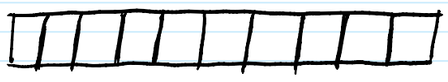
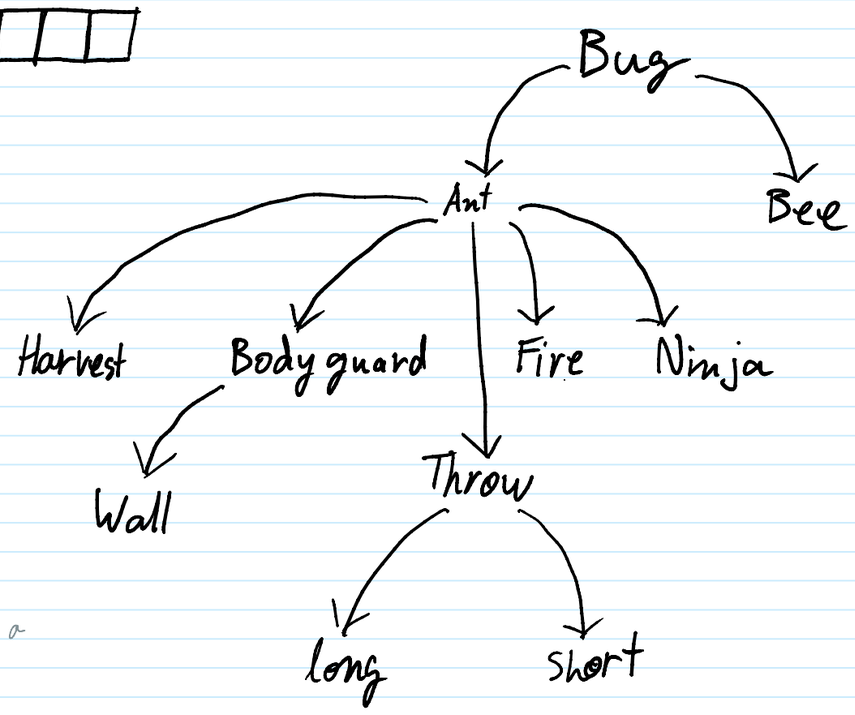
Design 4

Wednesday, May 16, 2018 12:58 PM

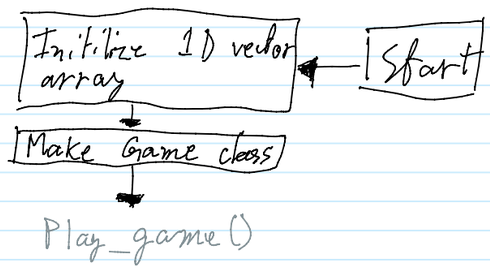


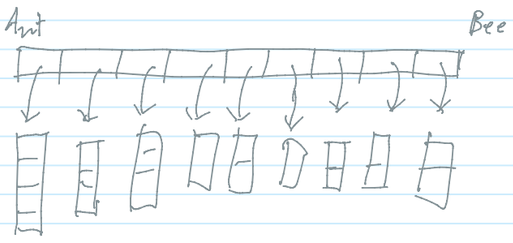
```
class Bug {

    int health // technically Armor
    int location // 0 - 9
    Bool Attack
    int Atack_Range
    Bool Attack_All
    Bool visible

  virtual char Return_typle() {
        return 'B';
    }
};
```

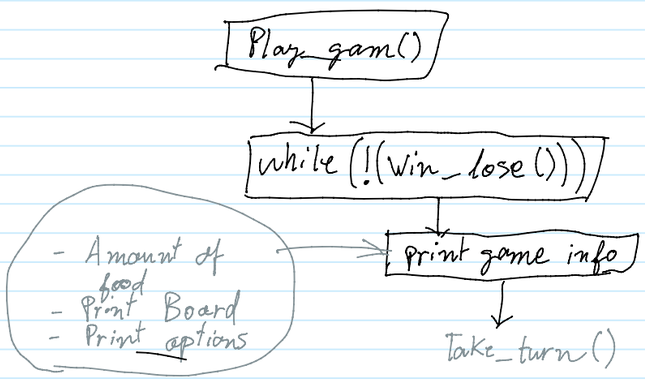



```
Class Ant : Public Bug {

    int cost
    int Produce // Harvest

};

class Harvest : Public Ant {
    Harvest() {
        cost = 2
        Produce = 1
        Visible = True
        Attack = False
        Health = 1;
    }
};

Class Bodyguard : Public Ant {

    Bodyguard() {
        Cost = 4
        Visible = True
        Attack = True
        Attack_Range = 2
        Attack_all = False
        Heath = 2
    };
};

Class wall : Public Bodyguard {

    wall {
        Health = 4
    }

Class Throw : Public Ant {

    Throw() {
        Cost = 4
        Visible = True
        Attack = True
        Attack_Range = 1
        Attack_all = False
        Health = 1
    }
};

class Short_Throw : Public Throw {

    Short_Throw() {
        cost = 3
        visible = True
        Attack = True
        Attack_Range = 3
        Attack_All = False
        Health = 1
    }
};

class Long_Throw : Public Throw {

    Long_Throw() {
        cost = 3
        visible = True
        Attack = True
        Attack_Range = 5
        Attack_all = True
        Health = 1
    }
};
```

Technically a 2D Array

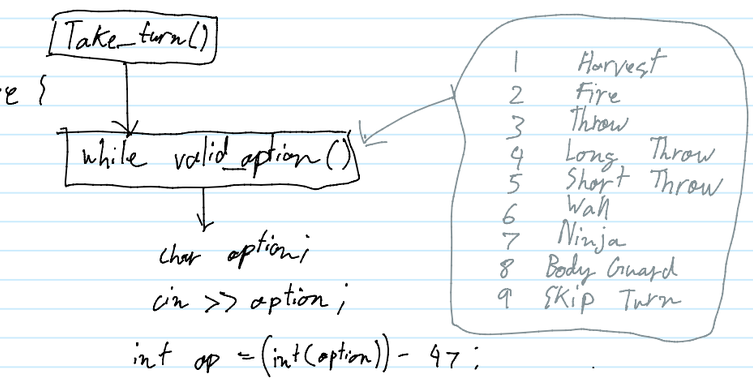






Testing

input	expect
option 0-9	Reprompt if not valid.
Index 1-10	Continue if valid.




```
Class Bee : Public Bee {

    Bee() {
        Heath = 1
        Attack = True
        Attack_Range = 1
        Attack_All = False
        visible = True
    }

Class Fire : Public Ant {

    Fire() {
        cost = 4
        visible = True
        Attack = True
        Attack_Range = 1
        Attack_all = False
        Heath = 1
    }
};

class Ninja : Public Ant {

    Ninja() {
        cost = 6
        visible = False
        Attack = True
        Attack_Range = 1
        Attack_all = False
        Health = 1
    }
};
```

```
char option;
cin >> option;
int op = (int(option)) - 47;
if (op != 9) {
    while (!(check_Valid_Place())) {
        cin >> option;
        int place = (int(option)) - 47;
    }
    Place_ant(op, place)
}
```