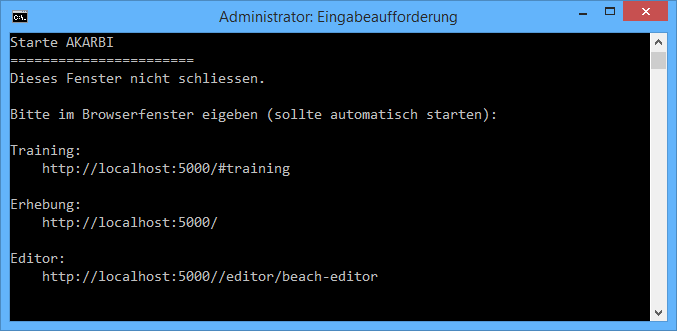
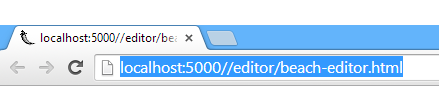
# Start des Programms

Um das Programm zu starten wird die Datei AKARBI.BAT durch einen Doppelklick ausgeführt.

Es erscheint das folgende Fenster, das während der Erhebung nicht geschlossen werden darf:

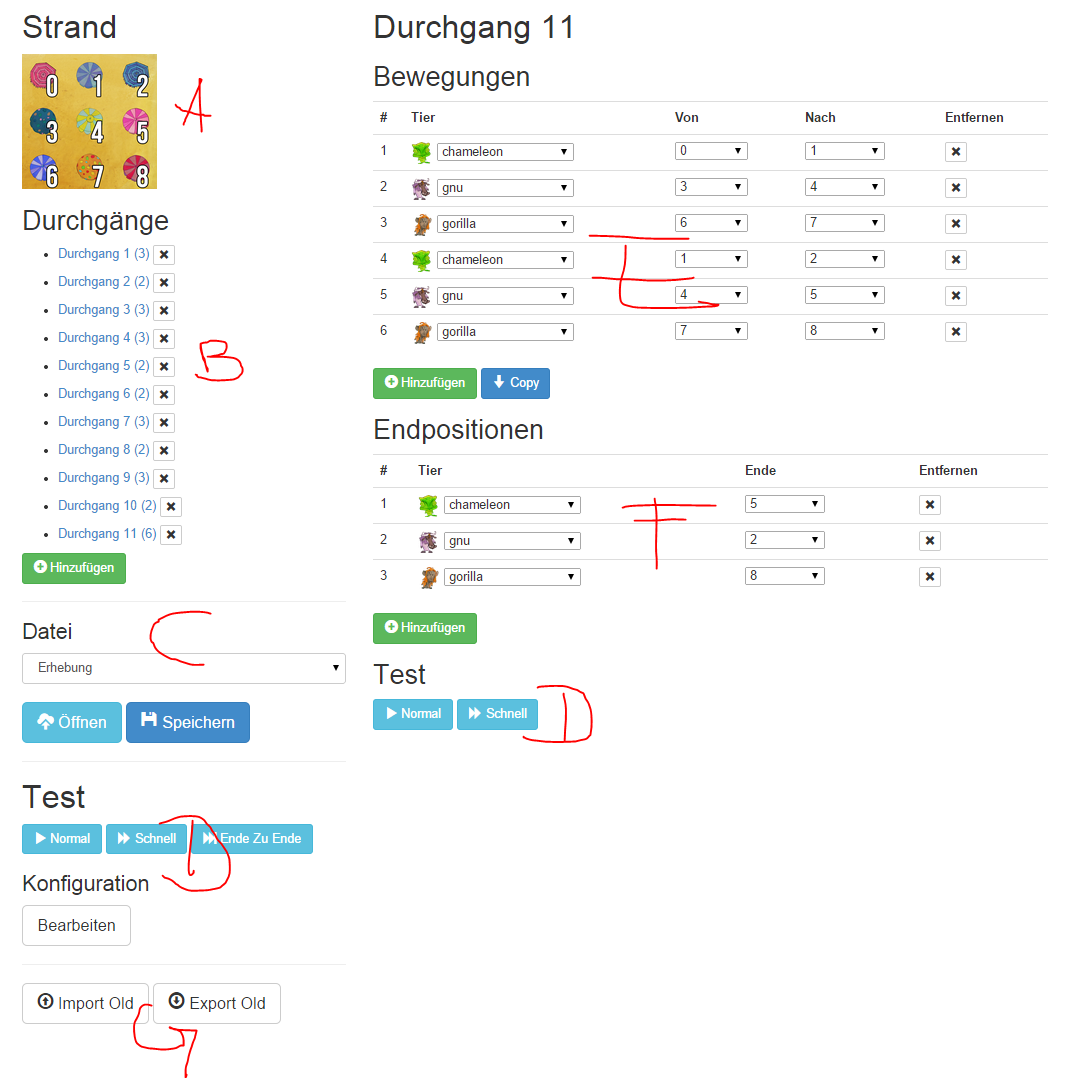


Es sollte automatisch ein Browserfenster aufgehen, das bereits den Editor und die Strandaufgabe lädt. Sollte das nicht der Fall sein, öffnen Sie ein Browserfenster und fügen Sie die folgenden URLS in die Adresszeile ein:



|  |  |
| --- | --- |
| URL |  |
| http://localhost:5000/ | Öffnet die Strandaufgabe |
| http://localhost:5000/#training | Öffnet die Strandaufgabe im Trainingsmodus |
| http://localhost:5000//editor/beach-editor | Öffnet den Editor |

