„Insel“ - Realitätsausschnitt für Menschen und Lebensumwelt von PSI

Frank Detje
Institut für Theoretische Psychologie, Universität Bamberg
frank.detje@ppp.uni-bamberg.de

1. Einleitung

Erste Implementationen des Computerprogramms ”Insel” wurden im Rahmen des DFG-Projektes ”Konstruktion einer Theorie zur Erklärung und Prognose menschlichen autonomen Handelns in komplexen Realitätsbereichen” (kurz: „Autonomie“-Projekt) entwickelt. Grundlegend für die Konzeption des Computerprogramms ist die zentrale Anforderung, dass sowohl verschiedene menschliche Versuchspersonengruppen als auch die Implementation der PSI-Theorie diese Umweltsimulation selbständig bewältigen können. Damit soll sichergestellt werden, dass Vergleiche zwischen menschlichem Problemlöseverhalten bei der Bearbeitung dieser Simulation mit dem Verhalten von PSI möglich sind und wir Aufschluss für Verbesserungen der PSI-Theorie bzw. deren Implementation bekommen. Weiterentwicklungen des ”Insel”-Programms erfolgten, um das Szenario unter unterschiedlichen Versuchsbedingungen einsetzen zu können und um das Programm auch für die Folgeprojekte des ”Autonomie”-Projektes ”fit” zu machen, namentlich das ”Sozionik”-Projekt und das ”Plastizitäts”-Projekt.[1]

Die wichtigsten Ziele des „Autonomie“-Projektes waren, ein Computerszenario zu konstruieren und zu verwenden, dass als „Umwelt“ in Form einer „Insel“ gleichermaßen von „realen“ Menschen wie von künstlichen Agenten (hier: die Implementation der PSI-Theorie) bearbeitet werden kann. Durch Vergleiche des Verhaltens (PSI versus Mensch) sollte die PSI-Theorie geprüft und erweitert werden (hierzu siehe Detje, 2000b).

Zusätzlich möchten wir im Rahmen des Sozionik-Projekts untersuchen, unter welchen Bedingungen sich aus einer Ansammlung individueller künstlicher Agenten interessante makroskopische Strukturen herausbilden, welcher Art sie sind und wie diese zu erklären sind. Dazu muss eine Umwelt für die Agenten geschaffen werden, die sehr unterschiedliche Interaktionsmöglichkeiten bietet und die Ausbildung sehr vielfältiger Verhaltensmuster ermöglicht. Die Erweiterung von ”Insel” zum Multi-Agenten-Betrieb wird ebenfalls angestrebt, ist aber noch nicht abgeschlossen.

Um einen ersten Eindruck über die Insel-Realität zu erhalten, die die verschiedenen Anforderungen an menschliche Versuchspersonen, und natürlich auch PSI, vermittelt, eignet sich die Versuchsinstruktion „Nukleotidios“ gut, die wir den Versuchspersonen vor der Bearbeitung des Szenarios vorlegen. Die Versuchspersonen schlüpfen in diesem „Spiel“ in

[1] Wir danken der Deutschen Forschungsgemeinschaft, die diese Projekte unter den Az. DO 200/14-1 und 2, DO 200/20-1, DO 200/21-1 finanziell förderte und fördert. „Sozionik-Projekt“ ist die Kurzform für „Die Entstehung und Änderung sozialer Strukturen in Gruppen intelligenter, multimotivierter, emotionaler Agenten“; „Plastizität-Projekt“ steht für: „Die aktualgenetische Plastizität des menschlichen Denkens“.

die Rolle eines simulierten Roboters, den wir „James“ nannten. Die PSI-Implementation hingegen „ist“ dieser Roboter, d.h. auch PSI füllt diese Rolle autonom aus. Die Versuchsbedingungen sind also für die menschlichen und die künstlichen Agenten identisch.

2. Die Versuchsinstruktion ”Nukleotidios”

Mit der Kernkraft ist das schon so ein Problem! Wohin mit dem strahlenden Abfall? Und was geschieht bei einem Kernkraftunfall? – Auf der anderen Seite: Kernkraft belastet die Umwelt nicht mit Verbrennungsprodukten und hat auch sonst noch einige Vorteile. Wenn nur die Strahlung nicht wäre!!

Nun hat man neuerdings Nukleotide entdeckt. Dies kann man in Kernkraftwerken in den Reaktoren als Heizelemente benutzen, aber sie lösen sich während des Betriebes vollkommen in Wasser auf. Es gibt also keine schädlichen Nebenprodukte. Auch ein Kernkraftunfall ist mit den Nukleotiden ausgeschlossen; sobald sie mit normaler Luft in Berührung kommen, werden sie zu Wasser. Während des Kernzerfalls werden auch keine schädlichen Gammastrahlungen frei, nur eben Hitze. Die Nukleotide sind also wahrhaft ideale Brennelemente. Es gibt sie leider nicht sehr häufig in der Natur; man hat sie aber auch noch nicht herstellen können.

Eine Lagerstätte für Nukleotide befindet sich auf der kleinen Südseeinsel Nukleotidios. Sie haben nun die Aufgabe, auf dieser Insel Nukleotide zu sammeln. Allerdings können Sie das gar nicht selbst tun, denn auf der Insel befindet sich ein kleiner, aktiver Vulkan, der auch ständig die Nukleotiden aus dem Erdinneren auf die Erdoberfläche befördert. Sie sollten sich im Interesse Ihrer eigenen Gesundheit dem Ascheregen, den Schwefeldämpfen und den bis zu faustgroßen Lavabrocken, welche der Vulkan ausstößt, nicht aussetzen, daher haben Sie einen kleinen Roboter namens ”James” zur Verfügung, der diese Aufgabe für Sie übernimmt.

Abbildung 1: ”James”.

Sie müssen diesen Roboter steuern, was Sie von einem Schiff aus, welches vor Nukleotidios vor Anker liegt, tun können. Sie sehen jetzt vor sich die entsprechende Steuerungseinrichtung für James.

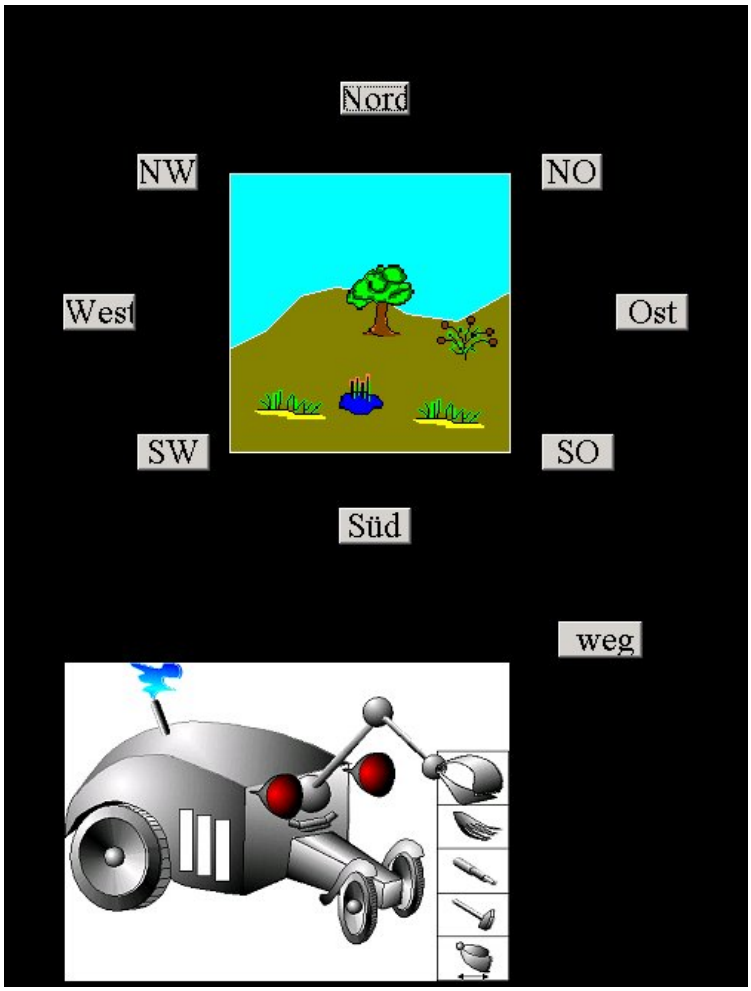

Abbildung 2 : Das Panel.

Auf ihrem Monitor sehen Sie direkt in der Mitte eine Abbildung der Landschaft, wie sie sich James‘ Kameraauge darbietet. Auf dem kleinen Sichtfenster sehen Sie jeweils die Situation, in der sich James gerade befindet mit ihren verschiedenen Komponenten, wie Dünen, Höhlen, Pflanzen, Sträucher, Bäume usw. Die Nukleotide sind kleine sechseckige bis runde Elemente, die Sie unschwer entdecken werden. Sie können sie dann jeweils einsammeln und mitnehmen. Manchmal aber sind die Nukleotide auch verborgen, und Sie müssen sie suchen. Dazu können Sie James umherfahren oder auch die in einer Situation vorfindbaren Objekte manipulieren lassen. Durch Mausklick können Sie einzelne Objekte (zum Beispiel einen Baum) auswählen, und James damit befehlen, sich diesem anzunähern. Anschließend können Sie James Operationen ausführen lassen. Dazu dienen die Werkzeuge an der rechten Leiste. James hat einen Greifarm, mit dem er Dinge aufnehmen kann. Dazu klicken Sie den Greifarm von James mit der Maus an. Soll James anstelle des Greifens andere Operationen ausführen, so klicken Sie die entsprechenden Werkzeuge an. Unter dem Greifarm

befindet sich beispielsweise ein Sieb, das bei Bedarf ausgefahren werden kann. Außerdem befindet sich im Greifarm auch ein ausfahrbarer Saugdorn, mit dem James Flüssigkeiten ”saugen” kann.

Soll James diese Operation durchführen, so klicken Sie mit der Maus auf den Saugdorn. Der Arm kann auch einen Hammer führen und ”hämmern”. Mit dem untersten Werkzeug kann James ”Schütteln”. Welche Operationen bei welchem Objekt sinnvoll sind und welche nicht, das werden Sie im Laufe des Versuchs selbst herausfinden. Wenn James sich wieder von dem ausgewählten Objekt entfernen soll, klicken Sie einfach mit der Maus auf den Button ”weg”. James gelangt dann wieder zu seinem Standort, von dem aus er den ganzen Ort gut übersehen kann.

Soll James sich weiter über die Insel bewegen, so können Sie über die Richtungsbuttons bestimmen, in welcher Himmelsrichtung James den Ort verlassen soll. Sie können James in eine von acht Richtungen bewegen, wenn hier ein einigermaßen befahrbarer Weg existiert. Nicht in jeder Situation können Sie sich überall hin bewegen, da mitunter dichtes Gestrüpp oder Felsgeröll die Vorwärtsbewegung in einer bestimmten Richtung unmöglich macht. Das werden Sie aber merken.

