Peer Evaluation for GridWorld

Your name	Bresia Prudente
ID of submission	710

For each of the following, provide a score on a 1-5 point scale. 1 being very poor, 5 being outstanding. For each give constructive feedback .

Data Decomposition/Organization: How well does the code organize the data structures? Do the members of structs for example have clear logical relationships to each other?

score	5
feedback	Student displays a clear understanding of structs (their functionality, how they work, etc) and properly organizes them in an understandable manner for the reviewer.

Functional Decomposition: Is the code decomposed into logically coherent subroutines/functions? Can the expected behavior of functions be easily understood (or inferred) with minimal examination of the body of the function (i.e., abstraction has been done welll)? Are functions of appropriate length?

score	5
feedback	The functions are of appropriate length. Each function have a decent amount of coding (and does what it should do). This shows that the student clearly understands how and why each function does whatever it's doing.

Naming: Were good names chosen for variables, functions and types? Are there cases where the type of word used (e.g., noun, verb, adjective, etc.) seems to not fit the meaning of the item? Is a plural form used when the singular form makes better sense (or vice-versa)? Are some names too long? Is there some kind of consistency in naming conventions?

score	4
feedback	Student uses very distinct variable names. There were some that might not be clear (such as “int popl” and “struct PEOPLE *peepl”), but overall a majority of the variable names are understandable.

Formatting: Is the code appropriately indented? Is the formatting consistent?

score	5
feedback	Student’s formatting is excellent and has made the code very easy to review.

Comments/Documentation: Are comments used appropriately and actually add to understanding? Are the comments clear ,unambiguous and concise? There is such a thing as over-commenting; does this submission suffer in this way? For example:

```
int a = 7;
a++;   // add one to a
```

score	3
feedback	Student could have used comments describing what certain statements do or what kind of purpose many of the more complex functions do. This way, the reviewer doesn’t have to look through the entire code to search for a certain function that may not work.

Runtime Requirements: To what degree does the code appear to meet the runtime requirements? Your answer may be “code too hard to understand to determine if runtime requirements are met” or similar.

score	5
feedback	The code appears to closely fit the runtime requirements as the functions aren’t very complex in nature and appear to perform correctly.