Computer-Aided Game Inventing

Ingo Althofer
Faculty of Mathematics and Computer Science
Friedrich-Schiller University Jena
07740 Jena -- Germany
althofer @ minet.uni-jena.de

01 October, 2003

Abstract

"Morphling" is a computer program for designing and playing whole classes of combinatorial 2-person games. Programs like Morphling help to speed up the process of inventing new games.

Key Words: combinatorial games, automatic game evaluation, computer-aided inventing;

1 Introduction

Tom Werneck has written the nice paperback "Leitfaden fuer Spiele-Erfinder und solche, die es werden wollen" (title translated to English: "Introduction for Game Inventors and those who want to become it"). In the meantime it has become a classic with five editions [Wer 2002]. For Werneck the following point is key:
Before a new game is offered to some publisher the inventor has to arrange intensive test-playing, with several rounds of improvements, modifications, and retesting. Friends and relatives of game inventors can tell long stories how they enjoyed (or better "disjoyed"?!) such testing marathons. The main problem is that testing tends to become dull when lots of variants (most of them only micro mutations of each other) come on the table evening for evening, week for week. At the end testers may have completely lost their fresh minds and no longer be able to realize new facettes.

My idea was to design a computer program which can do a large part of the testing job. This paper is a very concrete follow-up of the rather abstract pamphlet [Alt 2002a].

The paper is organized as follows. Section 2 contains a list of criteria for automatically measuring the interestingness of a game. In Section 3 we describe the multi game program "Morphling", using the little new game "SideKicker" as an example to demonstrate the process of computer-aided evaluation and invention of combinatorial 2-person games. In Section 4, SideKicker's interestingness is analysed with the help of Morphling. The final section contains concluding remarks and open problems. All parts may be read independently of each other.

2 Criteria for Automatic Evaluation of the Interestingness of a Game

Assume that a specific game A is given, and a (multi game) computer program X which allows to play A. The list below contains criteria for measuring the **interestingness of A (modulo X)**. Having in mind the computer-aided invention of games, my main intention was to include only aspects which may be evaluated automatically. A program like Morphling may play long series of automatic games against itself, and afterwards game A (i.e. its rules set) is evaluated by statistic analysis of the records.

*** Average Length of Games**
Players don't like when games are too short or too long. The inventor may enter margins for acceptable average game length and also fix to which extent fluctuations in the game length are acceptable or welcome.

*** Drawing Quota**
Many players don't like games which are too drawish. Nine Men's Morris, Checkers, and International Draughts on the 10x10 board are prominent problem cases in this sense. The inventor may enter intervals of acceptable drawing quota. By the way, drawing quota clearly above zero are not necessarily bad, because several players do not like when there has to be a winner and a loser all the time.

*** Balanced Chances or Advantage for a Certain Player**
In some games a certain player, for instance the one with the first move, has a decisive advantage. The inventor may give a margin M, and only games are accepted where each side wins at most 50+M percent of the games.
In some games there are several ways to gain a win, for instance "win of Player 1 by mate", "win of Player 1 by stalemate", "win of Player 2 by breakthrough", and so on. Given winning conditions W(1), W(2), ..., the inventor may enter a threshold T and demand that amongst all games at most T percent are finished by conditionW(i), for each i.

*** Variability**
People don't like when a computer adversary is completely deterministic, always making the same move in a position. Programs like Morphling or Zillions-of-Games [LM 1998++] allow to adjust a parameter called "variability". Only games with a certain variability may be accepted. This may be tested not so much by executing complete automatic games, but more by doing repeated computer searches in single positions.

*** Performance Loss by Large Variability**
Often large variability has a price in terms of diminished playing strength. The inventor may enter a margin of acceptable performance loss, when playing with large variability versus small variability.

*** Deepening Positivity**
From computer chess practice it is known that searching deeper in the average improves the performance of a program [Hei 2000]. The phenomenon is called deepening positivity. One may accept only games (= rules sets) which show deepening positivity at least to a certain extent. Of course, this criterion depends heavily on the program X.

*** Smoothness of Move Candidates in Iterative Deepening**
Normal game tree search is done in an iterative deepening manner, see description and experiments for instance in [Hei 2000]. It may happen that the move candidates change often and also the evaluations make big swings when going from one search depth to the next. Typically, this indicates that program X does not understand game A well.

In the list we always wrote that a game A would be acceptable if the corresponding parameter would lie within a given interval. Of course, the data may also be used to compare different game variants with

