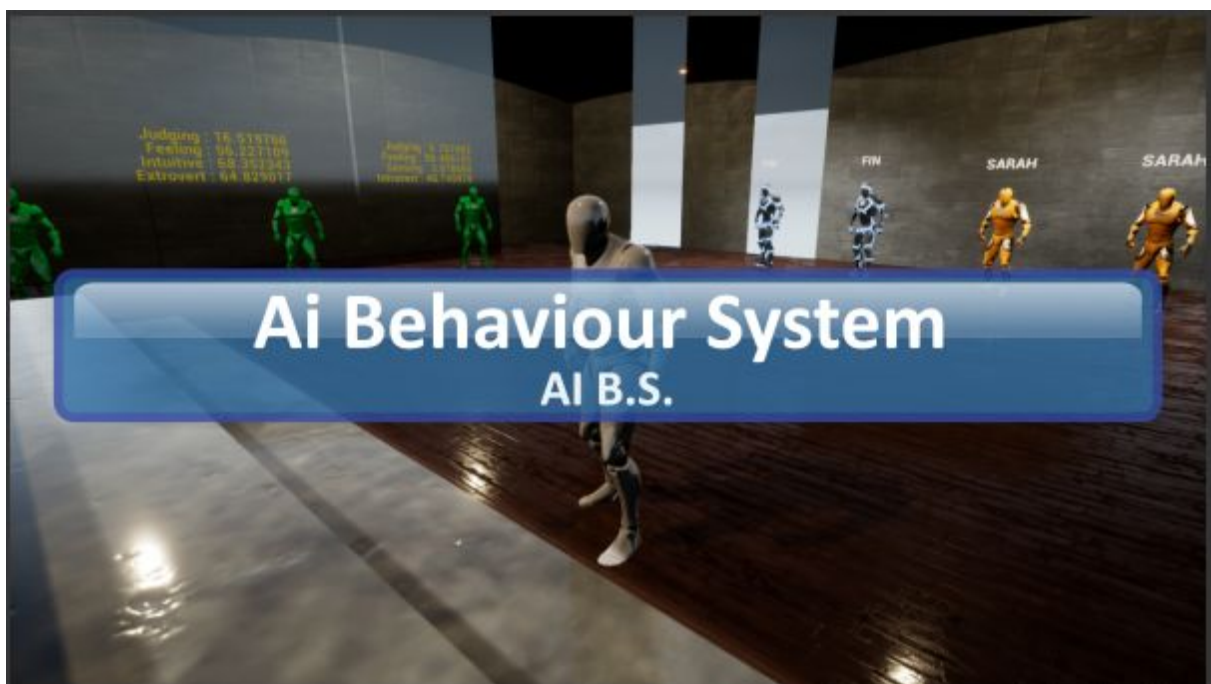


AI Behaviour System



Abstract:

A plugin that will enable easy dynamic ai behaviour creation. It's to help make the gaming worlds in rpg games feel more alive and realistic while allowing for more replayability by applying modifiers based on the Myer Briggs Personalities. This will make ai's behave and do actions differently based on their personality loaded. Take for example a boar in a game, most of the time games will have them all behave the same way, but with this it would be possible to make so they have variations in their actions to the same triggers based on a modifiable factor.

Internal Team Members: Richard De La Merced, Nathaniel Mentis, Dong Hyeong Lee

External Members: Peter Adams, Arielle Vachon

Project Lead/Owner & (Scrum Master): Richard De La Merced

Developers: Richard De La Merced, Nathaniel Mentis, Dong Hyeong Lee

Introduction:

There is yet to be a plugin specifically for making AI behaviours dynamic that revolve around the concept of having AI agents having "personalities." Most RPG games the enemies have a set pattern which can get easy to predict over time leaving little to no replayability when fighting the same enemies over and over again.

We'll be improving on how NPC behave in most games, by adding a plugin that could integrate with a generic AI and just by having multiple instances of said NPC, each instance would behave differently from the one before.

Examples of dynamic behaviours that are similar to what we're trying to accomplish are the "Sims" games; and "Shadow of War : Lord of The Rings" nemesis system.

<https://www.carls-sims-4-guide.com/traits/>
<https://shadowofwar.fandom.com/wiki/Nemesis>

If we have enough time, we could potentially make this a modular plugin allowing for other features to be added on to this, and potentially leaving this open for a modding community to improve.

Background Information

The Myers–Briggs Type Indicator is an introspective self-report questionnaire with the purpose of indicating differing psychological preferences in how people perceive the world around them and make decisions.

~Wikipedia.

