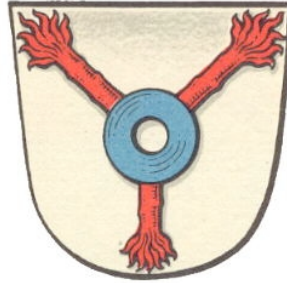## Jäger: The Hunt for Crazy Gert's Gold!

1. Background

In the late 1800s, the crazy Baron Gert Lotz von Wallroth, a wealthy bachelor of impeccable taste, lived on an eight-hectare estate. The townspeople of Wallroth called him "crazy" because of his obsession with cats. In fact, it was rumored he had no less than 500 cats on his estate! The Baron was somewhat of a recluse, and so not much is known about how wealthy he really was, nor exactly how many cats he owned.

As the years passed, Baron Lotz became obsessed with what would happen to his wealth after his passing into the next world. Having neither heir, nor relatives (that he much cared for, anyway), and even fewer friends, he decided he would hide his riches from the rest of the world. It is rumored that he constructed a labyrinth on his property in which to hide his gold and other riches. Furthermore, it is said that, shortly before his death, the Baron sealed his cats inside the labyrinth to "protect" his riches. Now, more than 100 years hence, the existence and exact location of the Baron's fabled labyrinth have still not been verified.

2. The Game

Having just graduated from college, and with little else to occupy his time, our hero, Hedi, has decided to try to solve the mystery of Crazy Gert's Gold. He will either get rich from the finds he makes, or die trying!

Hedi first heard about Crazy Gert while doing his IQP at the Project Center in Frankfurt, Germany. While there, he worked at a financial institution (N.B. Frankfurt is the banking center of Europe) on a project to design a multilingual user interface for clients from various European countries. This is not really important though.

3. The Assignment

You will write code to provide a first-person experience for our hero, Hedi, to explore Crazy Gert's labyrinth. Armed only with a flashlight (isn't that the way it *always* is in horror films?), Hedi must navigate through the dark maze of passages, collecting **gold pieces**. The goal is for Hedi to collect as many gold pieces as he can, and then to exit the labyrinth.

4. Jellybeans

Hedi loves **jellybeans**, and carries an unlimited supply with him. As the labyrinth can get quite confusing, Hedi has decided to drop jellybeans in order to help him remember his way. He only has red, green, and blue jellybeans, and uses one color to help him find his way back out of the labyrinth, and another to mark areas where he has already been (*e.g.*, dead ends). Then, when he walks into a new part of the labyrinth, if there is a jellybean there, he knows he has been there before. When it is time to leave the labyrinth, he just needs to follow the trail of appropriately colored jellybeans. Only one jellybean can occupy a cell of the labyrinth at any given time. If a jellybean is dropped in a cell that already contains a jellybean, it replaces the one that is already there. To remove a jellybean from a cell, Hedi must either replace it with another, or eat the one off the ground (YUCK!).

5. Goolies

Because most of the cats (if there ever were any cats) have been dead for more than a century, Hedi will need to avoid any Goolie remnants of their existence.

Throughout the duration of the game, the Goolies will move about the labyrinth in a pseudo-random manner. The only constraints on Goolie movement are that they can only move one cell at a time, and that no two Goolies can occupy the same cell. Therefore, before making a move, a Goolie will check to see if the cell to move to is occupied by another Goolie. If it is, the Goolie will stay put until the next time it is time for it to decide on a move. Of course, these are spirits, so walls do not get in their way. They cannot, however, leave the labyrinth.

If a Goolie appears in front of Hedi, he must throw three jellybeans in the proper color order for that Goolie. Each Goolie has a three-tuple associated with it, denoting the order that jellybeans must be thrown in order to appease

it. Once appeased, the Goolie can finally rest in peace, and disappears forever. For example, if a Goolie with the triple <R,R,G> appears, Hedi must throw a red jellybean, followed by another red jellybean, followed by a green jellybean, and the order matters. This must be done quickly; otherwise the Goolie steals some pieces of gold from Hedi, and disappears into an adjoining cell of the labyrinth. The gold pieces stolen by the Goolie are magically redistributed to a random cell in the labyrinth. The amount of time Hedi has to throw the jellybeans in the correct order varies from one Goolie to another.

