

BAUEN SIE IHR ERSTES SPIEL: „DAS VERRÜCKTE LABYRINTH“ (MAZE CRAZE)

Ziel dieser Dokumentation ist es, Sie beim Erstellen Ihres ersten Spiels mit der Software AgentSheets unterstützen. Am Ende dieses Leitfadens haben Sie ein funktionierendes Computerspiel erstellt, das Abbildung 1 ähneln sollte. Es kommen einige Begriffe aus der Computersprache vor, die allerdings erklärt werden, bevor sie im weiteren Kontext wiederverwendet werden.

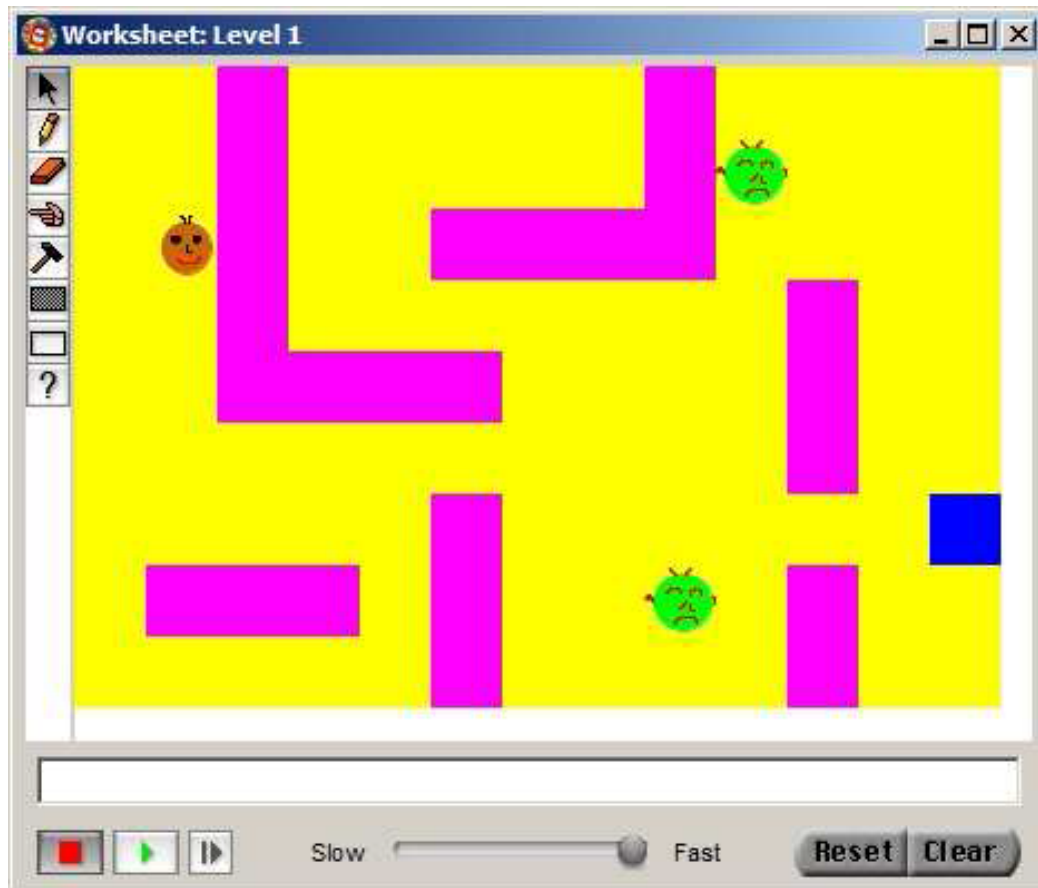


Abbildung 1. Das fertige „Verrückte Labyrinth“-Spiel

Inhalt

1. Ein neues Projekt entsteht
2. Das Erstellen von Agenten
3. Wir erstellen ein Worksheet
4. Wir programmieren die Bewegungen des „Ich“-Agenten
5. Das „Ich erreicht das Ziel – Spiel gewonnen!
6. Wir erstellen die restlichen Agenten
7. Das „Ich“ wird attackiert – Spiel verloren!

1. Ein neues Projekt entsteht

Mit einem Doppelklick auf das AgentSheets Zeichen (icon)  startet das Programm.

Selektieren Sie File->New Project... , um ein neues AgentSheet-Projekt zu starten.

Danach öffnet sich ein Dialogfenster, in welchem dem Projekt ein Name gegeben werden kann und man definieren kann, wo man das Projekt speichern möchte.