Leider können Sie Ihre Tätigkeiten nicht auf das Aufsammeln von Nukleotiden beschränken. James braucht nämlich Energie. James’ Kraftzentrum ist eine Dampfmaschine; diese Dampfmaschine braucht Wasser und außerdem braucht der Brenner der Dampfmaschine Brennstoff. Die Dampfmaschine ist allerdings sinnreicherweise so ausgestattet, dass sich James mit den Ressourcen, die er auf Nukleotidios vorfindet, versorgen kann. Er kann also ölhaltige Pflanzenteile aufnehmen und ”verdauen”. Der ”Verdauungsprozess” besteht darin, dass das Öl ausgepresst wird und dann in einen Brennstoffbehälter für die Heizung der Dampfmaschine zur Verfügung steht. Außerdem kann James mit Hilfe einer Saugpumpe Wasser aufnehmen. Meerwasser allerdings ist ihm unzuträglich. Wo genau sich auf Nukleotidios Wasserstellen befinden und wo es ölhaltige Pflanzen gibt, ist allerdings gänzlich unbekannt. Sie müssen das herausfinden, und Sie sollten dafür sorgen, dass James zum Zwecke der Erfüllung seiner Nukleotidensammelaufgabe immer über genügend Brennstoff und Wasser verfügt. James kann allerdings auch zu Schaden kommen. Schlechte Wegstrecken, schwefelhaltige Dämpfe oder andere Einflüsse können ihn beschädigen. Zum Glück kann James in weiten Grenzen die Schäden selbst reparieren, indem er entsprechende Spülflüssigkeiten durch sein Körperinneres pumpt, die die Schadstoffe beseitigen. Die entsprechende Spülflüssigkeit kann James selbst herstellen, wenn er die entsprechenden Pflanzen findet und zu sich nimmt. Nun weiss man nicht genau, welche Pflanzen auf Nukleotidios die für die Spülflüssigkeit notwendigen Wirkstoffe enthalten; Sie müssen das herausfinden!

Eine Anzeige am rechten Rand des Bildschirms gibt Ihnen den jeweiligen Stand an ”Durst”, ”Hunger” und ”Schaden” an, so dass Sie ungefähr wissen, wann Sie sich wieder um die Funktionstüchtigkeit von James kümmern sollten. Außerdem können Sie sehen, wieviele Nukleotide Sie bereits gesammelt haben. Aus dem Labor wurden Ihnen 5 Vergleichstücke mitgegeben, damit Sie wissen, wie Nukleotide aussehen.

Abbildung 3: Anzeige der Zeit, des Hungers, Dursts, Schadens und der Nukleotide.

Sie sollen also nun Nukleotide sammeln und zwar so viele wie möglich. Auf der anderen Seite müssen Sie zu diesem Zwecke James funktionstüchtig erhalten. Bedenken Sie: James ist teuer. Wenn Sie James aufgrund von Wasser- oder Brennstoffmangel oder aufgrund allzu großer Schäden zugrunde richten, können wir Ihnen einen neuen ”James” zur Verfügung stellen. Aber das sollten Sie vermeiden, da das sehr kostspielig ist. Sie müssen jeden verbrauchten James mit 5 Nukleotiden bezahlen. Gesammelte Nukleotide bekommen Sie am Ende des Versuchs mit 20 Pfennig pro Nukleotid vergütet. Beachten Sie unbedingt, dass Sie während des Versuchs ”laut denken”. Wir benötigen das, um Ihre Entscheidungen nachvollziehen zu können und für die Forschung verwertbar zu machen. Wenn Sie noch Fragen haben, so wenden Sie sich an den Versuchsleiter. Ansonsten alles Gute für die Revolutionierung der Kernkraft!

3. „Insel“ – Beschreibung des Versuchspersonenspiels

3.1 Notwendigkeit zur Herstellung eines eigenen Szenarios / einer eigenen Umwelt

Unsere Forschungsziele und die damit verbundenen Anforderungen an die zu konstruierende Umwelt (das Szenario) machen es notwendig, ein Simulationsprogramm in eigener Regie zu erstellen, denn auf vorhandene Simulationsprogramme kann aus verschiedenen Gründen nicht

zurückgegriffen werden. Die aus der Robotik und angrenzenden Bereichen bekannten Tools zur Konstruktion von Umwelten gehen zu sehr auf eine getreue Abbildung real-physikalischer, technischer Randbedingungen ein, die für die vorliegenden Ziele unwichtig sind und sie erlauben zum anderem die Konstruktion einer bestimmten Umwelt nur durch hohen zeitlichen Aufwand, wobei die Variation dieser einmal geschaffenen Umwelt wiederum auch nur durch sehr hohen zeitlichen Aufwand geleistet werden kann. Die aus der Simulation makroskopischer Phänomene bekannten Programme sind, was die simulierte Umwelt angeht, zu abstrakt, d.h. sie sind zu wenig konkret in ihren vorhandenen Umweltelementen und den Bedingungen ihrer Manipulation (oder sie verzichten sogar ganz darauf), als dass die einzelnen Agenten, wie in unseren Projekten vorgesehen, individuell tätig werden könnten. Die aus der Kognitionswissenschaft (und Künstliche Intelligenz-Forschung, Psychologie) bekannten Programme und Tools bieten nicht die Möglichkeit, mehrere Agenten mit ihnen zu konfrontieren und / oder sind zu sehr auf ganz bestimmte zu erreichende Ziele, nicht aber auf motivnahe Anforderungen, ausgerichtet (Problemlöseszenarien), als dass sie für unsere Fragestellungen geeignet wären. Die im Handel erhältlichen Computerspiele haben die Nachteile, dass sie entweder ein nur sehr begrenztes Verhaltensspektrum erlauben, durch ihre beeindruckende Graphik zu hohe Anforderungen an die Wahrnehmungsleistung der künstlichen Agenten stellen, keine Protokollierung der getätigten Eingriffe und der eigendynamisch hergestellten Umweltänderungen erlauben und / oder nicht (ohne Weiteres) in den Umweltbedingungen variierbar sind.

Das Simulationsprogramm ”Insel”[2] ist durch seinen Programmaufbau eher als ein Konstruktionstool zur Konkretisierung verschiedener Umwelten (was die Agenten angeht) bzw. verschiedener Spiele (was die Versuchspersonen angeht) anzusehen. Die Variation der Umwelt ist in einem einfachen Texteditor möglich (die zu beachtenden syntaktischen Randbedingungen sind minimal), die Protokollierung ist vollständig und die integrierten Elemente sind auf die Fähigkeiten der Agenten und die Anforderungen an die Projekte abgestimmt.

Eine Version des „Insel“-Programms ist über die Homepage des Instituts unter URL „http://www.uni-bamberg.de/ppp/insttheopsy/“ erhältlich.

3.2 Systemvoraussetzungen

Um „Insel“ sinnvoll einsetzen zu können, benötigen Sie einen Pentium-PC mit einer Taktrate von 100 MHz oder mehr, ferner eine Grafikkarte, eine Soundkarte und Lautsprecher. Der Rechner sollte über mindestens 16 MB RAM und 10 MB freier Festplattenkapazität verfügen. Wir empfehlen, alle weiteren Programme vor dem Spielstart zu beenden, vor allem sämtliche Netzwerk- und Internetverbindungen. Das Spiel ist für eine Bildschirmauflösung von 1024 * 768 Pixel optimiert, aber auch eine Darstellung bei 800 * 600 ist problemlos möglich.

---

[2] Die Arbeiten an der PSI-Theorie haben zur Konstruktion verschiedener Simulationsprogramme geführt, die sowohl als Umwelten für PSI als auch als Untersuchungsgegenstand für menschliche Versuchspersonen dienen. Zu den schon früher eingesetzten Szenarien gehören „Garten“ (Schaub, 1993) und „BioLab“ (Bartl & Dörner, 1998).

3.3 Programmbeschreibung (Überblick)

Die ”Insel” besteht aus eine beliebigen Anzahl von Situationen (”Orte”) die auf beliebige Weise miteinander verbunden sein können (”Wege”). Für unsere ersten Untersuchungen konstruierten wir eine Insel, die aus 45 verschiedenen Orten und 150 verschiedenen Verbindungen zwischen diesen Orten besteht (siehe Abbildung 4). Der rote Punkt Nummer 8 zeigt den aktuellen Standort. Der Aufbau der Insel, d.h. die Anordnung der Orte und ihrer Verbindungen ist den menschlichen und künstlichen Versuchspersonen allerdings unbekannt. Die Abbildung 4 ist für die Agenten während der gesamten Bearbeitungszeit nicht zugänglich.

Eine jede Verbindung zwischen zwei Orten besitzt Angaben darüber, wie gut die Wegstrecke ist und wie hoch das Risiko ist, durch Steinschlag oder andere Ereignisse Schaden zu nehmen.

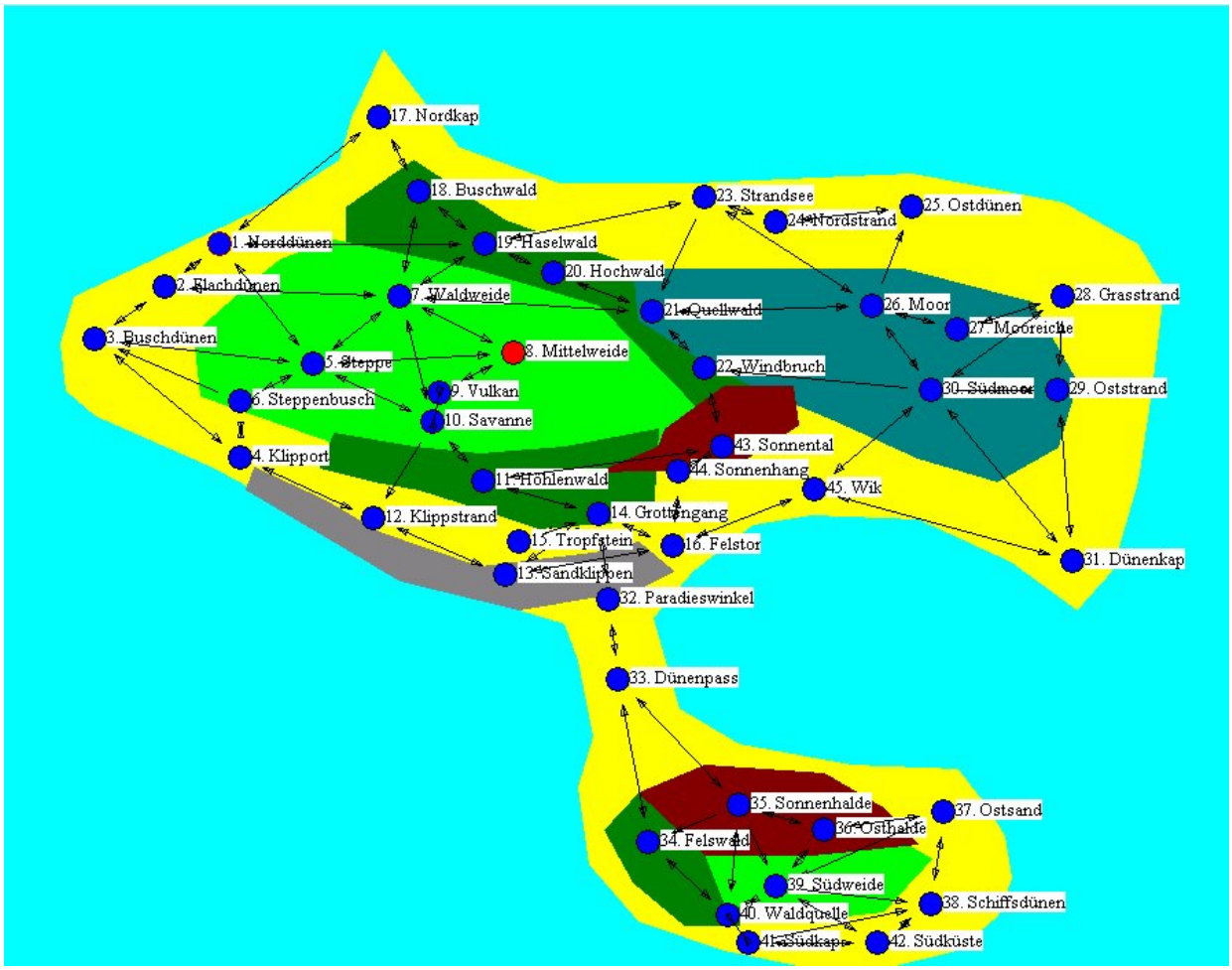

Abbildung 4: Die ”Insel” als Beispiel für eine konkrete Lebensumwelt.

