons + Bahn + Charakter)
Texte, Aufgabenstellungen	gemeinsam
Startseite	Thao
Begrüßung des Spielers	Lars
Home	Thao Session Storage: Thao, Moumita
Bibliothek	Yuliya (Rätsel) Moumita (Galgenmännchen)
Bahnsequenzen	Moumita
Drucker	Thao
Labyrinth	Lars Timer: Lars, Yuliya
Verabschiedung des Spielers + Credit	Lars
Game Over	Lars
Pause-Menü	Yuliya

Letztendlich kann man sagen, dass unsere Einteilung sehr gut war und besonders durch die einzelnen HTML-Seiten jeder in der Lage war, seine eigene Zeit produktiv einzuteilen, ohne Anderen in die Quere zu kommen.

# Technische Merkmale

Um das Motto eines Notizbuches aufzugreifen, haben wir uns aktiv gegen einen Sound entschieden.

## Datenstruktur

“Last Minute!” war anfangs viel kürzer geplant (*siehe Anlage*), doch mit weiteren Ideen wurde es viel komplexer. Dadurch haben wir uns entschieden das Programm auf mehreren HTML-Seiten zu strukturieren, um uns eine bessere Übersicht zu bewahren. So konnten wir die verschiedenen Minispiele einzeln programmieren.

Damit man nicht den Überblick verliert haben wir die Dateien in verschiedenen Ordner aufgeteilt. So sind CSS Dateien im Ordner “CSS”, Javascript Dateien im “JS”-Ordner und Bilder im Ordner “pic”, dies hat ebenso ein paar Unterordner für Bilder.

Zu jeder HTML-Seite gehört ihr zugehöriges CSS und Javascript. Jedoch haben alle eine Verlinkung zu einen allgemeinen CSS (*style.css*), damit die optische Fläche einheitlich ist, und eine Javascript (*storage.js*), sofern sie dies benötigen.

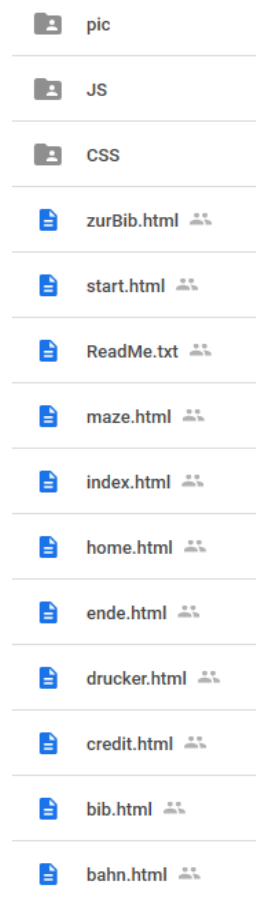
## Drag and Drop

Der Spieler befindet sich in seinem Zimmer und packt seine Tasche für den Tag. Er kann verschiedene Gegenstände mit Hilfe von drag ‘n drop in seinen Rucksack packen.

Die wichtigsten Felder des Home.HTML sind der *#Content* und das *#Inventar*. Im *#Content* befinden sich die Items, die der Spieler in das *#Inventar* ziehen kann. Jedes Item besitzt einen Wert, eine Klasse und eine ID. Damit ein Objekt gezogen und verschoben werden kann, müssen wir jedem das Attribut *draggable="true"* geben. Dies macht jedes Element ziehbar. Wenn das Item nun gezogen wird, wird eine Funktion aufgerufen (*ondragstart="drag(event)"*). Sobald ein Objekt mit gedrückter Maustaste gezogen wird, übergibt das Event die Information dem *dataTransfer-Objekt*. Es speichert die Eigenschaften des gezogenen Elements und der ID zum Zielort, wo es hingezogen wird.

Sobald sich ein gezogenes Item über der Zielzone befindet, wird die Funktion *“allowdrop(event)”* aktiviert. Damit werden die Standardwerte abgeschaltet und das Ablegen ermöglicht.