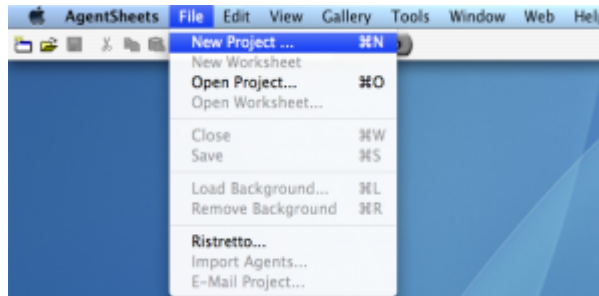


Abbildung 2. Wie man ein neues Projekt erstellt.

Wir nennen das Projekt „Labyrinth1“ und klicken *OK*. Um ihr Spiel zu personalisieren, wäre es auch empfehlenswert, die Anfangsbuchstaben Ihres Namens zu verwenden, z.B. *LabyrinthNAE*. Bestätigen Sie die Eingabe mit der Aufforderung *OK*.

Nun erscheint ein weiteres Dialogfenster, in dem die **Pixelgrösse der Agenten** definiert wird. Hier ist per *default* (= die Voreinstellung) die Pixelgrösse 32x32 hinterlegt. Diese übernehmen Sie mit *OK*. Nun sollte ein Fenster mit der Galerie (gallery) auf dem Bildschirm links oben erscheinen.

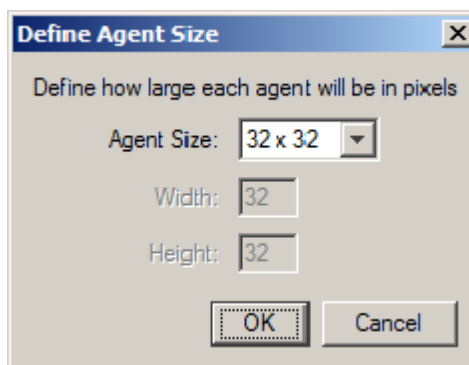


Abbildung 3. Die Pixelgrösse wird definiert

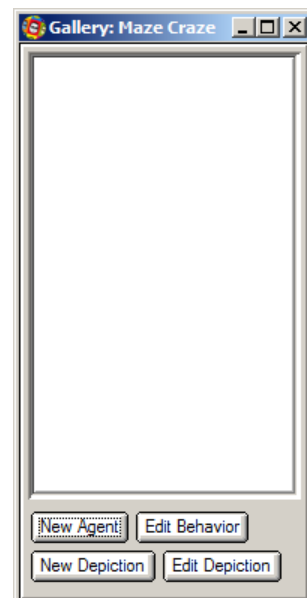


Abbildung 4. Die Galerie

2. Das Erstellen von Agenten

Im Folgenden erstellen wir graphisch verschiedene Objekte, in der visuellen Programmierung auch „Agenten“ (vom englischen Wort *agents*) genannt.

Wenn man im Menüfeld oben auf „**Gallery**“ **klickt**, erhält man eine Auswahl und wählt dort die Möglichkeit „**New Agent**“ (neuer Agent). Wir nennen diesen Agenten (= *agent*) „Ich“, es wird die Hauptfigur des Spiels. Sie sollte nun in dem neuen Dialogfenster der Galerie auf der linken Seite des Bildschirmes erscheinen (mit dem Bild von Abbildung 6 - als Voreinstellung hinterlegt). Nun gestalten wir das Objekt „Ich“.

Durch Doppelklick auf das Namensfeld „Ich“ (derzeit noch mit einem Gesicht) oder indem ich auf die Eingabeoption „Edit Depiction“ (editiere das Bild) klicke, öffnet sich ein neues Dialogfenster [Graphik Editor](#), in dem ich ein Objekt zeichnen und gestalten kann.



