3D Go: Traditional Board Game in Three-Dimension

Yun Miao
Graphics Lab, School of EECS
Oregon State University

August 30 2010

1 Introduction

3D Go envisions a new way to play the board game Go online. Players are provided with a three dimensional surface as their game board to replace the traditional two dimensional ones.
The game is a Java Applet running in a web browser, and employs client-server model to enable multiplayer functionality. In addition to the features provided by the existing online go games [1], 3D Go will give users the freedom of viewing the board from all angles and distances.
Go is a game of territory. By changing the geometric property of the board, we are really changing the terrain of a battlefield, making warfare more interesting and strategy more sophisticated. In doing so, we are hoping not only to provide more exciting games for Go players, but also seeking intriguing strategies and their correspondence to the mathematics of Go.

2 Game Rules

In this section, we will define the terms in 3D Go, and introduce the game rules, which are not greatly different from the traditional Go game.

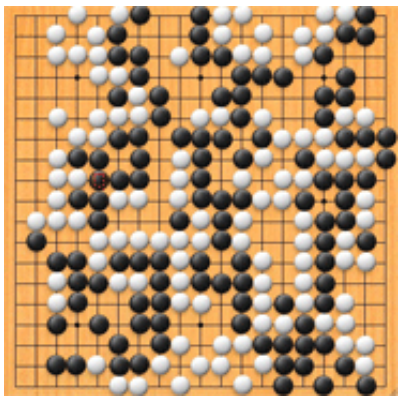


Figure 1: Traditional Go board

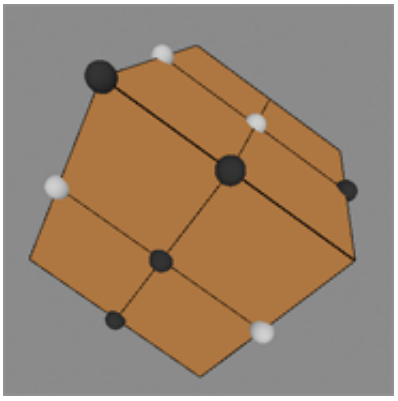


Figure 2: 3D Go board

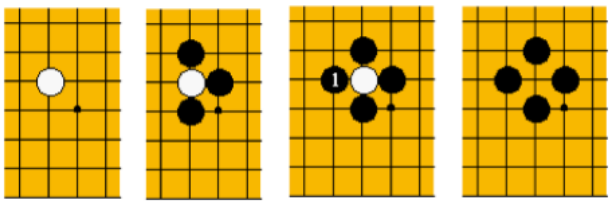
Game Board The fundamental difference between 3D Go and 2D Go is the game board. Traditional Go game uses a 19×19 grid (see Figure 1). 3D Go uses a 3D mesh as its board (see Figure 2). Since we can rarely find a mesh with exactly 361 vertices, 3D Go will simply use small meshes with number of vertices comparable to 361. Rather different from Chess, Go stones are placed at intersections instead of the inside of a grid. In this report, an intersection is called a vertex. The line connecting two adjacent vertices are called an**edge**.

Stones Each Go game, including 3D Go, involves two players. One holds black stones. The other holds white stones. The Black always plays first in Go.The players use the stones to claim territory, initiate attacks, and establish defenses.

Liberty A liberty is an unoccupied neighbor of a group. A group consists of one or more same colored stones. In Go, liberty accounts the "breathing space" or "life" of a group. A group is alive when it has one or more liberties. In traditional Go, a single stone can have either 4, 3, or 2 liberties. In 3D Go, a single stone cannot have fewer than 3 liberties due to the fact that a 3D mesh should not be flat.

Atari When a stone or a group of stones only has one liberty left, it is said to be in Atari (see Figure 3(b)).

Capture When a group of stones has no liberty, in other words, all neighbors are occupied by opponent stones, the members are captured and the group dies. Just like what would happen in a war, a group of troops would be captured or killed if they are surrounded by their enemies. [2]



(a) A stone with 4 liberties (b) Atari (c) Capture (d) Result

Figure 3: The concept of liberty, atari, capture in 2D Go

Illegal Moves

1. Superposition is the act of placing a stone over another. Only one stone is allowed per vertex.
2. Suicide. If the placement of a stone leads to its group's liberty reducing to zero, then the placement is forbidden (see Figure 4).
3. Ko (Eternity). A move is prohibited if it produces the same previous board position [4] (see Figure 5).