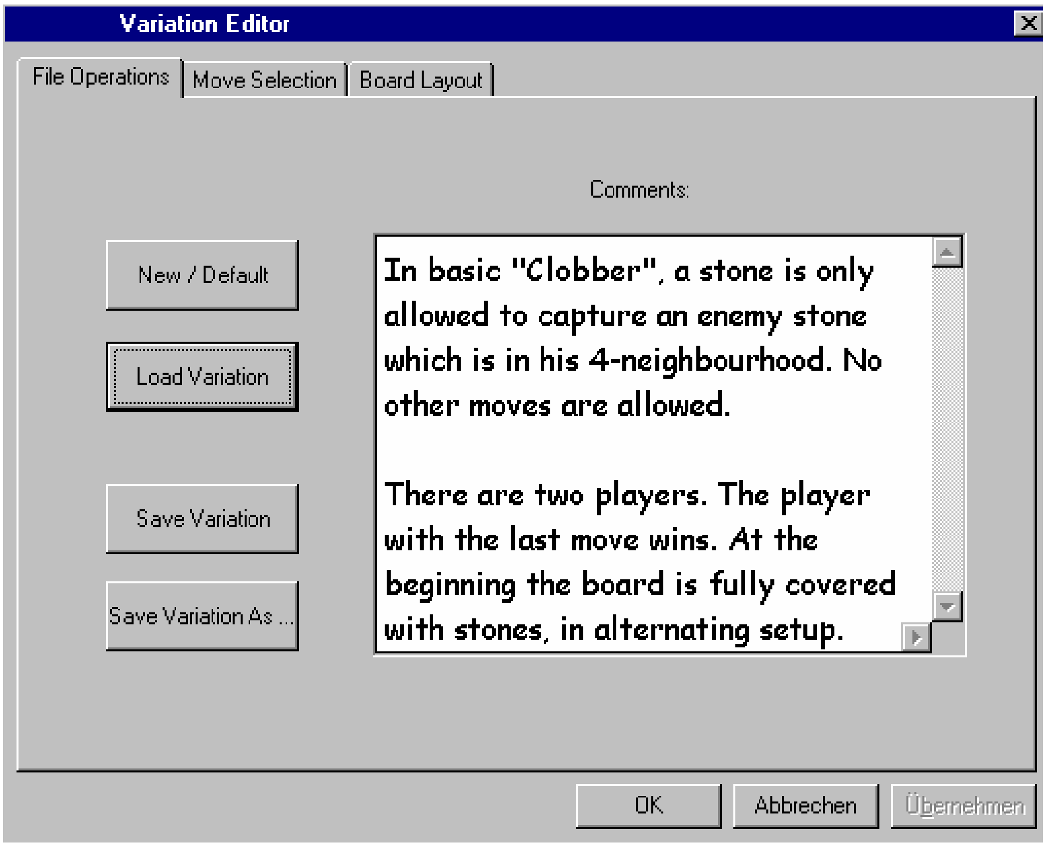
each other, for instance by defining an interestingness function f to be a weighted linear combination of several parameters. As an example, one may define
Interestingness = 0.5*Balancedness + 0.3*Niceness of average game length + 0.2*Variability.

In [ISHUH 2000] other criteria for automatic evaluation and classification of games are given. In [Wer 2002, pp. 54 and 58] criteria for **human** evaluation of games are listed.

3 The Multi Game Program "Morphling" and its Game Class "Clobber"

In his diploma thesis at Jena University the computer science student Thomas Rolle developed the multi game program Morphling [Rol 2003]. Morphling is not as general as Zillions-of-Games [LM 1998++] concerning the set of realizable games. But therefore Morphling has a very simple mechanism to design new game variants. Especially, in contrast to Zillions Morphling also allows intensive automatic testing of new game variants. T. Rolle himself implemented four different game classes for Morphling: **Blobs** (see [Bey 1997++] for the basic game), **Blocks** (as a generalization of Crosscram/Domineering, see [Ika 1999] for a Zillions realisation of the basic game), **Connect-N** (generalisation of Connect-4), and **Move-to-Four** (generalising the small game Mu-TicTacToe by H.D. Ruderman, see in Zillions [LM 1998++]). Peter Stahlhacke and Eiko Bleicher independently of each other implemented a fifth class, "Clobber", which is based on the very simple game Clobber [Alt 2002b].

Here we describe how a variant of Clobber (with name "SideKicker") may be defined within Morphling. In all Clobber variants, each player has only one type of piece, and in addition there may be some neutral walls on the board. (An explanation for non-native English readers: to "clobber" means to "capture".)



In basic "Clobber", a stone is only allowed to capture an enemy stone which is in his 4-neighbourhood. No other moves are allowed.

There are two players. The player with the last move wins. At the beginning the board is fully covered with stones, in alternating setup.

Figure 1: The folder "File Operations" in Morphling's variation editor for Clobber games.

