```cpp
#include "PetStructure.h"


void Init(TAMA* Pet)
{
    // this sets up all the initial values apart from the name so no issues occur when the code runs
    Pet->Levels[0] = 50;
    Pet->Levels[1] = 50;
    Pet->Levels[3] = 50;
    Pet->HAP = 0;
    Pet->isDead = false;
    Pet->isAsleep = false;
    Pet->Start, Pet->duration = 0;


}
void NamingPet(TAMA* Pet)
{
    //this will allow the user to name the pet if they want to and if not it will be called Mayberry by default.
    char Key;
    cout << "Do you want to name your pet?\n\n Y = Yes     N = No\n\n";

    Key = _getch();
    if (Key == 'y' || Key == 'Y')
    {
        system("CLS");
        cout << "What do you want to call it?\n\n";
        cin >> Pet->Name;
    }
    else
    {
        Pet->Name = "Mayberry";
    }
    system("CLS");
    Pet->Start = 0; // this is the only thing that isn't initialized in 'init' as it puts the timer start to before the pet actually exists
}

void DisplayImage(TAMA* Pet)// this will display art of the pet of whether it is sleeping/passed out or not
{
    switch (Pet->isAsleep | Pet->PassedOut)
    {
    case true:
        cout << "       |\\    _,,,---,,_" << endl;
        cout << "ZZZzz /,`.-'`'    -.  ;-;;,_" << endl;
        cout << "     |,4-  ) )-,_. ,\\ ( `'-'" << endl;
        cout << "    '---''(_/--'  `-'\\_)" << endl;
        break;
    case false:
        cout << " _._     _,-\'\"\"`-._" << endl;
        cout << "(,-.`._,'(      |\\`-/|" << endl;
        cout << "    `-.-' \\ )-`( , o o)" << endl;
        cout << "           `-   \\`_`\"'-\"" << endl;
        break;
    }
}

void DisplayStats(TAMA* Pet,HANDLE Cons)// This will be used to display the current values for the pets stats
{
#pragma region string array and HANDEL


    SetConsoleTextAttribute(Cons, 5);//sets the name to pink/purple
    cout << "Pet Name: " << Pet->Name << endl; // shows the name of the pet

    DisplayImage(Pet);
    string Displays[4]; // this will be used to display the state of the pet
#pragma endregion


#pragma region Hunger String
    /*

        this just sets the value of the string for the Hunger string and the relevant colour for the status

    */

    if (Pet->Levels[0] >= 80)
    {
        SetConsoleTextAttribute(Cons, 2);//green
        Displays[0] = "Well Fed";
    }
    else if (Pet->Levels[0] < 80 && Pet->Levels[0] >= 60)
    {
        SetConsoleTextAttribute(Cons, 3);//aqua

        Displays[0] = "Slightly Peckish";
    }
    else if (Pet->Levels[0] < 60 && Pet->Levels[0] >= 40)
    {
        SetConsoleTextAttribute(Cons, 7);//white

        Displays[0] = "Hungry";
    }
    else if (Pet->Levels[0] < 40 && Pet->Levels[0] >= 20)
    {
        SetConsoleTextAttribute(Cons, 6);//yellow

        Displays[0] = "Rather Hungry";
    }
    else
    {
        SetConsoleTextAttribute(Cons, 4);//red

        Displays[0] = "Starving";
    }

```

```cpp
    cout << "\n\nHunger: " << Displays[0];
#pragma endregion

#pragma region Hydration String
    /*

        this just sets the value of the string for the Hydration string and the relevant colour for the status

    */

    if (Pet->Levels[1] >= 80)
    {
        SetConsoleTextAttribute(Cons, 2);//green

        Displays[1] = "Hydrated";
    }
    else if (Pet->Levels[1] < 80 && Pet->Levels[1] >= 60)
    {
        SetConsoleTextAttribute(Cons, 3);//Aqua

        Displays[1] = "Quenched"; // fairly certain this means pretty much the exact same as hydrated
    }
    else if (Pet->Levels[1] < 60 && Pet->Levels[1] >= 40)
    {
