BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

Achieving Unpredictable and Complex AI for a 3D game

Nikola Machálková

Abstract

This work aims to achieve unpredictable and realistic AI of both unplayable and playable characters for a 3D game that is being created as a Bachelor thesis. It is based on already complex AIs used in the gaming industry. To achieve a similar result the work uses finite state machines, decision trees and predictions. However, it also uses these mechanics to elevate the already existing ideas and create a new, innovative way of how game AI works.

xmacha80@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Today, most games with the genre 'relaxing' use a non-playable low-level AI. The usual AI has only a few states with no difference between characters. This can create a boring and repetitive game-play. The idea behind this project's AI is to create complex characters with different metrics influencing reactions and behaviors, so no playthrough is the same. To achieve this, it uses finite state machines and decision trees, and also uses a spectrum of different mechanics, which are new in this genre.

The playable characters uses a system based on the RainWorld's[1] AI Figure 1 – it is using metrics as chaos, karma and hunger to determine a behavior. However, these metrics also determine how different not-player-visible variables increase/decrease and in which states will the character be in. The unplayable characters are much more complex Figure 2 – they have different scripts determining their state based not only on the metric of chaos, but also on their behavior and current mood status and they also include reactions to the playable character.

2. AI of playable character

Playable character, for the most part, is controlled by the player. There are however a few instances where the character behaves on its own. One of these cases is when its hunger goes too low. The second is when its karma is too high – this indicates that the character made a lot of mischievous choices.

2.1 Hunger Mechanic

Character behaves normally until its hunger hits H_{min} and lower. At that moment, the playable character moves on its own to the nearest food, until its hunger is above the threshold again. However, hunger is influenced by different values. The character has belly size, which indicates how much it was starved or fed by the player. Each food that the player can find gives different amount of hunger and some food can also decrease the hunger even lower. Hunger and belly size also influence stamina for running, so if the duck is too full or is starving, the stamina decreases faster. On top of that, the character can build a resistance to the bad food, so it eventually does not decrease hunger. Figure 9

2.2 Mischievous behavior

The character starts performing one of the different mischievous behaviours: dropping the item it is holding, making sounds on its own, not making any sounds and more. This behavior happens every set amount of seconds which is determined by all the player-visible metrics. If the behavior that is randomly picked can't be performed (e.g. dropping an item the character is holding while it is not holding anything), it generates a new action until the chosen behavior can be done. Figure 9

3. AI of non-playable character

The non-playable characters have two finite state machines; one to showcase in which state they are currently in, and second to determine their mood at

the moment. Figure 7 Together with two decision trees (one for distractions and one for movement patterns), they choose how the character will behave.

3.1 States

All states Figure 7 the character can achieve are listed below, together with what they mean:

- **idle** – nothing is happening, the default state
- **talking** – if the character has a quest and the playable character is near
- **distracted** – the playable character caused distraction (sound, throwing item)
- **moving** – the character is changing its location; can be distracted based on the *mood*
- **sleeping** – the character is sleeping; can't be distracted
- **eating** – the character is eating; can't be distracted

3.2 Behaviour

The character can have one of three types of behaviour. These behaviours determine their reaction to a distraction. Figure 6 The behavior has two parts – primary and secondary trait. If the chaos is low, the primary trait is the one used for detection. If the chaos is high, the secondary trait is used.

3.2.1 Low Chaos – primary traits

- **Fearful** – 'snoops' around the area
- **Cheerful** – returns after a few seconds back to original position
- **Aggressive** – follows player around for a few seconds

3.2.2 High Chaos – secondary traits

- **Confused** – only turns to the place of impact
- **Scaredy** – changes position based on the movement pattern
- **Ignorant** – cannot be distracted

3.3 Mood

All the moods the character can achieve and what they mean are listed below. While they do not have their own figure, the most important of them – grumpy – is showcased in Figure 3, 4, 6 and 7:

- **nothing** – default mood
- **grumpy** – cannot be distracted, after certain amount of time (based on chaos) turns to mood **nothing**
- **hungry** – looking for food
- **sleepy** – cannot be distracted

3.4 Hunger

Like a playable character, non-playable ones also have hunger, albeit it is not as complex. They have a hunger which slowly decreases with every frame. When it goes below H_{min}, it might start to eat food near it if it is an appropriate one. If the hunger goes even lower, the character starts looking for the nearest food outside of its area. If there is no food at the moment, the hunger is increased by 1.0, because there is a chance that food will spawn during that time.

3.5 Movement Patterns

Based on the type of the character, it also has a certain movement pattern. Figure 8 There are a few areas on the map where the character can go. The position within the area is randomized, however – certain types of characters do not like each other, so they will avoid going to the same area. And on the opposite spectrum, some types of the characters like to be together. The character also goes to sleep after a certain time of being awake. Figure 7 During the time going to its sleeping place, it cannot be distracted.

3.6 Detection

Based on the behaviour, as stated in the previous section, it changes the reaction of the character to the action. However, all the non-playable characters have a shared mechanism of this detection. The character first determines what kind of the action is happening – the player threw an item Figure 3, there was a sound Figure 4 or the player is holding a 'bad' item Figure 6. If the item hit the ground near the character, it checks if the player is in the blind spot. This blind spot is based on two nodes, FrontPoint and BackPoint Figure 5. If the player's position is outside of this triangle, the character doesn't do anything. If the player is inside, it behaves depending on the behavior. The character can be distracted only in some states and moods.

Acknowledgements

I would like to thank my supervisor Ing. Tomáš Milet for his help, guidance and patience. I would also like to thank all people that helped me with testing, mostly my younger brother, who enjoyed breaking it way too much.

References

[1] Rain World Wiki. Lizards — rain world wiki,, 2024. [Online; accessed 19-April-2024].