Personality Type Distribution for Males

Type	Frequency in Population	
ISTJ	■■■■■■■■■■■■■■■■■■■	16.4%
ESTJ	■■■■■■■■■■■	11.2%
ISTP	■■■■■■■■■	8.5%
ISFJ	■■■■■■■■■	8.1%
ISFP	■■■■■■■■■	7.6%
ESFJ	■■■■■■■■■	7.5%
ESFP	■■■■■■■■■	6.9%
ENFP	■■■■■■■	6.4%
ESTP	■■■■■■■	5.6%
INTP	■■■■■	4.8%
INFP	■■■■	4.1%
ENTP	■■■■	4.0%
INTJ	■■■	3.3%
ENTJ	■■■	2.7%
ENFJ	■■	1.6%
INFJ	■	1.2%

Personality Type Distribution for Females

Type	Frequency in Population	
ISFJ	■■■■■■■■■■■■■■■■■■■	19.4%
ESFJ	■■■■■■■■■■■■■■■■■	16.9%
ESFP	■■■■■■■■■	10.1%
ISFP	■■■■■■■■■	9.9%
ENFP	■■■■■■■■■	9.7%
ISTJ	■■■■■■■	6.9%
ESTJ	■■■■■■■	6.3%
INFP	■■■■■	4.6%
ENFJ	■■■	3.3%
ESTP	■■■	3.0%
ENTP	■■	2.4%
ISTP	■■	2.3%
INTP	■■	1.7%
INFJ	■■	1.6%
INTJ	■	0.9%
ENTJ	■	0.9%

Data source: "MBTI Manual" published by CPP

Each letter in these personality types has its own separate meaning.

I/E: Introversion or **E**xtraversion

The Introversion/Extraversion describes how a person manages their energy.

Introverts recharge by spending quiet time alone or with a small group. Generally tend to be more reserved and thoughtful.

Extraverts recharge by spending time with people and in busy, active surroundings.

Generally tend to be more expressive and outspoken.

S/N: Sensing or iNtuition

The Sensing/Intuition describes how an individual processes information.

Sensors rely on their five senses and care about what they can directly see, hear, feel, and so on. They are often described as "practical."

Intuitives rely on a more abstract level of thinking; such as theories, patterns, and explanations. They are often more concerned with the future than the present and are often described as "creative."

T/F: Thinking or Feeling

The Thinking/Feeling describes how people make decisions.

Thinkers tend to make decisions with their heads; they are interested in finding the most logical, reasonable choice.

Feelers tend to make decisions with their hearts; they are interested in how a decision will affect people, and whether it fits in with their values.

J/P: Judging or Perceiving

The Judging/Perceiving describes how people approach structure in their lives.

Judgers leans towards structure and order; they like things planned, and dislike last-minute changes.

Perceivers appreciate flexibility and spontaneity; they like to leave things open so they can change their minds.

Game Engine Architecture

«Game Specific Subsystems»

«AI»

«Encapsulated Personality Framework»
PersonalityFrameWork

Personality

«Personality List»

«Introverted/Extroverted»
E

«Judging/Perceiving»
JP

«Sensing/Intuition»
SN

«Thinking/Feeling»
TF

Behaviour

Stats

Imported Stats

Generated Stats

Status

Relationship

Affinity

Intimacy

Relationship Status

«Data Structures»

IDStore

«Generic HashTable»
IDHashTable

«Game Specific Subsystems»

«AI»

«Encapsulated Personality Framework»
PersonalityFrameWork

Personality

«Personality List»

«Judging/Perceiving»
JP

«Sensing/Intuition»
SN

«Thinking/Feeling»
TF

Behaviour

Stats

Imported Stats

Generated Stats

Status

Relationship

Affinity

Intimacy

Relationship Status

«Data Structures»

IDStore

«Generic HashTable»
IDHashTable

Example of Implementation

How the AI will Change Based on Personality

Movement/Patrol:

IE -> Speed, Range of Movement, Frequency, Sporadically i.e. [slow] vs [fast]

SN -> Movement Area/Patterns available i.e. [only visible areas, areas less accessible/visible]

TF -> Patterns/Routes chosen i.e. [the same paths] vs [different paths each time]

JP -> Variance in areas and speed i.e. [Small areas] vs [Large areas]

Attack:

IE -> Number of Attacks in a Combo i.e. [1 , 1 , 1] vs [1 , 1 , 1, 1, 1, 1]

SN -> Variance of attack modes i.e. [attack, defend] vs [attack, cautious, defend]

TF -> Frequency of attacks i.e. [1 combo/second] vs [1 combo every 3 seconds]

JP -> Variance of attacks used in combo i.e. [1 , 2, 1] vs [1, 9, 2]

I

Search/Chase/Aggro

IE -> How far they go from origin i.e. [short aggro range] vs [large aggro range]

