# Paranoia

Daniel Glasscock

University of North Texas
Summer, 2005

## Abstract

Paranoia is a game played by a group of people in which targets (other players) are assigned to each player. “Kills” are made between the players and targets are discarded and transferred based on the specific rules of the kill. An effective model for the game is established using directed graphs, and the facets of the game are thoroughly developed in this model. The majority of the work focuses on attaining the most desirable Paranoia game based on four primary concerns: equivalency among the players, non-degeneracy, and maximal self and duplicate distances. Each concern is related to the directed graph model and explored. Vertex transitive digraphs are utilized to ensure equivalency, games with specific cycle eliminating sets are proved to be non-degenerate, and values are assigned to help compare self and duplicate distances between similar games. After discussion of the four primary concerns, an applications section explains the practicality of the work to creating and maintaining a Paranoia game.

# Paranoia

Daniel Glasscock

August 11, 2005

## Contents

# 1 Overview

In this chapter, the game of Paranoia is described and the general rules of the kill are introduced. The main concerns for creating an acceptable Paranoia - equivalency, non-degeneracy, self distance, and duplicate distance - are outlined.

## 1.1 What is Paranoia?

Paranoia is a strategic, live-action game that challenges players in evasion, secrecy, and tactics. The game can be played indoors or out and last for an afternoon or a few months. While, in theory, Paranoia can be played with any number of competitors, the game's size is usually best when tailored to the specific playing environment. Practical information on creating and maintaining a Paranoia game can be found in chapter 6.

Before a Paranoia game begins, a game overseer assigns to each player a set of targets (via target cards) that indicate other players (usually via a picture). According to specific rules, kills are made between combatants, eliminating players from the game. The game ends when no more kills are possible.

Because kills are the only mechanism by which a game changes, general rules about the kill exist in all Paranoia games. More specific rules, depending on factors of the game such as location and scope, set the exact guidelines for what constitutes a kill. Allow *Miss* and *Tim*, two players in a Paranoia game, to aid in the discussion of the general rules of the kill:

1. **Permission**: If *Miss* holds a target card indicating *Tim*, *Miss* may kill *Tim*.
2. **Consignation**: If *Tim* is killed by *Miss*, *Tim* gives his set of target cards to *Miss*.
3. **Elimination**: If *Tim* is killed, *Tim* is removed from the game and every target card indicating *Tim* is discarded.

## 1.2 The Problem

There are certainly several factors that should be considered when creating an acceptable Paranoia game. A more detailed discussion immediately follows each qualification:

1. **Equivalency**: Every player's initial position in the game is equivalent.
   - The most important aspect of an acceptable Paranoia game is *equivalency* between the players. Imagine a Paranoia game is first created with numbers (player 1 targets players 2, 3, and 4, player 2 targets players 3, 4, and 5, etc.). The players should have confidence that, regardless of

which number they choose to assume, their initial position in the game will not be helped or hurt. *Equivalency* is explored in chapter 3.

2. **Non-Degeneracy**: The game cannot end with more than one surviving player.

    - It is important that a Paranoia game is created to ensure *non-degeneracy*, or a guarantee that the game will end with only one winner. Suppose player 1 and player 7 are both able to collect the target cards that indicate each of them. If neither player can be killed, the game will end with both players still alive. The ideas of *invincibility* and *degeneracy* are explored in chapter 4.

3. **Self Distance**: The distance between any player and the target cards that indicate him is maximized.

    - After ensuring equivalency and non-degeneracy, it is advantageous to make the separation between any player and the target cards that indicate him as large as possible. As mentioned before, if a player is able to collect all of the target cards that indicate him, he will win the game. Making it difficult for a player to collect the target cards that indicate him by maximizing the number of kills required to acquire them (*self distance*) is essential. The concepts of maximizing *self distance* are explored in chapter 5.

4. **Duplicate Distance**: The distance between any player and duplicate target cards is maximized.

    - Lastly, the distance between any player and duplicate target cards should be maximized. A player gains no advantage in attaining duplicate target cards; in fact, it is the player that is being targeted that has the advantage. Once two or more identical target cards (or any target cards, for that matter) are held by the same person, they will be held by one person for the entirety of the game. The chance for a player to attain duplicates should be minimized. *Duplicate distance* is defined and explored in chapter 5.

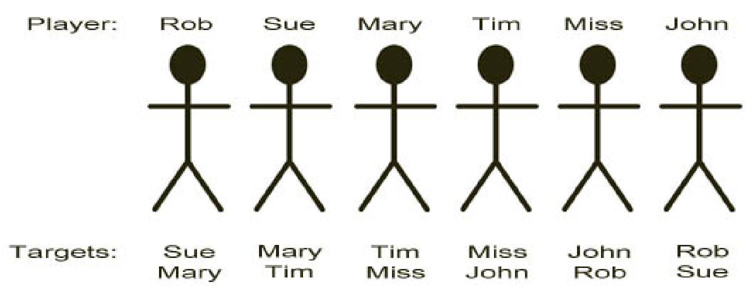
Unique difficulties arise when attempting to attain each of these qualities. The majority of this work deals with examining each of these qualifications and understanding how to attain the most desirable Paranoia game.

# 2 A Paranoia Game

Directed graphs are utilized as the most effective model for Paranoia games. In this chapter, several of the game's features are related to directed graphs. Game states and the game state tree are also introduced as theoretical models more useful in understanding many of the games more complicated facets.

## 2.1 The Model

Consider this Paranoia game (the original Paranoia game):




The goal is to develop a model that allows for a natural way to represent the game. The first step is to eliminate the names and replace them with numbers, as a player's number would be just as unique as his name. The graphics could become cumbersome in games with as few as ten players, and are, on the whole, unnecessary. This may lead to representing the game like this:

| Player | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Targets | 2 | 3 | 4 | 5 | 6 | 1 |
| | 3 | 4 | 5 | 6 | 1 | 2 |

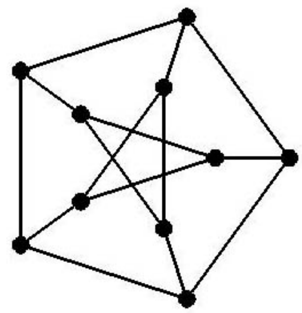
Noticing that this bears a striking resemblance to a matrix, the information may be reorganized one more time into a players by players, binary matrix in which a 1 in the $a, b$ position would mean that player $a$ targets player $b$:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The last step in the development of an effective model is to pick up a book on graph theory and notice that the matrix above could be the adjacency matrix for a directed graph.

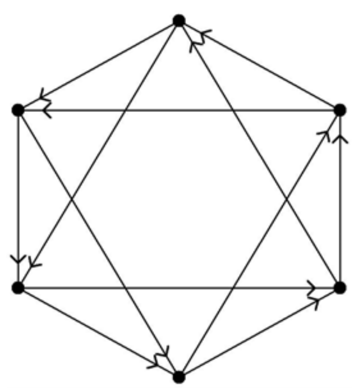
## 2.2 Graphs and Directed Graphs

A *graph* is composed of *vertices* and a collection of *edges* that connects the vertices. The graph below (the Petersen graph) has 10 vertices and 15 edges:



Notice that the graph's vertices are not labeled. If a graph's vertices are labeled, the graph's *adjacency matrix* can be defined as a vertices by vertices, binary matrix in which a 1 in the $a, b$ position represents an edge connecting vertices $a$ and $b$.

