In my project design I used the encapsulation to keep in private the attribute

sorted\_word that were random sorting on the class puzzles.

Below is the designer example for my project.

1. Identify the objects in your program.

* *puzzle*
* *player*
* *parachute*
* *director*
* *terminal*

1. Define the responsibility, behaviors and state for each object.

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Responsibility | Behavior’s | State |
| puzzle | Keep on secret the word sorted, and show the letters | -Keep secret sorted\_word | -listwords  -getword  -isfound |
| player | Get the word from player inputted | -Show letterinput | Getwordplayer |
| parachute | Show parachute and remove character when the letter is wrong | -Show parachute | drawingparachute  removeparachute |
| Director | To control the sequency of play | -To instance parachute, puzzle, player class | startgame  showparachute  removecharacter  getpuzzle  do\_output |
| Terminal | To provide input and output operations for the Terminal |  | read\_input\_text  read\_word  writeword  displaymessage |

1. Identify the relationships between your objects.

Director > terminal>parachute

> terminal>player>

> terminal>puzzle>

1. Translate your object designs to class designs.

*Diagram below*

Document your objects and classes so that everyone can refer to them throughout the project. The graphical notation that follows is often the easiest way. It really doesn't matter how you do it though. It just matters that you do.

*Diagram below*