Ein jeder Ort der Insel besteht aus einer beliebigen Anzahl von Objekten (für ein Beispiel siehe Abbildung 5), die von den Agenten (Menschen oder PSIs) auf beliebige Weise manipuliert werden können und diese Manipulationen können beliebige Auswirkung auf den Agenten und / oder die Umwelt haben. Die Details der konkreten Beschaffenheit der verwendeten Situationen auf der eingesetzten Insel werden weiter unten beschrieben.

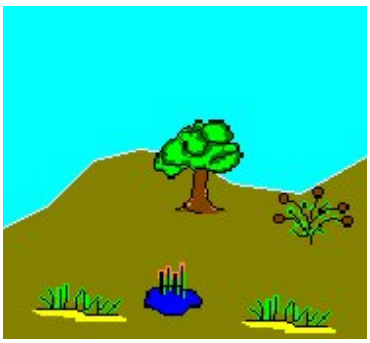

Abbildung 5: Eine Situation (ein Ort) auf der Insel.

Die Agenten (auch der ”autonome Agent” PSI, hier in Form seiner Inkarnation ”James”; siehe Abbildung 6) haben die Aufgabe, James existentiellen Bedürfnisse nach Nahrung, Wasser und Schadensvermeidung zu befriedigen, gleichzeitig aber auch nach den sogenannten ”Nukleotiden” zu suchen und diese aufzusammeln.

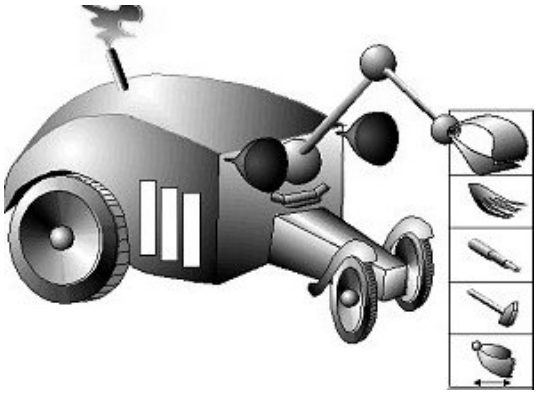

Abbildung 6: ”James”, eine PSI-Inkarnation.

Zur Befriedigung der existentiellen Bedürfnisse gibt es auf der Insel Orte, an denen für den Betrieb von James notwendiges Wasser und Brennstoff vorhanden ist. Nukleotide sind z.B. in Dünen oder Felsen ”versteckt”, diese Objekte geben Nukleotide erst nach einer entsprechenden Manipulation frei.

Die Wege zwischen zwei Orten zeichnen sich durch unterschiedlich gute Bodenverhältnisse aus (schlechter Boden verursacht Schaden) und durch ein gewisses (von Weg zu Weg variierendes) Risiko von Steinschlag aus, der ebenfalls zu Schaden führt. Schwefelgase, die durch Manipulation bestimmter Objekte der Orte freigesetzt werden, verursachen ebenfalls Schaden.

Im Laufe des Spiels erlernt man, die Objekte der Umwelt zu manipulieren und die Orte zur Befriedigung der Bedürfnisse anzustreben bzw. jene, die Gefahr bedeuten, zu vermeiden. Um in dieser Umwelt tätig zu werden, stehen verschiedene Operatoren zur Verfügung (siehe Abbildung 7). Man kann sich durch Betätigung der Richtungsoperatoren von Ort zu Ort bewegen. Man kann sich den Objekten eines Ortes nähern (dies ist notwendig, um sie z.B. zu manipulieren) und sich von diesen Objekten auch wieder entfernen.

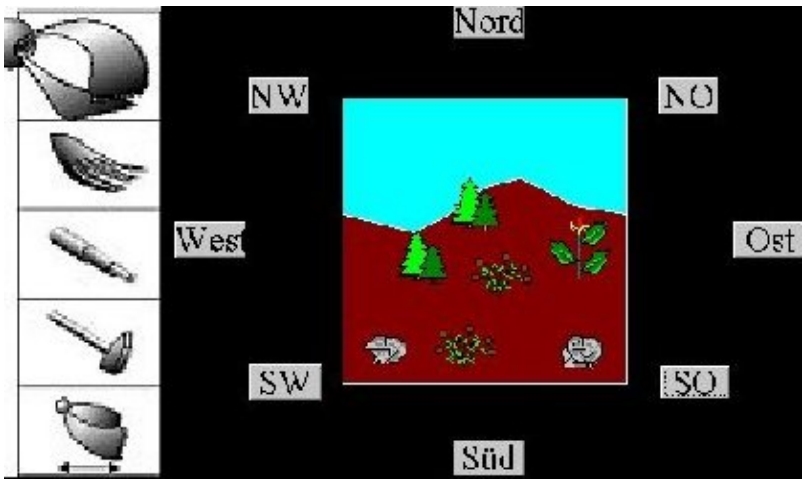

Abbildung 7: Die Manipulationsmöglichkeiten (Multi-Funktionsarm und Richtungsoperatoren).

Ferner ist James mit einem Multi-Funktions-Arm ausgerüstet, der es erlaubt, die Objekte auf verschiedene Weise zu manipulieren. So kann man sieben, greifen, trinken, hämmern, schütteln. Die spezifische Wirkung (z.B. ob bedürfnisbefriedigend, bedürfniserzeugend, erfolglos oder umweltändernd) einer Manipulation eines bestimmten Objekts ist frei vorgebbar.

Das Programm ”Insel” liest jeweils die Spezifikationen für eine konkrete Insel ein, lediglich semantische Kriterien beschränken die Möglichkeiten (so sollten Gegebenheiten, die dem ”gesunden Menschenverstand” widersprechen, vermieden werden).

3.4 Programmbedienung

3.4.1 Verzeichnisaufbau

Das Simulationspaket ”Insel” besteht in der Regel aus zwei Spielen, einem Übungsspiel (”DInselUeben.exe”) und dem Hauptspiel (”DInsel.exe”). Das Programm liest verschiedene Dateien ein und besitzt zu diesem Zweck zwei Unterverzeichnisse: 1) ”\bmp\”, in dem sich die Bilddateien (Bitmaps) befinden, 2) ”\wav\”, in dem sich die Sound-Dateien befinden. Fehlen diese Unterverzeichnisse und die dort erwarteten Dateien, ist ein Programmstart nicht möglich.

Das Programm erstellt automatisch für jedes Spiel mehrere Protokolldateien. Wenn nicht vorhanden, erstellt das Programm ”Insel” automatisch das Protokollunterverzeichnis ”\prt\”, in die alle Protokolldateien geschrieben werden. Folgende Protokolldateien werden erstellt: a) eine Protokolldatei des Spielablaufs, b) eine für das Versuchspersonenverhalten, c) eine für die Mausbewegungen, d) eine für die Motivwertänderungen und e) eine mit einem groben Überblick über die erzielte Leistungen. Die Protokolldateien werden einzeln numeriert, es werden damit keine vorhandenen Dateien überschrieben, es können also auch keine Protokolldateien verloren gehen (ausser man löscht sie später). Der Aufbau der Protokolldateien ist immer derselbe: ”Insel”, gefolgt von der Kennzahl, gefolgt von der Dateikennung (”Maus” für die Maus-Dateien, ”Motiv” für die Motivdateien, ”Verhalten” für die Verhaltensdateien und ”Auswertung” für die Auswertungsdateien, gefolgt von der einheitlichen Dateiendung ”.prt”). Da mit jedem Spielstart ein Satz Protokolldateien erstellt

wird, empfiehlt es sich, ab und zu das Verzeichnis aufzuräumen, d.h., nicht gebrauchte Dateien zu löschen.

Im Hauptverzeichnis des Spiels wird ausserdem bei Bedarf eine Datei ”InselFehler.prt” erstellt, wenn Fehler bei der Konstruktion von Spielen festgestellt werden. Sie enthält Hinweise auf die entsprechenden Konstruktionsfehler, wenn diese festgestellt werden konnten.

Die Datei ”neverdie.dat” darf auf *keinen Fall* gelöscht werden, sie zählt die Anzahl der Spielstarts mit.

Der Aufbau der mitgelieferten Initialisierungs-, Insel- und Manipulationsdateien werden weiter unten erläutert.

3.4.2 Unterschied zwischen ”Hauptspiel” und ”Übungsspiel”

Der wichtigste Unterschied besteht darin, dass das Übungsspiel keine Sounds abspielt. Übungsspiele sind zudem üblicherweise so konzipiert, dass sie nur eine kleine Insel mit wenigen Orten und wenigen Objekten beinhalten, um nicht zu viel von dem eigentlichen Spiel zu verraten. Es geht bei dem Übungsspiel in der Hauptsache darum, die Versuchspersonen an die spezifische Oberfläche und die Bedienung zu gewöhnen.

3.4.3 Programmstart

Starten sie das Spiel durch einen Doppelklick auf die Datei ”DInsel.exe”. Das Programm zeigt in den ersten sechs Sekunden eine zweiteilige Titelsequenz (siehe Abbildung 8).