A graph's edges can be directed so that the edge originates on one vertex and terminates on another (like an arrow). A graph that has directed edges is called a *directed graph* or a *digraph*. Here is a directed graph with 6 vertices and 12 directed edges:



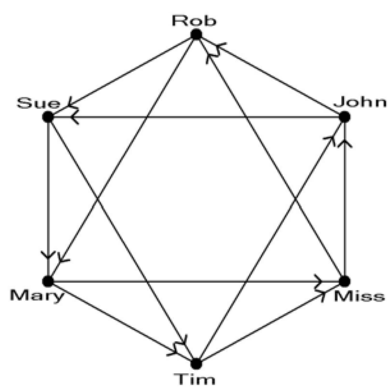
To come full circle, the adjacency matrix for a directed graph is nearly the same as the adjacency matrix for an undirected graph, except that a 1 in the $a, b$ position represents a directed edge originating from vertex $a$ and terminating on vertex $b$. If the vertices were labeled clockwise, the adjacency matrix for the directed graph above would look like this:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

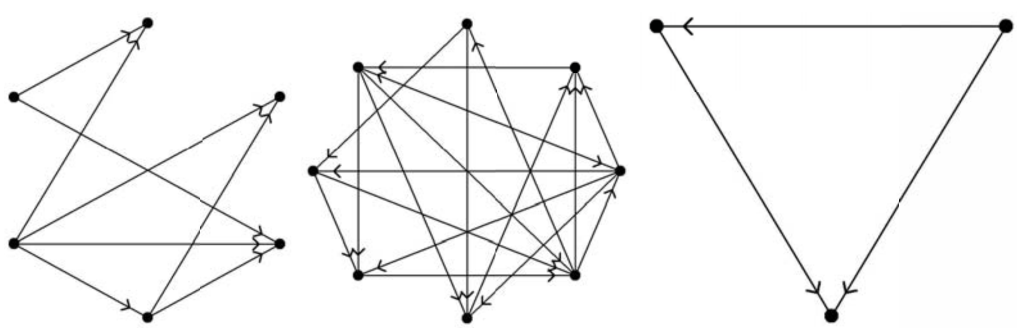
Notice that this matrix is exactly the players by players, binary matrix that was used to describe the original Paranoia game! Directed graphs are the most effective model for Paranoia games in the following way:

- Vertices in a digraph represent players in a Paranoia game
- Directed edges in a digraph represent targets in a Paranoia game (on an edge, the originating player targets the terminating player)

Looking back to the original Paranoia game, directed graphs are a perfect fit. Every player is represented by a vertex and the edges represent who those players target:



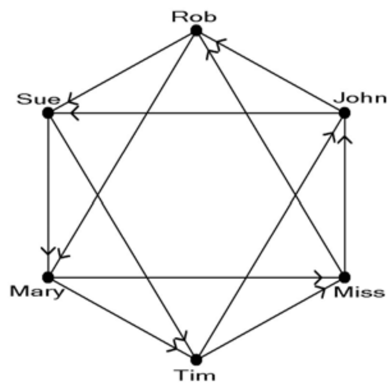

You may have noticed that this Paranoia game has a particularly nice structure in that each player targets and is targeted by two other players, and that no player has any advantage (assuming that the skill level of each player is the same). Paranoia games don't have to look this nice. Below are a few examples of Paranoia games that lack the structure of the first example:



All of these examples come from the same 20 player game in which multiple kills have been made. It may be surprising to learn that the 20 player game had a regular structure similar to that of the original Paranoia game, but despite that regular structure, the game on the left above has two players that cannot be killed.

### 2.2.1 Target and Attacker Sets

The *target set* of some player (call him *Tim*) in a Paranoia game is the set of players that *Tim* targets. The *attacker set* of *Tim* is the set of players that target *Tim*.



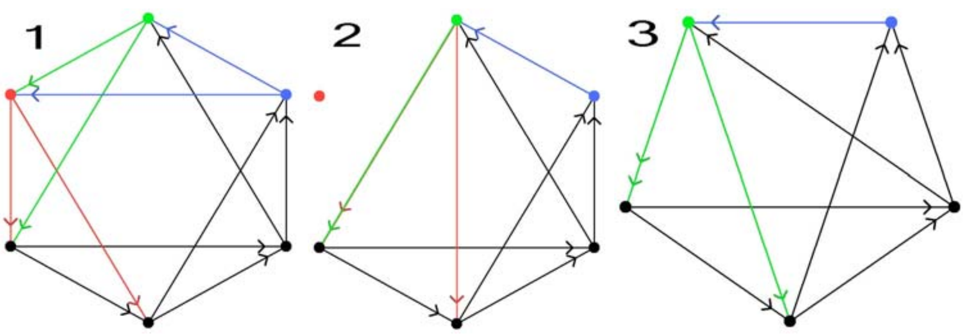

In the original Paranoia game above, we can see that the target set of *Tim* consists of *Miss* and *John* while the attacker set of *Tim* consists of *Sue* and *Mary*.

### 2.2.2 The Kill

Kills are the only mechanism by which the game can change. Remember the three rules by which kills are made: permission, consignation, and elimination. These rules can now be related to the digraph model that has been chosen to represent Paranoia games. The rules, in terms of directed graphs, read as follows (with help from *Miss* and *Tim*):

1. **Permission**: If there is a directed edge originating from *Miss* and terminating at *Tim*, *Miss* may kill *Tim*
2. **Consignation**: If *Tim* is killed by *Miss*, the originating edges from *Tim* now originate from *Miss*
3. **Elimination**: If *Tim* is killed, *Tim* is removed from the game and every edge terminating at *Tim* is removed

What does this mean in terms of the actual graph?



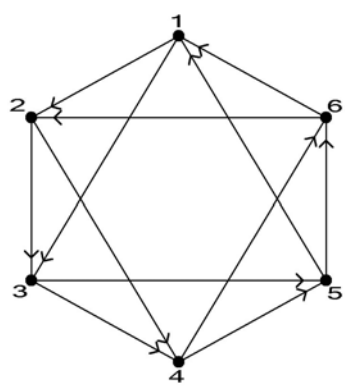

In the games above, the green, red, and blue players will all be involved in a kill. Green will kill red, and blue will lose a target due to that kill.

1. Green has permission to kill red (green targets red).
2. Green has killed red, and the edges that originated from red now originate from green (green acquires red's targets).
3. Red has been eliminated from the game, along with the blue edge that terminated on red (all target cards that indicate red are discarded).

This is the basic mechanism of a kill.

### 2.2.3 Paths, Kill Paths, Cycles, and Distance

A *path* in a graph or digraph is a series of distinct, consecutive vertices and edges of the form: *vertex*, *edge*, *vertex*, *edge*, ..., *vertex*, *edge*, *vertex*. It is still a path, but more specifically known as a *cycle*, if the first and last vertices are the same.




A path in the digraph above could be $4 \to 6 \to 1 \to 2$, and a cycle could be $4 \to 6 \to 1 \to 2 \to 4$.

In a Paranoia game, a path can represent a series of kills called a *kill path*. In the game above, the kill path $4 \to 6 \to 1 \to 2$ represents a series of kills. The arrow

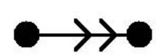
(→) can be thought of as an operator (left kills right), but the representation of the kill path is still slightly ambiguous: does player 4 kill player 6 then player 1 then player 2, or some other order. Because the arrow operator is associative, the order in which those kills takes place does not matter (like adding a set of numbers). In other words, the result of the kill path $4 \to 6 \to 1 \to 2$, regardless of the actual order of the kills, will be the same as if 4 had killed 6, then 1, then 2.

In graphs or digraphs, the notion of distance can be useful. When traversing from one vertex to another, the *distance* traveled is the number of edges traversed. For instance, there can be several distances associated with traveling from vertex 4 to vertex 2 in the graph above depending on which path is taken. The set of vertices a distance 1 away from vertex 4 would be $\{5, 6\}$ in the graph above. Distance is useful when determining the self and duplicate distance for a player in a Paranoia game.