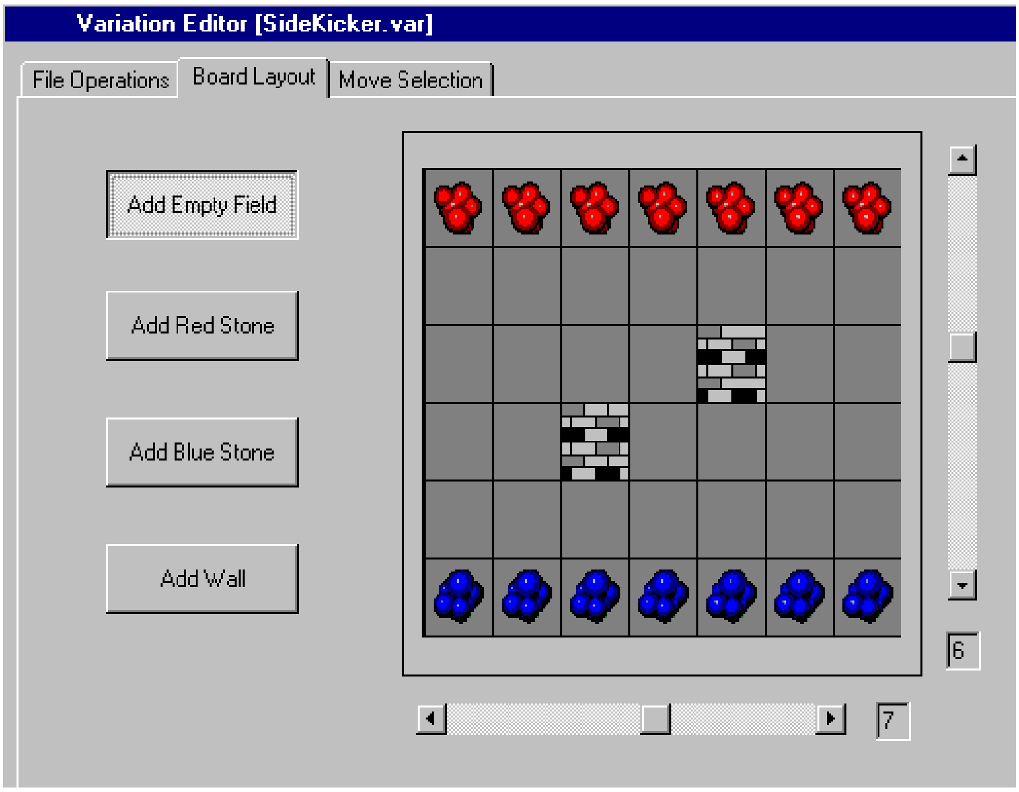


Figure 2: Board layout for a new game. The rectangular board size may vary between 3x3 and 10x10 squares. The figure shows the starting position for SideKicker.

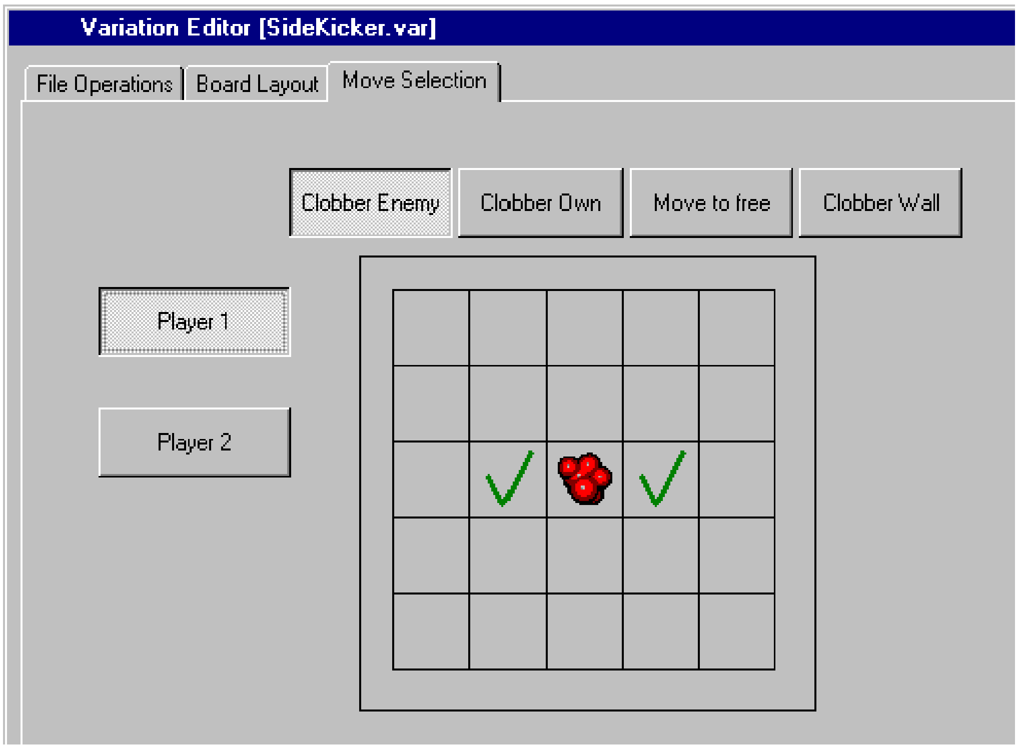
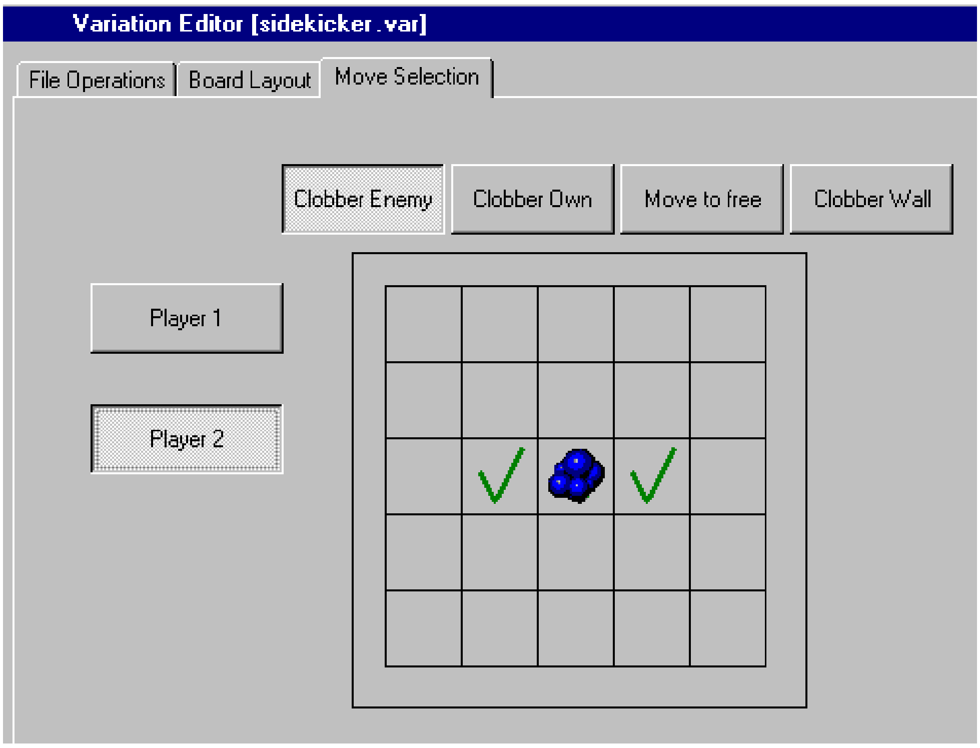
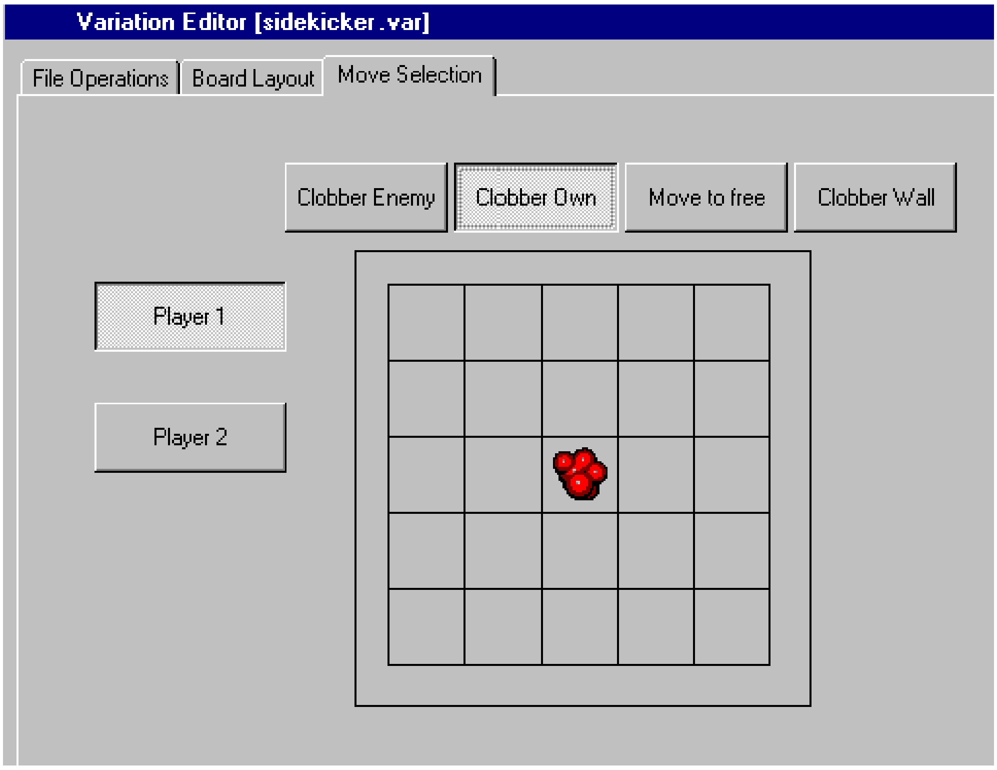