Wenn jetzt etwas abgelegt wird, wird die zweite Funktion gestartet (*ondrop="drop(event)"*). Diese hat mehrere Aufgaben. Zuerst werden die Standardeinstellungen ausgeschaltet. Wir erstellen eine Variable *“Data”*. Diese soll die Informationen des Objekts erhalten die wir bereits beim *ondragstart="drag(event)"* gespeichert hatten. Befindet sich ein Element bereits im Zielort, kann es nicht abgelegt werden und es kehrt zurück zum Ursprung. Sonst erhält das Zielelement ein



Kindelement, welches als Objekt bei "data" gespeichert wurde. Zudem werden die ID der Zielzone, des Images und die Klasse des Ziels an *addItem()* weitergegeben.

## Minispiele

Bei den Minispielen war uns besonders wichtig den Spaß am Spiel auch nach mehreren Durchläufen zu erhalten.

Deshalb haben wir uns entschieden, mehrere mögliche Wörter beim Galgenmännchen zu Verfügung zu stellen. Wobei das Programm beim Laden der Seite immer eines dieser zufällig auswählt.

Auch beim Labyrinth und beim Bücherwurm-Minispiel wollten wir es abwechslungsreich gestalten. Beim Laden des Labyrinths bewegt sich ein Feld zufällig, mithilfe der "*random*" - Funktion über das Feld, bis es in eine Sackgasse läuft. Passiert dies, läuft er alle Felder ab und die läuft zu Feldern die nicht besucht wurden. Bei dem Bücherwurm wird das Essen ebenfalls mit der "*random*" - Funktion innerhalb des Canvas hingelegt.

Für die Darstellung des Canvas von dem Bücherwurm-Minispiel und des Labyrinths haben wir die Bibliothek p5.js genutzt, da dies die Darstellung des Canvas erleichtert hat. Die Funktionen von p5.js vereinfachen es bestimmte Objekte zu "Zeichnen", wie beispielsweise Rechtecke, Kreise oder ähnliches.

# Post Mortem

## Lars Obist

Das Projekt, welches aus einem relativ simplen Gedanken entstanden ist gefällt mir sehr. Vom Inhalt sowie vom Aussehen ist es ein gelungenes Spiel. Die Art und Weise wie in unserem Projekt gearbeitet wurde hat mir sehr gefallen. Zu Beginn gab es einige Probleme und Uneinigkeiten zu lösen, dies lag aber auch daran, dass man keine Erfahrungen in diesem Bereich hatte. Nachdem wir gemeinsam die Rahmenbedingungen geschaffen haben, und die Aufgaben verteilt hatten konnte jeder selber für sich arbeiten, was mir sehr entgegen kam da man sich zunächst nicht in die Quere kam. Falls Probleme auftraten, war dies nie ein großes Problem da die Gruppe sich jederzeit unterstützt hat.

Wenn man ein ähnliches Projekt erneut starten würde, gäbe es wenig, was ich ändern würde. Kleinigkeiten die ich ändern würde wären, dass man stärker auf Responsive Design achten, dies haben wir nur teilweise beachtet.

Im großen Ganzen bin ich sehr zufrieden mit dem Projekt, und stolz auf unsere Gruppe.

## Yuliya Litvin

Vorab möchte ich gerne bemerken, dass ich stolz auf unser Projektergebnis bin. Noch vor wenigen Monaten hätte ich mir persönlich nicht vorstellen können, in nur so kurzer Zeit ein Spiel auf die Beine zu stellen. Alleine wäre das nicht zu schaffen gewesen, insbesondere, wenn man sich in diesem Themenbereich vorher kaum – bis eigentlich fast gar nicht - ausgekannt hatte. Und da kann ich, denke ich, für meine Gruppenmitglieder sprechen, wenn ich sage, dass wir in diesem kurzen Zeitraum eine Menge Neues kennengelernt, uns beigebracht und bewältigt haben, auch wenn der Eine oder Andere zum Teil schon etwas Vorwissen besaß.