#### 2.2.4 Duplicate, Self, and No Targets

Throughout the game (especially near the end), a player may be left with no targets, duplicate targets, or perhaps targets that indicate himself. Fortunately, directed graphs are able to effectively model each of these occurrences:

1. **No targets**: Suppose that a player lies dormant for a long while, not killing or being killed, and in the mean time, all of the players that he targets are killed. Suppose that that player wanted to start playing, only to find that he has lost all of his targets due to the rule of elimination - in the digraph, there are no edges originating from him. A more practical look at how to deal with this problem can be found in chapter 6.

2. **Duplicate targets**: Probably the most common occurrence of the three (though hopefully not too common), if two players share a common target and one kills the other, the killer attains duplicate targets. The player holding two of the same target cards is not at an advantage - he doesn't gain any special privileges because of the duplicates. In the digraph, duplicate target cards are represented with two edges, each originating and terminating on the same vertices (shown with an edge and two arrows):  .

3. **Invincibility**: A player is said to be *invincible* if no other players target him. If a player is able to acquire all of the target cards that indicate him, he becomes invincible. A player that holds a target that indicates himself is represented in the digraph by an edge that originates and terminates on that player's vertex:  .

#### 2.2.5 The Winner(s) and the End of the Game

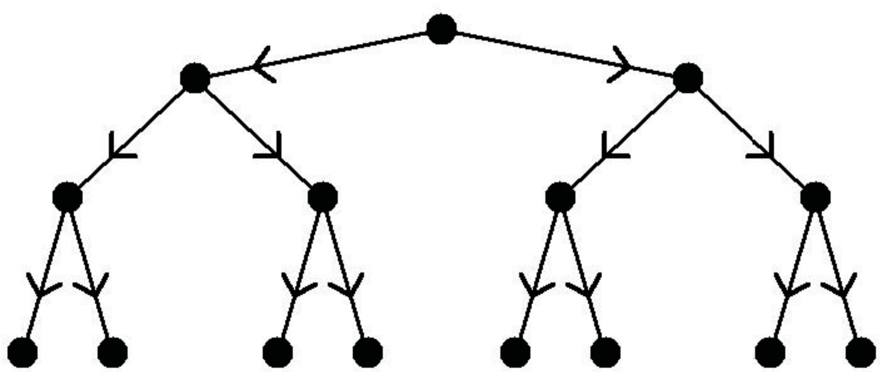
When no more kills are possible, the game is over and the surviving player(s) wins. In the digraph, the end of the game is signified by complete disconnectedness between the remaining vertices: the only edges that remain are those that originate and terminate on the same vertex. In other words, the remaining players are invincible. This game has ended, , and both remaining players win.

### 2.3 Game States

The *state* of a Paranoia game refers to the current number of players and the target arrangement that connects them. Consequently, a Paranoia game has many *game states* - every time a kill is made, the game enters a new state. A Paranoia game consists of an *initial state*, several *intermediate states*, and an *end state*. Before any kills are made, the game is in its initial state; when no more kills are possible, the game is in its end state; every other state the game is in is an intermediate state.

#### 2.3.1 Game's Path and the Game State Tree

A *tree* is a graph with no cycles. Trees can be utilized to organize all of the possible game states of a Paranoia game.



The directed graph above is a tree. Notice that the top vertex has only originating edges (the *root node*) and the bottom vertices have only terminating edges (*leaf nodes*).

The *game state tree* is a tree in which the vertices represent game states and the edges represent potential kills. The root node of the tree is the initial game state and the leaf nodes of the tree are the possible end game states. The vertex on the top level (distance 0 from the root node, a.k.a. the root node) is the initial state (a game can only have one initial state). The vertices at distance 1 from the root node are all of the possible game states after one kill. In general, the vertices at a distance of $n$ from the root node represent all of the possible game states after $n$ kills.

A Paranoia game can be said to start at the root node of it's game state tree and travel away from the root node to a leaf node. A complete game will follow a path of this form: *initial game state*, *kill*, *intermediate game state*, *kill*, . . . , *intermediate game state*, *kill*, *end game state*. If the path is not meant to show an entire game, the path can start or end on any game state. This type of path is called the *game's path*.

### 2.3.2 Reaching a Game State

The vertices in the game state tree are not distinct, thus there can be several game paths to a particular game state. For instance, if we are trying to reach the end state where player 1 wins in a game with three players who all target each other, 1 could kill 2 then 3, or 1 could kill 3 then 2, or 2 could kill 3 then 1 could kill 2, and so on. The kill paths $1 \rightarrow 2 \rightarrow 3$ and $1 \rightarrow 3 \rightarrow 2$ represent all of the possible game paths in which player 1 wins.

Kill paths can be used to represent overall changes in the game, or a transition from one game state to another. The kill paths mentioned in the previous paragraph both represent a transition from the initial state of the game to the end state of the game where player 1 wins. As mentioned before, the representation for a kill path can be purposefully ambiguous because it doesn't matter in what order the kills are made, just that the game transitions from one state to another.

A *kill path to invincibility* for some player is a kill path that, when taken, results in that player becoming invincible. A *game path to invincibility* for some player is a series of game states (a path through the game's game state tree) that concludes with a game state in which the specified player is invincible. Kill paths and game paths to invincibility are important in understanding and utilizing the procedure to determine degeneracy and non-degeneracy.

# 3 Ensuring Equivalency

Equivalency between players is the most important aspect of creating an acceptable Paranoia game. This chapter focuses on creating Paranoia games from circulant digraphs, a class of vertex transitive digraphs. By restricting the creation of Paranoia games to circulant digraphs, equivalency between players can be assured.

## 3.1 Vertex Transitive Graphs

A *vertex transitive graph* is a graph that has vertices that are essentially all alike. Imagine standing on a vertex and looking out over a graph. If the graph is vertex transitive, the view would look the same no matter which vertex you were standing on.

Vertex transitive digraphs can be utilized to create equality between players in a Paranoia game. Because each player assumes a vertex, and all vertices have the same role in the overall structure of the digraph, every player is equivalent.

There are several different types of vertex transitive graphs, but, unfortunately, not every vertex transitive graph can be *oriented* (edges changed from undirected to directed) to form a vertex transitive digraph.

The focus for creating Paranoia games will now be limited to vertex transitive digraphs since they nicely satisfy the first condition for acceptable Paranoia games.

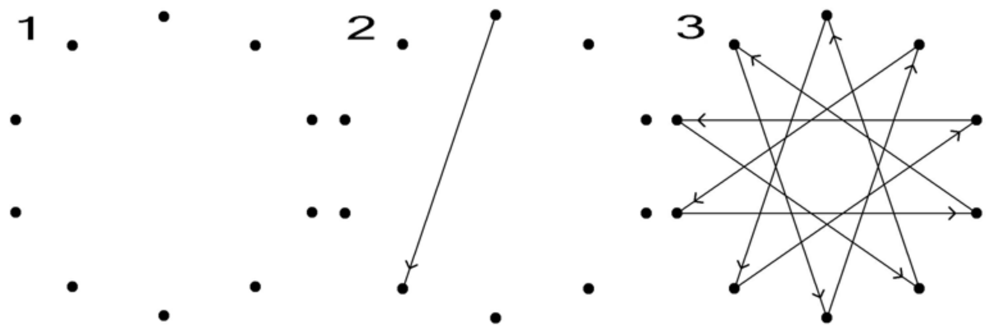
## 3.2 Circulant Digraphs

*Circulant digraphs* are the class of vertex transitive digraphs that were chosen to help create and study the majority of Paranoia games throughout the body of this work.

### 3.2.1 Target Numbers

Circulant Paranoia games are created by connecting players with combinations of *target numbers*.

