Malloc();

Team Brain Surgery Production

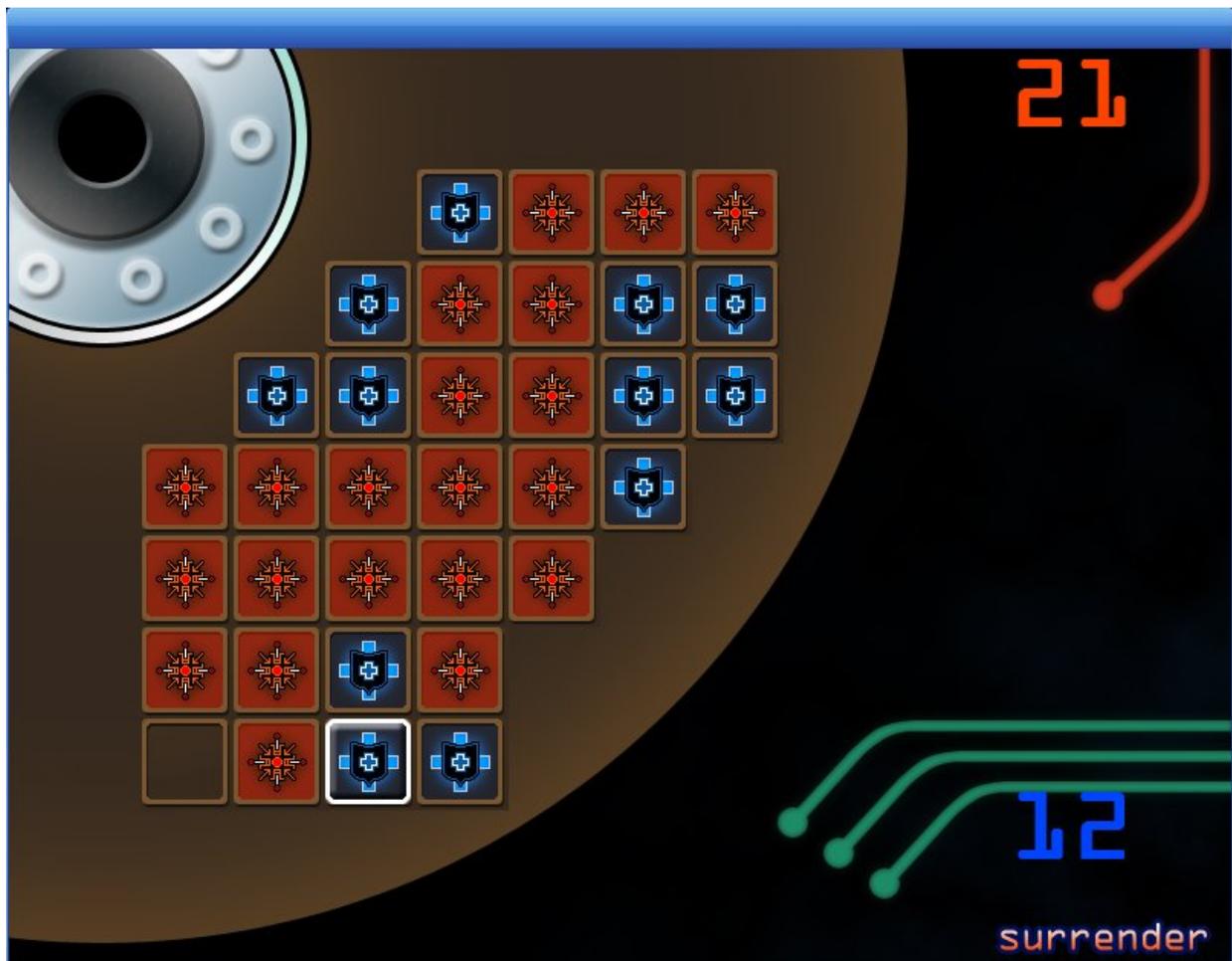Matvei Stefarov <mstefaro@ucsc.edu>

Geon Lee <ghlee@ucsc.edu>

What is this game about?

