|  |  |
| --- | --- |
| Full Name | Caitlin Burns |
| Student ID | 1903398 |

Final design

In this game, the main character is sent to another universe, and they must go through the game to collect the pieces of the map to make it back to their own world.

* Create a name string and have user input their name
* Print hello name
* Have user input a number 5-10 and set it as an integer for the number of map pieces needed
* Convert map pieces to an integer
* If input is between 5 and 10
  + Output number of map pieces needed from input
* If the input is not between 5 and 10
  + Output invalid input
* Create a float to determine journey length
* Define short and long journey lengths
  + If input is 1.0-5.0
    - Journey length is short
    - If journey is short
      * The user will have 4 days in the game to collect all pieces of the map
    - Otherwise
      * The user will have 8 days in the game to collect all pieces of the map
  + If input is 6.0-10.0
    - Journey length is long
* Prompt user to pick between two character options followed by two pet options depending on character choice
  + If user selects Pirate
    - If journey length is long
      * User will be assigned a parrot as a pet
    - Otherwise
      * user will be assigned a cat as a pet
    - otherwise
      * unknown animal
  + otherwise if user selects fairy
    - if journey length is long
      * user will be assigned a dog as a pet
    - otherwise
      * user will receive a mouse as a pet
    - otherwise
      * unknown animal
* Prompt user to input a number 1-9 to determine what world they will be in
* Set world choice string to an int
  + If input is equal to 1,2, or 3
    - The user’s world is enchanted forest
    - Set map pieces collected equal to zero
  + If input is equal to 4,5, or 6
    - The user’s world is pirate beach
    - Set map pieces collected equal to zero
  + If input is equal to 7,8, or 9
    - The user’s world is fairy castle
    - If user selects fairy castle, prompt user to choose from two different treasure chests
      * If user chooses option 1
        + The user receives no map pieces
        + Set map pieces collected equal to zero
      * Otherwise if user chooses option 2
        + The user receives two map pieces
        + Set variable map pieces collected equal to 2
      * Otherwise
        + Output invalid choice
* Ask user if they want to rest or continue searching
  + If user has less than two map pieces or the journey length is set to short
    - Output that the user can continue searching and ask is user would like to continue searching
    - If yes
      * Output that the user has found one more map piece and add that to the total map pieces collected
      * Output total pieces collected
      * After, tell the user they should rest now
    - Otherwise
      * Output that the user should rest
  + Otherwise
    - Output that the user cannot continue searching and should rest
* Output a statement saying the user found two more map pieces
* Ask user if they want to store map pieces in their bag or their pocket
  + If pocket
    - Output a statement telling the user that one of the map pieces has fallen out, and they have one less
  + Otherwise
    - Map piece total stays the same
* Prompt user to choose from two different treasure chests
  + If user chooses option 2
    - The user receives no map pieces
  + If user chooses option 1
    - The user receives two map pieces
  + Otherwise
    - Output invalid choice
* Ask user if they want to rest or continue searching
  + If user has less than four map pieces or the journey length is set to short
    - Output that the user can continue searching and ask is user would like to continue searching
    - If yes
      * Output that the user has found two more map pieces and add that to the total amount of map pieces collected
      * Output to the total pieces collected
      * Tell user they should rest
    - Otherwise
      * Output that the user should rest
      * Output how many pieces the user has collected
  + Otherwise
    - Output that the user cannot continue searching and should rest
    - Output how many pieces the user has collected
* If not map pieces is less than pieces needed
  + Output you have completed the game
* Otherwise
  + Output game over