Figure 3a: In this window the user defines, on which squares in the 5x5 neighbourhood a stone of Player 1 (=Red) may clobber an enemy stone.



	✓	●	✓	

Figure 3b: Here the user defines, on which squares in the 5x5 neighbourhood a stone of Player 2 (=Blue) may clobber an enemy stone.



		●		

Figure 4: Here the user defines, on which squares in the 5x5 neighbourhood a stone of Player 1 may clobber an own stone. In "SideKicker" the clobbering of own stones is for both players not allowed. Therefore the corresponding diagram for Player 2, which is also empty, is omitted here.

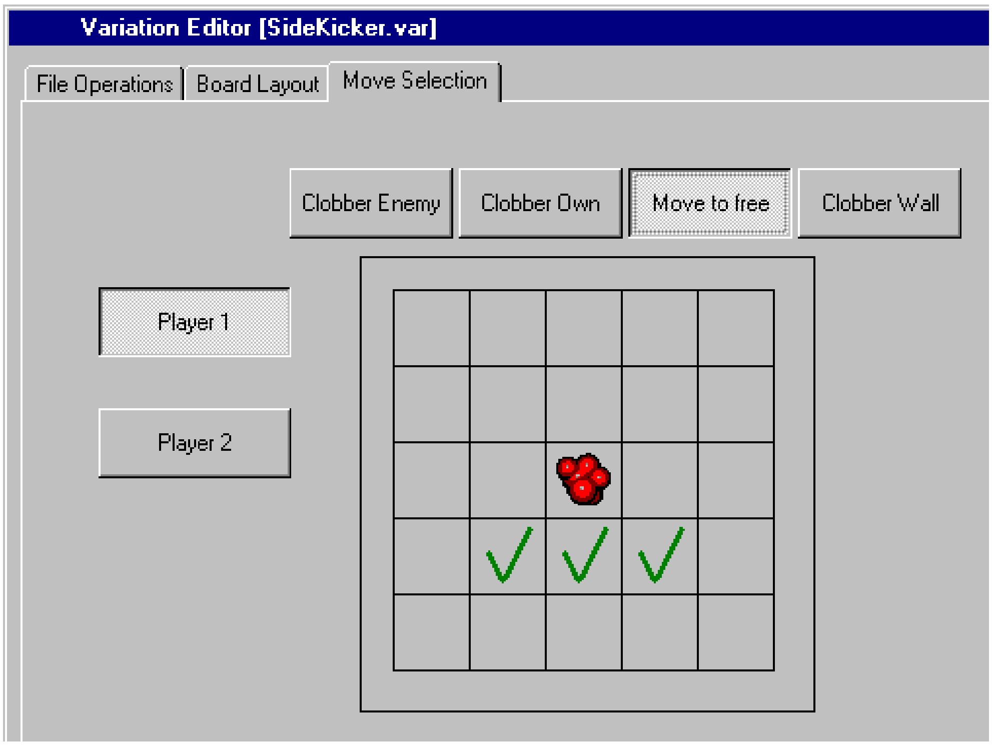


Figure 5a: Here the user defines, to which free squares in his 5x5 neighbourhood a stone of Player 1 may jump. Hence, in SideKicker Red may jump one square to the south or the south-west or the south-east.