SN -> How they start chase/search/aggro i.e. [sees player then aggro] vs [random proximity distance from player or areas that players may be in potentially]

TF -> How they search after losing sight i.e. [immediately go back to patrol] vs [search and look longer]

JP -> How cautious they are when searching/chasing/aggro i.e. [slowly search] vs [quickly search]

Skills:

IE -> level of skills vs number of skills

SN -> types of skills i.e. [active] vs [passive]

TF -> which types of skills used i.e. [predictable use of skills] vs [random use of skills]

JP -> time it takes for stage rotation i.e. [one stage] vs [multiple stages] (like boss stage rotation when health changes)

Interact with Interactables/Items Usage (if applicable)

IE -> Hoards items vs Uses items freely

SN -> Item style -> Item equipped[active] vs unequipped[inactive] (drops)

TF -> Item priority (list of items AI can choose from depending on type from SN)

JP -> How often they switch items. [not a lot] vs [frequently switching to match the situation]

```
Instantiating People  
ID: 00000000  
Name: <TemplatePerson>
```

```
Stats:  
Strength: 21  
Dexterity: 21  
Intelligence: 2  
Wisdom: 8  
Charisma: 11
```

```
PersonalityType:  
Introvert: 44.8809  
Intuitive: 79.7325  
Feeling: 71.9149  
Judging: 19.4425
```

```
ID: 00000001  
Name: <TemplatePerson>
```

```
Stats:  
Strength: 7  
Dexterity: 7  
Intelligence: 6  
Wisdom: 1  
Charisma: 18
```

```
PersonalityType:  
Extrovert: 98.7733  
Sensing: 49.9264  
Feeling: 73.2307  
Judging: 20.7362
```

Explanation of Implementation

The PersonalityFrameWork is our wrapper that wraps the data that is used to modify a base AI behaviour. It contains four Personalities, each corresponding to a category from the Myer Brigg's Personality Test (Introvert/Extrovert, Sensing/Intuitive, Feeling/Thinking, Judging/Perceiving). These Personalities are wrapped in a Behaviour to prevent anyone from accessing how these numbers are generated. Along with these Data Structures, there are two complementary supporting Data Structures: Stats, and Relationships.

Personalities

Each of the four Personalities are generated through a random gaussian number generation (a bell curve). Each Personality corresponds to the four traits found in each personality type. Each trait is subdivided into 2 traits, each with opposite ends, this allows us to use a scale value between 0 - 100 with 50 being the middle ground of each opposite trait. For example,

“Introvert” would be valued between 0 - 49, and “Extrovert” would be valued between 50 - 100.

Behaviour

The Personalities would then be wrapped in the behaviour where they would be modified to be able to be a modifiable behaviour package.

Relationships

In real life, behaviour is not just solely based on personality, relationships with others play a factor as well. For instance, if a person knows someone or not will change how their behaviour works. “Relationship” holds these key information that allows the developer to change the behaviour appropriately.

Stats

Stats are used in the event that the developer doesn't want to make stats to link with said Personality. This is a generic random number generator to help support the behaviour in the AI's. The stats are as follows:

- Strength
- Dexterity
- Constitution
- Wisdom
- Intelligence
- Charisma

Each one portrays how well each of the personality traits will manifest in an AI.

Preset Modifiers

We categorized AI behaviour into 3 categories: movement, dialogue, and actions. Using the values generated from the Personalities and Behaviour, we can further categorize these values into two preset modifiers. Using how Myer Brigg's categories are sorted out in the table below we can deduce how it could potentially affect certain aspects in an AI behaviour.

Myers-Briggs		
	Subjective	Objective
Deductive	Intuition/Sensing	Introversion/Extraversion
Inductive	Feeling/Thinking	Perception/Judging

We set the movement modifiers to act as multipliers that will decrease or increase movement variables by a set amount. Dialogues have a weighted value that corresponds to how a dialogue library is weighted on (Refer to Dialogue System in Showcased Implementation.)

Current Implementation V1

Movement System Modifiers

```
//Speeds
float movementSpeed;
float searchSpeed;
float patrolSpeed;
float interceptSpeed;
float reactionSpeed;
//Frequencies
float patrolFrequency;
float interceptFrequency;
float reactionFrequency;
float actionFrequency;
//Distances
float movementDistance;
float searchDistance;
float patrolDistance;
float interceptDistance;
float reactionDistance;
//Variances
float movementVariances;
float searchVariances;
float patrolVariances;
float interceptVariances;
float interactVariances;
```

Dialogue System Design

Dialogue Information Data Structure Template:

