Tools

* IDE of choice: Visual Studio
* Repository: GitHub
  + **What to install**: GitHub for Desktop (this will make your lives 100x easier because now you don’t have to type your commands into the terminal. You can just click buttons to make any commands ☺
    - Where to find it: <https://desktop.github.com/>
  + Now, we’re going to clone the Team’s repository onto our respective desktops.
    - Why do we need to do this? This is important because it allows you to make changes to a file on your computer, and then those changes will eventually sync up to everyone in the group’s computers and we will all get each other’s file updates ☺ Hope this simplification made sense
    - Where do we start? Start by cloning our team’s repo onto your computer:
      * Here is the link to our repo: <https://github.com/teamq20/Software-Design>
      * Here is a quick guide on how to clone: <https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository#cloning-a-repository-to-github-desktop>
    - Now that the repo is cloned, you want to create the files/folder for the game itself. When you open Visual Studio and you’re creating your project, it’s going to ask you to save it somewhere. This part is crucial. Remember that repo you just cloned onto your system? Locate it in your Files and go to the folder that is labeled “Tic-Tac-Toe Game” and save it in there. If you fail to save it in the cloned repo folder, none of your edits will be accessible to GitHub, nor to the team.
  + **What happens when I want to start coding?**
    - We want to keep this project organized. We can do so by using branches. Never make changes in the master branch, I am begging you not to. Always create a new branch for every new feature you’re working on. Working on the main menu? *Create a new branch*. A new day rolls around and you’ve decided to fix some errors on your array? *Create a new branch*. Someone reported a bug and you have to go back in and fix something? *Create a new branch*.
    - Branch naming conventions
      * Try to give your branch a meaningful name. Don’t just name It “Branch 1,” or else we will not necessarily know what it’s about. Try to stick to the following exemplary naming conventions:
        + “Nkengfack\_Main\_Menu”
        + “Nolasco\_Array\_Errors”
        + “Schembri\_Imput\_Validation”
    - How do I know I’m done with that branch?
      * If you created a branch with the intention of using it just to create the Main Menu of the game, then whenever you’ve determined you are done with the Main Menu, you’re done with that branch.
      * Tip: make commits *often*. Even if you’re not done with a branch yet, always save your progress by committing your changes and writing meaningful comments to describe what you did/accomplished. Also, make sure to add a link to your Git commits in the Jira comments of our Project (in the respective issue/task/bug/etc.)
        + Either link your Git commits, or copy down the commit # (ID) and paste it into the comments so that we know what you’re referring to in the Repo
      * Tip: do not ever return to an old branch to make an edit. If you made a branch for the Main Menu and decided you’re done with that branch, do not go back to that branch if you have bug fixes to make on the Main Menu. *Create a new branch* and go from there, because this is a new issue/feature after all.
    - What do I do when I’m done with a branch?
      * Make a pull request
        + Who approves pull requests? (i.e. who is that person that determines that the code you are submitting looks good, has no errors, and can be synced/merged back up into the Master branch?)

Ever will approve Michael’s pull requests, and Michael will approve Ever’s pull requests for this project. Do not approve your *own* pull requests: it’s always a good idea to have a second opinion before anything is merged back into the master branch.

* + **Summary**: Whenever you make changes to a file, be sure to consistently commit, and then push those changes back up to GitHub by creating a pull request so that once it is approved, it can be merged back into the Master branch and everyone will have access to your updated files on their machines as well. If all of this GitHub lingo sounds foreign to you, feel free to text/call me (Chariane) and I can help you out. Also, be sure to check out this website (<https://guides.github.com/activities/hello-world/>); they do a great job of explaining it all. GitHub Desktop also does a great job of simplifying the process of making pull requests, pushing/pulling, and merging things back into master and back to the origin. Here’s a guide from GitHub: <https://docs.github.com/en/desktop/contributing-and-collaborating-using-github-desktop>
  + What happens if you need help with something code-related? For example, you want to implement some open-source code into your project?
    - I (Chariane) will be responsible for providing you with information on anything technical you might need during the project implementation process. Feel free to reach out ☺