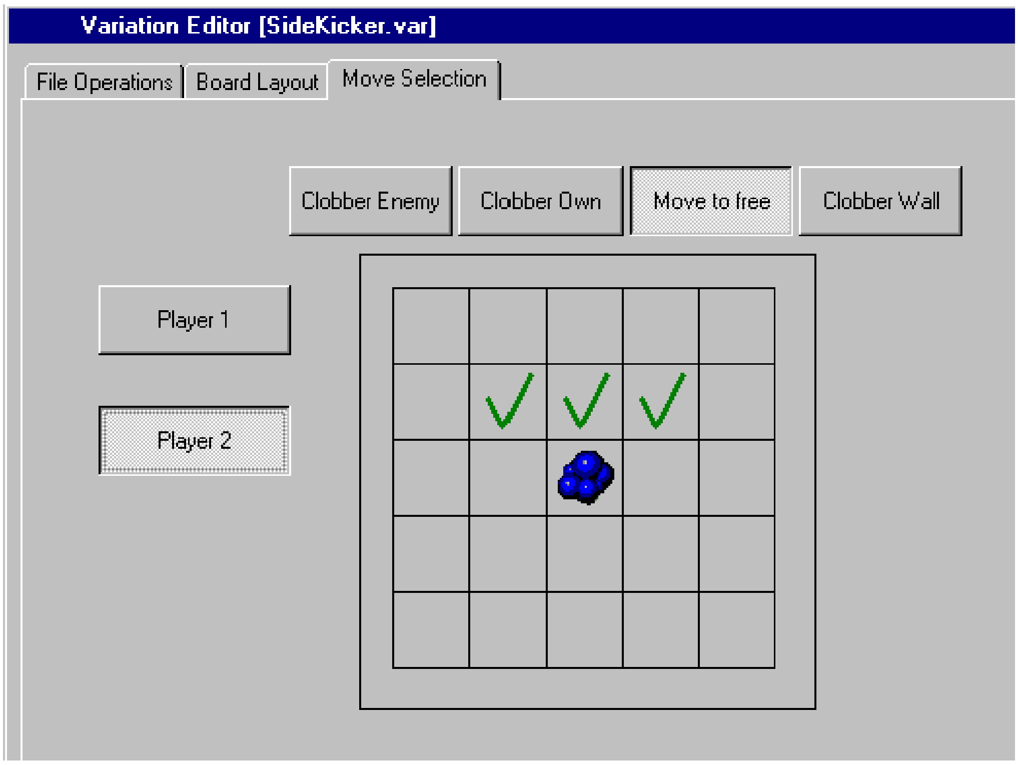


Figure 5b: Here the user defines, to which free squares in his 5x5 neighbourhood a stone of Player 2 may jump.

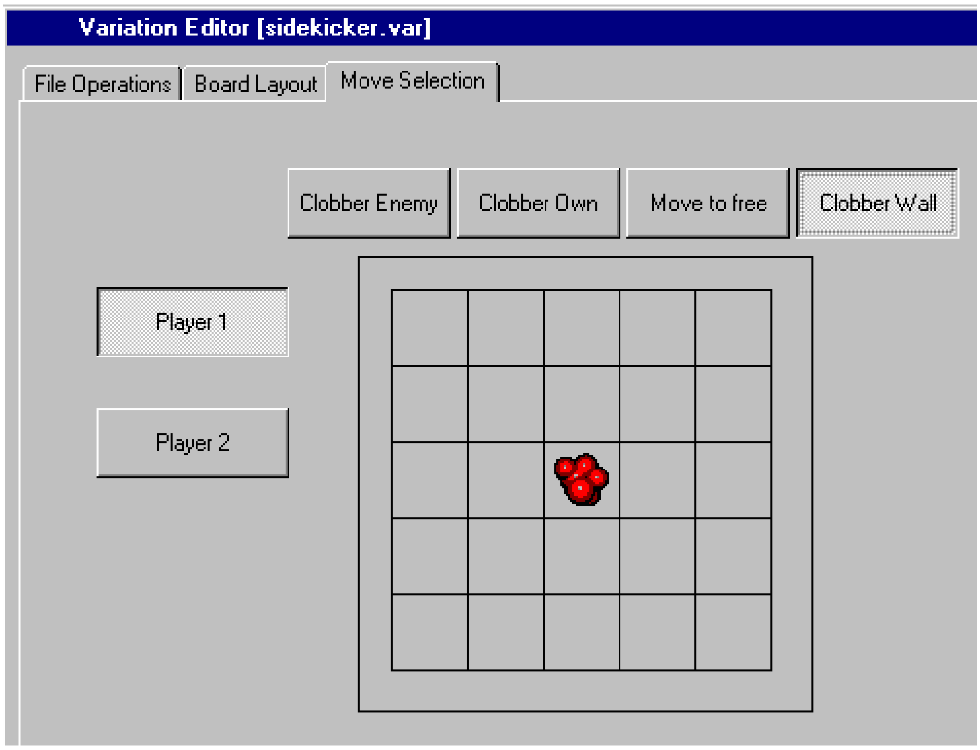
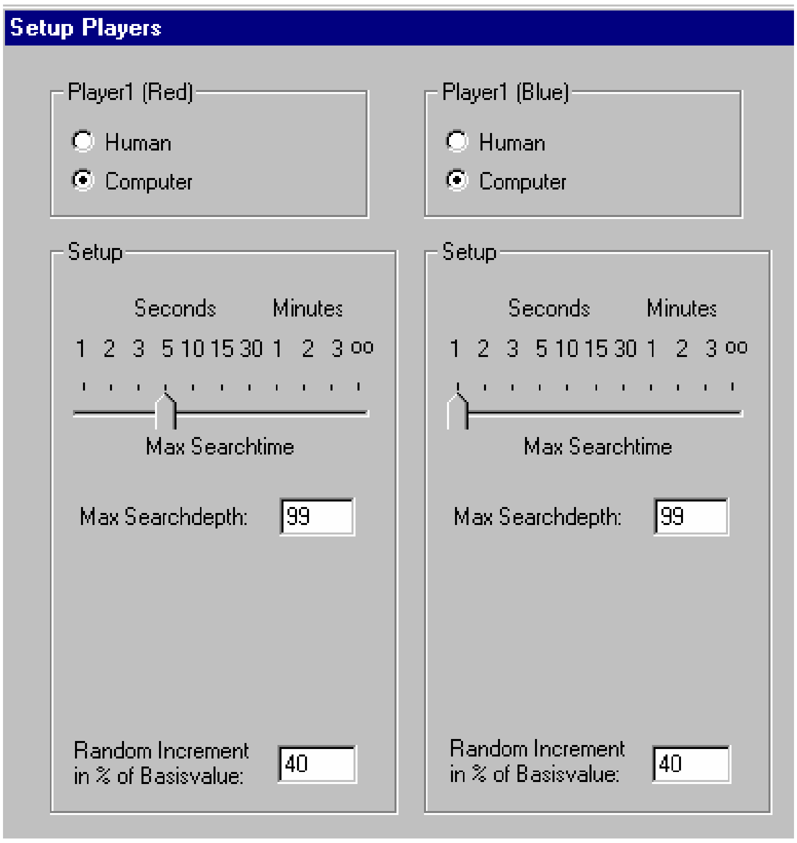