Mir hat das Projekt sehr gefallen und vor allem Spaß gemacht, was nicht wenig insbesondere an meinen Gruppenmitgliedern lag. Ich finde, wir haben als Gruppe ausgezeichnet funktioniert, auch, wenn es natürlich hin und wieder Unstimmigkeiten, Diskussionen und stressige Situationen gab. Aber gerade aufgrund unserer tollen Zusammenarbeit gab es keine Probleme, die wir zusammen nicht bewältigen konnten. Wenn man zum Beispiel bei seiner zugeteilten Aufgabe Schwierigkeiten hatte – da wir ganz am Anfang die Aufgaben untereinander, so fair wie möglich, aufgeteilt hatten -, konnte man sich an die Gruppenmitglieder wenden und bekam ohne Weiteres Hilfe und Unterstützung. Dadurch haben wir unter anderem auch viel voneinander gelernt. Das einzige worauf ich beim nächsten Mal, wenn wir ein Spiel programmieren sollten, mehr achten würde, wäre das „Responsive Design“, weil wir uns dieses Mal nicht besonders intensiv damit beschäftigt hatten.

Insgesamt empfand ich das Projekt als eine gemeinsame schöne, spaßige Zeit mit meiner Gruppe.

## Moumita Ahmad

Unser Projekt startete Mitte November und nachdem wir uns über unseren eigenen Vorlieben klar wurden, stand auch schnell das Spielkonzept.

Ein Punkt, mit dem wir anfangs mehr Schwierigkeiten hatten, war mit der eigentlichen Programmierung anzufangen. Dies lag vorrangig an unseren geringen Vorkenntnissen, die es uns erschwerten einen Anfang zu finden und richtig abzuschätzen, wie lange viel für bestimmte Aufgaben brauchen würden. Dadurch waren wir am Ende etwas mehr im Stress, als nötig gewesen wäre.

Dennoch haben wir es die ganze Zeit über geschafft gut zusammen zu arbeiten und größere Probleme gemeinsam auch zu lösen. Durch dieses gute Gruppenklima waren wir alle sehr motiviert und wussten, dass wir uns aufeinander verlassen können.

Trotzdem, eine Sache, die aus diesem Projekt gelernt habe, ist es mehr eine Bibliothek wie "jQuery" zu nutzen. Dies hätte wahrscheinlich auch unsere Arbeit etwas vereinfacht.

Doch letzten Endes kann ich sagen, dass mir dieses Projekt viel Spaß gemacht hat und ich auch mit dem Ergebnis sehr zufrieden bin. Ich habe, wie alle aus meiner Gruppe, denke ich, sehr viel aus diesem Projekt gelernt.

## Thao Nguyen

Dieses Projekt entstand durch eine Aufgabe vom Labor, eine Idee die durch einen einfachen Impuls entstand und ein Team das sich erst paar Wochen kannte. Durch unseren unterschiedlichen Wissensstand war es schwierig uns gegenseitig einzuschätzen. Anfangs wusste ich nicht mal was JavaScript überhaupt war. Ich machte es mir schwieriger obwohl ich es mir hätte leichter machen können.

Mit den Erweiterungen kam es zu mehr Arbeit und umso unübersichtlicher wurde die HTML-Seite. Deswegen teilten wir das Programm auf mehreren HTML-Seiten auf, damit jeder arbeiten konnte. So hatte jeder seine Aufgabe und war auf sich allein gestellt. Trotz Schwierigkeiten brachten wir uns die Fähigkeiten selber bei, um voranzukommen. Herausfinden was das Problem sei und daraus lernen, um weitere zu vermeiden.

Am Ende gab es noch Bugs und die Zeit drängte, vielleicht sollte man beim nächsten Projekt die Zeit besser einteilen. Aber auch damit machten wir kurzen Prozess. Man konnte sich stets auf die Gruppe verlassen, falls man nicht weiterkam.

Aus einer hoffnungslosen Situation, haben wir unsere Strategie verändert und uns selber übertroffen. Zusammen sind wir mit dem Spiel gewachsen und wurden weiser. Das Spiel ist bei Weitem nicht perfekt, aber wir hielten zusammen und versuchten unser Bestes. Doch das macht es so verdammt schön.

Wir können stolz sein auf das was wir erreicht haben.