Abbildung 5. Galerie mit Agent

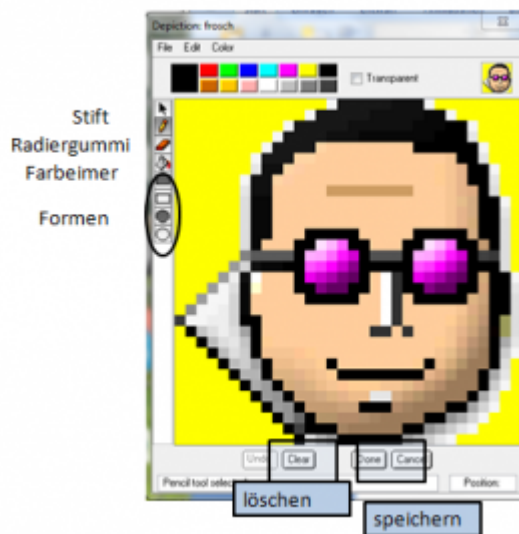


Abbildung 6. Graphik Editor

Klicken Sie zuerst auf *Clear*, damit der Zeicheneditor für eine neue Bildeingabe bereit ist. Der Mann mit der lila Brille ist nur ein „*default*“ Bild (default = Voreinstellung) und kann gelöscht werden, so dass ein leeres Graphikfenster neue Zeichnungen erlaubt.

Nun wollen wir unser „**Ich**“ **zeichnen**: Wählen Sie eine Farbe aus und klicken Sie auf den „*Stift*“, um mit dem Zeichnen zu beginnen. Wenn man eine Umrandung gezeichnet hat, kann man die Fläche innerhalb dieser Umrandung mit dem sogenannten „*Farbeimer*“ füllen. Der Farbeimer ist das 4. Symbol (= *icon*) auf der linken Menüseite des Editors. Wenn man eine Eingabe rückgängig machen möchte, kann man dies mit dem Eingabefeld *Undo* machen. Außerdem hat man die Möglichkeit, Formen zu zeichnen (runde mit dem Kreissymbol oder rechteckige mit dem Rechteck). Es gibt auch den sogenannten „*Radiergummi*“ (das 3. Symbol auf der linken Seite), mit dem Fehler entfernt werden können. Interessant für ein graphisch zufriedenstellendes Ergebnis ist noch die Eingabe *Color* → *Mask Color* → *White* im oberen Menü des Editors. Damit kann man den Hintergrund transparent machen, was im fertigen Spiel eine bessere Graphik ergibt. Außerdem hat man bei „*Color*“ die Möglichkeit, auf eine größere Farbpalette zuzugreifen.

Klicken Sie auf die Schaltfläche *Done* im unteren Bereich des Editors, wenn der Frosch fertig gezeichnet ist. Damit ist die erste Graphik gespeichert, kann aber jeder Zeit über *Edit Depiction* in der Galerie verändert werden.

Wiederholen Sie nun den Vorgang, um zwei weitere Agenten zu erstellen: den „Hintergrund“ und das „Ziel“. Ihre Agenten-Galerie sollte nun in etwa so aussehen:

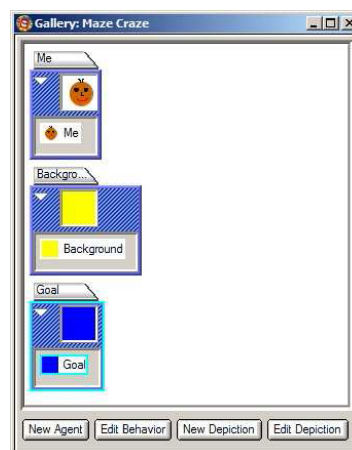


Abbildung 7. Galerie mit Agenten „Ich“, Hintergrund“ und „Ziel“.

3. Wir erstellen ein *Worksheet* (Spielfeld)

Das Worksheet ist unser Spiel-Level bzw. das Spielfeld, auf dem wir unsere Agenten platzieren.

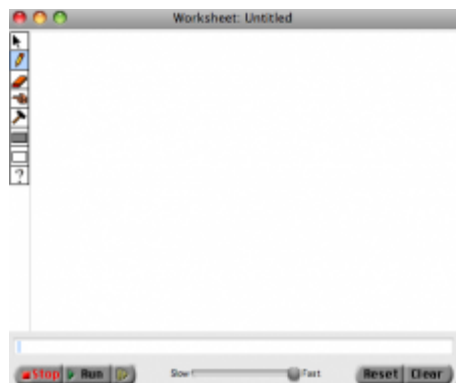


Abbildung 8: Leeres Spielfeld (worksheet)

Gehen Sie zum Menü in der oberen Leiste des Bildschirms und wählen Sie „File → New Worksheet“, um ein neues Spielfeld zu erstellen. Es erscheint ein **neues Dialogfenster mit einem leeren Spielfeld** (*Worksheet: Untitled*). Sie können die Größe des Spielfeldes anpassen, indem Sie den Rand des Spielfeldes mit dem Pfeil in die gewünschte Grösse ziehen. Auf dieses leere Spielfeld platzieren wir unsere verschiedenen Agenten (agents), so dass wir am Ende ein fertiges „Labyrinth“-Spiel (wie in Abbildung 1) erhalten. Wir beginnen mit dem „Hintergrund“ und ergänzen dann das „Ich“ und das „Ziel“.

