

Technical Design Document

**Dominion AI Using Monte Carlo Tree Search (MCTS)**

Version 1.0

|  |  |  |
| --- | --- | --- |
| **Position Title** | **Name** | **Signature** |
| Programmer | *Jonathan Sarasua* |  |
| Advisor | *Matt Butler* |  |

* ***Delete all instructional information before turning in final version.***
* *Replace the image above with an appropriate image.*
* *Replace names above.*
* *Update the Header and Footer with the correct information.* 
  + *Lock the date to when the version is completed, not the date of its opening.*

Table of Contents

[Table of Figures 4](#_Toc62576215)

[Introduction 5](#_Toc62576216)

[Overview 6](#_Toc62576217)

[Scope 6](#_Toc62576218)

[End Product 6](#_Toc62576219)

[Deliverables 7](#_Toc62576220)

[Table of deliverables 7](#_Toc62576221)

[Technology Sources 8](#_Toc62576222)

[Theory 9](#_Toc62576223)

[Monte Carlo Tree Search 9](#_Toc62576224)

[Selection 9](#_Toc62576225)

[Expansion 9](#_Toc62576226)

[Simulation 9](#_Toc62576227)

[Backpropagation 9](#_Toc62576228)

[Background 9](#_Toc62576229)

[Implementation 10](#_Toc62576230)

[Chance Nodes 10](#_Toc62576231)

[Selection 10](#_Toc62576232)

[Expansion 10](#_Toc62576233)

[Simulation 10](#_Toc62576234)

[Backpropagation 10](#_Toc62576235)

[Architecture 10](#_Toc62576236)

[Results 11](#_Toc62576237)

[Selection 11](#_Toc62576238)

[Expansion 11](#_Toc62576239)

[Simulation 11](#_Toc62576240)

[Example Images Section 12](#_Toc62576241)

[Example Image 1 12](#_Toc62576242)

[Example Image 2 13](#_Toc62576243)

[Example Image 3 14](#_Toc62576244)

[Example Citations Section 15](#_Toc62576245)

[Example Citation 1 15](#_Toc62576246)

[Example Citation 2 15](#_Toc62576247)

[Example Citation 3 15](#_Toc62576248)

[Bibliography 16](#_Toc62576249)

# Table of Figures

[Figure 1: Example Image 1 13](#_Toc61607850)

[Figure 2: Example Image 2 14](#_Toc61607851)

[Figure 3: Example Image 3 15](#_Toc61607852)

# Introduction

*Add your usual thesis introduction. Should be a short paragraph between 3 – 5 sentences in length, use common industry terms, and reference important existing algorithms or games.*

Dominion is a deckbuilding game where every turn you try to play and buy cards to make your deck better with the eventual goal of buying more victory point cards than your opponents. The artifact is both a playable game of Dominion and an AI using MCTS as its algorithm. The AI is implemented with chance nodes to handle the hidden information and uses Upper Confidence Bounds Applied to Trees (UCT) as the selection method. The AI is tweakable to show how changes to the AI change the result. Simple state machine strategies for Dominion like “Big Money” and “Single Witch” are used to test the strength of the AI.

# Overview

*Add a more detailed overview to this section explaining why this project is interesting to do and describe the level of detail needed in this document in order to assure the success of the project.*

Here I will talk about my interest in deckbuilding games and desire to make a competitive AI for them. Deckbuilding games are not obvious in their strategies, and most want to add cards. MCTS pulled me with the concept of not needing domain knowledge which would allow for making an AI that could be used with minor changes for an entire genre.

## Scope

*Give a brief statement of the scope of the project from a technical point and view. Tie this to the design vision and introduce the overarching development plan for this project. Insert as much detail as necessary.*

Scope limitations such as not implementing all cards and not handling different sets of cards.

## End Product

*Describe the pieces that are going to be built in order to develop the final game as defined in the Game Design Document. This sets expectations for amount of work necessary.*

*Gameplay*

* *List separately all game mechanics that need to be implemented*

*Game Objects*

* *List separately all game objects that need to be implemented*
* *For example:*
* *Player Character(s)*
* *Enemies (list with description)*
* *Pick-ups*
* *Destructible objects*
* *Vehicles*
* *Etc.*

*HUD & UI*

* *All GUI attributes implemented*
  + *List each aspect of the UI separately*

*Menu Systems*

* *All systems implemented*
* *List menus separately, with purpose and placement*

*Etc.*

# Deliverables

## Table of deliverables

Table of all of the features in a similar format to what Danny did

# Technology Sources

I will say I used Visual Studio with TinyXml2, STBI, and potentially ImGui

# Theory

*Include as many sections as necessary to cover all theoretical background information that is considered a prerequisite to the thesis. Describe important math and algorithms.*

## Monte Carlo Tree Search

Have a fancy graphic of the four phases of MCTS

### Selection

The method most commonly used for MCTS is UCT

##### Upper Confidence Bounds Applied To Trees (UCT)

Have a fancy formula here and explain it

### Expansion

### Simulation

### Backpropagation

# Background

Here I can talk about AlphaGo and other examples of MCTS usage

# Implementation

Here I will talk about my implementation of MCTS and include where I differed from pure MCTS. Using heuristics as the simulation method, not allowing all expansion paths, etc.