# Editor



Fügen Sie einzelne Durchgänge hinzu, indem sie links im Menü den Button Hinzufügen Nutzen (B).

## Bewegungen Hinzufügen

Auf der rechten Seite kann dann der Durchgang bearbeitet werden (E). Wählen Sie ein Tier, und wo dies auftauchen soll. Verwenden Sie die Übersicht (A) um ein Feld auszuwählen. Fügen Sie zunächst alle Bewegungen hinzu.



In diesem Beispiel bewegt sich der Affe von Feld 0 links oben, um ein Feld nach rechts (vgl. Übersicht A). Danach bewegt sich das Chameleon von Feld 1 nach Feld 2.

## Endpositionen bestimmen

Am Ende eines Durchlaufs werden die Tiere aufgedeckt. Wo sie sich dann befinden wird bei den Endpositionen (F) festgelegt. Sie können die Liste der Bewegungen kopieren um die Endpositionen anpassen. Verwenden Sie dazu den Copy Button.

## Testen

Um die Konfiguration des Durchgangs im Lifesystem zu überprüfen, können Sie ihn individuell laufen lassen (D). Zuvor müssen Sie die Daten jedoch abspeichern (C).

# Fehlertoleranz

Die Software wurde aus Kostengründen etwas simpel gehalten, daher hat sie nicht viele Extras. Daher müssen Sie darauf achten, dass die Bewegungen, die sie einstellen auch Sinn ergeben.

Es gelten dabei folgende Regeln:

* Sie können die Tiere nacheinander „aufrufen“, also beispielsweise erst einen Affen, dann einen Chamäleon laufen lassen.
* **Es gibt ein Tier nur einmal.** Wenn sie einen Affen hinzufügen, der sich von 0 🡪 1 bewegt und dann von 1 🡪 2 ist dies derselbe Affe, egal ob dazwischen ein anderes Tier bewegt wird.
* „Auslassungen“ werden dabei ignoriert. Wird ein Affe wie folgt bewegt:   
  0🡪1, 1 🡪 2, 6 🡪 7, dann bewegt sich der Affe 0 🡪 1 🡪 2 🡪 7. Das Feld 6 wird dabei ignoriert.
* **Die Bewegungen sind immer gleich schnell**. Daher sollten Sie darauf achten, dass ein Tier nicht 2 Felder oder mehr in einer Bewegung läuft. Bewegungen sollten eine Feld-Differenz von 1 aufweisen, da sonst in zu kurzer Zeit zu viel Strecke abgelaufen wird. Diese Bewegungen sind dann also sehr schnell. Beispiel: Bewegungen von 0 🡪 1 dauert eine Sekunde und ist „weich“, von 0 🡪 8 ist mehr Strecke und die Bewegung ist dann recht hektisch.
* **Tiere schließen sich nicht aus**. Sie überdecken sich! Wenn sie also in der Konfiguration einen Affen 0🡪1 und ein Chamäleon 0🡪3hinzufügen, werden beide unter Schirm 0 aufgedeckt und man erkennt nur das (farblich dominante) Chamäleon.
* Es wird nicht verhindert, dass zwei Tiere unterm selben Schirm landen.

# Speichern

Wenn Sie eine Liste mit Durchgängen (B) erstellt haben, können Sie diese als Datensatz für das Training oder als Datensatz für die eigentliche Erhebung abspeichern (D). Passen Sie dabei auf, dass sie nicht aus Versehen die Dateien überschreiben.

# Testen

Am Ende können Sie den gesamten Ablauf testen. Speichern sie zunächst den Datensatz ab (C).

Rufen Sie dann den Test auf. Bei **Normal** handelt es sich um einen regulären Durchlauf. **Schnell** lässt die Zeit schneller ablaufen und spielt die Animationen schneller ab. Es wird kein Sound wiedergegeben. Dieser Modus ist sinnvoll, um die Endpositionen zu prüfen. **Ende zu Ende** spielt den kompletten Satz an Aufgaben durch und bestätigt jede Frage mit einem „Positiv“- Haken, bis die Aufgabe zu Ende ist. Am Ende der Aufgabe wird eine Ergebnisdatei geschrieben.