# Quellen

Rätsel:

[Eingabefeld](#)

p5.js:

[Grundlagen](#)

[Create Canvas](#)

Labyrinth:

[Wikipedia Artikel - Maze Algorithm](#)

[Maze & Timer Code Beispiel](#)

[Maze Algorithm Teil 1](#)

[Maze Algorithm Teil 2](#)

[Maze Algorithm Teil 3](#)

[Maze Algorithm Teil 4](#)

[ASCII-, DOS-Latin-1-, Windows-1252- und HTML-Zeichencodes sowie deutsche Tastatur-Scancodes](#)

Timer:

[Timer Progress Bar](#)

[Timer Tutorial](#)

[Div Element drehen](#)

[Timer Funktion](#)

[W3School - Progress Bar](#)

Bücherwurm:

[How to build a Snake game](#)

[Snake Tutorial](#)

HTML-CSS-JavaScript-Research:

[CSS animation-delay Property](#)

[CSS z-index • Übereinander positionierte Blöcke](#)

[HTML-Elemente durch JavaScript ändern](#)

[Modal Popup From Scratch](#)

[How to disable HTML button using JavaScript?](#)

[HTML Attribut onkeydown: Auszuführender JavaScript-Code bei Drücken einer Taste](#)

[HTML Attribut onload von HTML Tag frameset: Auszuführender JavaScript-Code, wenn alle Frames geladen wurden](#)

[HTML Attribut onsubmit von HTML Tag form: Auszuführender JavaScript-Code bei Absenden eines Formulars](#)

[HTML <input> align Attribute](#)

[HTML onclick Event Attribute](#)

[HTML5/CSS3 Modal Dialog Box; no JavaScript](#)

[Javascript keyCode, charCode, which • Tastatureingaben](#)

[JavaScript Timing Events](#)

[JavaScript/DOM/Document/getElementById](#)

[JavaScript/Tutorials/Zeitberechnung](#)

[Keyboard and Mouse buttons events](#)

[Class KeyEvent](#)

[onblur Event](#)  
[onfocus Event](#)  
[onmouseover Event](#)  
[target Event](#)

Browser Logo:

[Favicon](#)

Storage

[Webstorage](#)  
[Webstorage 2](#)  
[sessionstorage](#)  
[localStorage](#)  
[JSON](#)

Drag and Drop:

[drag and drop tutorial](#)  
[HTML DOM Events](#)  
[ondragover Event](#)

Intro/ Arrow:

[Animation](#)

