

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[System.Serializable]
public class BehaviourBase {

    public string name;
    public Enums.BehaviourTypes behaviourType;
    public bool isValid;
    public int behaviourPriority = 1;

    //Ref
    public CritterBase critterBaseRef;
    public CritterBehaviourController behaviourControllerRef;

    public Vector2 minMax;

    public Vector2 minMaxWaitAfterBehaviour;

    public float behaviourRange;

    //public BehaviourBase behaviourCopy;

    public virtual void InitializeBehaviour(CritterBase critterBase) {
        name = this.GetType().ToString();
        critterBaseRef = critterBase;
        behaviourControllerRef = critterBaseRef.behaviourControllerRef;
        SetPriority();
    }

    //Sets the default priority for behaviours
    void SetPriority() {
        //Na types need to have a custom priority
        if (behaviourType != Enums.BehaviourTypes.Na) { //Else priority is the amount it is set
            if (behaviourType == Enums.BehaviourTypes.Habitat)

```

```

        behaviourPriority = 700;
    else if (behaviourType == Enums.BehaviourTypes.Defend)
        behaviourPriority = 600;
    else if (behaviourType == Enums.BehaviourTypes.PredatorSighted)
        behaviourPriority = 500;
    else if (behaviourType == Enums.BehaviourTypes.Water)
        behaviourPriority = 400;
    else if (behaviourType == Enums.BehaviourTypes.Food)
        behaviourPriority = 300;
    else if (behaviourType == Enums.BehaviourTypes.Reproduction)
        behaviourPriority = 200;
    else if (behaviourType == Enums.BehaviourTypes.Social)
        behaviourPriority = 100;
    else if (behaviourType == Enums.BehaviourTypes.Idle)
        behaviourPriority = 0;
    }
}

public virtual bool CheckIfValid() {
    return false;
}

public virtual void BehaviourUpdate() {
    //If no longer valid set is valid False
}

public virtual void SetBehaviourValid () {
    behaviourControllerRef.SetAfterBehaviourWaitTime(minMaxWaitAfterBehaviour);
    isValid = true;
}
}

```