# Paintball Frenzy: Graphical Turn-Based Game With an Optimized Minimax AI Agent

Christopher Goss

TJHSST 2004 - 2005

## 0.1 Abstract

Paintball Frenzy is a unique, newfangled turn-based game of my creation. The game is played on a four by four grid layout that is initially a neutral grey color. Two to four players that are either human or computer controlled posess one colored paintball that originates at one of the board's four corners. The players take turns moving about the grid one horizontal or vertical space at a time. Players also have the option to jump over one player's paintball in an adjacent space. There is a time limit that each player must choose their move before. Whenever a paintball moves into a space on the grid, that space becomes the color of that paintball. Whichever player has the most colored spaces at the end of the turn limit wins the game.

My final project was the creation of this game Paintball Frenzy. Firstly, I had to invent the game and all of it's particularities and balance issues. Of course, I had to program this game's framework, first with a standard terminal display and keyboard input, and later with graphical gameboard display and mouse controls. Additionally, I had to program several artificial intelligence agents for my game including an agent that moves randomly and an agent that performs optimally. To design the agent that moves optimally, I researched the minimax algorithm and used it to search through all the possible game-choice options and choose the best one. Combining all of these aspects together yielded a fully enjoyable and informative senior research project.

## 0.2 Introduction and Background

I am pleased to introduce Paintball Frenzy in all its glory. Originally, I had a massively ambitious project proposal to design a computer programming language designed specifically for the creating of video games. I was quick to dispatch this proposal towards the end of my junior year. Also in my junior year I took the Artificial Intelligence class and I learned of my deep interest in the Artificial Intelligence field and the difficult minimax algorithm in particular (I was never able to master it in the class). Therefore, I was quick to consider the minimax algorithm for my final project. But I wanted to do more than just Artificial Intelligence, I also wanted to design my own game. So I did, and Paintball Frenzy was born.

Background on game design.

Background on graphics.

Background on game AI.

## 0.3 Theory

My research was a three-pronged proccess. First, I chose to formulate an innovative turn-based game. Game Design Theory.

Graphics Theory

Intitially, I delved into my artificial research with impressive haste. I acquired a green artificial intelligence book from Mr. Torbert and I've been researching the minimax algorithm ever since. The turn-based, deterministic style of my program makes the minimax algorithm ideal. However, the multiplayer asect of the game will greatly slow down the algorithm and it'll most likely force it to use a cut-off test before the entire tree has been searched. This means, unfortunately, that at the faster time limits, my AI agent won't operate optimally. More AI Theory.

## 0.4 Design Criteria

The progress of my project is three-pronged. In the first quarter, I researched game design and programmed a working version of Paintball Frenzy withoutgraphics or an AI Agent. I used C++ to program. My Main function first initializes global variables like the board matrix. It then calls functions to display the title and generate the menus. After that, Main runs a do-while loop until the number of turns has expired. In this loop, Main calls functions to display the board and prompt the current player's move. Then Main repeats the loop with the next player, incrementing the turn counter as required. After the loop, Main calls a function to evaluate and display the results of the game.

In the second quarter, I used OpenGL graphics to make my program more aesthetically pleasing. I needed to make several design modifications to allow for a graphical interface. My main function is only responsible for initilizing variables and initiating the OpenGL display cycle. My display function (RunGame) regulates the game loop with a series of if else statements that assess the current game progress via mouse clicking history. In my main game loop, I designed to have a second set of if statements to track the state of each particular turn. First, I start a timer, then I get mouse input until the timer has expired. At the end of the time, I excecute the players' move and signal the change in player or turn. I store the mouse clicking history by saving phases of the game as integers in a global variable. I also designed DispBoard to display the gameboard in the left part of window. The upper right side displays the current turn and player. Below that is dirctional pad that records mouse input. I created three auxilary functions to handle mouse clicking, the mouse menu, and text output. I also designed the menu screen layout that would allow for alterable data and buttons.

The second semester is consumed with my attempts to add an AI agent to my program. This includes the research of Game AI and more specifically the minimax algorithm. Techniques such as alpha-beta pruning where also researched. In the end, an optimized AI Agent was produced. I've redesigned my RunGame function to call the AgentMove function at the start of its time limit to allow for the most time. I designed the first version of a recursive

minimax function that I call maximax. My AgentMove function now simply sets some variables and calls maximax to get the new board position. First, maximax tests the turn limit and time limit constraits. If the time is up or the simulated game is over, the program calls the terminal function entitled EvalFunc. Next, I increment the simulated player or turn to delve further into the maximax tree. Then, I determine if the four directional moves are possible, and recurse in each of the legal directions. Then, I decide which one is the best move, and return the changed board. I also programmed the EvalFunc.

## 0.5 References

(this is merely a reference list right now, not a bibliography)
http://www.latex-project.org/
http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html
http://www.msoe.edu/eecs/cese/resources/stl/string.htm
http://www.tjhsst.edu/ rlatimer/
http://www.opengl.org
http://www.tjhsst.edu/ rlatimer/assignments2004/independenceday.txt
The AI book
http://ai-depot.com/LogicGames/MiniMax.html