Parametres	Data Type	Function Description
Tag	Int/String	used to identify dialogue choice or dialogue, used to help navigate through dialogues
Dialogue File Type	extension	used to link corresponding reader for dialogue
Filepath	string	used for reference to access filetype
Current Stage	int/string	placeholder for in game progress of dialogue, used to help navigate through dialogues
Transition Stage	int/string	placeholder for in game transition for dialogue choices, used to help navigate through dialogues
Transition Stage Tag	int/string	Used to reference next dialogue data structure node, optional parameter
Weight	unsigned int	Used to place a valued placeholder to help determine what type of dialogue choice actor will take

Current Implementation V2

Movement System Modifiers

Currently the base values of the movement values in the behaviour tree are multiplied by the modifiers and then added onto the base value allowing for the personalities

to affect the base value according to the personality.

```
USTRUCT(BlueprintType)
struct FModifier
{
    GENERATED_USTRUCT_BODY()

    //Speeds
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float movementSpeed;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float searchSpeed;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float patrolSpeed;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float interceptSpeed;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float reactionSpeed;
    //Frequencies
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float patrolFrequency;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float interceptFrequency;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float reactionFrequency;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float actionFrequency;
    //Distances
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float movementDistance;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float searchDistance;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float patrolDistance;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float interceptDistance;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float reactionDistance;
    //Variances
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float movementVariances;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float searchVariances;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float patrolVariances;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float interceptVariances;
    UPROPERTY(VisibleAnywhere, BlueprintReadWrite, Category = "Modifiers")
    float interactVariances;
}
```

Dialogue System Design

The Dialogue System was redesigned to allow for linear and non-linear dialogue. With weight's being attached to the dialogue nodes, we can use the personalities to

determine which dialogue type they will adhere to while deciding which dialogue nodes to select.

ID	Tag	Weight	DialogueLine	CurrentStage	TransitionStage	TransitionID	DialogueKey	IsLinear
00000001	Sarah	INFJ	Hey.	00	01	00000033	Hi	True
00000002	Sarah	INFP	H-hey.	00	01	00000034	Hi	True
00000003	Sarah	INTJ	Excuse me.	00	01	00000035	Yes?	True
00000004	Sarah	INTP	E-excuse me.	00	01	00000036	Yes?	True
00000005	Sarah	ISTJ	Hey you.	00	01	00000037	What is it?	True
00000006	Sarah	ISTP	H-hey you.	00	01	00000038	What is it?	True
00000007	Sarah	ISFJ	Hi.	00	01	00000039	Hi	True
00000008	Sarah	ISFP	H-hi.	00	01	00000040	Hello	True
00000009	Sarah	ENFJ	Hi there.	00	01	00000041	Hey	True
00000010	Sarah	ENFP	Sup.	00	01	00000042	Hey	True
00000011	Sarah	ENTJ	Hello.	00	01	00000043	Hey	True
00000012	Sarah	ENTP	Hi!	00	01	00000044	H-hi	True
00000013	Sarah	ESTJ	Yo.	00	01	00000045	Hey	True
00000014	Sarah	ESTP	Hi?	00	01	00000046	...Hi there.	True
00000015	Sarah	ESFJ	Hello there.	00	01	00000047	Hi there	True
00000016	Sarah	ESFP	Hey~?	00	01	00000048	Yes?	True
00000017	Fin	INFJ	Tsk. hey.	00	01	00000049	Uh. hi?	True
00000018	Fin	INFP	... Hey.	00	01	00000050	...Hi	True
00000019	Fin	INTJ	Pardon me.	00	01	00000051	Yes?	True
00000020	Fin	INTP	Greetings.	00	01	00000052	Hello	True
00000021	Fin	ISTJ	You!	00	01	00000053	Hm?	True
00000022	Fin	ISTP	You there!	00	01	00000054	Hm. may I help you?	True
00000023	Fin	ISFJ	Psst.	00	01	00000055	What is it?	True
00000024	Fin	ISFP	Psst. h-hey	00	01	00000056	What is it?	True

Future Implementations

Personalities

We hope to have the personality modules to increase, allowing for other methods of determining personalities and potentially combining other personality methods.

Behaviours

We also hope to have the behaviour system designed to be more optimized.

Stats

In addition the stats that are currently in a work in progress requires implementation, we wish that these stats will be able to be linked with the personality generated.

Relationships

Lastly we hope that the relationships would be dynamically updated as well as pre-generated based on the corresponding factors such as stats, and personalities.

Future Improvements

1. Fuzzy Logic with the way these modifiers are created and decided.
2. Review on algorithms that create modifiers
3. Inclusion of additional Personality Indicators.

4. Balancing of calculations for modifiers.



Resources

https://github.com/Dlee4428/Albs_Demo

<https://github.com/PeteAdams/Albs>