Figure 6: Here the user defines which walls a stone of Player 1 may clobber in his 5x5 neighbourhood. In SideKicker clobbering of walls is not at all allowed, therefore no hooks. The corresponding diagram for Player 2 looks analogously.

When all the rules of a new Clobber variant are fixed, the user may save the rules file in the "File Operations" window, see Figure 1. Looking only at the definition of moves, the Figures 3a to 6 show that there are ((2 to the power 24) to the power 8) = 2 to the power 192. This number is larger than leading 1 with 50 zeros. SideKicker is only one of these many possibilities!

The evaluation function for Clobber and its variants is very simple in Morphling: it only counts the numbers of red and blue pieces and determines the difference. In addition, random increments to the evaluations do not only lead to variable play, but may also help to identify positions with higher mobility for the own pieces. The (often positive) influence of random increments on the quality of evaluation functions has been experimentally investigated: in [Rol 03] for 2-person games and in [Heu 2003] for single agent search.

Having defined a (new) game, Morphling allows to set up the characteristics of the players for sparring matches. Observe that the settings do not have to be identical for the two sides.



Setup Players

Player1 (Red)

○ Human

● Computer

Player1 (Blue)

○ Human

● Computer

Setup

Max Searchdepth: 99

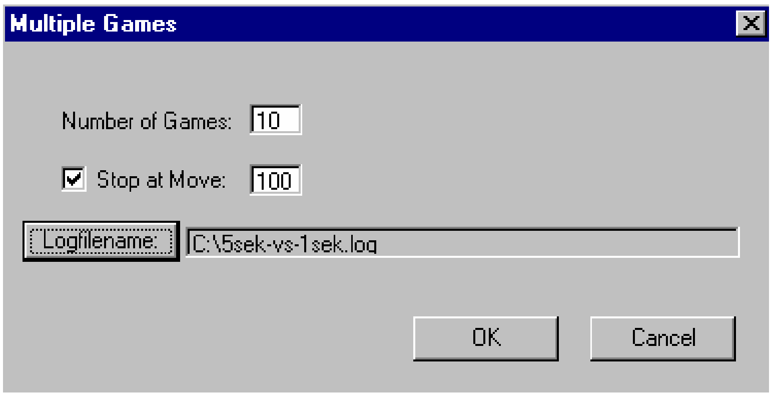
Random Increment in % of Basisvalue: 40

Setup

Max Searchdepth: 99

Random Increment in % of Basisvalue: 40

Figure 7: In this example both sides shall be played by program Morphling. Red will compute 5 seconds for each move, and Blue only 1 second per move. The maximum search depth is set to (unreachable) 99. So the search will never stop by this criterion but only by the time limits. "Random Increment" is the parameter for the variability of the program. 0 % would mean completely deterministic play up to hardware-based discrepancies, and 40 % (like above) means a good portion of random influence.



Multiple Games

Number of Games: 10

☑ Stop at Move: 100

Logfilename: C:\5sek-vs-1sek.log

Figure 8: With the settings from above (Figure 7) an autoplay session may be run and recorded. In this example 10 games shall be played, and the data will be saved to a logfile. The feature "Stop at move xyz" is meant for games which may not necessarily terminate. It is possible to define such rule sets, for instance within the class Clobber. In SideKicker infinite games can not happen.

We conclude the section by an explanation of SideKicker in normal text mode:
SideKicker is a board game for two players, called Red and Blue. In its basic form SideKicker is played on a rectangular grid of size 7 times 6, with two permanent walls on the squares c3 and e4. The players move in turn, with Red to start. The last player to move is the winner. Draws are not possible.
Each team has seven stones. In the beginning the red stones are placed on the northern back rank a6, b6, ..., g6, and the blue ones on the southern back rank a1 to g1. In his turn, a player has to move forward one step with one of his stones. "Forward" means in direction south or south-west or south-east for player Red, and in direction north or north-west or north-east for player Blue. Or a player may capture an enemy stone by sidekicking to the west or east with one of his stones. By these rules a player can make at most 7*5 forward moves and 7 sidekicks (one for each opponent stone), hence altogether at most 42 moves. So, the game is definitely finite.