Selektieren Sie auf dem Spielfeldmenü in der linken Menüleiste den Stift, um **Objekte auf das Spielfeld (Worksheet) zu platzieren**. Nun wählen wir in der Galerie das Objekt „Hintergrund“ aus, indem wir es anklicken. Dabei wird es hellblau umrahmt. Nun können die „Hintergrund“-Elemente einzeln auf dem Spielfeld platziert werden. Schneller geht es, wenn man in der Menüleiste des Spielfeldes das ausgefüllte Rechtecksymbol wählt. Damit kann man das gesamte Spielfeld mit einer Aktion ausfüllen. Wie auch schon beim Graphikeditor kann man mit dem Radiergummi eventuelle Fehler löschen.

Ergänzen Sie nun auf gleiche Weise das „Ich“ und das „Ziel“ auf dem „Hintergrund“. Das Spielfeld sollte anschliessend in etwa so aussehen:

Nun ist ein guter Zeitpunkt, Ihr *Worksheet* zu **speichern**, denn andernfalls verschwinden alle Neuerungen beim Drücken der *Reset*-Taste. Gehen Sie dafür, während das Worksheet-Fenster selektiert ist, in der Menüleiste auf **File → Save**. Geben Sie dann dem Worksheet einen passenden Namen (z.B. Labyrinth Level 1) und drücken Sie speichern (*Save*).

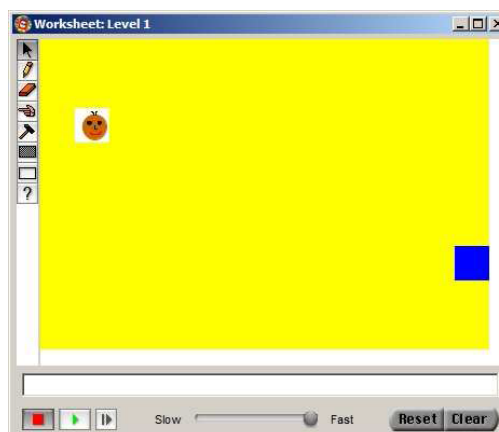


Abbildung 9. Das Spielfeld mit den Objekten „Hintergrund“, „Ich“ und „Ziel“.

4. Wir programmieren die Bewegungen des „Ich“-Agenten

Nun geht es darum, den Agenten ein bestimmtes Verhalten zuzuordnen. Wie bewegen sich die Agenten und wie interagieren sie während des Spiels? Wir beginnen mit dem „Ich“. Es soll sich in verschiedene Richtungen bewegen, wenn wir bestimmte Tasten auf der Tastatur bedienen.

Im Galerie-Fenster selektieren Sie den „Ich“-Agent und drücken dann auf „Edit Behavior“ unten im Galeriefenster. Dadurch werden drei neue Fenster geöffnet: die „Conditions“-Palette (blau), mit einer Liste von wählbaren Bedingungen, der „Behavior“-Editor, der noch ohne Befehle ist und die „Actions“-Palette (rot), mit einer Liste von wählbaren Aktionen.

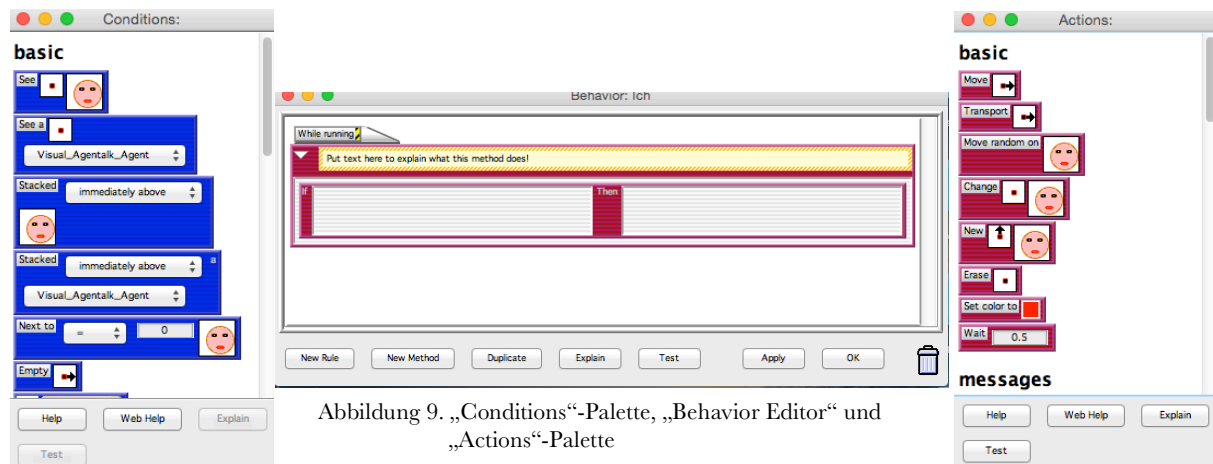
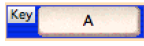