Exception to Illegal Moves A suicide move which will not result in the same board as the previous is the exception to the rule is considered legal(see Figure 6 and Figure 7).

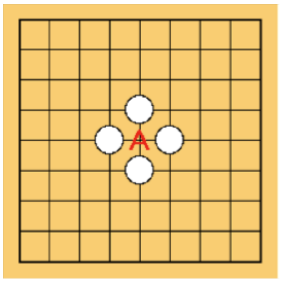


Figure 4: Black stone cannot place itself in A because there is no liberty for it.

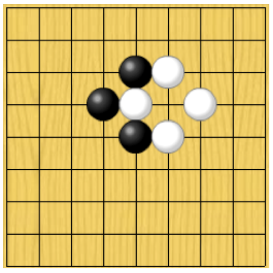


Figure 5: A white stone is in Atari. One can place a black stone to capture the white. However, this will put the new black stone in Atari. Black and white can keep capturing each other forever this way.This kind of eternal loop is forbidden. Yet the white can place stone somewhere else, then go back and sack the black stone.

3 Game Implementation

The Interface The game interface is built with Processing, an open source programming language and environment for sketching images and animations. And the game back end is supported by Java. Enabling picking by using an invisible buffer layer is an exciting progress when I was designing the interface. In the buffer layer, all vertices are colored by a gray scale color based on its index (index+1). During the game, the buffer layer would process the color picked and from the color, thus identify the index of the picked object [3].

The Board The implementation of the game board is inspired by a C program that constructs 3D mesh from PLY format file. The program treats each vertex, edge, and face in the 3D mesh as an object. And it stores each type of objects in a corresponding list. During initialization, the program constructs a large data structure called polyhedron storing the vertex list, the edge list and the face list, with pointers between the elements. For example, each vertex objects stores pointers pointing to the adjacent polygons/faces and adjacent edges. This logic transforms into the board creation seamlessly. Afterall, the board is a 3D mesh of a surface.

The Game States Go is a game of states. Whenever a stone is added to the board, the state of the game changes. To go from one state to another, two elements are taken in to consideration: the territory of oneself and the territory of the enemy. Territories are formed by same-colored stones and have liberties as their breathing space. To keep track of the territories on the board, there is a structure called Group. A Group always has a color. It stores the indices of adjacent stones of that color into a hash set, thus treats a region of stones as a whole. The reason for using hash set is not only due to its constant access time

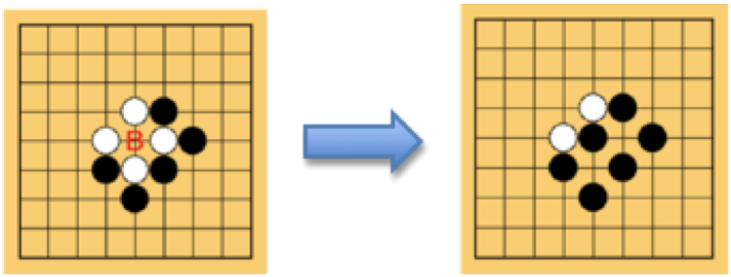


Figure 6: In this case, two white stones are captured by placing black at B. It is impossible to go back to the same board with just another whites move. So the blacks suicidal move is legal.

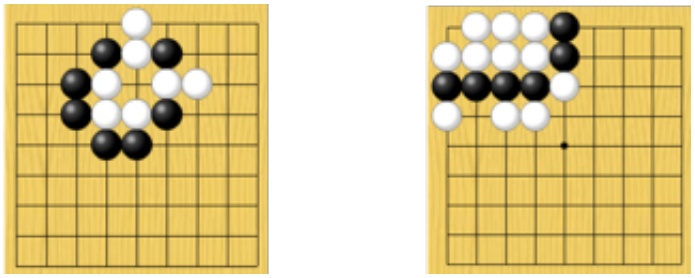


Figure 7: In both cases, if one can place a black stone in the suicidal position, a group of white stones will be captured. So the next move by white will never revert to the previous board.