Galgenmännchen

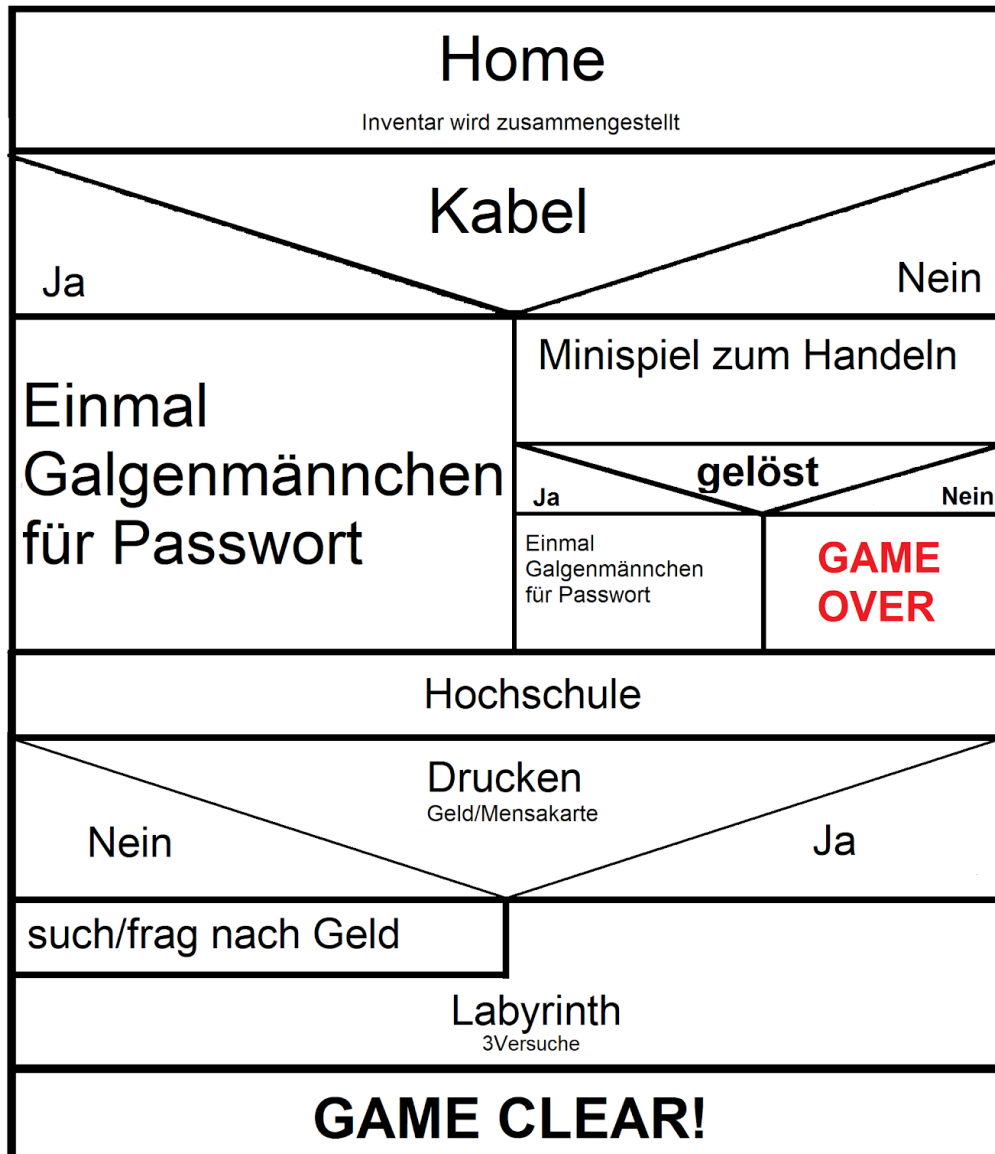
[Hangman Game](#)  
[Code example](#)  
[Writing Hangman](#)

Bahn-GIF:

[Frame by frame animation](#)