Abbildung 9. „Conditions“-Palette, „Behavior Editor“ und „Actions“-Palette

Wir nennen die Befehle, die wir dem Programm zuordnen (programmieren) Regeln. Regeln bestehen aus einem **wenn-dann (if-then)-Paar**.

Wenn wir das „Ich“ mit den Pfeiltasten bewegen wollen, dann lautet eine Regel: „Wenn die Pfeiltaste → gedrückt wird, dann soll sich das „Ich“ um einen Schritt nach rechts bewegen“. Insgesamt brauchen wir 4 Regeln, welche die Bewegungsrichtungen anordnen (oben, unten, rechts, links).

Wir programmieren die „Tastatur-Eingabe“ (key condition) des „Ich“. Klicken Sie auf Symbol *Key A*  in der Bedingungen(Conditions)-Liste und ziehen Sie diesen Programmierbaustein mit der Maus (*drag & drop*) in den Programm-Editor (*Behavior Ich*) in die linke Spalte mit der *If(wenn)*-Bedingung. Aus dem „A“ können wir durch einen Klick auf das Buchstabensymbol eine beliebige Tastatureingabe machen und wählen in diesem Fall die **Pfeiltaste** →. Die Regeln bestehen aus zwei Teilen, es fehlt nun noch die *Then(dann)*-Aktion.

Wir ergänzen die Bewegung in den Programmiereteil für „Aktionen“ (Actions): Dazu ziehen Sie den Programmierbaustein *Move*  aus dem Aktionen(Actions)-Liste der in die rechte Spalte des Programm-Editors mit der Bezeichnung *then* (dann). Durch einen Klick auf den Pfeil im Programmierbaustein *Move* kann die Bewegungsrichtung entsprechend der Tastatureingabe angepasst werden. Nun sollte der Programmiereditor (*Behavior editor*) des „Ich“ wie folgt aussehen.

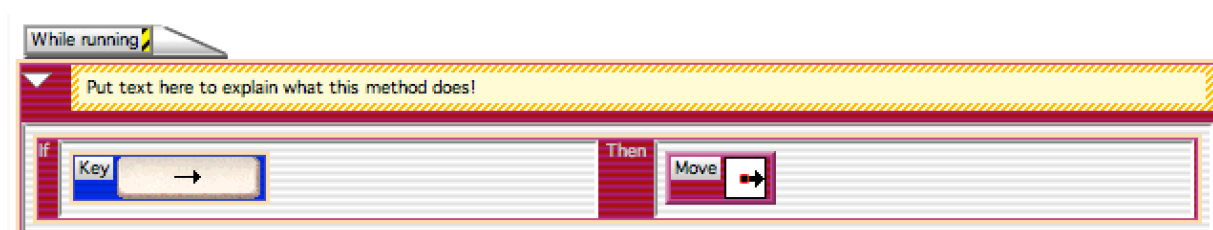
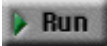


Abbildung 10: Das „Ich“ bewegt sich nach rechts: „Wenn die Pfeiltaste → bedient wird, dann bewege dich nach rechts“

Klicken Sie auf „**Apply**“ (entspricht speichern) im unteren Bereich des Programm-Editors, um die Eingaben zu speichern. Anschliessend testen wir das Programm, indem wir auf  im Worksheet-Fenster klicken. Damit initiieren Sie das Programm. Was passiert?

- „Ich“ sollte sich nach rechts bewegen, wenn Sie die rechte Pfeiltaste auf ihrer Tastatur bedienen. Sollte sich „Ich“ nicht bewegen, schauen Sie in der Dokumentation nach, woran es liegen könnte.
- Wenn wir die anderen Pfeiltasten drücken, passiert allerdings noch nichts. Was müssen wir tun, damit das sich ändert und das „Ich“ in alle Richtungen geht?