### Chance Nodes

### Selection

### Expansion

### Simulation

### Backpropagation

# Architecture

Here I will talk about how the code is laid out. MCTS is separate from the Dominion game and must ask the game for what is playable and what isn’t. Also all moves still go through the game. The basic methods for MCTS match up to the four phases

#### *Engine*

#### *Game*

# Results

Here I will talk about lessons learned and what worked the best. I will give data on how much better with tables.

### Selection

### Expansion

### Simulation

# Example Images Section

## Example Image 1

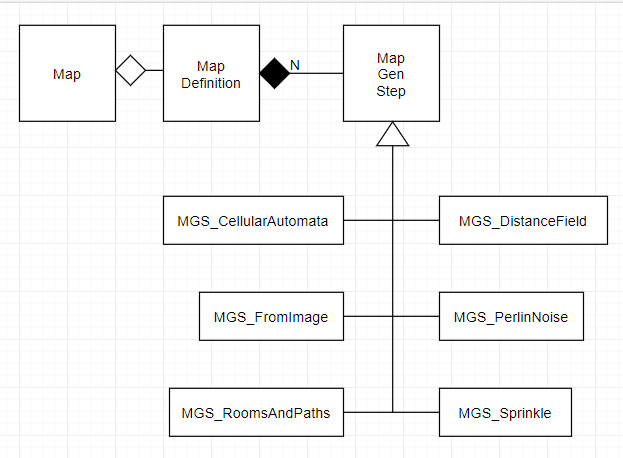


Figure : Example Image 1

## Example Image 2

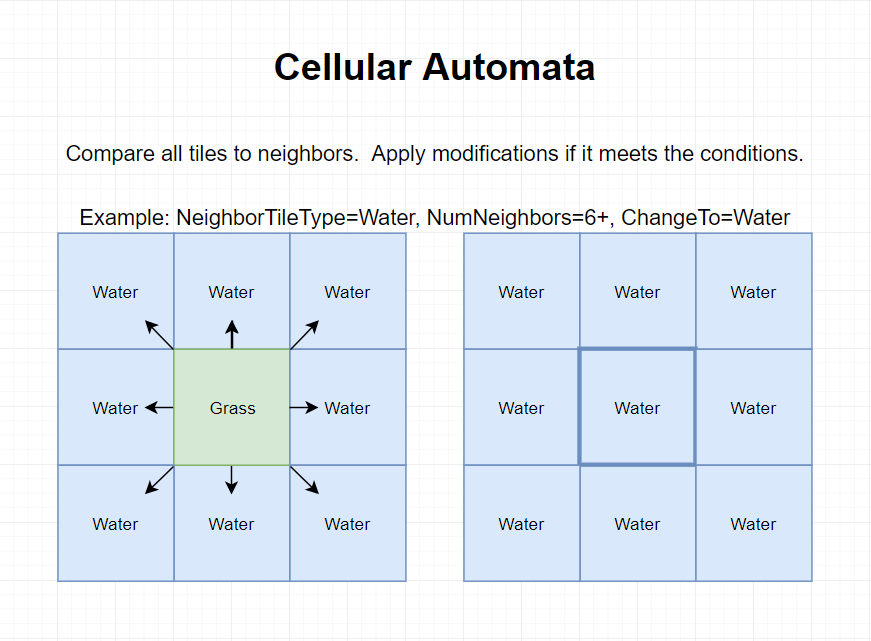


Figure : Example Image 2

## Example Image 3

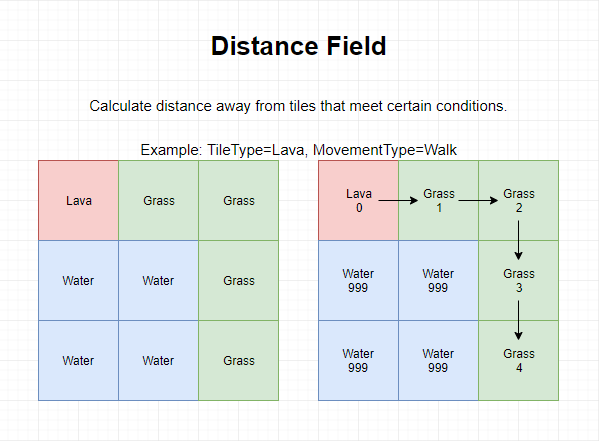


Figure : Example Image 3

# Example Citations Section

## Example Citation 1

Perlin Noise is a means of producing randomized heat maps using numbers in the negative one to positive one range [1].

## Example Citation 2

*Tiled* [2] is a map editor with many features.

## Example Citation 3

The *Tanagra* [3] map editor also has many features.

# Bibliography

|  |  |
| --- | --- |
| [1] | Wikipedia, "Perlin noise," 28 February 2020. [Online]. Available: https://en.wikipedia.org/wiki/Perlin\_noise. [Accessed 28 March 2020]. |
| [2] | T. Lindeijer, "Tiled: Map Editor," 26 March 2020. [Online]. Available: https://www.mapeditor.org/. [Accessed 28 March 2020]. |
| [3] | G. Smith, J. Whitehead and M. Mateas, "Tanagra: A Mixed Initiative Level Design Tool," 2010. [Online]. Available: https://people.engr.ncsu.edu/dlrober4/gamesreading/papers-s11/4-5.smith.10.pdf. [Accessed 28 March 2020]. |