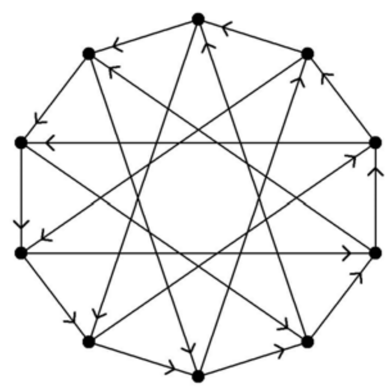
First, an example. Imagine that you have a group of players and wish to create a circulant game.




1. Have the players create a circle.
2. Tell the top most player to target the fourth player going counter-clockwise.
3. Give that instruction to every player.

Game number 3 above is denoted $C_{10}^{4}$ and is said to have 10 players and the target number 4.

The circulant Paranoia game below is denoted $C_{10}^{1,4}$ and is said to have the target numbers 1 and 4.



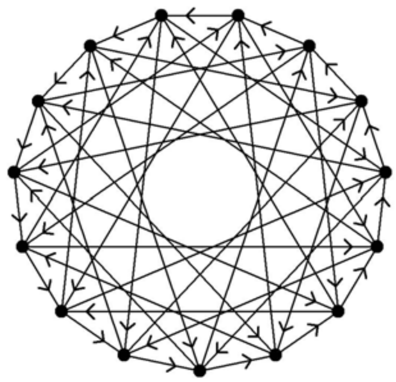
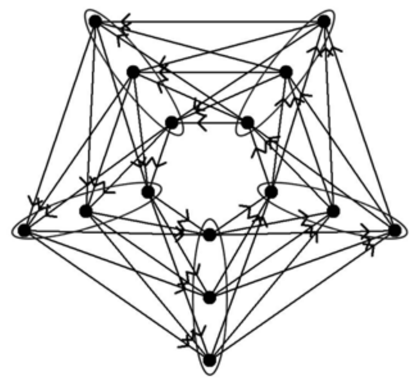
The target number $x$ in a circulant game means that each player targets the $x$th player counter-clockwise (or clockwise, as long as it is consistent). In a game with $p$ players, the set of possible target numbers is 0 (players target themselves) and 1 (target one adjacent player) to $p-1$ (target the other adjacent player).

The original Paranoia game is circulant and is denoted $C_{6}^{1,2}$.

In the most general case, the circulant game $C_{p}^{x,y,z}$ has $p$ players and the target numbers $x$, $y$, and $z$.

### 3.2.2 Spokes

In many cases, games like the one on the left below can become too complicated to study in their original form. It is often advantageous to rearrange the vertices to better understand the game's structure.

Both games above are $C_{15}^{1,6,11}$ - the game on the right is the game on the left with the vertices rearranged. The structure of the game quickly becomes apparent when analyzing the rearranged game: there are five groups of three players, every player in each group targeting every player in the next group.

These groups are called *spokes*, and, like spokes on a bicycle tire, they radiate outward from the center of the circle. The five spokes are circled in the graph on the right above.

A circulant Paranoia game can be reorganized into $\frac{p}{a}$ spokes when $a$ is some proper divisor of $p$ (in other words, $a \neq p$). When the players are numbered, players $x, y$ are on the same *spoke* if $x \equiv y$ $(mod\ \frac{p}{a})$.

Remember that a cycle in a graph or digraph is defined as a path where the first and last vertices are the same. A *spoke cycle* is a cycle in a Paranoia game that visits more than one spoke. In the discussion of degeneracy, spoke cycles are, in a sense, non-trivial cycles compared to cycles that solely reside on one spoke.

# 4 Combating Degeneracy

Understanding and avoiding degeneracy is key in creating an acceptable Paranoia game. The game overseer should have complete confidence that, without intervention, the game he creates will finish with only one winner. This chapter introduces a procedure for determining degeneracy or non-degeneracy and describes cycle eliminating sets and their importance to the study of degeneracy.

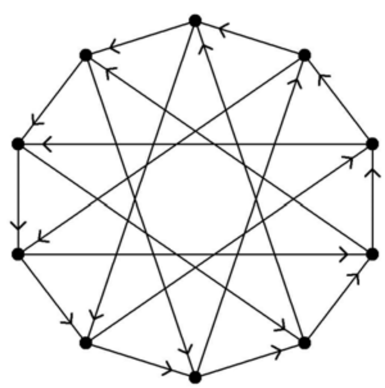
## 4.1 What is Degeneracy?

A Paranoia game is said to be *degenerate* if it is possible for two or more players to be alive in the game's end state. In other words, if there can be more than one winner in a Paranoia game, the game is degenerate. If a game is guaranteed to have only one winner, the game is said to be *non-degenerate*. Degenerate and non-degenerate are abbreviated $D$ and $ND$, respectively.

## 4.2 Determining Degeneracy: Invincible Players

Understanding why some games are degenerate and others are not is a matter of understanding what happens after players achieve invincibility. Remember that a player is said to be invincible if no other players target him.

To begin, consider the game from the previous section ($C_{10}^{1,4}$):



To determine whether this game (or any game, circulant or not) is degenerate or non-degenerate, a fail-safe procedure is necessary. The procedure is as follows:

1. Choose some player (call him *Tim*) to achieve invincibility; if all possible players have been tried, go to step 5.

2. Allow *Tim* to achieve invincibility (only *Tim* is allowed to kill other players in this step); if all possible kill paths to invincibility for this player have been tried, go back to step 1 and select another player.

3. Select a player that *Tim* does not target (call her *Miss*); if no suitable player exists, go to back to step 2 and choose another kill path to invincibility.

4. If *Miss* is able to achieve invincibility (only *Miss* kills in this step), the game is degenerate; if not, go back to step 3 and select another player.

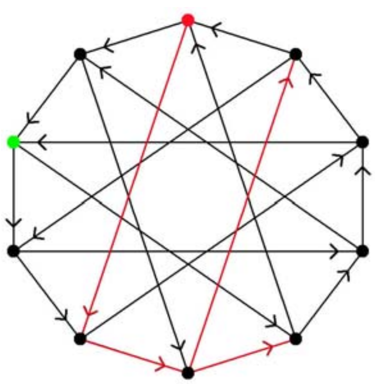
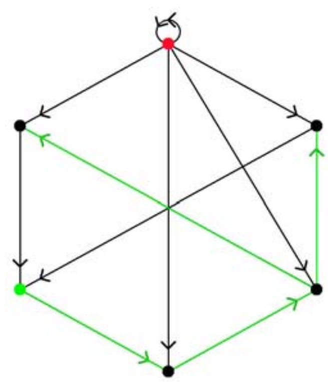
5. The game is non-degenerate.

In case you are skeptical as to whether or not this procedure is mathematically sound, see section 4.2.1.

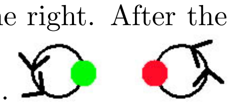
Returning to the question as to whether or not $C_{10}^{1,4}$ is degenerate, apply the procedure:

1. Because circulant digraphs are vertex transitive graph (all of the players are equivalent), it does not matter which player we designate to achieve invincibility first. Let's choose the top most player, red.

2. Allow red to make the kills highlighted in red in the game on the left below. The game after red makes those kills is on the right (notice that no player targets him):

3. Due to the kill path red made, there is only one player that red does not target after becoming invincible. We must choose the player on the lower left (green) as a candidate for invincibility.

4. Green is able to become invincible by making the kills marked by the green edges in the game to the right. After the green makes the kills in that path, the game looks like this:  .

This game is clearly degenerate: the game is in its end state and two players remain.