for most element, but also the fact that no duplicates are allowed in a Group, since a stone cannot be counted as a Group member twice. In addition, Group keeps track of the liberties of a territory: another hash set of all the neighboring empty vertex indices of that group. A single stone can form a group. But more likely, more same colored stones will be added to the single stone's neighborhood to enlarge the group. Using hash set allows the group to grow flexibly with a small cost.
A game usually involves more than one regions of stones. So the program have another important list to keep track of all the groups, namely GroupList. GroupList stores all the groups. Each game would have two GroupList, one for Black, one for white. whenever a move happens, the program updates the GroupList to eliminate dead Groups. Speaking of the moves, there is another list that stores each move by its vertex index for future regeneration of the Game board.
To summarize, Figure 8 at the end of this report is an illustration of the data structures used by the game and their relations.

The Back-end The game logistic is the most difficult part of the project, because it involves building a client-server application, which is left undone at this stage. The design is lacking due to my lack of knowledge of the client-server applications. So I am leaving this part as the future work of the project.

4 Result and Future Work

I have designed and implemented the game interface on a small 3×3×3 cube (see Figure 2). The program can be narrated by two diagrams: data structures (see Figure 8) and the flow diagram(see Figure 9). The game now can run in a Firefox browser smoothly, and the

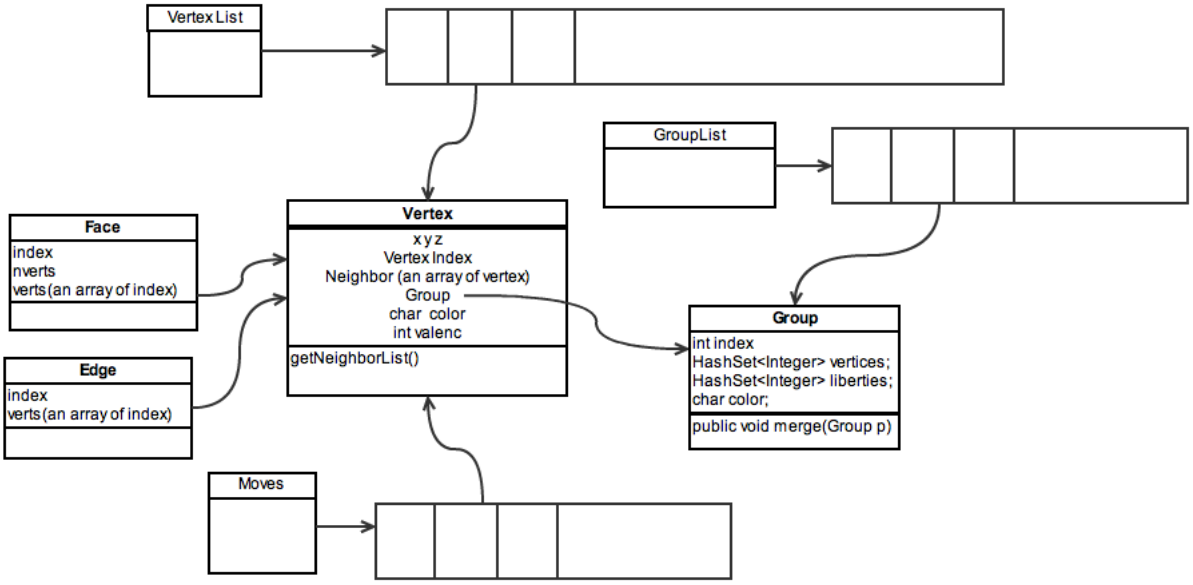


Figure 8: The data structures used in the game.

user can place the stone with no problem. Yet bugs do exist. And as previously mentioned, the back-end of the game is a vast blank. The design of the data structures is subject to change. Reconstructing the game relies on the move list and therefore cannot be done fast. In conclusion, I have started the project and give it a general direction. But The heavy-lifting job is yet to be finished.

References

[1] Go, an online board game. `http://uk.games.yahoo.com/online-games/board/games_go.html`, October 2010.

[2] How To Play Go. `http://www.kiseido.com/ff.htm`, August 2010.

