Possible flow on control for the startup of the program based on the current class design

**Bold represents classes**

*Italics represents functions*

1. User starts program (aka ./program\_name)
2. initialize **Interface**
3. check if **Server** is available
   1. if not then display error to User via **Interface** and exit program
4. ask user to *login* or *createCharacter* via **Interface**
   1. if *createCharacter*

* ask user to enter name and password via **Interface**
* check with **Server** if name is already in use
  + if already in use display error and ask again for name
* create **CharacterProfile** object based on the input from User
* store **CharacterProfile** on **Server**
* perform *login*
  1. if *login*
* ask User for username and password via **Interface**
* check with **Server** if data is valid
  + if invalid then display error and ask again
* create **CharacterProfile** object based on data from **Server**

1. get available worlds from the **Server**
2. display list of worlds to the User via **Interface**
3. ask User to select a world via **Interface**
   1. if invalid selection then display error and ask again
4. get data from the **Server** for the world selected by the User
5. create a **World** object based on data from **Server**
   1. unsure if User should have a complete world object, or only have data that the User currently needs
6. create **Character** object based on **CharacterProfile**
7. place **Character** in the world
   1. if first time entering world then place in a defined starting spot in the **World**
   2. else place in the same spot as when the User logged out last time they played that character
8. run *GameLoop*
   1. the main part of the game is performed here.