### 4.2.1 Why the Procedure Works

This is the most general procedure for determining whether any Paranoia game is degenerate or non-degenerate (integrated with the justification for each step):

1. Choose some player (call him *Tim*) to achieve invincibility; if all possible players have been tried, go to step 5.
    - In the end of every Paranoia game, one or more players will be invincible. For vertex transitive games, it is safe to check only one player for this step in the procedure since all of the players are essentially equivalent. For other games, a loop is established to check every player.
2. Allow *Tim* to achieve invincibility (only *Tim* is allowed to kill other players in this step); if all possible kill paths to invincibility for this player have been tried, go back to step 1 and select another player.
    - After choosing the player that we are assuming will achieve invincibility, a loop allows us to check every path to invincibility. The key lies within checking every path - if that player is to become invincible, one of the paths will have to be taken. Allowing the kills to be made by one player only is a matter of simplifying the procedure - it doesn't matter who kills who, as long as the kills are completed.
3. Select a player that *Tim* does not target (call her *Miss*); if no suitable player exists, go to back to step 2 and choose another kill path to invincibility.
    - After taking each kill path to invincibility for *Tim*, the game is at a state where it can be decided if any other player can simultaneously become invincible. The only requirement for this player is that he is not targeted by the player that just became invincible.
4. If *Miss* is able to achieve invincibility (only *Miss* kills in this step), the game is degenerate; if not, go back to step 3 and select another player.
    - If one of these players exists, and there is a kill path for which that player can become invincible, the game is degenerate. A specific example is all that is needed in order to prove that a game is degenerate.
5. The game is non-degenerate.
    - If all invincibility kill paths for every player have been tested, the game can only be non-degenerate.

## 4.3 ND: Too Many Target Numbers

There are certain cases where it can be proven that a game with certain attributes will be degenerate or non-degenerate. If Paranoia game has an excess of target numbers, for example, it can be proven, regardless of the arrangement of those target numbers, that the game will be non-degenerate.

**ND Theorem 1** *Let $\mathcal{G}$ be a circulant Paranoia game with $p$ players and $c$ target numbers. If $c \geq \lfloor \frac{p}{2} \rfloor$, then $\mathcal{G}$ is non-degenerate.*

**Proof** Because $\mathcal{G}$ is circulant and vertex transitive, it is safe to assume that some specific player *Tim* will achieve invincibility.

Contradiction. Suppose *Tim* achieves invincibility in the minimum number of kills ($c$ kills) and does not target some player *Miss*. The number of players eligible to target *Miss* remaining in the game is $p - c - 2$ (*Tim* does not target *Miss* and she does not target herself). *Miss* must be targeted $c$ times, but $p - c - 2 < c$ when $c \geq \lfloor \frac{p}{2} \rfloor$ and each of the $p - c - 2$ players may only target *Miss* once. There are not enough players remaining in the game to target *Miss*. This is a contradiction - *Tim* must have targeted all players after achieving invincibility, thus the game is non-degenerate.

■

## 4.4 ND: Cycle Eliminating Sets

A *cycle eliminating set* is a set of players in a Paranoia game that, when removed with every edge that terminate or originate on any vertex in the set, eliminates all spoke cycles in the game. Cycle eliminating sets are abbreviated *CES*.

### 4.4.1 Degeneracy and CES

It's easy to see what a cycle means in a graph: it's a closed path - if you were to walk it, you would walk in a loop. Understanding what cycles mean in a Paranoia game is important to understanding how CES affect degeneracy.

A cycle in a Paranoia game represents the ability for a player to acquire at least one target card that indicates himself. If there is not a path beginning and ending on a specific player, either that player is already invincible (there are no edges that terminate on him, thus there is not a path that can end on him), or he cannot attain invincibility (if there are terminating edges on him, he cannot kill the players from which those edges originate).

Let's now focus on what we know about certain players achieving invincibility. In a vertex transitive graph (such as the circulant graphs), we can assume that some specific player (*Tim* again) will achieve invincibility. The only players that are killed in *every* kill path to invincibility for *Tim* are the players in *Tim*'s attacker set. In other words, regardless of the other kills, *Tim* must kill his entire attacker set if he is to achieve invincibility. Suppose that *Tim*'s attacker set is a cycle eliminating set, so that when he achieves invincibility, there are no spoke cycles left in the game. No other player can become invincible and thus the game is non-degenerate.

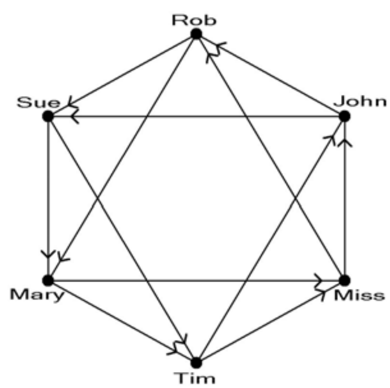
This leads us to the main result in this chapter:

**ND Theorem 2** *If the attacker set of some player in a circulant Paranoia game $\mathcal{G}$ with at least one spoke cycle is a cycle eliminating set, $\mathcal{G}$ is non-degenerate.*

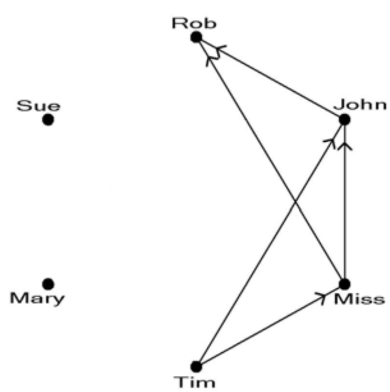
**Proof** Because $\mathcal{G}$ is vertex transitive, it is safe to assume that player *Tim* can be designated to achieve invincibility. Let *Tim* kill every player in his attacker set (a necessary move to achieve invincibility). By definition, a cycle eliminating set removes every spoke cycle. Because each player began the game with a spoke cycle on their vertex, elimination of those cycles means that no other player can achieve invincibility. $\mathcal{G}$ is non-degenerate.

■

As an example, the original Paranoia game is non-degenerate because the attacker set of any player is also a cycle eliminating set. The original Paranoia game is below:




Suppose we were to remove *Tim*'s attacker set:



Notice that removing *Tim*'s attacker set quite effectively eliminates every cycle in the game (and thus every spoke cycle as well). This means that no other player can become invincible simultaneously with *Tim* (or any player for that matter).

### 4.4.2 Technical Form Specifications

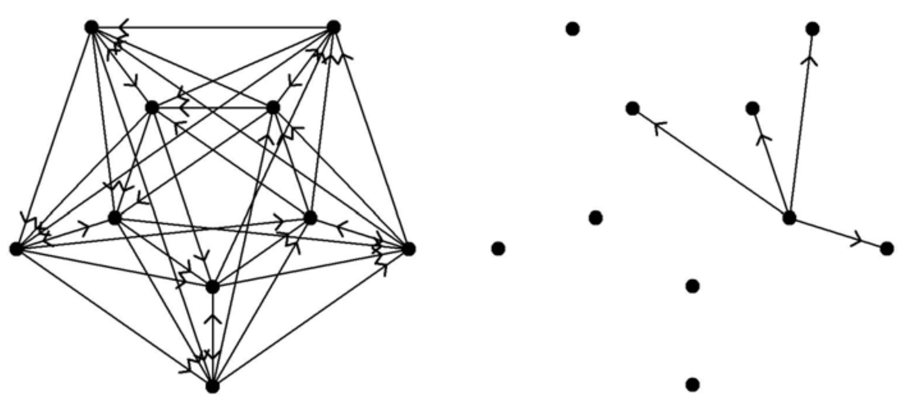
In section 3.2.2, the concept of rearranging the vertices of a game into spokes was introduced. In order to create games in which a player's attacker set will also be a cycle eliminating set, spokes are utilized. There are only two types of games (only one of which is explained here - see chapter 7) in which the attacker set of some player is also a cycle eliminating set.