 > *Malloc();* is an original board/puzzle game set in the world of computer hardware. In the primary storyline, player competes against the increasingly difficult AI on six boards. At the beginning of a new game, only two boards are open to the player, but more can be unlocked by winning the open levels.

 > The story is told mainly through the use of 3D rendered cutscenes. A computer in which the game is set becomes infected with a virus, and an antivirus is being deployed by the System Kernel to delete the malicious program. You take the role of Malloc, a young and inexperienced antivirus program, who is the last hope of the system. Each board represents an important component of a computer, such as RAM/HDD/etc., and winning against the AI signifies deleting the virus from that system component. The ultimate goal of the player is to remove the virus from the system completely by beating all six levels. After CPU level (the last one) is completed, another cutscene shows how last of the virus units are driven out. There is also a hidden randomly-generated bonus level that will be revealed to the player when storyline is completed.

 > Malloc multiplayer mode has the same core gameplay, but allows up to 4 people to play on the same board via LAN. There are many made maps a;ready included with the installation, and player can load custom maps from files (SDK is in development right now).

Where to start?

> Malloc(); consists of two parts: single-player storyline and multiplayer mode. There is also an interactive Tutorial accessible from Single Player menu.

> To begin the single-player campaign, click "Single Player," then click "New," and choose a difficulty level. Intro movie will play, and you will be presented with the main screen. Computer components that are highlighted blue are playable; you can unlock access to higher, more difficult levels by winning on lower levels.
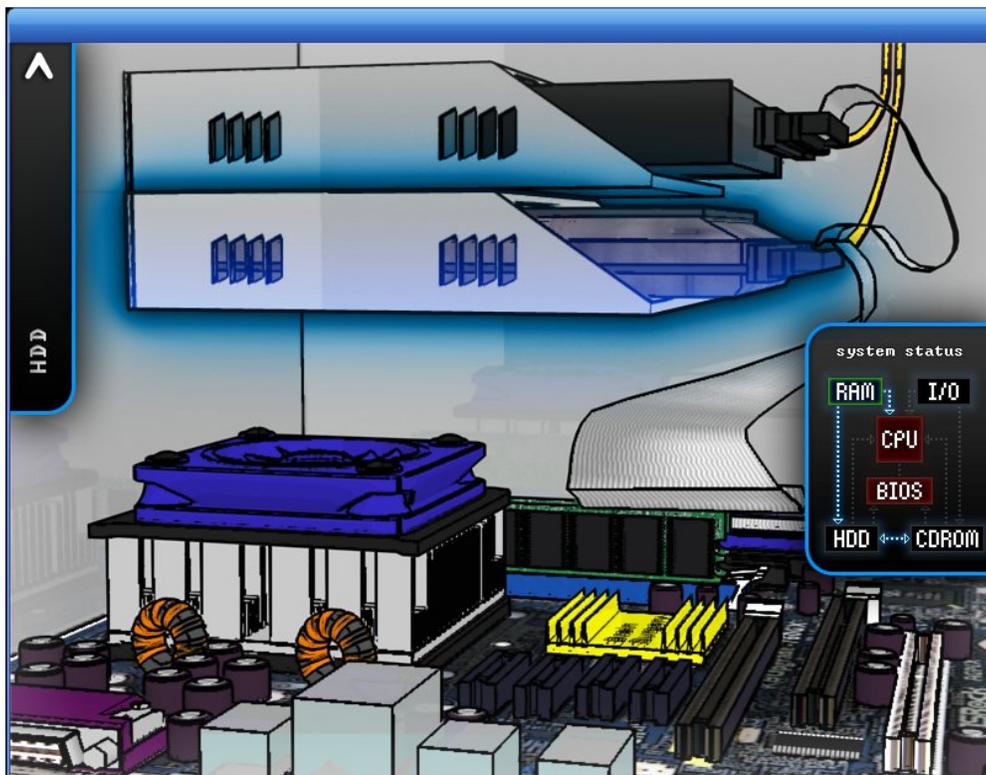
> Every time you beat a level, your progress will be saved. You can load a saved game by choosing "Load" from the single-player menu. *Malloc();* also keeps track of your high- and low-scores for each level.

