**Greedy Dwarf Detailed Map and Algorithm Scenario**

Group 9 - Alyssa Mihok, Diana Vera, Ivana Pavlovic and Jazbel Lopez

In the first Scenario, we got the basic function of the game set up: dragon movement, an easy map, a player who can move around, pick up the treasure and escape from the simple level. For this upcoming scenario, we will build upon those features. The dragon will no longer move randomly, the dragon will also be able to move towards the dwarf. The map will be a more complex maze with more treasure on it. Lastly, the dwarf will have some effects from all the treasure they pick up.

**Scenario - New Challenge**

In the last scenario, we planned for the dragon to move towards the player once it’s in range of the dragon. Unfortunately, we overestimated how much we could get done and had to leave this part out for our first scenario. In this Scenario, the dragon will move towards the player when in the range of the dragon and when the player picks up the treasure. While the dwarf is in the range of the dragon, it will move in the direction of the dwarf. If the player is no longer in the sight of the dragon, the dragon will move towards the last known location. After, the dragon will move randomly. If the dragon is able to reach the dwarf then the dragon will deal enough damage to kill the dwarf.

The Map will include new layouts with tiles the player will not be able to step on which form walls and other obstacles like rocks on the Map as seen in **Figure 1**. The Map will load the layout from a set of files allowing the player to access multiple levels when “escaping” from the dragon. The player will also be able to interact with more types of Pickup treasure objects like Gold, Diamonds and Rubies. In order to differentiate the types of treasure to the player there will be images displayed in treasure locations that indicate the type of treasure. All of the new types of treasure will also come with different weight and value variables that will affect the player’s movement speed on the Map.

The different treasures will put on different weights on the dwarf which will reflect on the dwarf's speed. The dwarf will slow down depending on the value of the treasure value picked up. In order to determine the speed the traveling salesman algorithm (TSA) will be implemented. The speed of the dwarf will be determined by the fractions of tiles per keyboard press. For example if the dwarf would pick up too many types of treasurers, the dwarf would move half a tile, instead of moving a tile. The TSA would determine all possible speeds the dwarf could have after collecting different values of treasures on the map.

**Figure 1** - Diagram of the New Map

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **W** | **T** | **W** |  |  | **D** |  |  | **T** | **W** | **T** |  |  |  |
|  | **W** |  | **W** |  |  |  |  |  |  | **W** |  |  | **W** |  |
|  | **W** |  | **W** | **W** |  |  |  |  | **W** | **W** |  |  | **W** |  |
| **R** | **W** |  |  |  |  |  |  |  |  |  |  |  | **W** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | **W** |  |
|  | **W** |  | **W** |  |  |  |  |  |  |  |  | **T** | **W** | **R** |
|  | **W** |  | **W** | **W** | **W** | **W** | **W** | **W** | **W** | **W** | **W** | **W** | **W** |  |
| **T** | **W** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **W** | **W** | **W** | **W** | **W** | **W** |  | **W** | **W** | **W** | **W** | **W** | **W** | **W** | **W** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | **T** |
|  |  |  |  |  |  |  |  | **R** | **W** | **W** | **W** | **W** | **W** |  |
|  |  | **R** |  |  |  | **P** |  |  | **W** |  |  |  |  |  |