Philip Warton
CS 162
February 23, 2020

Assignment 4 Design Document

Understanding the Problem:

Summary: In this assignment we are asked to create a program that is a game. There is a grid of points that serves as the game board. There are two types of hazards, with two instances of each, placed somewhere on the board. There is also a gold prize, and the Wumpus monster. Players can move in four directions and fire up to three arrows in any given direction. If the player retrieves the gold and escape, then they win.

The goal of this assignment is to use polymorphism on the event class and its child classes.

Assumptions:

- I am assuming that the player does not press 'ctrl-c' to exit the game
- I am assuming that the player does not input the enter key when prompted for input
- I am assuming that the size of the board does not exceed 10^5 x 10^5
- I am assuming that the size of the board is positive

Design:

(see next page)

Design - Hunt the Wumpus

```
Event Class {
        precept_message;
        name;
        symbol;                    } ~~private~~ protected
        Event();
        void show();
        void precept();            } public
        virtual void encounter() = 0;
}
```

~ Child Classes ~

Wumpus, Bats, Pit, Gold

```
Room Class {
        bool hidden;
        event *event;
        Room(bool, string);
        void print_room();
        void reveal();
}

Point Class {
        x;
        y;
}

Board Class {
        vector<vector<Room>> array;
        int size;
        bool debug_mode;
        void swap_rooms(a, b)
        void allow_turn();
}
```

Testing:

Function	*Case*	*Case Type*	*Outcome*
Board::allow_turn()	Input = " "	Bad	Error, must input direction or shoot an arrow
Board::allow_turn()	Input = "WA"	Edge	Error, must input one direction at a time
Board::allow_turn()	Input = "S"	Good	Move downwards if possible, otherwise nothing happens
Gold::encounter()	Player encounters gold	Good	Changes player has_gold member to true
Pit::encounter()	Player encounters bottomless pit	Bad	Game ends, message prints to console "You died."
Wumpus::encounter(Arrow a)	Player misses Wumpus	Edge	Wumpus lives and moves to an empty random room
Wumpus::encounter(Arrow a)	Player hits Wumpus	Good	Wumpus is removed from game board
Wumpus::encounter(Player p)	Player runs into the Wumpus	Bad	Game Over. Player dies.