Nach erfolgreichem Testdurchlauf gehen wir auf „**reset**“. Dadurch bringen wir das „Ich“ in die Ausgangslage zurück, denn *Reset* setzt das Spiel immer wieder auf die zuletzt gespeicherte Version zurück.

Duplizieren einer Regel: Wir ergänzen die restlichen Bewegungsrichtungen, indem wir 3 weitere Regeln im *Behavior-Editor* von „Ich“ hinzufügen. Durch Kopieren der ersten Regel können wir dies mit geringem Aufwand tun. Hierfür selektieren wir die bereits vorhandene Regel (sie sollte dann blau umrahmt sein) und klicken dann dreimal auf die Schaltfläche „**Duplicate**“. Anschließend passen wir die neuen Regeln an, indem wir die Richtungen der Tastatureingaben verändern und dann auf der Seite der Aktionen die Pfeile der Bewegung entsprechend anpassen. Dann klicken wir auf *Apply* und testen anschliessend im Worksheet, ob wir das „Ich“ mit den Pfeiltasten korrekt in alle vier Richtungen bewegen können. Schliesslich sollte der Programm-Editor (*behavior editor*) so aussehen:



Abbildung 8: Fertiger Programmausschnitt für alle vier Bewegungsrichtungen des „Ich“

5. Das „Ich“ erreicht das Ziel – Spiel gewonnen!

Im Programm-Editor (*Behavior editor*) des „Ich“ klicken wir nun auf „**New Rule**“, um eine neue, leere Regel zu erstellen. Selektieren Sie dann die *Next to*-Bedingung in der *Conditions*-Palette und ziehen Sie diese auf die *if*-Seite der neuen Regel. Klicken Sie dann auf das mathematische Symbol im Baustein und wählen das „>“-Zeichen. Schliesslich klicken Sie im selben Baustein auf das kleine Bild und wählen das „Ziel“-Bild.

Nun definieren wir die zugehörige Aktion im *Then*-Teil der Regel: Zunächst wird ein Geräusch abgespielt, z.B. „trara“, dann wartet das Programm einen gewissen Zeitraum (z.B. 2 Sekunden), damit der Ton vollständig abgespielt werden kann, und dann wird die Simulation gestoppt. Finde diese drei Aktionen in der *Actions*-Liste und ziehen Sie diese in der angegebenen Reihenfolge in den *Then*-Teil der Regel. Die Regel im *Behavior-Editor* sollte nun so aussehen:

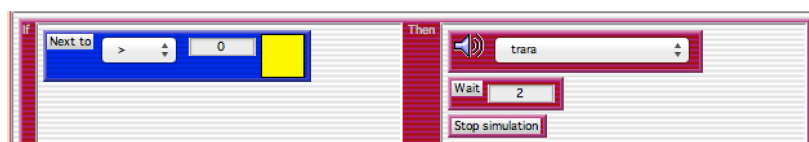


Abbildung 9: Fertige Regel für das Erreichen des Ziels

Jetzt testen wir das Verhalten des „Ich“, um sicher zu sein, dass alles so funktioniert, wie es soll. Klicken Sie nach dem Testen wieder den *Reset*-Knopf. Anschliessend schliessen Sie den *Behaviour-Editor* des „Ich“.

6. Wir erstellen die restlichen Agenten

Wir erstellen jetzt die restlichen Agenten, konstruieren das Labyrinth auf dem Spielfeld, platzieren die Angreifer-Agenten und definieren das Verhalten der restlichen Agenten. Zunächst erstellen und zeichnen wir die zwei neuen Agenten „Angreifer“ und „Mauer“. Auf dem *Worksheet* platzieren wir „Mauer“-Agenten, damit sie ein einfaches Labyrinth bilden. Dann platzieren wir ein oder zwei „Angreifer“-Agenten auf dem „Hintergrund“. Anschliessend unbedingt das *Worksheet* speichern, um den aktuellen Status des Spielfelds zu sichern!!

Das Spielfeld sollte nun ungefähr so aussehen:

Selektieren Sie nun in der Agenten-Galerie den „Angreifer“ und klicken Sie *Edit Behavior*. Der Einfachheit halber sollen sich die „Angreifer“-Agenten zufällig auf dem „Hintergrund“ bewegen. Ziehen Sie dazu den Programm-Baustein *Move random on* aus der *Actions*-Palette in den *Then*-Teil der leeren Verhaltensregel von dem „Angreifer“-Agenten. Klicken Sie nun auf das Bild im Baustein und wählen Sie das Bild des „Hintergrund“-Agenten. Der „Angreifer“-Agent wird sich nun zufällig und nur auf dem „Hintergrund“ bewegen.