        SetConsoleTextAttribute(Cons, 7);//White

        Displays[1] = "Slightly Parched";
    }
    else if (Pet->Levels[1] < 40 && Pet->Levels[1] >= 20)
    {
        SetConsoleTextAttribute(Cons, 6);//Yellow

        Displays[1] = "Parched";
    }
    else
    {
        SetConsoleTextAttribute(Cons, 4);//Red

        Displays[1] = "De-hydrated";
    }

    cout << "\nHydration: " << Displays[1];
#pragma endregion

#pragma region Happiness String

    /*

        this just sets the value of the string for the Happiness String and the relevant colour to the status

    */

    if (Pet->Levels[2] >= 80)
    {
        SetConsoleTextAttribute(Cons, 2);//Green

        Displays[2] = "Exstatic";
    }
    else if (Pet->Levels[2] < 80 && Pet->Levels[2] >= 60)
    {
        SetConsoleTextAttribute(Cons, 3);//Aqua

        Displays[2] = "Happy";
    }
    else if (Pet->Levels[2] < 60 && Pet->Levels[2] >= 40)
    {
        SetConsoleTextAttribute(Cons, 7);//White

        Displays[2] = "Neutral";
    }
    else if (Pet->Levels[2] < 40 && Pet->Levels[2] >= 20)
    {
        SetConsoleTextAttribute(Cons, 6);//Yellow

        Displays[2] = "Sad";
    }
    else
    {
        SetConsoleTextAttribute(Cons, 4);//Red

        Displays[2] = "Depressed";
    }

    cout << "\nHappiness: " << Displays[2];

    cout << "\nPet meter: " << Pet->HAP << "/10";
#pragma endregion

#pragma region Tiredness String

    /*

    this just sets the value of the string for the Tiredness string and the relevant colour for the status

    */

    if (Pet->Levels[3] >= 80)
    {
        SetConsoleTextAttribute(Cons, 2);//green

        Displays[3] = "Wide Awake";
    }
    else if (Pet->Levels[3] < 80 && Pet->Levels[3] >= 60)
    {
        SetConsoleTextAttribute(Cons, 3);//green

        Displays[3] = "Awake";
    }
```

```cpp
    else if (Pet->Levels[3] < 60 && Pet->Levels[3] >= 40)
    {
        SetConsoleTextAttribute(Cons, 7);//green

        Displays[3] = "Drowzy";
    }
    else if (Pet->Levels[3] < 40 && Pet->Levels[3] >= 20)
    {
        SetConsoleTextAttribute(Cons, 6);//green

        Displays[3] = "Tired";
    }
    else
    {
        SetConsoleTextAttribute(Cons, 4);//green

        Displays[3] = "Falling Asleep";
    }

    cout << "\nTiredness: " << Displays[3] << "\n\n";
#pragma endregion

#pragma region Controls

    //sets the colour for the controls and prints them

    SetConsoleTextAttribute(Cons, 1);//Blue

    cout << "Press f to feed the pet\nPress h to hydrate the pet\nPress p to pet the pet\nPress s to put the pet to sleep\nPress x to Euthanise the pet\n\n";
#pragma endregion
}

void Decrease(TAMA* Pet, int n)// this will pick either the hunger or hydration stat
{
    int Selection = rand() % n - 1;

    switch (Selection)
    {
    case 0:
        Pet->Levels[0] -= rand() % 2 + 0b1; //0b1 is used to get it to stop putting those annoying green lines under it as "doesn't like cast a 4 bit number to an 8 bit cast"


        break;
    case 1:
        Pet->Levels[1] -= rand() % 2 + 0b1;

        break;
    }

}

```

```
void StatCap(TAMA* Pet, int n)//will check to see that none of them has exceeded the max values of 100 and the min values of 0;
{
    for (int i = 0; i < n; i++)
    {
        if (Pet->Levels[i] > 100)
        {
            Pet->Levels[i] = 100;
        }
        if (Pet->Levels[i] < 0)
        {
            Pet->Levels[i] = 0;
        }
    }
    if (Pet->HAP > 10) // caps the HAP stat to 10
    {
        Pet->HAP = 10;
    }
    if (Pet->HAP < 0)
    {
        Pet->HAP = 0;
    }
}
```