# Anlage

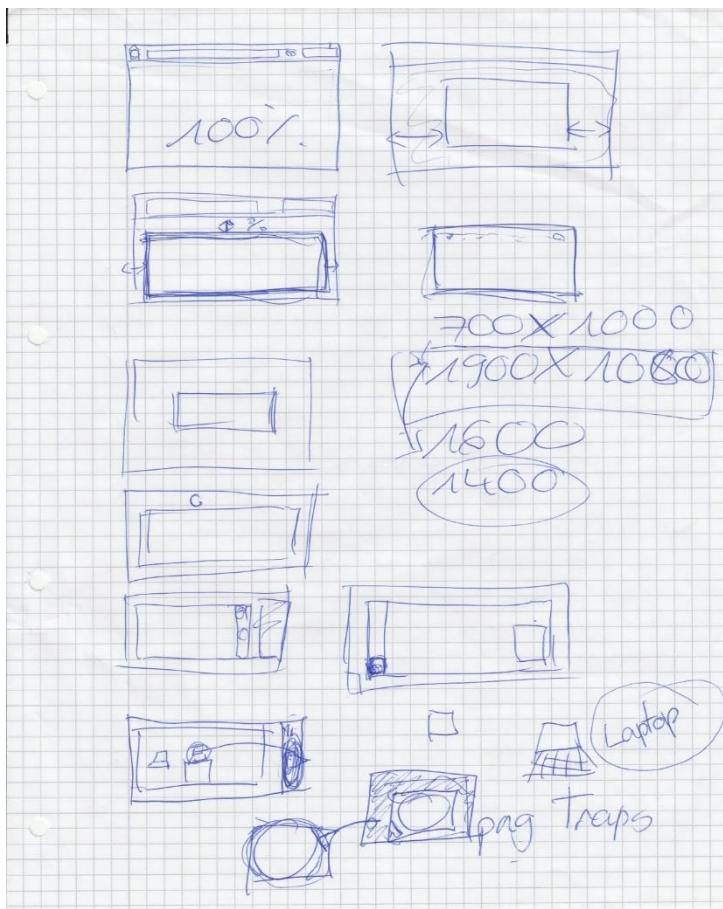
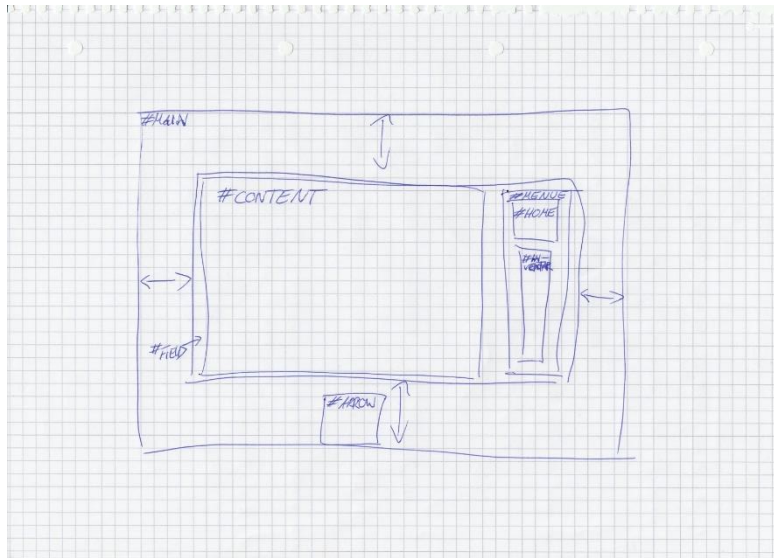
## Ursprünglicher Spielplan



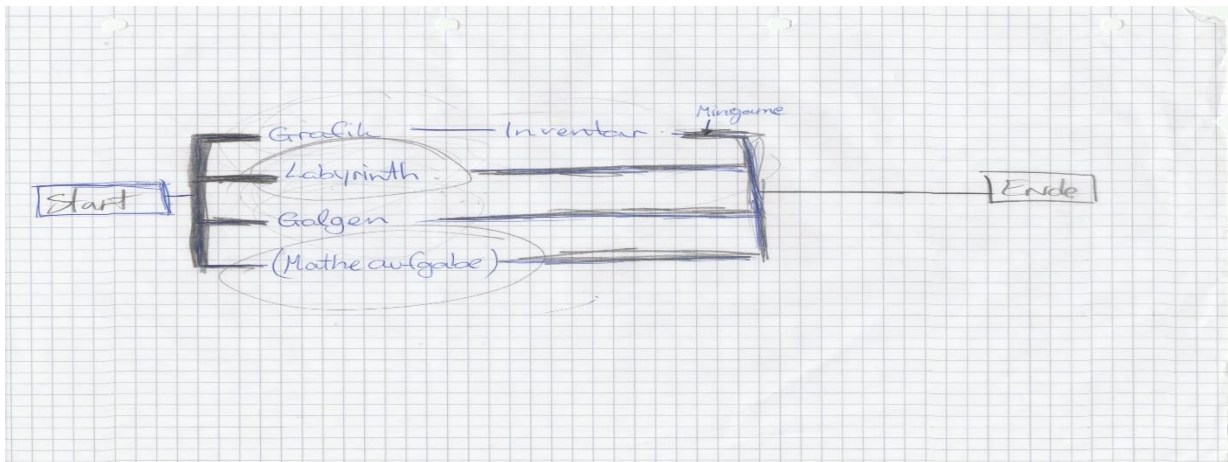
## Finaler Spielplan

Home					Snake	
zur Bibliothek						
Laptop Abfrage						
Nein			Ja			
Zurück zu Home			Bibliothek			
Tischauswahl						
linker Tisch			rechter Tisch			
Kaffee			Kabel			
Kein Item		Item		Item		
GAME OVER	Galgen- männchen		Galgen- männchen		Kein Item	
					Rätsel	
				gelöst	nicht gelöst	
				Galgen- männchen	GAME OVER	
Bahn						
Ticket						
Nein			Ja			
GAME OVER			Universität			
Drucker						
Geld						
Nein			Ja			
GAME OVER			Labyrinth			
			GAME OVER		Game clear!	