[3] Picking With a Colored Buffer. `http://wiki.processing.org/w/Picking_with_a_color_buffer`, October 2010.

[4] The Interactive Way To Go. `http://playgo.to/iwtg/en/`, August 2010.

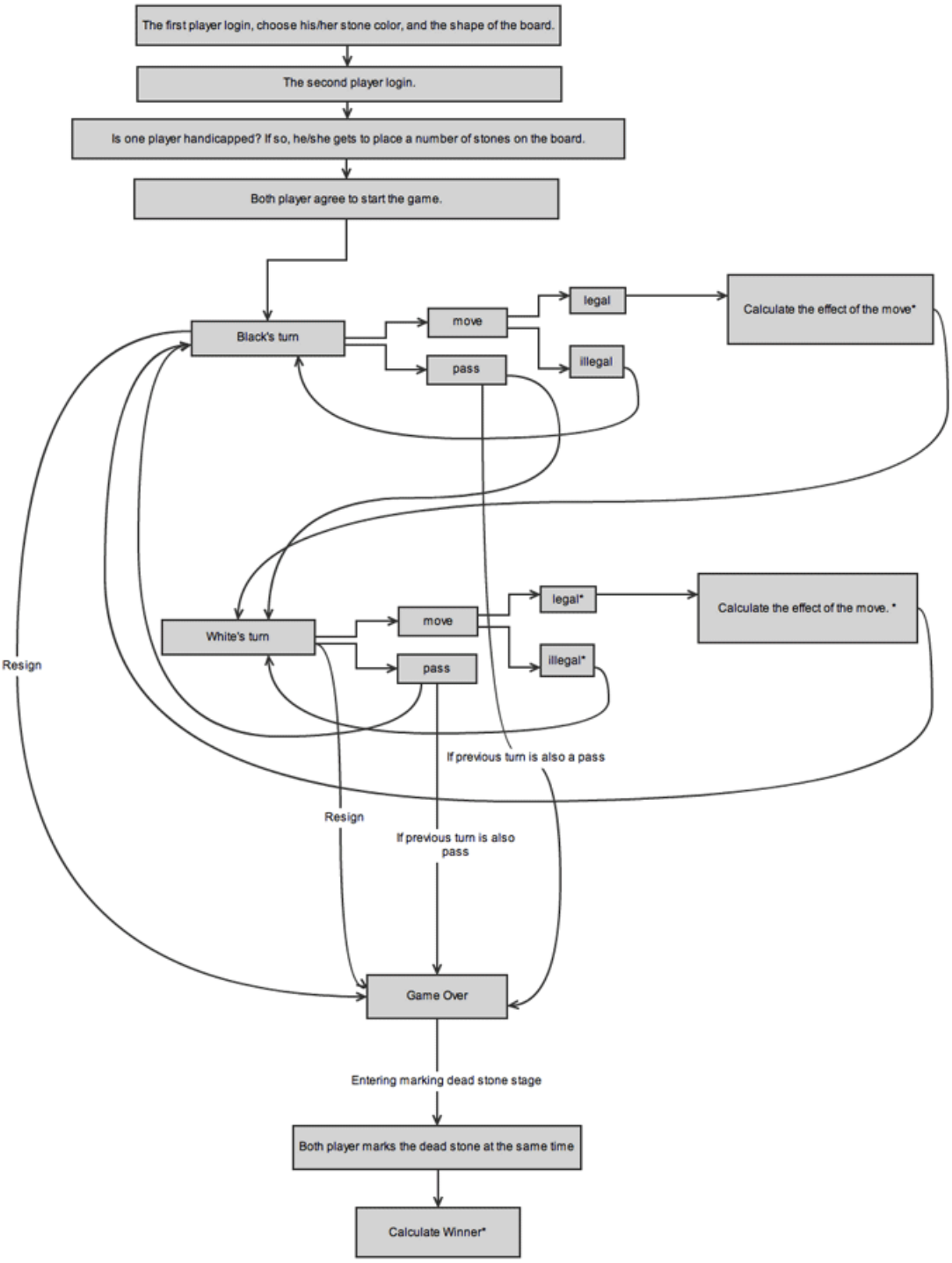


Figure 9: The flow diagram of the game.