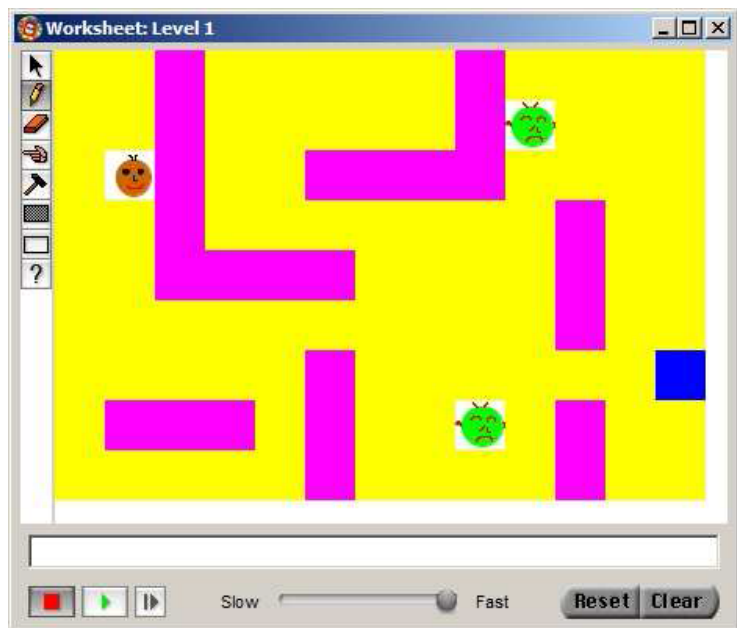


Abbildung 10. Komplettes Spielfeld.

Um die Bewegung der „Angreifer“-Agenten etwas zu regulieren, ziehen Sie nun den *once every*-Baustein aus der *Conditions*-Palette in den *If*-Teil der Regel und stellen den Wert auf 0.5 (Sekunden). Sie benötigen keine weitere Bedingung, da der „Angreifer“-Agent diese Aktion immer ausführen soll, solange das Programm läuft (*while running*). Testen Sie nun das Programm.

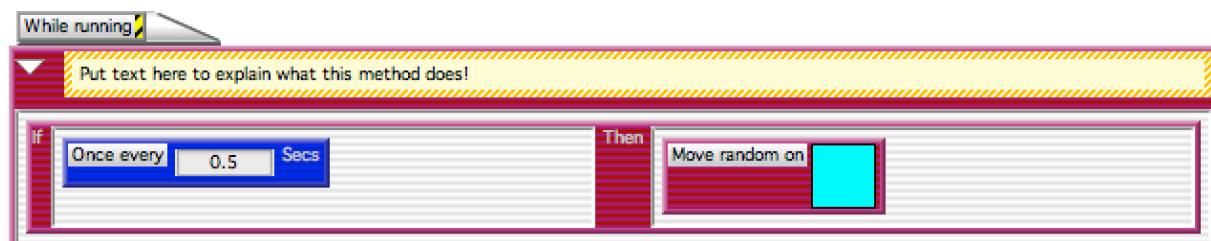


Abbildung 11. Vollständige Verhaltensregel des „Angreifer“-Agenten.

Der „Angreifer“-Agent kann sich lediglich auf dem „Hintergrund“ bewegen, nicht aber durch die „Mauer“ laufen. Das ist bei dem „Ich“-Agenten noch anders! Um zu verhindern, dass „Ich“ durch bzw. über die „Mauer“ gehen kann, müssen wir sein Verhalten entsprechend anpassen. Dafür öffnen Sie wieder den *Behavior-Editor* des „Ich“ und ziehen die *See a*-Bedingung in den *If*-Teil der ersten Pfeiltasten-Steuerungsregel. Wechseln Sie nun das Bild im *See a*-Baustein, damit es den „Hintergrund“-Agenten anzeigt und ändern Sie die Richtungsangabe, so dass sie der Angabe der Pfeiltaste entspricht.



Abbildung 12. Aufwärts-Bewegungsregel des „Ich“

Sie können diesen Baustein nun kopieren, in die anderen 3 Bewegungsregeln einfügen und entsprechend der Richtungen anpassen.