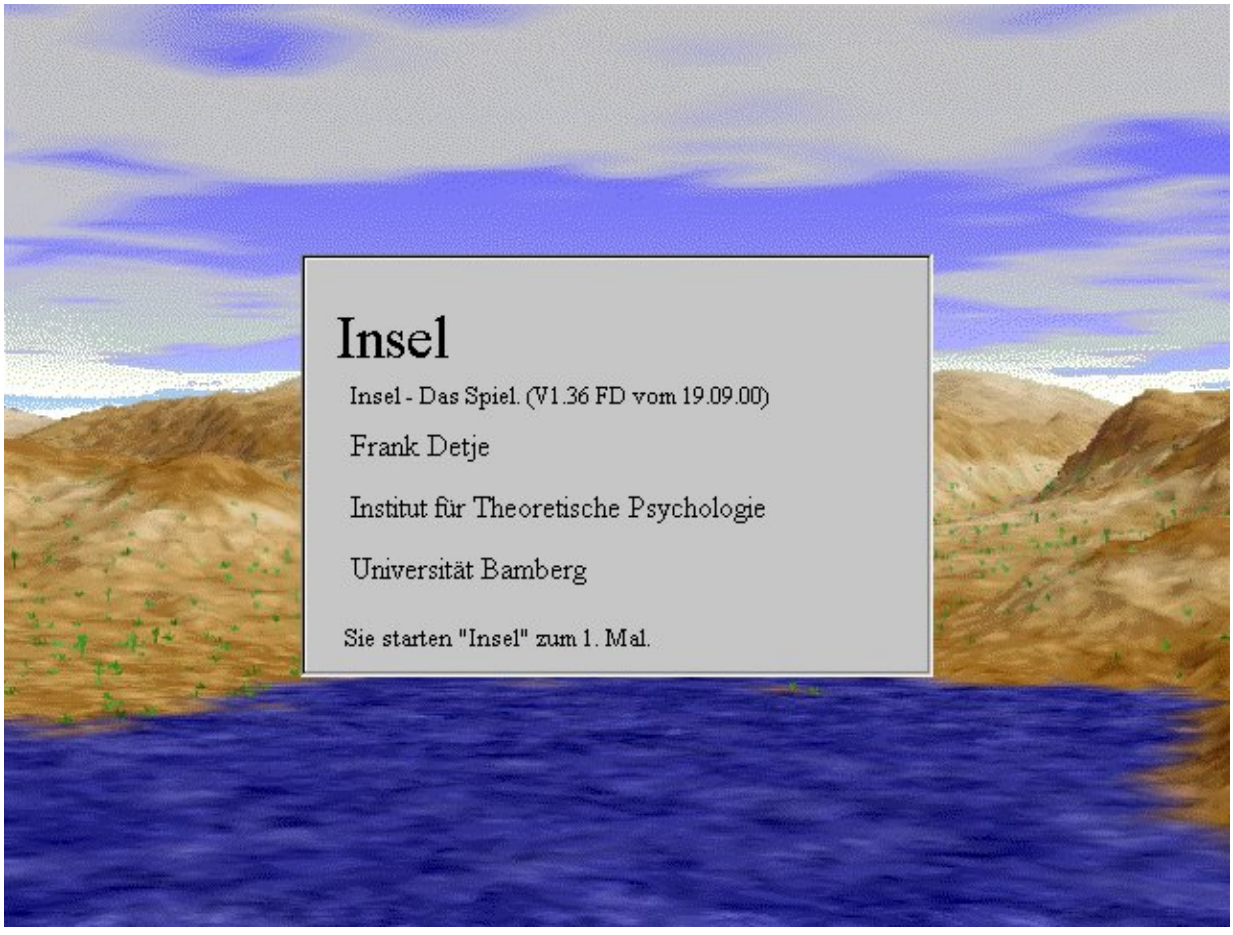

Abbildung 8: Insel – das Titelbild.

Es wird dann automatisch das Spielmenü angezeigt, das nur die beiden Optionen ”Start” und ”Ende” beinhaltet (siehe Abbildung 9). Mit ”Start” starten Sie das Spiel, mit ”Ende” beenden Sie das Programm.

Abbildung 9: Insel – die Menüleiste.

Zu Beginn des Spielstarts werde sämtliche Bild-Dateien aus dem Verzeichnis \bmp\ eingelesen. Das dauert etwas und dies wird in der Menüleiste gemeldet. Man wird also etwas warten müssen, ehe die anfängliche Spielsituation angezeigt wird (siehe Abbildung 10).

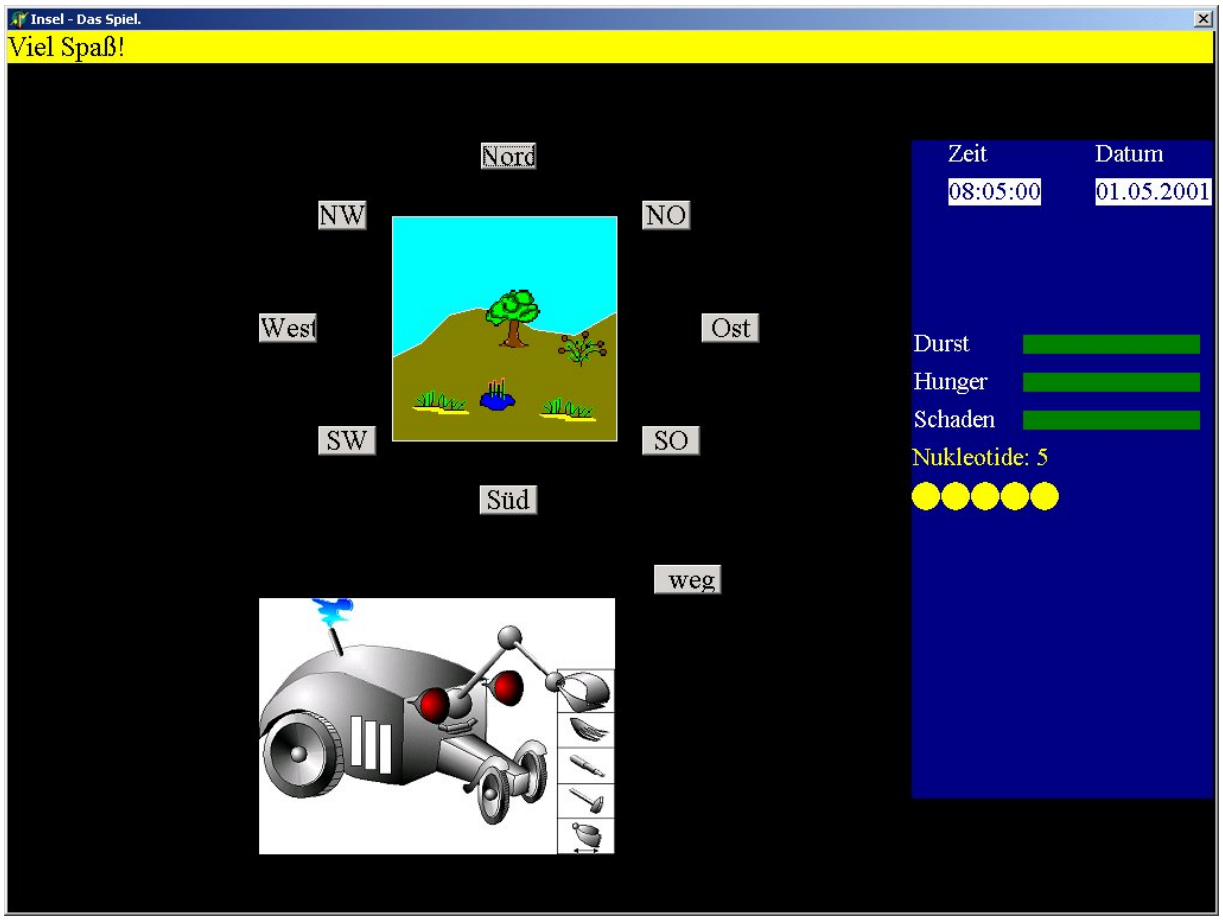

Abbildung 10: Insel – die Startsitation.

3.5 Spielkonstruktion

Insgesamt werden vom Programm drei Parameterdateien benötigt, um ein vollständiges Spiel aufbauen zu können, gleichzeitig heisst dies, dass es drei Parameterdateien sind, die in einem einfachen Texteditor manipuliert werden können, um eigene Spiele zu konstruieren. Es handelt sich um ”insel.ini”, ”Insel*.dat” und ”Manip*.dat”. Die Verwendung des Asterix ”*” bei der Angabe der Insel- und der Manipulationsdateien verweist darauf, dass Insel verschiedene Spielmodi zu konstruieren erlaubt. Das Programm kennt bestimmte Features für bestimmte Spielmodi, die anderen Spielmodi nicht zur Verfügung stehen.

3.5.1 Die verschiedenen Spiel-Modi

Das Programm ”Insel” erlaubt es, Spiele unter vier verschiedenen Spielmodi zu konstruieren:

a) Insel. *Standardspiel.*

b) InselLight. *Eine weniger ”stressige” Version. U.a.: Der Bedarfssound und der Belohnungssound sind ausgeschaltet. Die Zusammenbruchsmeldung klingt nicht so hart.*

c) InselFun. *Eine Spielversion, die zusätzliche Eigendynamik aufweist, der Anstieg der Bedürfnisse in der Zeit ist nicht gar so rasant. Die InselLight-Änderungen gelten auch hier.*

c) InselFunSozial. *Eine Spielversion, die neben der zusätzlichen Eigendynamik auch die Interaktion mit anderen (allerdings passiven) James‘ ermöglicht.*

Die genaueren Erläuterungen für das Standardspiel, das einige Versuchspersonengruppen und die PSI-Agenten spielten, erfolgen weiter unten; für die anderen Spiel-Modi siehe Detje (2000b).

3.5.2 Der Aufbau der Parameterdatei ”Insel.ini”

Diese Datei enthält die Konkretisierung eines spezifischen Inselvariablen-Sets. Sämtliche Parameter, die der Spielkorpus des ”Insel”-Programms bereithält, werden hier festgelegt. D.h. ohne diese Datei kann das Programm ”Insel” nicht gestartet werden.

```
"Insel.ini" KonfigurationsDatei (V1.36 FD+US+DD vom 21.09.00)
    {Mögliche Versionstypen: "Insel", "InselLight", "InselFun", "InselFunSozial"}
    {Wichtig: Die exakte Schreibweise des Typs, denn danach werden die Dateien eingelesen}
Insel
            {xxxxxxxxxxxx Zeiten xxxxxxxxxxxxxxxxxxxxxxxx}
08:00:00                    {fiktive Zeit im Spiel beim Start}
01.05.2001                  {fiktives Datum im Spiel beim Start}
60 15 29 45 0 0 0           {AutoEnde, 6 * Unterbrechungen (alles in Minuten)}
1000 5 60 15                {SpielTakt, MinProSpielZeit, Länge der Auszeit in Sekunden, schwer + leicht}
1 1000 0 10                 {2 * FahrenNachZeit, 2 * FahrenNachKlick}
200 10                      {DelayTime, ZoomDelayDiv}

            {xxxxxxxxxxxx Versuchsbedingung xxxxxxxxxxxxx}
J                           {ToolJames = 'J', Buttons = 'B'}
0                           {Versuchsbedingung (GoldManip): -x: wegnehmen, 0: nichts, +x, schenken}
10 2                        {Wann soll das geschehen (alle x Min.) und wann umgekehrt (x Min vor Ende)}
0 sprachlich neutral        {Versuchsbedingungen (SprachManip)}
Nukleotide
5                           {StartGold}
1 5                         {sollen Zusammenbrüche kosten? Wieviel?}

            {xxxxxxxxxxxx Eigendynamik xxxxxxxxxxxxxxxxxx}
0 1 1 1 1                   {Keine, ZeitVergeht, Motivabfall, Apfelwachstum, Brunnenwachstum}
1 1                         {EssenWachsChance, TrinkenWachsChance}
0.0002 0.0004 0.02 0.03 0.07 0.05       {Hunger-Weg, Durst-Weg, Hunger-Manip, DurstM, SchadenSteinschlag,
Schaden-Wasser}
Schwefeldämpfe haben Ihr Röhrensystem verletzt.

            {xxxxxxxxxxx Standort und Motive xxxxxxxxxxxxx}
8                           {LfdNr des StartPunktes}
4                           {Motive: N}
0 1 0.0023 0.0016 -1 0.6 Durst
Durst.wav
0 1 0.002 0.0013 -1 0.6 Hunger
Hunger.wav
0 1 0.002 0.0014 -1 0.6 Schaden
```

```
Schaden.wav
0 0 -0.001 -0.001 0 0 Regeneration
nosound.wav
Sie haben "James" zugrunde gerichtet.
Nun müssen Sie warten, bis Ihnen ein neuer Roboter zur Verfügung gestellt wird.
"James" hat eine seiner Ressourcen verbraucht.
Bitte warten Sie, bis Ihnen ein neuer Roboter zur Verfügung gestellt wird.

        {xxxxxxxx Bildschirmaufbau xxxxxxxxxxxxxxxxxxx}
18                      {PunktGroesse}
700 700                 {ConvertDDX, ConvertDDY);
-90 -120 190 190        {BMSit-Parameter: StartX, StartY, Width, Height}
50 50                   {BMPart-Parameter: Width, Height}
2                       {Anzahl "noway"-SoundDateien}
1 4 4 0 0 4 0           {StatusZeile: AnAus, Div X, Div Y, StartX, StartY, Width, Height}
20 30                   {ProtokollZeile: X, Y}
0                       {mindestDistanz zwischen zwei Punkten; -1 für ObjektDichteTest}
0                       {LKBox-Anzeigen: -1= nur Punkte und Linien, 0=minimaler Text, 1=erweitert}

        {xxxxxxxx Buttons xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
1024 x 768
200 600
350 600
250 645
330 550
400 655
180 535
490 560
470 505
520 610
270 485
545 450 weg
800 x 640
215 470
340 480
250 520
325 430
420 520
205 410
460 450
440 390
320 350
240 380
470 310 weg

        {xxxxxxxx ToolJames xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
160 350 300 216             {XStart, YStart, Breite, Hoehe}

{xxxxxxxx Steuerdateien werden jetzt automatisch aus Versionstyp generiert xxxxxxxxxx}
"Insel*.dat"
"Manip*.dat"
```

