### AdventureGame by Shannon Kirby and Doug McNally README ###

This game is a simple (mostly) text based adventure game. At the most rudimentary level, you, the brave adventurer, are on a quest to explore an unfamiliar world. In this world you will encounter various scenarios and items. At the beginning you can consider yourself home, and fundamentally you are tasked with bringing all of the items in this treacherous expanse back to your home location.

The configuration, rooms and items plain text files as well as the map image file should be stored in the running directory of the program. We used no outside coding material in our program.

Some specific features of this game include:

* A Graphical User Interface is wrapping the game so that you can see a visual map of where you are (if available) and controlling your movement through the world as well as viewing the contents of your inventory, your current score, and your current location is streamlined by graphical display and button clicks in place of manual text entry.
* In the GUI we implemented the use of images by putting the map of the rooms on the left hand side to make navigation for the user easier from room to room. In the configuration file, the file name of the map image should appear on the third line. However, the program still behaves normally if no map image file is provided. For optimal behavior the image file should not be larger than 550 x 500 pixels.
* Games in progress can be saved and resumed later by loading them (in the GUI, File>Save or Save As.. and File>Open)
* Two distinct scenarios are available for the game, the first is on Miami University’s beautiful campus with a graphical map included and the second is a haunted mansion, also with an artfully constructed map image. At the beginning you will be prompted for a configuration file and the various scenarios can be access by specifying unique configuration files.
* In playing the scenarios you may randomly encounter an authority figure (for instance a park ranger) that will cause you to lose the game because you have an object in your possession that, in his opinion, you ought not to have taken. If you are caught you will lose the game and the GUI will freeze allowing no further progress in the game. The information for a random encounter is stored in the Items plain text file at the end. To code for a Random Event you must enter RANDOM for a line at the end of your items followed by a line indicating the name of the event, a line for a description of what message is displayed when the event is triggered, the item name that causes the random event to take place when you pick it up and the probability between 0 and 1 that the event will take place each time you move. Make sure to return questionable items home quickly to prevent this from happening!
* If you happen to be a dinosaur aficionado then you’re in luck! The names of various dinosaurs will provide you with some “magical” behavior of the game. Beware of raptors, they are very dangerous killing machines and feel no remorse! If the user inputs “raptor” in the command line, a message will appear warning the player of a dangerous raptor approaching in 5 seconds, the GUI will then freeze for 5 seconds and then display another message that the player has died and their score will show “EPIC FAIL”. If the user inputs “triceratops” a lovely triceratops will transport the user to the highest numbered room on the map. If the user inputs “pterodactyl #” where the # is replaced with an integer in the range of room numbers, the player will be flown to the room with that number.