Suppose $\mathcal{G}$ is a Paranoia game with $p$ players. The game is non-degenerate if it has target numbers of the form $\{0_g, 1, 2, \ldots, q_h\}$ $(mod\ \frac{p}{a})$ where $g = \{0, 1, 2, \ldots, a-1\} \backslash h$, $h = \{0\frac{a}{d}, 1\frac{a}{d}, \ldots, (d-1)\frac{a}{d}\}$, $d$ is some divisor of $a$, $a$ is some proper divisor of $p$, and $1 \leq q < \frac{p}{a}$.

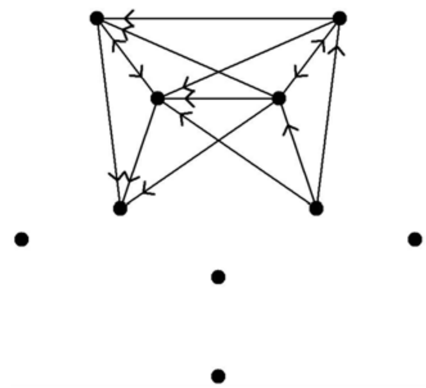
There are two types of form numbers: with subscripts and without subscripts. The form numbers without subscripts represent *every* possible target number (0 to $p-1$) that is equivalent to that form number $(mod\ \frac{p}{a})$. The form numbers with subscripts (specific subscript $i$, for example) represent the $(i+1)$th possible target number (0 to $p-1$) that is equivalent to that form number $(mod\ \frac{p}{a})$.

In games with this form, any player targets $a-d$ players on his own spoke, $d$ players on the $q$th spoke, and every player on every spoke in-between for a total of $aq$ targets per player.

As an example, see $C_{10}^{1,2,5,6}$:



For this game, $p = 10$, $a = 2$, $d = 1$, and $q = 2$. This game's target numbers have the form $\{0_1, 1, 2_0\}$ $(mod\ \frac{10}{2})$. Notice that $\{5\}$ is the $(1+1)$nd possible target number equivalent to 0 $(mod\ \frac{10}{2})$, $\{1, 6\}$ are all of the possible target numbers equivalent to 1 ($mod\ \frac{10}{2}$), and $\{2\}$ is the $(0+1)$st possible target number equivalent to 2 ($mod\ \frac{10}{2}$). Notice that when some player's attacker set is removed, the game is left without any spoke cycles (although it is left with trivial cycles that reside on single spokes):



The form discussed here is a sort of "minimal" form in the sense that target numbers equivalent to 0 ($mod\ \frac{10}{2}$) can be added (not subtracted) from any of these games without affecting their non-degeneracy. Adding target numbers equivalent to 0 ($mod\ \frac{10}{2}$) adds only trivial cycles to the game.

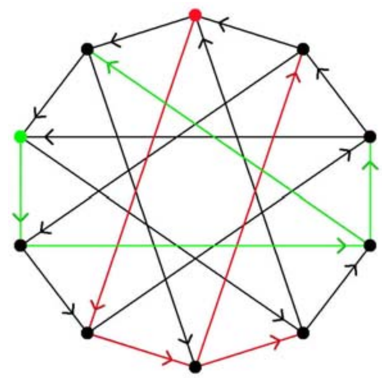
## 4.5 D: All Other Games?

Although thousands of example games suggest that it is true, it remains to be proven that if the attacker set of some player is not a cycle eliminating, the game will be degenerate (given that there are a sufficient number of players).

### 4.5.1 Disjoint Forks

*Fork* is the term used to describe the shape of a kill path to invincibility for some player in a Paranoia game. The player would ideally kill other players up to a point (the handle) and then branch out to kill those in his attacker set (the prongs). In degenerate games, there are two *disjoint* (not sharing any common players) forks that can make two players invincible simultaneously.

In the game below, red and green can both follow their forks to become invincible:



Because there exists two disjoint forks, this game is degenerate. Notice that green and red are at least a distance 2 from any vertex that exists in the other players path. This is true because the two cannot target each other after achieving invincibility.

Proving the existence of two or more disjoint forks in a game may be the key to proving a necessary condition for non-degeneracy. Work still remains to be done in this area.

# 5 Self and Duplicate Distance

Up to this point, two vertex transitive, non-degenerate Paranoia games would appear to be equally desirable. This chapter outlines some of the addition considerations that should be made after ensuring that equivalency and non-degeneracy have been met.

## 5.1 Self Distance

If a player is able to become invincible in a non-degenerate game, he will be the sole survivor. For this reason, it is of utmost importance to maximize the separation of a player and the target cards that indicate him.

*Self distance* is an integer used to quantify the "distance" between a player and the target cards that indicate him. The greater the self distance, the longer it takes and the more kills required to become invincible. Each player has his own self distance, but, in a vertex transitive game, every player's self distance is the same making it a property of the game instead of individual players.

### 5.1.1 Technical Definition

Define the potential self distance ($PSD$) function as a function of a game's path to invincibility for a specific player. Suppose $v : \{v(1) \rightarrow v(2) \rightarrow \ldots \rightarrow v(m)\}$ is a game's path to invincibility for some player where each $v(i)$ is a game state. The PSD function is then defined:

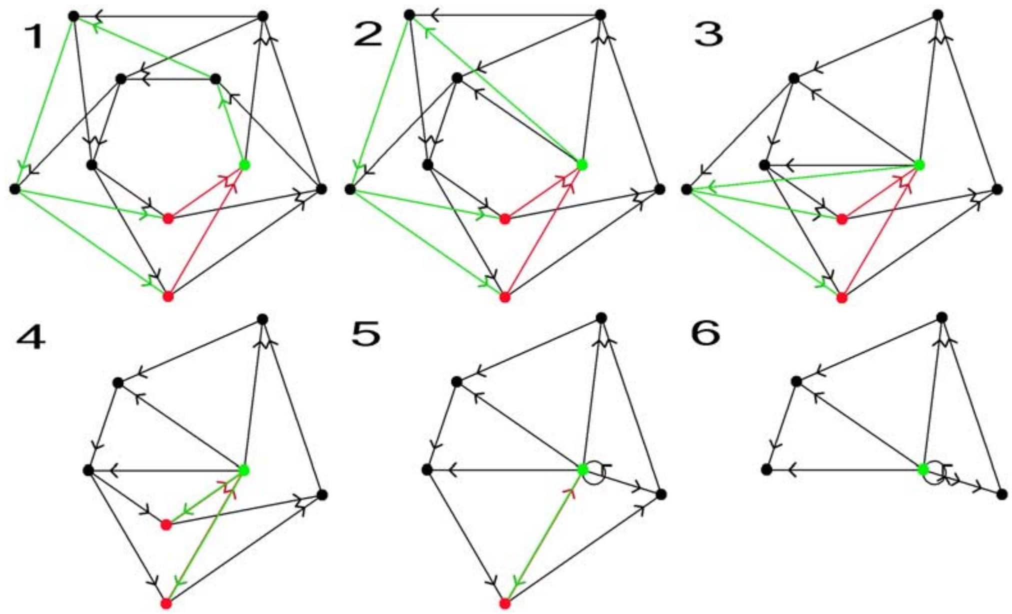
$$PSD(v) = \sum_{i=1}^{m} wt(v(i))$$

The weight function $wt()$ is a function of a game state that returns the number of terminating edges on the specific vertex of the player that is becoming invincible (it disregards self targets, or the edges that originate on that vertex).

Self distance for a specific player can finally be defined as the minimum number that the $PSD$ function assumes when taken over all kill paths to invincibility for that specific player.

### 5.1.2 An Example

The Paranoia game $C_{10}^{1,6}$ is below. We will find the self distance number for this game (remember that self distance is a property of the game when the game is vertex transitive). Call the game's path to invincibility for the green player below $v$, and notice that the red players compose the attacker set of green.