Abbildung 11: Die Datei ”insel.ini”.

Die ersten drei Zeilen enthalten Bemerkungen. Sie werden beim Programmstart vom Programm überlesen.

Die 4. Zeile enthält die Angabe über den Spieltyp. Dieser wird verwendet, um a) entsprechende Features für den Programmablauf zur Verfügung zu stellen, b) Änderungen des Spielablaufs gegenüber der Standardversion einzuführen und c) entsprechende Manipulations- und Inseldateien einzulesen. Für jeden der vier Spieltypen wird eine eigene Insel- und eine eigene Manipulationsdatei erstellt (dazu siehe unten).

Wichtig ist es bei der Angabe der Motive auch die Anzahl der Motive zu ändern (2. Zeile im entsprechenden Abschnitt), wenn neue Motive (wie z.B. ”Affiliation” für die InselFunSozial-Version) hinzugefügt werden. Die letzten vier Zeilen dieses Abschnitts werden je nach Spieltyp verwendet. Für den Spieltyp ”Insel” werden die ersten zwei Zeilen verwendet, für die übrigen Spieltypen die anderen beiden.

Die nun folgenden Angaben zur Konstruktion einer Insel und den erlaubten Handlungsmöglichkeiten beziehen sich sowohl auf das Inselspiel für Versuchspersonen als auch auf die PSI-Lebensumwelt. Das PSI-Programm liest vor seinem Start identisch aufgebaute Dateien ein, in denen die konkrete Umwelt und die konkreten Handlungen und ihre Auswirkungen definiert sind.

3.5.3 Der Aufbau der Parameterdateien ”Insel*.dat”

Die Dateien ”Insel*.dat” beinhalten die Konkretisierung einer Insel, d.h. Angaben über ihre Oberflächenstruktur für die Landkarte, der Orte (Situationen) und der Wege (Verbindungen).

```
Realität "Insel" (V1.31 FD+US+DD vom 25.04.00)
--------- Insel -----------------
  192  33  230  102  280  127  347  127  457  120
  533  140  570  170  583  200  577  273  570  337
  557  400  540  427  507  393  460  363  407  360
  377  367  347  400  327  433  340  497  370  523
  427  537  480  540  493  563  503  587  507  617
  500  643  480  667  443  680  407  680  363  667
  320  647  295  607  290  553  297  517  290  463
  283  437  260  427  210  407  107  327   47  290
   33  273   30  240   37  207   73  183  137  140
  170  110  176   80

---------- Landschaften ----------
Mittelwald
  207  110  253  153  323  187  373  230  383  273
  353  287  310  267  303  220  247  207  173  180
  173  143
  32768
Steppe
  237  187  300  213  350  287  320  333  237  327
  183  317  100  277   97  233  123  200  140  180
  167  167
  65280
Suedwald
  167  303  207  313  240  317  283  313  330  300
  327  363  270  370  243  357  210  343  187  337
  160  327
  32768
Moor
  317  187  453  187  517  210  540  267  530  313
  500  337  460  320  417  293  377  270  350  250
  320  223
  8421376
Steinstrand
  127  327  163  350  190  373  240  397  297  387
  320  380  337  400  320  413  260  427  200  407
  123  343
  8421504
Sonnenfeld
  300  330  367  327  400  297  397  270  363  270
  333  300  330  313
  128
```

```
Westwald
  320  557   347  563   370  610   370  650   343  650
  317  620   303  583
  32768
Suedsteppe
  350  590   400  580   450  590   467  603   443  640
  400  647   363  633
  65280
Sonnenhalde
  320  557   353  537   413  543   443  567   460  593
  407  600   353  600
  128

-------- Orte ------------
Norddünen    // 1
 110  170
 0 140   30 115   40 105   60 100   80 70   100 45   130 15   159 10
 65535   16711680    // gelb blau  (Meer)
 Due1 0 100  Duene 30 80  Due1 70 55   Duene 105 35  Due1 110 5  Duene 90 80 Fisch 20 20
 SW 21 Flachdünen 1 0.9 NO 60 Nordkap 5 0.8  O 66 Haselwald 0 1  SO 48 Steppe 0 1
Flachdünen    // 2
 82  200
 0 145   20 125   40 115   60 90   80 75   100 45   130 15   159 10
 65535   16711680   // gelb blau (Meer)
 Duene 0 110  Due1 30 85  Duene 100 10  Pfef 50 110  Busch 110 110
 SW 25 Buschdünen  0 0.9  NO 21 Norddünen 0 0.9  O 59 Waldweide 0 1
Buschdünen    // 3
 47  237
 0 145   20 125   40 115   60 90   80 75   100 45   130 15   159 10
 65535   16711680   // gelb blau (Meer)
 Duene 0 110  Due1 30 85  Duene 100 10  Busch 50 110  Hasel 95 65  Busch 110 110
 SO 55 Klipport  3 0.9  NO 25 Flachdünen 0 0.9  O 56 Steppe 0 1

[analog geht es weiter]
```

Abbildung 12: Die Datei ”Insel.dat”.

Die Dateien ”Insel*.dat” besteht neben dem Header (”Realität...”) aus drei Abschnitten.

1. ”--------- Insel ----------”. Hiernach wird der Umriss der Insel als Punkte eines Polygons festgelegt (erst x-Wert, dann y-Wert eines Punktes).

2. ”---------- Landschaften ----------”. Die Insel wird in einzelne Landschaften unterteilt. Diese erhalten einen Namen (erste Zeile), Angaben über den Polygonzug der Landschaft innerhalb der Insel und in der letzten Zeile die Farbe mit der sie dargestellt werden sollen (siehe Abbildung 13).

```
Sonnenhalde
  320  557   353  537   413  543   443  567   460  593
  407  600   353  600
  128
```

Abbildung 13: Ein Beispiel für eine Landschaft der Insel.

3. ”-------- Orte ----------”. Für die Versuchspersonen sichtbar sind aber nur die Orte. Diese werden im dritten Abschnitt nacheinander definiert und müssen sehr sorgfältig erarbeitet werden. Die hier vorgestellten Konventionen sind auf jeden Fall zu beachten, denn nicht alle Fehler können abgefangen und in der Fehler-Protokolldatei ”InselFehler.prt” festgehalten werden.

Jeder Ort erhält in der ersten Zeile einen Namen (sinnvoll ist es, die Orte durchzunumerieren, erforderlich ist dies nicht). In der zweiten Zeile stehen die x- und die y- Koordinate dieses Ortes auf der Insel (siehe Abbildung 14).

```
Norddünen    // 1
 110  170
 0 140   30 115   40 105   60 100   80 70   100 45   130 15   159 10
 65535   16711680    // gelb blau  (Meer)
 Due1 0 100  Duene 30 80  Due1 70 55   Duene 105 35  Due1 110 5  Duene 90 80 Fisch 20 20
 SW 21 Flachdünen 1 0.9 NO 60 Nordkap 5 0.8  O 66 Haselwald 0 1  SO 48 Steppe 0 1
```

Abbildung 14: Ein Beispiel für einen Ort auf der Insel.

Die Darstellung des Ortes als Situation beginnt in der dritten Zeile. Der Polygonzug, der eine Situation in einen oberen und einen unteren Abschnitt teilt (Horizont), wird in der dritten Zeile definiert.

Die vierte Zeile enthält die Farbe des Untergrundes (1. Wert) und die Farbe des Himmels (2. Wert), gleichbedeutend mit der Farbe unterhalb bzw. oberhalb der Horizontlinie. Optional können nicht weiter verwendete Angaben wie Bemerkungen hinzugefügt werden.

Die Definition der Objekte einer Situation erfolgt in Zeile 5. Jedes Objekt besteht aus einem Namen. Dies ist der Namen der Bitmap, die eingelesen werden soll, d.h. es muss von dem ”Programmierer” vorher sicher gestellt sein, dass Dateien dieses Namens existieren. Fehler bei der Definition können zu unvorhersehbaren Programmfehlern führen. Zudem werden hier die Koordinaten der Bitmaps in der Situation angegeben (x- und y-Angabe). Die einzelnen Objekte werden nacheinander in einer Zeile definiert.

Die sechste Zeile enthält die Verbindungen von diesem Ort zu anderen Orten (die Definition erfolgt unidirektional, bidirektionale Verbindungen müssen also an beiden Orten als unidirektionale Verbindungen definiert werden).

Dazu wird a) der Richtungsbutton angegeben, der die erfolgversprechende Änderung des Ortes angibt, b) die Entfernung, die durch den Ortswechsel zurückgelegt werden soll, c) der Namen des Ortes, zu dem die Verbindung hergestellt wird (Achtung: der Name muss in genau dieser Schreibweise als Ort der Insel definiert sein bzw. werden), d) das Risiko für einen Steinschlag oder für austretende Schwefelgase (die entsprechende Meldung, und der Anteil Schaden, der entsteht, werden in der ”Insel.ini” festgelegt); es bedeuten: ”0” = keine Wirkung, ”1” bis ”n”, jedes n-te Mal, das diese Verbindung gewählt wird, kommt es zu diesem Ereignis; und e) die Güte der Wegstrecke im Bereich [0..1]. Dieser Wert geht in den Schaden mit ein, insofern, als dass ”1 - Wert” dem Schaden hinzu addiert wird, d.h. ein Wert von 1 verursacht keinen zusätzlichen Schaden. Die einzelnen Verbindungen werden in einer Zeile definiert.

