Designs 1# and 2#
CS 202: Programming Systems

Alves Silva, Otavio Augusto
PSU ID: 902840168

Design #1 and 2#

The program #1, which will represent a maze game, will have classes to represent the maze construction as well as the players, points, player life and the data structures that will be necessary to create the application.

1. Class player

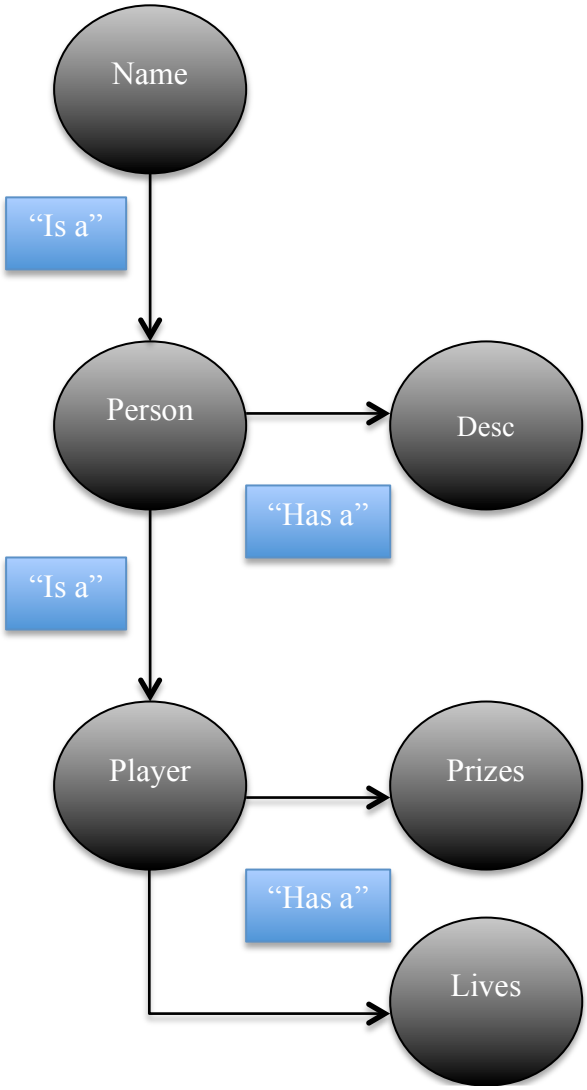
All game needs a player. Therefore, the first class that will be created is the class player. The player needs to have a name, a past prizes won, brief description about him, the amount of lives that the player has and probably the player is a person. So right now, we have five nouns that we have to create for build a support for the player class. The following descriptions of each class explain each responsibility and function of the classes that will be related with the player class.

1. **Description**: This class will hold the background information about the player. It will have the following functions and data members.
    a. Functions
        i. Change description
        ii. Set description
    b. Data members
        i. Description
2. **Name**: This class will hold the player's name. It will have the following functions and data members.
    a. Functions
        i. Change name
        ii. Set name
    b. Data members
        i. First name
        ii. Last name
3. **Person**: This class will represent a person in the real world. We are assuming that class person is derived from the name class and it has a description. It will be done at that way because the class name will be used more frequently than the description class. It will have the following functions and data members.
    a. Functions
        i. Change name
        ii. Change description

b. Data members
    i. Description

4. **Prizes**: This class will hold the amount of points that represent the prizes won by a player. It will have the following functions and data members.
    a. Functions
        i. Increase points
        ii. Decrease points
    b. Data members
        i. Points
5. **Life**: This class will hold the amount of lives that the player still has to continue playing the game.
    a. Functions
        i. Start the amount of lives
        ii. Decrease the amount of lives
    b. Data members
        i. Amount of lives

After set the classes, which have a relationship with the player, we have to explain the relationship between those classes. The following UML diagram can show a brief idea about the design of the player class.



2. Class Maze

The maze will be path, which the player will walk through, created randomly each time that the player starts a new game. It has just one finish line and every other leaf is dead-end. Therefore, it must have a secret (The Code), which will be the randomly builder of the paths, and each path(Node) must have a specific kind (Trap or Prize). So right now, we have four nouns that we have to create for build a support for the maze class. The following descriptions of each class explain each responsibility and function of the classes that will be related with the maze class.

1. **The Code**: This class will have a random algorithm to create and build the first and the second level of the maze. Therefore, it will create a binary tree (Level 1) or the graph (Level 2), where each path can be a trap or a prize. It will have the following functions and data members.
    a. Functions
        i. Create maze
        ii. Set path kind
        iii. Destroy maze
    b. Data Members
        i. Number of paths
2. **Prize**: This class will represent a prize for the player. Therefore, it will give points for the player when the player passes through this path. It will have the following functions and data members.
    a. Functions
        i. Set points of the prize
        ii. Give points for the player
    b. Data Members
        i. Number of points
3. **Trap**: This class will represent a trap for the player. Therefore, it will take off one life of the player when the player passes through this path. It will have the following functions.
    a. Functions
        i. Take life from the user
4. **Node**: This class will represent the paths through the binary tree(Level 1) and graph(Level 2). Each node will have a specific kind. Therefore, it can be a trap or size. Call a function from the prize or trap class depending of The Code algorithm.
    a. Functions
        i. Go to the left
        ii. Go to the right
        iii. Connect to the left
        iv. Connect to the right
        v. If Kind is 1

1. Call trap functions

vi. If Kind if 0

1. Call prize functions

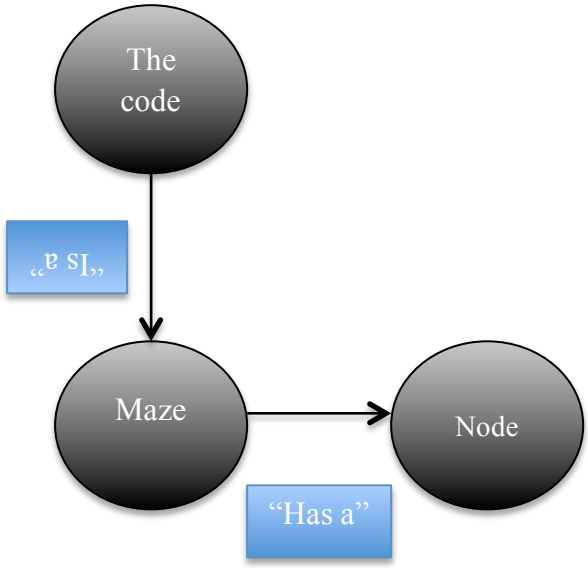
b. Data Members

i. Left

ii. Right

iii. Kind (1 – Trap or 0 – Prize)

After set the classes, we need to explain the relationship between those classes. The following UML diagram can show a brief idea about the design of the maze class.



This design can resolve the assigment, but somethings may change over the develop time.