6. Glowing Mushrooms

The labyrinth is a damp place, just perfect for mushrooms to grow! There are two types of mushrooms that grow in the labyrinth, good ones and bad ones. Unfortunately, they look identical. If Hedi comes across a mushroom, he can pick it up, or leave it alone. Good mushrooms repel Goolies, and bad mushrooms (you guessed it!) attract Goolies. If he picks up a good mushroom, any Goolies who choose to enter the cell currently occupied by Hedi will not do so, and any Goolies that are in a cell Hedi enters will flee to a neighboring cell. If Hedi picks up a bad mushroom, however, any Goolies in cells neighboring Hedi's current cell will choose Hedi's cell as their next cell. If the cell is currently occupied by another Goolie, then the Goolie will choose to stay put until the next round. The power of the mushrooms only lasts for a few seconds. If Hedi picks up another mushroom while already holding one, it replaces the first one, and the clock starts again. Once a mushroom is picked up, it cannot be dropped. There can be at most one mushroom per cell.

7. The Labyrinth

The labyrinth consists of a grid layout of M x N cells. The list of possible cells (**Figure 1**) is made up of types of passages that fit together to form a "legal" labyrinth.

Cross	Vertical	Horizontal	SouthWest	SouthEast

NorthWest	NorthEast	NorthTee	EastTee	SouthTee

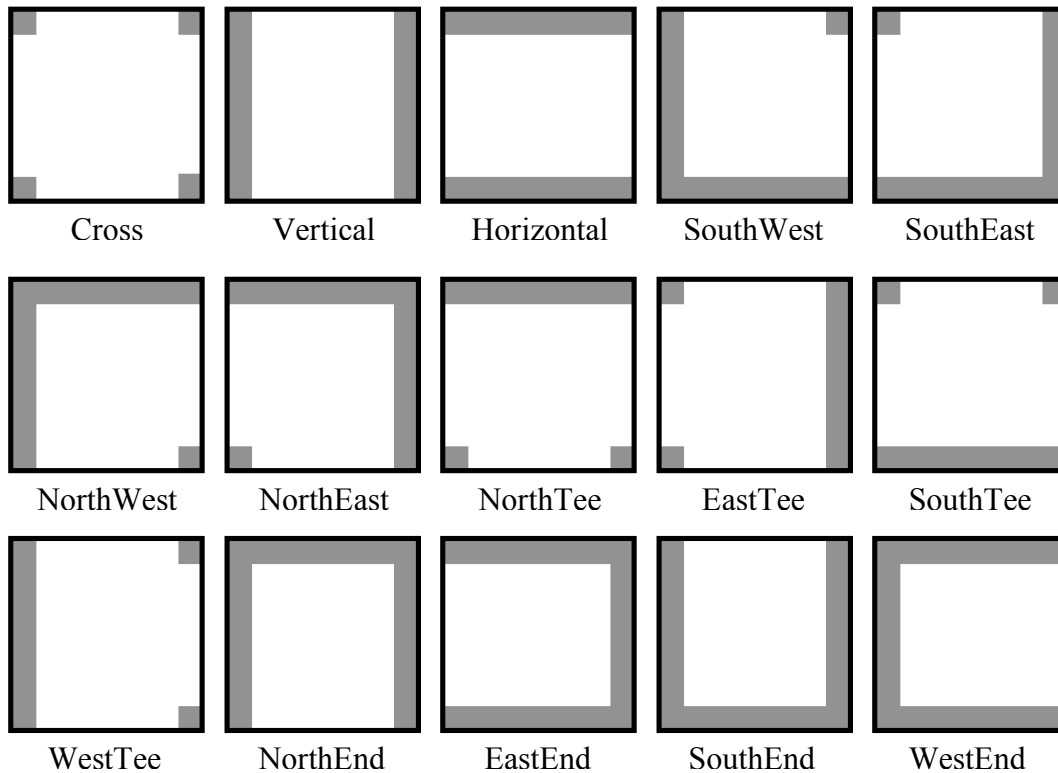| WestTee	NorthEnd	EastEnd	SouthEnd	WestEnd

Figure 1: Legal cells.

Each time a new game is started, a random number seed will be used to randomly generate the layout of the cells. Each cell must be reachable from every other cell. One edge cell is randomly chosen to be the labyrinth entrance/exit.