4 Evaluating the Example Game "SideKicker"

Several Morphling autoplay series on a PC with 900 MHz AMD-Athlon processor were performed for the game SideKicker. The results were as follows.

Series 1
Red: 1 second per move, randomization switched on.
Blue: 1 second per move, randomization switched on.
Total result "Red - Blue" after 20 games: 11 - 9. So, no side seems to have a clear advantage.

Series 2a
Red: 1 second per move, randomization switched off.
Blue: 1 second per move, randomization switched on.
Total result "Red - Blue" after 20 games: 6 - 14.
Series 2b
Red: 1 second per move, randomization switched on.
Blue: 1 second per move, randomization switched off.
Total result "Red - Blue" after 20 games: 13 - 7.

Series 2a and 2b together result in a 27-13 win for the side with randomization. So, randomization seems not to be a disadvantage for Morphling-SideKicker, but to help instead.

Series 3a
Red: 5 second per move, randomization switched on.
Blue: 1 second per move, randomization switched on.
Total result "Red - Blue" after 10 games: 7 - 3.
Series 3b
Red: 1 second per move, randomization switched on.
Blue: 5 second per move, randomization switched on.
Total result "Red - Blue" after 10 games: 4-6.

Series 3a and 3b together result in a 13-7 win for the side with 5 seconds per move. So, more computing/thinking time seems to help.

Average Game Lengths

Altogether 80 games were played. The average number of moves per game was 42.5 . 41 games were won by Red (= player to move first), with an average move number of 41.6 . The other 39 games were won by Blue, with an average move number of 43.4 .

Except for the very first duel, all matches were "strong vs. weak" in one or the other way - either long time vs. short time, or high variability vs. low variability. In the 40 games with a win for the "strong side" the average move length was 41.6 . In the remaining 20 games the weak side won, and the average move length was 45.5 . My interpretation: in the average the stronger side was able to gain quicker wins and to set up stiffer resistance when losing.

Variance of Game Lengths

The shortest one of the 80 games took 29, the longest one 55 moves. But these were single outliers. Almost all games had between 35 and 50 moves. (Observe that single moves and not move pairs were counted.) For a little game of entertainment, and SideKicker is meant to be one, this means a pleasant length. Setting about 15 seconds for each move in play "human vs. human" the total game length will be something between 7 and 15 minutes with very high probability.

The Other Criteria

In SideKicker there are no draws, so the "drawing percentage" criterion did not apply. The variability of computer play was high (enough): there were no identical games in the total sample of 80. The smoothness of move candidates in iterative deepening was not tested, because Morphling does not collect these data in its autoplay mode.

What the Computer did not find out ...

Smart readers will have realized already during Section 3 that the second player can secure a win by a simple mirror strategy. Namely, Blue may easily copy each move of Red by turning it by 180 degree around the center of the board. The starting position is mirror-symmetric. Therefore the mirror move is always feasible. As stalemate is the only possible end of a game Red cannot avoid a loss. Morphling did of course not understand this meta-principle, but SideKicker's human inventors saw it.

How to deal with this little catastrophe? SideKicker was made public under Zillions-of-Games [AB 2003] in three variants: the first one is the basic game to show that the computer is a strong adversary with Blue even without understanding the mirror strategy. In the second variant the starting position (with seven versus seven stones) is slightly randomized by putting one of the stones on a random square on the second rank. Several test games "human versus computer" showed that in this variant computers are much too strong for normal human brains. So in the third variant the computer has to give handicap, playing with six stones against seven in a slightly randomized setup.

5 Concluding Remarks

* Besides SideKicker, in the reference list of this paper eight more games are mentioned which were designed and tuned with the help of Morphling: Moray Eels [Alt 2003a], Redcappies and the Wolves [Alt 2003c], Cannibal Clobber [Alt 2003d], Jeremy's Nightmare [Alt 2003e], Permission Denied [Alt 2003f], Wall-Eaters [Alt 2003g], Roll-Ing to Four [AR 2003], and Kick and Run [AS 2003].

* Automatic checking of a game's interestingness may become much more reliable when done with several (multi game) programs, including matches between the different programs. We did this (by hand simulation) in the chess variant Stroebeck Crisis Chess [Alt 2003b], using the multi game program

Zillions of Games and a quickly adapted version of Stefan Meyer-Kahlen's strong commercial chess program Shredder.

* Besides Morphling, there are other multi game programs like Awale [GG 1996++] for Kalaha type games and the often-mentioned Zillions of Games for almost arbitrary board games. Their disadvantage with respect to game evaluation is that Awale has none and Zillions only very limited autoplay facilities.

* As described already in [Alt 2002a], large sets of game variants may be defined by product spaces of micro mutations (mutation 1 yes/no, mutation 2 yes/no, ...). Local search in such a space based on automatic game evaluation allows to invent "locally optimal game variants" in fully automatic mode. A warning for game inventors, especially for those with commercial interests: Be very careful not to publish too many similar games! In the market related games may start cannibalizing each other.

* Thomas Rolle's program Morphling should be extended by including more game classes for 2-person games, a module for (1-person) puzzles, and modules for games with three or more players.

* Other criteria for the automatic evaluation of the interestingness of games are welcome. The list in Section 2 is meant only as a starting point.

