Assignment Description

The purpose of this assignment is to increase your familiarity with the version control program, git, and the build program, make. In order to accomplish this we are going to develop a suite of command line games together. Each student will create their own game using C and the ncurses library. Git will be used to control and track changes to the package, and make will be used to build the individual games, as well as the game suite. **Note:** When you see `cs202xx`, mentally replace it with your username.

Project Description

The project is composed of the following components:

- A `main.c` file that serves as the entry point to the game suite. This code is responsible for loading all the necessary student modules and allows the user to choose which game to play.
- A `Makefile` for the whole project. Running make in the root directory will archive all of the static libraries and compile the game suite into an executable called `202games`.
- `common.c` and `common.h`. A set of functions that can be used by anyone who wishes to use them in their game. There are only a few functions now, but more may be added.
- A subdirectory for each class account containing a module of code. This is the portion of code that each student will be in charge of changing in order to create their game. Each student module is archived into a static library called `libcs202xx.a`. The reason for using static libraries is to allow each student freedom for structuring their code. The only requirements will be to implement the function `cs202xx_game`, which will serve as the "main" function for each game. Each static library is linked with the main project code during compile time.

Directions

Follow these directions to complete the assignment.

- Navigate to `https://gitlab.indstate.edu/ggolish/202games` and request access to the repository. You will be given 'Developer' status on the project. This means you will be able to checkout the code and make changes to the code; however, you will not be able to push to the main branch. More on this below. **Note:** Your login for gitlab.indstate.edu should be your university username and password.
- Checkout the project from gitlab.indstate.edu using the following command:

```
git clone https://gitlab.indstate.edu/ggolish/202games
```

- Create and checkout a new branch of the repository called `cs202xx` using the following command:

```
git checkout -b cs202xx
```

All of your changes should be made to this branch.

- Edit the module of code in the subdirectory of the project that corresponds to your cs login to create a curses game. The function `cs202xx_game` in the file `cs202xx.c` will serve as the entry point (read "main" function) for your game. You can also change the `name` variable in this file to something more relevant to your game. For instance, if you were making a Pacman clone, you could change it to `static char name[] = "pacman clone (cs202xx)"`. This is the name that will appear for your game in the main menu of the game suite.
- Keep the file `cs202xx.c` as brief as possible. You should create at least one other set of source and header files to contain most of the code for your game. Update the `Makefile` in your subdirectory so that it will generate the static library for your game properly.
- Make incremental changes when building your curses games. When you feel you have reached a checkpoint, you should commit your changes.
- Once your game is complete and all of your changes have been committed to your branch, you can push your branch to the remote repo on gitlab:

  ```
  git push origin cs202xx
  ```

- Navigate to the repo homepage on gitlab and submit a merge request to have your branch merged with the main branch.

Useful Links

- 202games
- The GNU Make manual
- A git tutorial
- Making a merge request on gitlab
- Ncurses howto
- Brief static library explanation