1. $v(1)$: The initial state and the beginning of this specific game path to invincibility for green. $wt(v(1)) = 2$ because of the two terminating edges on green (red edges).

2. $v(2)$: Green makes 1 kill, and $wt(v(2)) = 2$ because there are still two edges terminating on green.

3. $v(3)$: Another kill and $wt(v(3)) = 2$.

4. $v(4)$: Another kill and $wt(v(4)) = 2$.

5. $v(5)$: This time, green has killed a red player, reducing the number of terminating edges on green by 1 (disregarding the loop on green). Now, $wt(v(5)) = 1$.

6. $v(6)$: At this stage, the end state, green has killed his entire attacker set, there are no more edges terminating on him, and $wt(v(6)) = 0$. In fact, the weight of every end state for every kill path to invincibility will be zero because the player in question is invincible - he has no terminating edges on him.

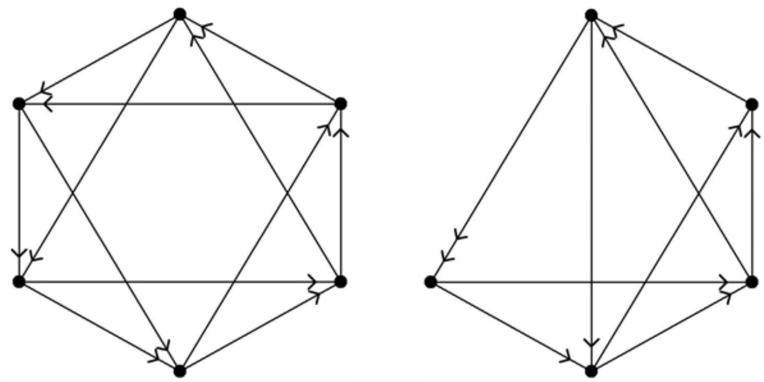
The potential self distance function gives this kill path a value of $2 + 2 + 2 + 2 + 1 + 0 = 9$, and as it turns out, this is the minimum value that the potential self distance function will return when sampled over every game path to invincibility for green. The self distance of this game is 9. This number is completely relative to the self distance numbers for other games. In other words, 9 does not mean much unless compared with another game's self distance number.

The goal is to make this self distance number as large as possible. The best way to do that is obvious: expand the distance between any player and the players that target him.

(Warning: As of now, this number should only be compared between games with the same number of players and assigned targets. Do not compare self distance with duplicate distance. See section 7.1).

## 5.2 Duplicate Distance

Suppose that the top most player in our original Paranoia game $C_6^{1,2}$ (below on the left) kills the player adjacent to him. He acquires the killed's targets which includes one that he already had (below on the right).



Clearly, this situation should be avoided. The top most player does not gain anything by holding duplicate targets; in fact, the player that he targets twice has the advantage: there is one less player targeting him than before.

*Duplicate distance* is an integer used to quantify the distance between a player and duplicate target cards. Duplicate distance is a property of single players, but in vertex transitive games, it is a property of the game since all players are essentially the same.

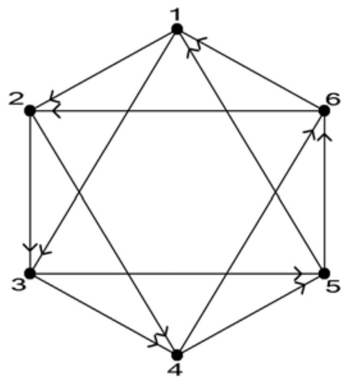
### 5.2.1 Technical Definition

Define $\mathcal{T}(d)$ to be the set of players at a distance between 1 and $d$ from some specified player. Duplicate distance for a player can then be defined simply as the number of unique players in $\mathcal{T}(d)$ for some specified distance $d$.

The distance $d$ should be small enough so that $\mathcal{T}(d)$ does not include every player in the game, but large enough to accurately compare two games. The distance $d$ should usually be $d = \lfloor \frac{p}{c} \rfloor - 1$ where $p$ is the number of players in the game and $c$ is the number of targets that each player is assigned, but it is acceptable if $d$ must be changed to get a more accurate comparison (as long as it is consistent).

### 5.2.2 An Example

The original Paranoia game is numbered below. Consider that, for player 1, $\mathcal{T}(1) = \{2, 3\}$ because players 2 and 3 are at a distance of 1 from player 1.




If we take $d = \lfloor \frac{6}{2} \rfloor - 1 = 2$, the self distance for player 1 (and thus for the entire game) is the number of unique players in $\mathcal{T}(2) = \{2, 3, 4, 5\}$, or 4.

The goal is to make this number as large as possible by separating players from duplicate targets. Similar to self distance, 4 does not mean anything unless it is compared with other games.

(Warning: As of now, this number should only be compared between games with the same number of players and assigned targets. Do not compare duplicate distance with self distance. See section 7.1).

# 6 Applications - Creating and Maintaining a Paranoia Game

This chapter applies the results of this work by guiding the creation and maintenance of a Paranoia game. As a game overseer, read this chapter to gain insight into applying this work to your game.

## 6.1 Creating

Creating a Paranoia game is no small undertaking. As it should be clear by the concerns addressed in this paper, there are several important factors to consider.

### 6.1.1 Participants and Targets

Before a game can be created, the total number of participants must be known. If there is any control over how many people play, always aim for a highly composite number. The more composite the number of players that play, the more non-degenerate games there will be to choose from (more spoke possibilities).

The next decision is much more within the game overseer's control: how many targets should each player begin the game with. In other words, how many target numbers should the game have.

Once the number of players and the number of target per player is known, it is time to begin considering specific games. A Mathematica (version 5.1 only) notebook designed to list all of the games of the form described in section 4.4.2 was created to aid in this step*. When considering the list of games, remember that a high self distance number usually means a low duplicate distance number. Try to find a good balance between the self distance number and the duplicate distance number when choosing a game. Although the two cannot be directly compared, try to choose a game with a relatively high value for each.

* Download - "http://people.unt.edu/dgg0009/Paranoia.zip". Follow the instructions in readme.txt.

### 6.1.2 Kill Mechanisms, Duration, and Location

In order for a game to be playable, specific rules about what constitutes a kill need to be made. Usually, a kill is some harmless action made by one player against another. The specific rules about the kill are best when tailored to the duration and location of the game.

## 6.2 Maintaining

As the game overseer, maintaining the Paranoia game is as important as ensuring the discussed qualities in its creation. There are a few steps that can be taken to ensure that the game will play out according to plan:

- **Kill Log**: A public kill log that lists the players that have been killed and the players that killed them is generally a good idea - players can check to see which of their targets are still alive and some of the more involved players can gain clues as to the game's overall structure.
- **Game Structure**: It is important to keep the form of the game hidden from players. As a player, part of the game is figuring out who targets you and how to kill them by theorizing about the structure of the game. If the structure of the game is released, players can learn how to become invincible more quickly, and the strategy of the game will be hurt.
- **No Targets Problem**: Sometimes, as mentioned before, a player can find himself in a position where he has no other players to target. As the game overseer, this problem must be dealt with by giving that player additional targets. Unfortunately, this problem did not get consideration in the work.

# 7 Conclusions and Future Work

## 7.1 The Future

Though an entire summer was spent exploring the multi-faceted game Paranoia, much more work has yet to be done. The items below require additional work:

- **Non-Degeneracy Fixation**: The focus for creating Paranoia games was limited to non-degenerate forms only. The chance that a degenerate game will end with more than one player can be calculated. Perhaps the advantages of using a degenerate game (much larger self and duplicate distances) would not be outweighed by the fact that the game has a very small chance of being able to end with more than one player. As the game overseer, perhaps there are measures that could be taken if it was foreseeable that a game would end with more than one player.