* Open problem: Is there a way to find out automatically whether the rules set of a game allows successful mirror strategies for one of the players?

* It is not enough to invent a game by defining a rules set. A new game also needs nice screen design and an attractive title. Maybe, graphics and names can also be found in computer-aided mode, for instance by a generative design approach.

Having collected some experience as a multiple game inventor I conclude with a provocative claim: **Computers are only number crunchers. They are unsuited to invent completely new nice games in fully automatic mode. Game inventing will always remain a job for creative humans!**

Acknowledgements

Student Thomas Rolle put a lot of work in the program Morphling. Independently of each other, Peter Stahlhacke and Eiko Bleicher realized the game class "Clobber" under Morphling. Jeroen Donkers informed me about the multi game program "Awale" by the Guillion brothers. Multi game inventor Karl Scherer participated in some internet discussions on computer-aided game inventing, and Peter Stahlhacke (again) in long discussions on lots of variants of the classic game "Nine Men's Morris". Stefan Meyer-Kahlen provided a special version of his strong commercial chess program Shredder for "Stroebeck Crisis Chess". The attorney Julia Bezzenberger, representing her client Nina Hagen, gave me an unasked and expensive lesson on legal matters concerning game invention. One of the results was [Alt 2003f]. Max Kobbert brought the expression "cannibalization of related games" to my attention. Thanks to all of them!

Thanks to the Zillions company with Mark Lefler, Jeff Mallett, and Ed van Zon for publishing all the new games on their website.

References

[Alt 2002a] I. Althofer. Inventing game variants with computer help. Proceedings “Workshop on Computer Games within the Seventh Computer Olymiad” (Editor J.W.H.M. Uiterwijk), University of Maastricht, Report CS 02-03 (June 2002), 1-10.

[Alt 2002b] I. Althofer. Clobber - a new game with very simple rules. ICGA Journal 25 (2002), 123-125.

[Alt 2003a] I. Althofer. 2-Person-Game **Moray Eels.** Online description at http://www.zillionsofgames.com/games/MorayEels.html, May 2003.

[Alt 2003b] I. Althofer. Chess variant **Stroebeck Crisis Chess.**
Online description at http://www.zillionsofgames.com/games/StroebeckCrisisChess.html
German article with title "Stroebecker Krisenschach" in Schachmagazin-64 (July 2003), 357.

[Alt 2003c] I. Althofer. 2-Person-Game **Redcappies and the Wolves.** Online description at http://www.zillionsofgames.com/games/Redcap-and-Wolf.html, August 2003.

[Alt 2003d] I. Althofer. 2-Person-Game **Cannibal Clobber.** Online description at http://www.zillionsofgames.com/games/CannibalClobber.html, August 2003.

[Alt 2003e] I. Althofer. 2-Person-Game **Jeremy's Nightmare.** Online description at http://www.zillionsofgames.com/games/JeremysNightmare.html, August 2003.

[Alt 2003f] I. Althofer. 2-Person-Game **Permission Denied.** Online description at http://www.zillionsofgames.com/games/PermissionDenied.html, July and August 2003.

[Alt 2003g] I. Althofer. 2-Person-Game **Wall-Eaters.** Online description at http://www.zillionsofgames.com/games/Wall-Eaters.html, September 2003.

[AB 2003] I. Althofer and E. Bleicher. 2-Person-Game **SideKicker.** Online description at http://www.zillionsofgames.com/games/SideKicker.html, August 2003.

[AR 2003] I. Althofer and T. Rolle. 2-Person-Game **Roll-Ing to Four.** Online description at http://www.zillionsofgames.com/games/Rolling24.html, June 2003.

[AS 2003] I. Althofer and P. Stahlhacke. 2-Person-Game **Kick and Run.** Online description at http://www.zillionsofgames.com/games/Kick-and-Run.html, June 2003.

[Bey 1997++] A. Beyrand. Website on Ataxx and Blobs. http://www.pressibus.org/ataxx/indexgb.html, since 1997 with occasional updates.

[GG 1996++] D. Guillion and O. Guillion. Computer game program Awale (current version 3.4 in August 2003). http://www.myriad-online.com/enindex.htm

[Hei 2000] E.A. Heinz. Scalable Search in Computer Chess. Vieweg, Braunschweig, 2000.

[Heu 2003] S. Heuser. Randomized evaluation functions in single agent search. Diploma Thesis, Friedrich-Schiller-University Jena, Faculty of Mathematics and Computer Science, August 2003. Pdf-file available from althofer @ minet.uni-jena.de.

[Ika 1999] J.-P. Ikaheimonen. Zillions-program for Crosscram/Domineering. http://www.zillionsofgames.com/games/crosscram.html, 1999.

[ISHUH 2000] H. Iida, N. Sasaki, T. Hashimoto, J.W.H.M. Uiterwijk, and H.J. van den Herik. Towards a Classification of Games using Computer Analyses. Proceedings "Board Games in Academia III - an interdisciplinary approach" (Ed. N. Neuwahl), 2000, pp. 139-152.

[LM 1998++] M. Lefler and J. Mallett. Multi game program **Zillions of Games**. http://www.zillions-of-games.com; since 1998, with weekly updates.

[Rol 2003] T. Rolle. Development of a Multi-Game Engine. Diploma Thesis, Friedrich-Schiller-University Jena, Faculty of Mathematics and Computer Science, May 2003. Pdf-file available from althofer @ minet.uni-jena.de.

[Wer 2002] T. Werneck. Leitfaden für Spieleerfinder und solche, die es werden wollen (paperback, in German). Ravensburger Spieleverlag, Ravensburg, fifth edition, 2002.