Die verschiedenen Orte werden einfach nacheinander ohne Leerzeile definiert.

3.5.4 Der Aufbau der Parameterdateien ”Manip*.dat”

Diese Dateien enthalten nach dem Header (”ManipulationsFile...”) in jeder Zeile die Eingriffsmöglichkeiten und die Folgen eines Eingriffs. Die Angaben müssen den unten festgelegten Konventionen folgen, Fehler können zu unabsehbaren Programmfehlern führen.

ManipulationsFile vom 2.1.99						
Sonblu	mN	Sonblu-	Hunger	-0.4	Fressen.wav	Nehmen
Hasel	mN	Hasel-	Hunger	-0.3	Knabbern.wav	Nehmen
Lache	mS	Lache-	Durst	-0.5	Schlurf.wav	Saugen
Schilf	mS	Schilf-	Durst	-0.4	Saugen.wav	Saugen
Pfef	mN	Pfef-	Schaden	-0.4	Rascheln.wav	Nehmen
Baum1	mM	Baum2	nilmotiv	0	Raschplong.wav	Schütteln
Gras	mN	Gras1	nilmotiv	0	Rascheln.wav	Nehmen
Blu	mN	Blu1	nilmotiv	0	Rascheln.wav	Nehmen
Busch	mN	Busch1	nilmotiv	0	Rascheln.wav	Nehmen
Bauma	mM	Baum3	nilmotiv	0	Rascheln.wav	Schütteln
Baum	mM	Baum-	nilmotiv	0	Rascheln.wav	Schütteln
Tannen	mM	Tannen1	nilmotiv	0	Rascheln.wav	Schütteln
Fels1	mC	Fels1-	nilmotiv	0	Schlagen.wav	Hämmern
Fels2	mC	Fels2-	nilmotiv	0	Schlagen.wav	Hämmern
Klippe	mC	Klippe1	nilmotiv	0	Schlagen.wav	Hämmern
Due1	mI	Due1-	nilmotiv	0	Sand.wav	Sieben
Duene	mI	Duene-	nilmotiv	0	Sand.wav	Sieben
Blumen	mN	Blumen1	Schaden	0.2	Ersticken.wav	Nehmen
Fels1-	mN	Fels3	Nukleo	-0.4	Kugel.wav	Nehmen
Due1-	mN	Due2-	Nukleo	-0.4	Kugel.wav	Nehmen
Aeste1	mN	Aeste	Nukleo	-0.4	Kugel.wav	Nehmen
Baum2	mN	Bauma	Nukleo	-0.4	Kugel.wav	Nehmen
Hoehl1	mN	Hoehle	Nukleo	-0.4	Kugel.wav	Nehmen
Wrack2	mN	Wrack1	Nukleo	-0.4	Kugel.wav	Nehmen
Fisch	mN	Fisch1	Nukleo	-0.4	Kugel.wav	Nehmen

Abbildung 15: Die Datei ”Manip.dat”.

Zuerst wird der Name des Objekts angegeben, das manipuliert werden kann (= Name der Bilddatei aus dem Verzeichnis \bmp\, allerdings ohne Dateiendung). Dann folgt ein Kürzel (= Name der Manipulation), das für den Spielablauf ohne Interesse ist (dies ist der Operatorname, den PSI verwendet). Das Ergebnis der Manipulation wird als dritter Eintrag festgehalten, wieder als Objektname, der einer Bitmap-Datei entspricht (er kann identisch bleiben, d.h. die Manipulation bleibt dann ohne Umweltänderung). Es folgen die beiden Angaben der Wirkung für James: zuerst das Motiv, auf das der Eingriff wirkt (”nilmotiv” für keine Auswirkung), dann die Stärke der Änderung (-1..0..+1), wobei gilt: negative Werte senken den Bedarf (bedürfnisbefriedigend), positive Werte steigern den Bedarf (bedürfniserzeugend), ”0” steht für keine Auswirkung. Als sechster Eintrag wird die Sounddatei festgelegt, die bei diesem Eingriff abgespielt wird (Datei aus \wav\; Angabe mit Dateiendung). Zuletzt steht der Name des Buttons und der Name der Teilfunktion von James Multifunktionsarm, der bzw. die angeklickt werden muss, um diese Manipulation auszuführen.

4. Die Untersuchungen mit Menschen

Das Szenario ”Insel” diente uns zu erst als Realitätsausschnitt, um das Verhalten von Versuchspersonen unter verschiedenen experimentellen Bedingungen zu studieren und zu vergleichen. Insgesamt haben wir bisher das Verhalten von 7 Gruppen untersucht. Wir wählten folgende Versuchsmanipulationen, die Antworten auf unterschiedliche Forschungsfragen geben sollten:

1. Zum Einfluss der Emotionen auf die Handlungsorganisation: zwei Gruppen (N jeweils 10) erhielten unterschiedliche Spielvarianten.

2. Zum Einfluss der Sprache auf die Handlungsorganisation: zwei Gruppen erhielten unterschiedliche Versuchsinstruktionen (N jeweils 10) (zu den Ergebnissen siehe Bartl, 2000).
3. Zur Struktur der Handlungsorganisation: eine Gruppe wurde gebeten, während der Bearbeitung laut zu denken (N = 10).
4. Zum Einfluss von Stress auf die Handlungsorganisation: eine Gruppe erhielt eine einfachere Spielversion, nämlich ”InselLight” (N = 20).
5. Kontrollgruppe: eine Gruppe spielte ”Insel”, wie vorher beschrieben ohne, dass es Manipulationen in den Versuchsbedingungen gab (N = 10).

4.1 Die protokollierten Daten

Das Szenario ”Insel” und sein Versuchsaufbau erlaubt es, eine ganze Reihe von unterschiedlichen Indikatoren, die Relevanz für die Handlungsorganisation besitzen (können), für Vergleiche zwischen den Versuchspersonengruppen und mit PSIs heranzuziehen. Darunter befinden sich z.B. die folgenden Indikatoren:

A. Kategorien der Auswertung

- Versuchsbedingung

B. Allgemeine Daten

- Alter
- Geschlecht

C. Leistung Exploration

- Anzahl der versuchten Ortswechsel (erfolglose und erfolgreiche)
- Anzahl verschiedener besuchter Orte
- Anzahl verschiedener benutzter Wege

D. Leistung Nukleotide

- Nukleotide: Haben (Endergebnis an Nukleotiden)
- Nukleotide: Nehmen (Anzahl der Aktionen ”Nehmen”, d.h. Anzahl der erfolgreichen Versuche, Nukleotide aufzusammeln)
- Nukleotide: Sammeln (Anzahl der tatsächlich aufgesammelten und damit auch auf der Insel gefundenen Nukleotide. Diese Zahl weicht von ”Haben” gegebenenfalls ab, da die Versuchsbedingungen Einfluss auf das Endergebnis haben, und ein Zusammenbruch Nukleotide kostet, diese Zahl weicht von ”Nehmen” gegebenenfalls ab, da unterschiedlich viele Nukleotide durch ”Nehmen” aufgesammelt werden können, je nachdem, ob ein Objekt ein oder mehrere Nukleotide enthält.

E. Leistung James (jeweils bezogen auf eines der Bedürfnisse Durst, Hunger, Schaden und einen Gesamtwert)

- Anzahl der Zusammenbrüche
- Mittlerer Bedarf an Ressourcen
- Durst (Anzahl wie oft der Bedarfssound ertönte, Sekunden, die er zu hören war, „Rettungen“, d.h. Bedarfsstillungen, während der Sound zu hören war)
- Hunger (SoundNAn, SoundSekAn, Rettungen)

- Schaden (SoundNAn, SoundSekAn, Rettungen)
- mittlerer Bedarf an Durst (1. Viertel, 2. Viertel, 3. Viertel, 4. Viertel)
- mittlerer Bedarf an Hunger (1. Viertel, 2. Viertel, 3. Viertel, 4. Viertel)
- mittlerer Bedarf an Schaden (1. Viertel, 2. Viertel, 3. Viertel, 4. Viertel)

F. Eingriffsverhalten (jeweils getrennte Daten für Gesamtklicks, erfolgreiche und erfolglose Eingriffe, Fehler)

- Anzahl der Klicks
- Anzahl der Objektannäherungen („Zoom“)
- Richtungswahl
- Sieben
- Schütteln
- Saugen
- Nehmen
- Hämmern
- Alle Manipulationen
- Alle Operationen
- Fehlklicks insgesamt

4.2 Unterschiede zwischen den Versuchsgruppen

Wie bereits genannt, haben wir die Untersuchung mit ”Insel” unter verschiedenen Versuchsbedingungen durchgeführt. Neben einer Kontrollgruppe verfügen wir über folgende Experimentalgruppen:

- ”Ärger”; dieser Gruppe wurden regelmäßig Nukleotide „geschenkt“
- ”Freude”; dieser Gruppe wurden regelmäßig Nukleotide „weg genommen“
- ”Selbstbefragung”; diese Gruppe wurde aufgefordert, Fragen über ihr Vorgehen zu beantworten
- ”Kommentieren”; diese Gruppe wurde aufgefordert, zu erzählen, was sie sah und tat
- ”Laut Denken”; diese Gruppe wurde aufgefordert, zu erzählen, woran sie während des Spielens dachte
- ”leichte Insel-Version”; diese Gruppe bekam eine andere Inselversion

Tabelle 1 zeigt die Mittelwerte für einige Leistungs- und Verhaltensvariablen der verschiedenen Gruppen . Neben vielen statistisch signifikanten Unterschieden zwischen den Gruppen, die hier im Einzelnen nicht geschildert werden können, ist besonders beachtenswert, dass die Versuchspersonen, die die leichte Insel-Version spielten, im Mittel nicht weniger Zusammenbrüche, sondern mehr als die meisten anderen Gruppen hatten. Sie sammeln aber auch mehr Nukleotide, so dass als Ergebnis festgehalten werden kann, dass diese Versuchsmanipulation zu einer Verbesserung bezüglich des einen Instruktionsziels (Sammeln von Nukleotiden) führte, aber zu einer Verschlechterung bezüglich des anderen Instruktionsziels (James am Leben halten).