# Spiellayout - Skizzen

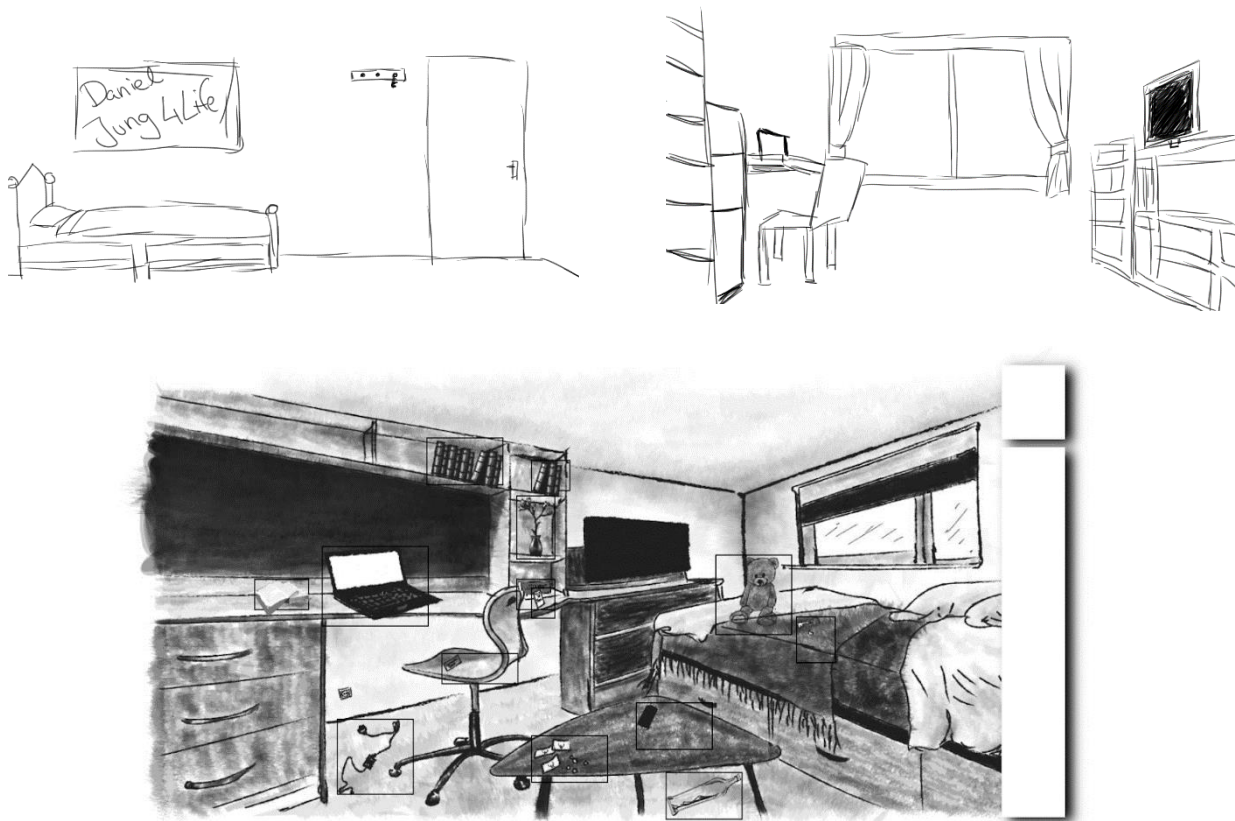


## Einteilung der Aufgaben mit Gantt-Diagramm



## Entwicklung eines Kapitels

Beispiel home.html



# LAST MINUTE!