7. Das „Ich“ wird attackiert – Spiel verloren!

Nun müssen wir nur noch eine Regel im Verhalten des „Ich“-Agenten ergänzen, damit das Spiel vollständig ist. Selektieren und duplizieren Sie zunächst die vollständige Regel, die definiert, was passiert, wenn „Ich“ das „Ziel“ erreicht und das Spiel gewonnen ist. Anschliessend wechseln Sie in der neuen Regel im *Next to*-Baustein das Bild zu dem des „Angreifer“-Agenten und ändern Sie das Geräusch im *Play sound*-Baustein, so dass es eine nachteilige Situation suggeriert, z.B. „scream“ oder „explode“.

Zu guter Letzt müssen die Regeln noch in eine gute Reihenfolge gebracht werden, damit das Programm tadellos läuft. Selektieren Sie dafür nacheinander die beiden Regeln mit dem *Next to*-Baustein und ziehen Sie diese an den Anfang der Verhaltensregeln-Liste. Dadurch testet das Programm die Ausnahme-Regeln, die das Spiel beenden, und führt diese aus vor den Regeln der Pfeiltasten-Eingaben.

Falls nämlich der „Ich“-Agent mit einem „Angreifer“ kollidiert und gleichzeitig eine Pfeiltaste gedrückt wird, stellt diese Reihenfolge sicher, dass man das Programm nicht ‚ausricksen‘ kann.

Und so sieht das vollständige Verhalten des „Ich“-Agenten aus:

Sie haben nun ein einfaches „Labyrinth“-Spiel gebaut!

Und nun...viel Spass beim **Testen Sie Ihres Spiels!**

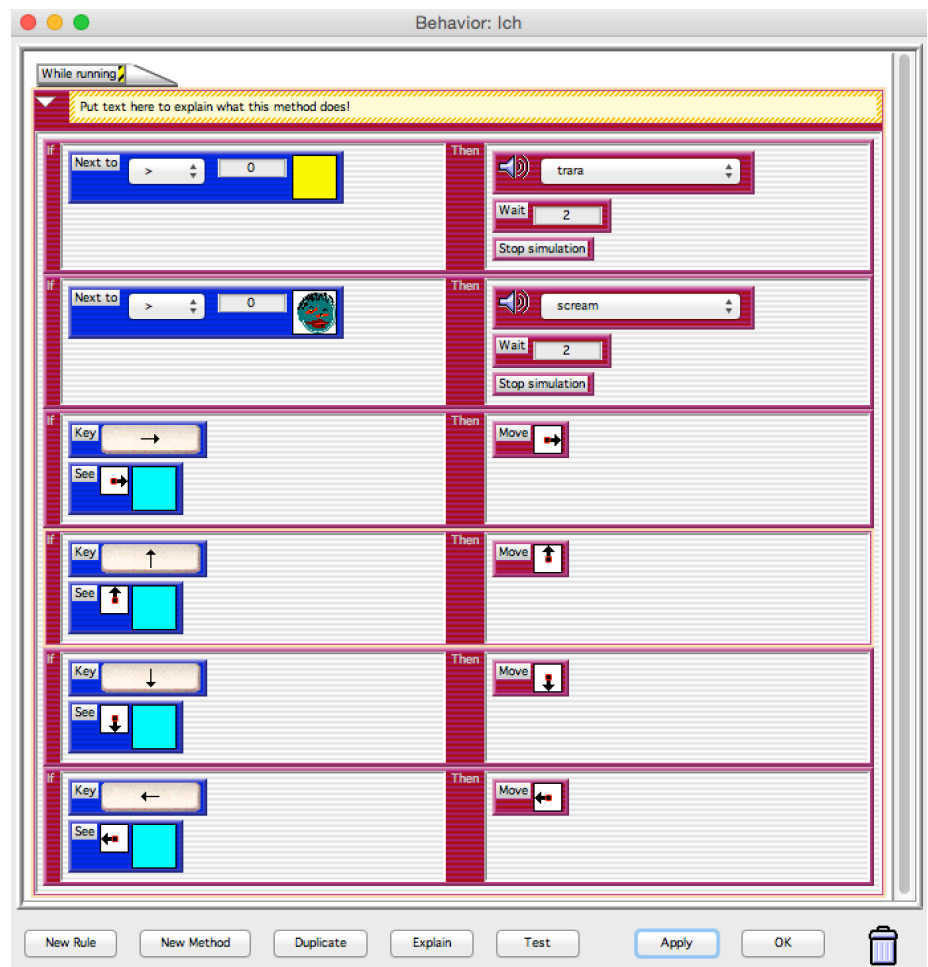


Abbildung 13. Vollständige Verhaltensregeln des „Ich“