	Kontrollgruppe	**Ärger**	**Freude**	**Selbstbef**	**Kommentieren**	**Laut Denken**	**Light**
bisherige Gruppengröße	N=10	N=10	N=10	N=10	N=10	N=9	N=20
Versuchsbedingung	0,000	-2,000	2,000	0,000	0,000	0,000	0,000
Spielbeginn	10:58:53	11:25:20	10:49:05	14:17:00	14:37:54	11:46:49	13:40:42

Orte gesehen	31,500	32,400	30,900	33,400	31,000	31,556	33,000
Wege gegangen	63,500	65,200	60,700	65,600	59,900	62,444	69,600
Nuk: Aufsammeln	32,900	30,000	27,900	35,800	29,900	26,222	38,450
Nuk: Aufgesammelt	44,900	40,300	38,400	50,200	39,700	37,556	52,850
Nuk: Haben	23,700	13,400	20,200	23,800	13,200	17,778	21,500
Nuk: Schuldenaufnahme	0,900	1,900	1,500	0,100	3,200	0,889	0,850
Nuk: Schulden	3,300	5,100	3,500	0,100	10,000	2,111	2,400
Z Durst	3,400	4,900	3,300	4,300	6,000	3,111	4,250
Z Hunger	1,800	1,000	2,100	1,500	1,900	1,778	2,350
Z Schaden	0,700	0,600	1,100	0,500	0,400	0,444	1,150
Z gesamt	5,900	6,500	6,500	6,300	8,300	5,333	7,750
weg-NKlick	164,600	188,800	166,700	181,000	161,400	159,222	199,450
weg-NPLus	154,600	168,800	156,400	174,200	149,300	137,333	190,050
AlleRichtungen-NKlick	246,100	250,700	276,200	278,100	230,000	246,444	320,550
AlleRichtungen-NPLus	110,600	114,000	115,700	115,600	100,000	117,111	143,950
AlleRichtungen-NMinus	135,500	136,700	160,500	162,500	130,000	133,222	176,600
AlleOP-NKlick	261,800	275,400	266,000	277,400	295,900	210,000	316,500
AlleOP-NPLus	172,300	174,300	163,600	188,700	160,600	159,222	206,500
AlleOP-NMinus	43,700	71,800	71,200	73,800	109,100	40,778	82,550
AlleOP-NFehler	45,800	29,300	31,200	14,900	26,200	15,623	27,450
GesamtOP-NKlick	842,000	901,200	882,200	922,600	864,700	784,889	1045,850
GesamtOP-NPLus	592,300	626,100	592,700	653,200	559,600	549,111	730,750
GesamtOP-NMinus	203,900	245,800	258,300	254,500	278,900	215,667	287,650
GesamtOP-NFehler	45,800	29,300	31,200	14,900	26,200	15,683	27,450
Durst (Motivmittel)	0,416	0,407	0,387	0,406	0,405	0,384	0,401
Hunger (Motivmittel)	0,366	0,337	0,350	0,361	0,352	0,329	0,354
Schaden (Motivmittel)	0,252	0,255	0,254	0,285	0,242	0,265	0,256

Tabelle 1: Mittelwerte verschiedener Versuchsgruppen bezüglich einiger Variablen (aus Detje, 2000b).

4.3 Diskussion

Die durchgeführten Untersuchungen zeigen insgesamt betrachtet, dass „Insel“ als Szenario eine sehr große Verhaltensvariabilität zwischen den TeilnehmerInnen erlaubt. Es lassen sich verschiedene Spielstile identifizieren. Im Prinzip ist das Spiel, wie es für die meisten Versuche verwendet wurde, jedoch als zu „stressig“ für die TeilnehmerInnen zu bezeichnen, auch wenn sich keine Deckeneffekte zeigten. Eine Neujustierung der Spielparameter führte zu der „Insel-Light“-Version, die als angenehmer zu spielen empfunden wird.

Die bisherigen Ergebnisse erlauben zwar aufgrund der sehr kleinen Gruppengrößen keine weiterführenden statistischen Analysen, zeigen jedoch die oben angesprochenen großen Unterschiede zwischen den Versuchspersonen in Bezug auf eine sehr große Vielfalt unterschiedlicher Indikatorvariablen.

Was insbesondere die Untersuchung zum Einfluss der Emotionen betrifft, so muss festgestellt werden, dass die erhofften Verhaltensunterschiede nicht gefunden werden konnten, was, wie dargelegt, wohl auf die Belastung durch das recht schwierige Spiel selbst zurückzuführen ist.

5. Die Insel von PSI und PSIs Bewährung in dieser Umwelt

Von zentralem Interesse für das Institut für Theoretische Psychologie und auch erster Grund für die Konstruktion des Computerszenarios „Insel“ ist die Weiterentwicklung und Überprüfung der Theorie menschlicher Handlungsregulation, der PSI-Theorie (Dörner, 1999)

und ihrer Implementation[3]. In Abbildung 17 ist dargestellt, wie sich die weiter oben vorgestellte Insel bei PSI darstellt. Im Gegensatz zum Versuchspersonenspiel ist die Insel mit ihren Orten, Verbindungen und dem aktuellen Aufenthaltsort von PSI / James ständig zu sehen. Dies gilt allerdings nur für den Betrachter des Computerbildschirms; PSI selber sieht nur jeweils die Situation des aktuellen Aufenthaltsortes.

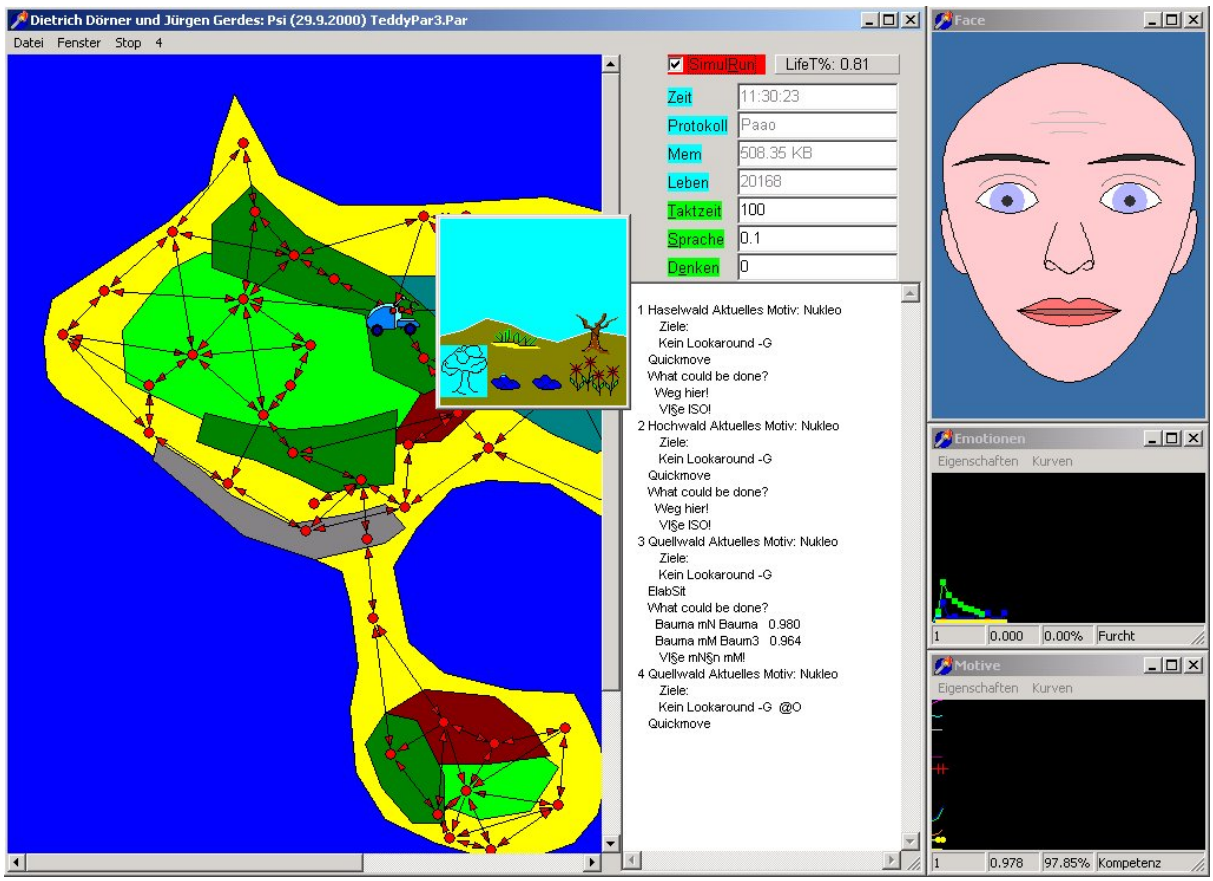

Abbildung 17: Die Oberfläche des PSI-Programms

Weitere Fenster, die ein Gesicht zeigen, den Verlauf der Werte für Emotionsparameter und Motive und das Protokoll der PSI-Prozesse dienen der On-Line-Analyse von PSIs Verhalten. PSI ist in der Lage, das Szenario autonom zu bewältigen.

Verschiedene methodische Zugänge erlauben es, das Verhalten von PSI z.B. auf Konsistenz, Plausibilität oder hinsichtlich der Vergleichbarkeit mit menschlichem Verhalten zu analysieren. Grundsätzlich gilt, dass sozionische Forschungsziele anzustreben, für agentenorientierte Modellarchitekturen erst sinnvoll ist, wenn die einzelnen Agenten sich als hinreichend „menschlich“ erwiesen haben.

Deshalb überprüften wir die Implementation der PSI-Theorie bzw. das von ihr gezeigte Verhalten bei der Bearbeitung der Computersimulation „Insel“ daraufhin, ob die Leistungs- und Verhaltensdaten mit denen menschlicher Versuchspersonen übereinstimmen. Die Ergebnisse sind sehr zufriedenstellen: In der Mehrzahl der herangezogenen Indikatorvariablen des Verhaltens und der Leistung finden sich zwischen PSIs und Menschen keine nennenswerten Unterschiede (s. Detje, 2000a). Die Implementation der Theorie ist also nicht

[3] Ebenso wie das „Insel-Spiel“ ist auch PSI erhältlich über unsere WWW-Seiten unter URL: www.uni-bamberg.de/ppp/insttheopsy/.

nur prinzipiell in der Lage mit der komplexen Umwelt „Insel“ umzugehen, sondern auch mit Menschen vergleichbare Bearbeitungsleistungen zu erzielen. Tabelle 2 zeigt einige Mittelwertsvergleiche zwischen unseren Versuchspersonen und den PSIs und das Signifikationsniveau ermittelt durch den T-Test (N=30) (ausführlich in Detje, 2000a).

	Besuchte Orte				**Aufgesammelte Nukleotide**				**Anzahl benötigter Roboter**			
a)	*mean*	*sd*	*T*	*sig*	*mean*	*sd*	*T*	*sig*	*mean*	*sd*	*T*	*sig*
Vpn	31,84	6,63			42,70	17,13			7,70	1,66		
PSI	33,23	4,35	-1,025	0,308	38,00	5,25	1,459	0,149	6,70	0,99	2,997	0,004
	Anzahl der Ortswechsel				**Objektannäherungen**				**Objektmanipulationen**			
b)	*mean*	*sd*	*T*	*sig*	*mean*	*sd*	*T*	*sig*	*mean*	*sd*	*T*	*sig*
Vpn	111,18	27,54			161,04	27,97			245,82	57,04		
PSI	126,43	30,41	-2,306	0,240	161,03	13,43	0,001	0,999	203,37	15,76	7,415	0,000

Tabelle 2: a) Werte der wichtigsten Leistungsindikatoren; b) Werte anderer Verhaltensvariablen (“mean” = Mittelwert; “sd” = Standardabweichung”, “T” = T-Test, “sig” = Signifikanzniveau, N jeweils 30).

Tabelle 2 zeigt, dass für vier der sechs betrachteten Hauptindikatorvariablen keine signifikanten Unterschiede zwischen menschlichen und künstlichen Gruppen bestehen. Jedoch weisen die Anzahl benötigter Roboter und die Objektmanipulationen bedeutsame Unterschiede zwischen den beiden Gruppen auf. PSI braucht weniger Roboter, erleidet also weniger Zusammenbrüche und die Versuchspersonen manipulieren die Objekte wesentlich häufiger, als es das PSI-Programm tut.

