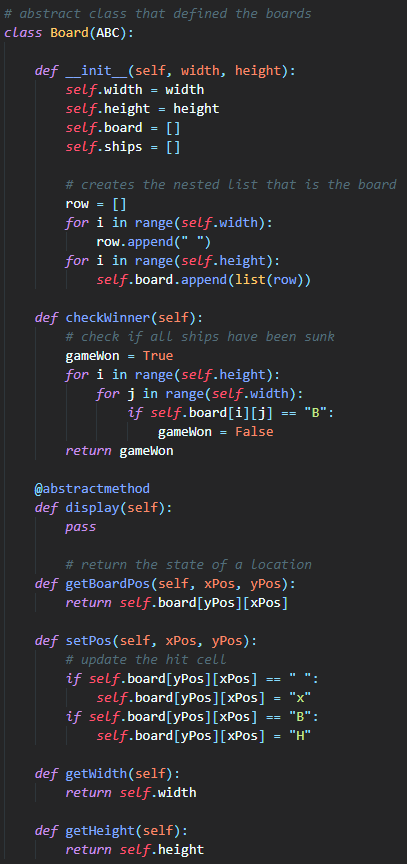
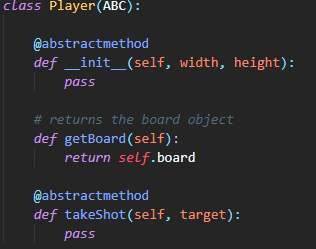
link to example video:

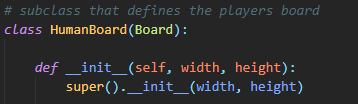
<https://youtu.be/H62YHzVq8dE> - setup of the game

<https://youtu.be/oj6yKoq47jU> - winning the game

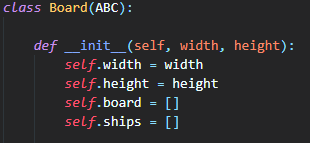
I attempted to make it so that it would display a sunken ship but I ran out of time

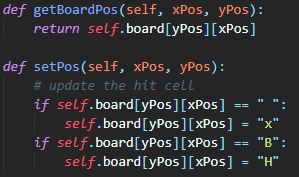
OOP concepts:



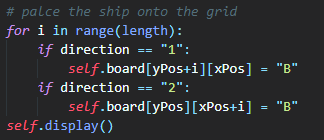
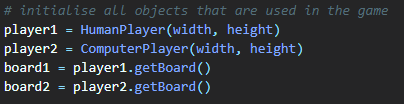
I used abstract classes for both my players and boards. It allows me to have methods that are the same for both the computer and player while still having different methods in the subclasses. It was more useful for the board class as that had more in common between the computer and the player.

To initiate the boards classes the super method was used. This was because some of the initiation was the same however the second half was different as the two classes had different methods of placing the ships.



Simple methods and attributes were used to give the classes memory and functionality. The methods are public. The attributes were also public but used as if they were private because python does not have inbuilt private attributes.

Getters and setters were used to update or retrieve attributes from outside of the class. The getters return a simple value whereas the setters have multiple options to set the cells to depending on their previous condition.



Methods are called from both outside of the classes using their objects but also from within a class calling its self-function. The self-function is also used for all attributes.