# Erhebung

Die Erhebung selbst läuft unter <http://localhost:5000/> ab.

Dieser URL können verschiedene Parameter übergeben werden. Ein Parameter wird übergeben, indem der Anwendung ein HASH (#) hinzugefügt wird. Wenn Sie beim Testen auf die URL achten, sehen Sie, wie sich diese verändert.

|  |  |  |
| --- | --- | --- |
| Parameter | Beispiel | Beschreibung |
| #debug | <http://localhost:5000/#debug> | Alles wird ganz schnell. Kein Sound. |
| #<ZAHL> | <http://localhost:5000/#1> | Spielt einen Durchlauf mit der Nummer x ab. |
| #auto |  | Alle Aufgaben werden mit OK bestätigt |
| #autodebug |  | Schnelldurchlauf bis zum Ende |
| #training |  | Trainingsmodus |

Die Parameter können kombiniert werden, indem man sie einfach hintereinander weg schreibt:

<http://localhost:5000/#trainingautodebug2>

Spielt also Trainingsdurchgang Nummer 2 ab, läuft dabei sehr schnell und wird am Ende immer mit OK beantwortet. Wenn Ihr Büro mal kalt ist, können Sie diesen Parameter verwenden und der Rechner spielt so lange diesen Durchgang ab.

# Ergebnisdaten

Die Daten werden mit einem Timestamp im ISO Format abgespeichert und beinhalten JSON Daten.

 Sie befinden sich nach der Erhebung im Ordner „data“

# Weitere Konfiguration

Um ein wenig Feintuning durchzuführen, können sie die Parameter im Code des Programms anpassen. Öffnem Sie die Datei /static/js/game.js. Zu Beginn der Datei finden Sie folgende Stelle:

var config = {

ANIMAL\_WIDTH: 110,

ANIMAL\_HEIGHT: 110,

PARASOL\_SIZE\_PERCENT: 0.60,

ANIMAL\_SIZE\_PERCENT: 0.4,

PLAY\_SOUND: true,

RESOLUTION\_WIDTH: 2880,

RESOLUTION\_HEIGHT: 1800,

HIDE\_FEEDBACK\_AFTER: 2000,

HIDING\_TIMEOUT: Phaser.Timer.SECOND \* 2,

HIDING\_DURATION: 500,

MOVEMENT\_TIMEOUT: Phaser.Timer.SECOND \* 6,

MOVEMENT\_DURATION: 1500,

WAIT\_BETWEEN\_MOVES: 1000,

WAIT\_BEFORE\_DISCOVERY: 1000

};

Hier können Sie Werte anpassen, wenn es denn sein muss. Diese Tätigkeit sollte aber besser einem Programmierer ihrer Wahl überlassen werden, der sich mit JavaScript auskennt. Trotzdem seien hier einige Werte erklärt:

|  |  |  |  |
| --- | --- | --- | --- |
| Schlüssel | Datentyp | Default | Bedeutung |
| PARASOL\_SIZE\_PERCENT | Number | 0.6 | Prozentuale Größe der Schirme |
| ANIMAL\_SIZE\_PERCENT | Number | 0.4 | Prozentuale Größe der Tiere |
| PLAY\_SOUND | Boolean | True | Sound abspielen? |
| RESOLUTION\_WIDTH | Number | 2880 | Qualität des Hintergrundbildes |
| RESOLUTION\_HEIGHT | Number | 1800 |
| HIDE\_FEEDBACK\_AFTER | Number | 2000 | Wie lang soll das Feedback angezeigt werden? |
| HIDING\_TIMEOUT | Number | 2000 | Nach wie vielen Sekunden startet das Verstecken? |
| HIDING\_DURATION | NUMBER | 500 \*\* | Wie “schnell” verstecken sich die Tiere? |
| MOVEMENT\_TIMEOUT | Number | 6000 | Nach wie vielen Sekunden solls losgehen? |
| MOVEMENT\_DURATION | Number | 1500 | Wie lange ist ein Tier sichtbar? |
| WAIT\_BETWEEN\_MOVES | Number | 1000 | Wie lange bis zum nächsten Tier |
| WAIT\_BEFORE\_DISCOVERY | Number | 1000 | Wie lange wird nach der nächsten Bewegung gewartet? |

## Anmerkungen

Alle Zahlen beziehen sich auf Millisekunden. 1000ms sind 1 Sekunde. Die Bewegungsdauer ist also 1500ms = 1 ½ Sekunden.

\*\*) Das verstecken geschieht in 2 Schritten, ein „Sprung nach oben“ und ein Verschwinden unter dem Schirm. Diese beiden Bewegungen dauern je 1x HIDING\_DURATION und ergeben zusammen die Gesamtdauer der Versteckzeit eines Tieres. Bis ein Tier nicht mehr sichtbar ist, vergehen also 2x500ms = 1 Sekunde.