|  |
| --- |
| WEAPON |
| (ATTRIBUTES)  +NUMBER OF BULLETS  +FIRE RATE  +WEIGHT  +TYPE OF GUNS |
| (METHOD)  +SET\_WEAPON\_DETAILS |

CLASS 1

ENUM TYPE OF GUN(PRIMARY,SECONDARY)

CLASS 2

|  |
| --- |
| PLAYER |
| +VECTOR<WEAPON> ALL\_WEAPON  WEAPON  +CURRENT\_WEAPON |
| +FIRE(THIS IS A METHOD)  (IN THIS METHOD WE DECREASE THE NUMBER OF BULLETS BY FIRE RATE)  +EUIP\_GUN(METHOD)  (IN THIS ADD GUNS TO VECTOR ALL WEAPON)  +STOP\_FIRING |

CLASS 3

|  |
| --- |
| UI |
| (METHOD)  +CURRENT GUN(METHOD) |
| +TOTAL GUN(METHOD) |
|  |

Above is UML class diagram.

Flowchart: it goes downward.

FOR UI WE CAN CREATE STATIC FUNCTION FOR SHOWING CURRENT AND TOTAL WEAPON DETAILS

WE DO NOT NEED TO CREATE OBJECT OF UI CLASS AS IT IS INTERFACE.

OR

WE CAN MAKE 5 CLASS :

1.WEAPON CLASS.

2.PRIMARY GUN CLASS -> PARENT CLASS🡪WEAPON.

3.SECONDARY GUN -> PARENT CLASS🡪WEAPON.

4.PLAYER CLASS.

5.UI CLASS

PROCESS:

1.CREATE A POINTER OF WEAPON TYPE FOR GUN OBJECT TO BE CREATED.

2.PLAYER CAN CALL PLAYER->FIRE WHICH IN TURN CALL WEAPON->FIRE METHOD.BECAUSE POLYMORPHISM SPECIFIC CLASS FIRE METHOD WILL BE CALLED.

3.UI CLASS OBJECTS WILL NOT BE CREATED IT WILL JUST CALL ITS METHOD AND READ THE WEAPON CLASS AND DERIVED CLASS ATTRIBUTES.

END.