- **Other Digraphs**: Circulant digraphs were used to represent Paranoia games throughout this work, however, there is nothing that suggests that circulant digraphs would be better than any other class of vertex transitive digraphs. There are several other types of vertex transitive digraphs that may serve as acceptable Paranoia games. While vertex transitivity was a key factor in ensuring equivalency among the players, perhaps other properties of graphs would ensure the same.

- **Necessary Condition for Non-Degeneracy**: A sufficient condition for non-degeneracy was one of the main results in this work. It is believed that the sufficient condition is also the necessary condition, although the proof remained elusive.

- **The Stack Form**: There is one other game form that yields non-degenerate games, however, there was much difficulty in tying it in with the most general form already presented. The number of non-degenerate games omitted by excluding this form is minimal.

- **Self and Duplicate Distance**: While happy with the definition and results of self distance, duplicate distance is the most unsatisfying part of this work. Future work will make self and duplicate distances be comparable between games with differing numbers of players and target numbers.

- **The No Targets Problem**: The no target problem is a very real concern for a game overseer maintaining a Paranoia game. Specific instructions on how to handle this problem is most definitely of interest.

- **Software**: A comprehensive software solution would make creating and maintaining a Paranoia game much simpler.

### 7.2 The Scientific Method

This work made use of the scientific method in the following way:

- The idea for this work on Paranoia came from close *observation* and *questioning* of how the game was created and maintained. Questions such as how are targets assigned and why can't the game end with two people led to the establishment of several *problems*: what would be the best way to assign targets and what games can only end with one player.

- *Hypotheses* were made as potential answers to the problems: perhaps non-degenerate games are the games in which the attacker set of some player is also a cycle eliminating set. With the hypotheses formed, an overall direction was given to the research.

- *Predictions* were made after *analyzing* several thousand example game *experiments*. The samples were organized and analyzed, providing insight into the validity of the hypotheses: indeed, in all of the non-degenerate sample games found, the attacker set of some player was also a cycle eliminating set.

- Collecting and organizing the data and tested hypotheses helped to form a *theory* (although only partially developed in this work). Future work for this project includes additional hypotheses and data gather for different types of problems. Ideally, a unifying theory governing Paranoia will be developed.

### 7.3 Thanks

I'd like to thank Dr. Joseph Kung for his mentoring and for providing valuable insight into the organization of the mess of information that I presented to him. I'd like to thank him for the topics on abstract thinking, graph theory, and a free market society. This work was made possible through his guidance and assistance.

I'd like to thank Ian Haken not only for the personalized software that made creating and analyzing thousands of Paranoia games possible, but for the promptness and interest with which he with created them. All of the Paranoia games in this work were created using Ian's software.

I'd like to thank my parents for providing room and board at their house for one more summer, but more importantly for their continual support and encouragement.

## 8 Glossary

The terms used in this paper have been concentrated below for convenience.

**Adjacency Matrix**: The *adjacency matrix* for a directed graph is a vertices by vertices, binary matrix in which a 1 in the $a, b$ position represents an edge originating from $a$ and terminating on $b$ (in an undirected graph, the edge simply connects vertices $a$ and $b$).
**Attacker Set**: The *attacker set* of a player is the set of players that targets him.
**CES**: *See Cycle Eliminating Set.*
**Circulant Digraph**: A *circulant digraph* is a digraph constructed by connecting vertices with edges based on adjacency and a set of circulant numbers ($\mathrm{C}_p^{x,y,z}$ is a circulant digraph with $p$ vertices and circulant numbers $\{x, y, z\}$).
**Cycle**: A *cycle* is a path in which the first and last vertices are the same.
**Cycle Eliminating Set**: A *cycle eliminating set* is a set of players that, when removed with every edge that terminate or originate on any vertex in the set, eliminates all spoke cycles in the game.
**D**: *See Degenerate.*
**Degenerate**: A Paranoia game is *degenerate* if it is possible for two or more players to be alive in the game's end state.
**Digraph**: *See Directed Graph.*
**Directed Graph**: A *directed graph* is a graph with directed edges.
**Disjoint**: Two sets are *disjoint* if they share no common elements.
**Distance**: The *distance* between two vertices is the number of edges that are traversed in a path from one vertex to the other. If there are separate paths from one vertex to another, there may be different distances between the two.
**Duplicate Distance**: The *duplicate distance* of a player is an integer used to quantify that players distance from duplicate targets.
**Edge**: In a graph, *edges* connect vertices.
**End State**: A Paranoia game in which all possible kills have been made is in its *end state*. A Paranoia game can have only one end state, though which one depends on the order of the kills in the game.
**Fork**: The shape of a path to invincibility for some player resembles a *fork* with a handle and multiple prongs.
**Game Path**: A Paranoia game can take a *game path* through its game state tree. An entire Paranoia game will take only one path through its game state tree, beginning on its initial state (the root node) and ending on one of the end states (a leaf node).
**Game State**: A Paranoia games *game state* refers to the current number of players in the game and the target arrangement that connects them.
**Game State Tree**: The *game state tree* for a Paranoia game is a tree in which the vertices represent game state (not necessarily distinct) and the edges represent potential kills.
**Graph**: A *graph* is composed of vertices and a collection of edges that connects the

vertices.
**Initial State**: A Paranoia game in which no kills have been made is in its *initial state*. A Paranoia game has only one possible initial state.
**Intermediate State**: A Paranoia game that is not in its initial state or end state is in an *intermediate state*. A Paranoia game can have several intermediate states.
**Invincible**: A player is said to be *invincible* if no player targets him.
**Kill**: A player may *kill* any player that he targets. Permission, consignation, and elimination are the general rules governing the kill.
**Kill Path**: A path in a Paranoia game known as a *kill path* is a series of kills that transitions the game from one game state to another.
**Leaf Node**: The vertices in a tree that have no originating edges are known as *leaf nodes*.
**ND**: *See Non-Degenerate*
**Non-Degenerate**: A Paranoia game is *non-degenerate* if it is not possible for two or more players to be alive in the game's end state.
**Orient**: An undirected graph can be *oriented* by changing every edge into a directed edge.
**Paranoia**: *Paranoia* is a game composed of players and targets between the players in which kills are made to eliminate players from the game and the last surviving player(s) wins.
**Path**: A *path* in a graph is a series of distinct, consecutive vertices and edges (it is a path, more specifically a cycle, if the first and last vertices are the same).
**Player**: A *player* in a Paranoia game is a participant in the game who is bound by the rules of the game.
**Root Node**: The vertex in a tree that has no terminating edges is known as the *root node*.
**Self Distance**: The *self distance* of a player is an integer used to quantify that player's distance from the players that target him.
**Spoke**: A *spoke* is a group of players that are modularly equivalent in a Paranoia game that has been rearranged to aid in the games underlying structure.
**Spoke Cycle**: A *spoke cycle* is a cycle in a Paranoia game that visits more than one spoke.
**Target Numbers**: In a circulant Paranoia game, *target numbers* are the circulant numbers used to construct the circulant digraph to represent the Paranoia game.
**Target Set**: The *target set* of a player is the set of players that he targets.
**Tree**: A *tree* is a graph with no cycles.
**Vertex**: In a graph, *vertices* are connected by edges.
**Vertex Transitive**: A graph or directed graph is *vertex transitive* if a particular vertex is undistinguishable from another when considering its role in the overall structure of the graph.

# 9 References

Although neither of these books were cited in the work, they were useful in developing a working knowledge of graph theory:


- West, Douglass B. Introduction to Graph Theory. 2nd ed. Upper Saddle River: Prentice Hall, 2001.
- Wilson, Robin J. Introduction to Graph Theory. 4th ed. Harlow: Addison Wesley, 1996.

# Pictures