Was die Weiterentwicklung der Theorie und ihrer Implementation betrifft, arbeiten zwei Projekte an entsprechenden Vorschlägen: zum einen das „Plastizitätsprojekt“, dass sich der Frage widmet, wie PSI die Fähigkeit bekommen kann, natürliche Sprache zu verstehen und zu produzieren. Die hierzu als notwendig erachteten Voraussetzungen sind in Dörner (1999) beschrieben. Die spezifische Auswirkungen der Sprache auf das Denken werden innerhalb der Psychologie und den angrenzenden Wissenschaften kontrovers diskutiert und reichen von einer Ablehnung irgendeines direkten Zusammenhanges von Sprache und Denken (Anderson, 1996) bis hin zu einer Gleichsetzung von Sprache und Denken (Platon im Sophistes 263e). Verschiedene Untersuchungen des Instituts haben schon früh auf einen engen Zusammenhang hingewiesen, für neuere Arbeiten siehe z.B. Bartl (1997) und Bartl & Müller (2000). Die Fähigkeit des Menschen zur Selbstreflexion scheint für die Erforschung dieses Zusammenhangs ein wichtiger Aspekt zu sein, der auch mit dem Insel-Versuch näher untersucht wurde. Wie schon angesprochen wurden zwei Versuchsgruppen hinsichtlich ihrer Verbalisierungen während der Bearbeitung unterschiedlich instruiert: Die eine Gruppe „Selbstbefrager“ wurde eine Liste von Fragen zu ihrem Vorgehen bei der Problembearbeitung regelmäßig vorgelegt und die Personen wurden gebeten, diese Fragen laut zu beantworten. Eine zweite Gruppe, die „Kommentierer“, wurde gebeten, ausschließlich das zu Verbalisieren, was sie gerade sahen und taten. Über die Unterschiede im Vorgehen und der Leistung zwischen den beiden Gruppen, und über die Hinweise auf die Weiterentwicklung

der PSI-Theorie, die durch eine gezielte und ausführliche Betrachtung von Einzelfällen gewonnen werden können, berichtet Bartl (2000).

In einer weiteren Studie wurde gezeigt, dass die innerhalb der PSI-Theorie angenommenen Parameter der Modulation des Verhaltens durch systematische Variation tatsächlich auch systematisch zu erwünschten Verhaltensänderungen führt (Detje, i.Vorb.).

Ein dritter Weg, den wir zur Überprüfung der Theorie zur Zeit beschreiten, besteht darin, einzelne Versuchspersonen in ihrem Verlauf durch die Implementation der PSI-Theorie nachzuzeichnen. D.h. es wird versucht, einen Parametersatz für die PSI-Implementation zu finden, der es erlaubt, die Leistungs- und Verhaltensdaten einzelner Versuchspersonen zu reproduzieren. Die Implementation wird dabei selber nicht verändert. Dies gelingt in überraschend gutem Ausmaß, wie Dörner (2000) nachweisen kann. Die Implementation der PSI-Theorie ist damit nachweislich in der Lage, ohne Veränderung der allgemeinen Architektur, verschiedene Verläufe bei der Problembearbeitung zu produzieren, die in sehr hoher Übereinstimmung mit den Verläufen einzelner, konkreter, realer Versuchspersonen sind. Parameter, die sich für die Retrognose von einzelnen Personen bisher als relevant erwiesen, sind z.B. die Kompetenzinkremente und –dekremente, die z.B. dazu führen, dass Resignation bei der Problembearbeitung früher oder später einsetzt.

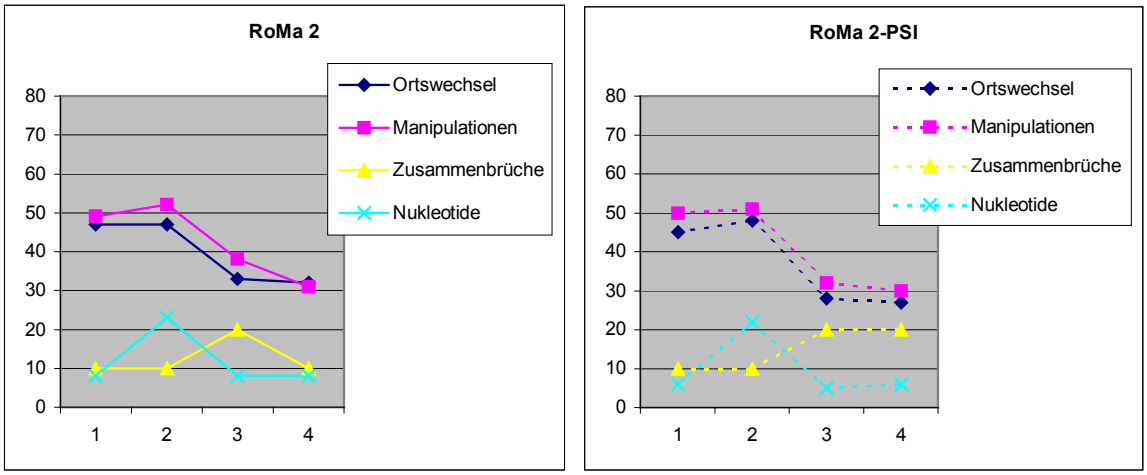

Abbildung 18: Eine menschliche Versuchsperson („RoMa2“) und ein PSI-Lauf („RoMa2‘).

Abbildung 18 (nach Dörner, 2000) zeigt, dass die menschliche Versuchsperson „RoMa2“ hinsichtlich der wichtigsten Leistungsindikatoren im Verlauf der Bearbeitungszeit durch die PSI-Implementation („RoMa2-PSI“) sehr gut abgebildet werden kann. Dörner (2000) wies dies für mehrere Versuchspersonen nach.

6. Ausblick

Die hier nur kurz geschilderten Versuchsergebnissen zu den Vergleichsstudien zwischen PSIs und Menschen zeigen, dass die PSI-Implemenation hinreichend gut ausgebaut ist, um menschliches Verhalten bei der Bearbeitung des „Insel“-Spiels abbilden zu können. Dies gilt sowohl hinsichtlich der erbrachten durchschnittlichen Leistungen als auch hinsichtlich des Bearbeitungsverlaufs für den Einzelfall.

Im nächsten Schritt werden wir die PSI-Implementation in Hinblick auf sozionische Forschungsfragen und Forschungsziele ausbauen. Dazu werden wir PSI zuerst mit einem

Affiliationsmotiv ausstatten (vereinfacht ausgedrückt ist dies ein Bedürfnis nach Geselligkeit), um zu analysieren wie sich diese Änderung auf PSIs Verhalten, die internen Prozesse und die Bedürfnisbefriedigung auswirkt. Von ersten diesbezüglichen Untersuchungen berichtet Detje (2001).

In einer weiteren Ausbaustufe werden wir schliesslich die PSI-Implementation auf einen Multi-Agenten-Betrieb umstellen und die „Insel“-Welt in eine 3D-Insel überführen. Diese Änderungen sollen es erlauben, mehrere PSIs in der selben Lebensumwelt agieren zu lassen, wobei sie sich frei bewegen können und bestimmte Probleme nur gemeinsam lösen können. Nach Abschluss dieser programmtechnischen Aufgaben können wir dann Gruppenbildungsprozesse, Stabilitätsbedingungen und Zerfallsprozesse von Gruppen und Subpopulationen in PSI-Gesellschaften unter sehr verschiedenen Bedingungen untersuchen. Diese verschiedenen Bedingungen betreffen unter anderem:

a) den Aufbau der Umwelt: sie kann ressourcenreich (paradiesisch) oder ressourcenarm sein, sie kann voller unerwarteter Ereignisse oder relativ konstant sein, die Umweltänderungen können schnell oder verspätet (lange Totzeiten) eintreten;

b) die Fähigkeiten der Agenten: wir können die PSIs so gestalten, dass bestimmte Objektmanipulation nur bestimmten PSIs möglich ist (einige können gut Steine zertrümmern, andere hingegen Dünen sieben), dass sie unterschiedlich viel Ressourcen speichern können (einige haben große Wassertanks, andere kleine, was bei Durchquerungen von Wüsten eine entscheidende Rolle spielen kann) oder einige PSIs können sich z.B. schneller als andere bewegen;

c) die Persönlichkeits- oder Temperamentseigenschaften der künstlichen Agenten: einige PSIs resignieren schneller als andere, weil ihr Kompetenzpegel durch Misserfolg schneller absinkt und vice versa: einige sind wagemutiger, weil sie durch Misserfolge nicht beeindruckt werden (geringes Dekrement der Kompetenz bei Misserfolg), einige PSIs vergessen schneller, was sie gelernt haben als andere oder einige PSIs prüfen genauer als andere PSIs die Bedingungen des Handelns und des Planens (hoher versus niedriger Auflösungsgrad).

7. Literatur

Anderson, John Robert (1996[2]): *Kognitive Psychologie*. Heidelberg: Spektrum.

Bartl, Christina (1997): *Sprachlos beim Denken – Eine Untersuchung zur Rolle des „inneren Gesprächs der Seele mit sich selbst" für das Problemlösen.* Bamberg: Lehrstuhl Psychologie II, Memorandum Nr. 23.

Bartl, Christina (2000): Metacognition in Complex Problem Solving: How to implement the effects of metacognition on goal selection in the architecture of PSI. In Niels Taatgen & Jans Aasman (Hrsg.): *Proceedings of the Third International Conference on Cognitive Modeling*. Veenendaal: Universal Press, S. 33-41.

Bartl, Christina & Dörner, Dietrich (1998): *Comparing the behaviour of PSI with human behaviour in the BioLab Game.* Bamberg: Lehrstuhl Psychologie II, Memorandum Nr. 32.

Bartl, Christina & Müller, Gerhard (2000): *Beeinträchtigung der Problemlösefähigkeit bei Broca-Aphasie*. Bamberg: Lehrstuhl Psychologie II, Memorandum Nr. 37.

Detje, Frank (2000a): Comparison of the PSI-theory with human behaviour in a complex task. In Niels Taatgen & Jans Aasman (Hrsg.): *Proceedings of the Third International Conference on Cognitive Modeling*. Veenendaal: Universal Press. S. 86-93.

Detje, Frank (2000b): *”Insel”. Dokumentation – Versuche – Ergebnisse*. Bamberg: Institut für Theoretische Psychologie, Memoramdum Nr. 39.

Detje, Frank (2001): Sozionik: *PSI - Erste Schritte in Richtung sozialen Verhaltens*. Bamberg: Institut für Theoretische Psychologie, Memoramdum Nr. 41.

Detje, Frank (i. Vorb.): *PSI-Vergleiche: Die Variation der Modulationsparameter*. Unveröffentlichtes Manuskript.

Dörner, Dietrich (1999): *Bauplan für eine Seele*. Reinbek: Rohwolt.

Dörner, Dietrich (2000): The Simulation of Extreme Forms of Behaviour. In Niels Taatgen & Jans Aasman (Hrsg.): *Proceedings of the Third International Conference on Cognitive Modeling*. Veenendaal: Universal Press, S. 94-99.

Schaub, Harald (1993): *Modellierung der Handlungsorganisation*. Bern: Huber.