> In order to participate in multiplayer games, you need to be on the same LAN (Local Area Network) as the other player(s). Up to 4 players can play against each other on the same server at the same time.

> To create a new server, click "Multi Player," enter your name (if prompted), and click "Create." A server is set up automatically, but you might need to add malloc.exe to your Windows Firewall exceptions list. Once the server is up, choose the map and map size you want, and wait for players to join.

> To join an existing game, click "Multi Player," then click "Join." The game will automatically search for any online servers within your network. If one or more servers are found, you will be presented with a list. Enter the number of the server that you want to join and press <Enter>. You're in!

> To exit the game in progress, whether in single- or multiplayer mode, simply press <Escape>.

How to play?

 > The board is a grid of "cells." Each cell can either be empty or occupied; color of an occupied cell indicated its affiliation. There can also be gaps in the grid.

 > Players take turns, moving one cell per turn. There are two types of moves: a "clone" and a "jump" move.

 > To make a clone move, click on a cell that belongs to you to select it; now click on an empty cell that is immediately adjacent; the target cell is now occupied by a copy of your original cell.

 > To make a jump move, click on a cell that belongs to you to select it; now click on an empty cell that is one cell away (vertically, diagonally, or horizontally). Target cell now belongs to you; but the original cell is now empty.

 > All enemy cells that touch the target cell of your move are instantly converted into your own cells.

 > Your score is the total number of cells that belong to you on the board. Player with the highest score in the end of the game wins.

 > The game ends when (1) there are no more empty cells left, (2) a player captured all of his opponents' cells, or (3) all but one player are cornered and have no possible moves left. The game calculated the possible number of moves automatically, so if the game is not over yet, it means that you still have at least one move somewhere on the board.

How do I...

> Toggle fullscreen mode? Press <F4>

> Mute / unmute the music? Press <Ctrl>+<M>. Press again at any time to unmute.

> Make my own maps for malloc? If you created a Start menu folder, go to *Start > Malloc > Map Editor*. Otherwise, go to the folder where Malloc is installed and start *meditor.exe*

> Get the soundtrack in MP3? Simply go to the folder where you installed Malloc (usually C:\Program Files\Brain Surgery\Malloc). MP3s of all music from the game can be found in the "music" subfolder.

> Interpret the game's title? There is a function named "malloc" in the "C" programming language. The function stands for Memory ALLOCation. The gameplay of this game resembles the fight between several programs to allocate the same sector of computer memory; hence the name "malloc."

> See what the AI is thinking? In single player mode, press <D> to toggle AI debug mode. Little triangles over empty cells show probability of AI making a move to that particular location. The white arrow shows the last move executed. White diagonal lines over a cell indicate that it has been converted from blue to red on the last move. Press <D> again several times to return to normal mode.

> Find out what's new in version 1.1? Go to http://www.fragmer.net/malloc/changelog.html

It's YOUR turn to move.   WootServer

f2k (blue) was randomly chosen
to go first

fragmer: it's pretty even so far

comradeb14ck: what are you
talking about I am pwning your
a$$

f2k: lol

>|

Who made this game?

> Team Brain Surgery

> **Matvei Stefarov** <www.fragmer.net>
project lead // all programming // backgrounds // sprites

> **Geon Lee** <ghlee@ucsc.edu>
creative lead // sprites // 3D art and animation // installer

> Contributors

> Sean Peterson Schnell <seansps@kneehighs.com>
music [main theme, tutorial, all levels except CDROM] // testing

> Andrew Cichowski
music [intro, ending, CDROM level] // testing

> Karl Schmidt
music [guitar, intro] // testing

> Randall Degges II
AI tweaking // testing

> Chris Leary (aka Ochre) <www.ochremusic.com>
music [victory theme]