Creating an Affective Fighting Game AI System with Gamygdala

Kaori Yuda s1240239 Supervised by Prof. Maxim Mozgovoy

Abstract

We aim to introduce human-like computer-controlled fighters into a popular fighting game engine UFE. This can be done with an emotional appraisal engine Gamygdala that assist creation of virtual characters, affected by emotions. While the goal of a fighting game is simple (to beat an opponent), fighters' behavior is altered with a variety of emotions such as fear, anger or exuberance. The use of Gamygdala in UFE requires us to solve a number of technical issues, such as establishing interface between these engines, and designing a method to apply Gamygdala as an auxiliary technology for fine-tuning the UFE-provided Fuzzy A.I. system. This work is dedicated to the practical implementation of emotional agents in UFE. We show how a relatively simple combinations of tools and technologies can be used to increase player enjoyment and immersion.

1 Introduction

Due to achievements in the development of game AI technology, non-player characters (NPCs) are already good enough to beat most human players. Therefore, we have to think in different direction if we want to improve AI-controlled characters further.

The primary goal of a game system is to entertain the player, thus "good AI" in this context is AI that facilitates fun. In turn, there is evidence that in games like one-vs-one fighting people enjoy playing against AI that behaves like a human [1]. There are different ways to implement believability (human-likeness) of AI behavior. One major trait of human-like behavior is affective (emotion-driven) decision making. If NPCs exhibit emotions during fights, the game should be more interesting for people.

In this paper we will describe how we designed human-like NPCs for Universal Fighting Engine environment [2].

2 Method

2.1 Gamygdala

We enrich NPC behavior with emotions using emotional appraisal engine Gamygdala [3, 4]. NPCs are controlled with an AI system, and of course they can't feel or think. Within Gamygdala paradigm, we set character's goals and annotate game events with relation to these goals, then Gamygdala produces emotional state of the character according to the model of Ortony, Clore and Collins (OCC) [5].

2.2 Interoperability with UFE

Using Gamygdala poses additional challenges for us, since it is written entirely in JavaScript, while UFE is developed with Unity using C# language. We achieved interoperability by employing Jurassic, an open source implementation of JavaScript for the .NET platform [6]. Gamygdala is implemented in a single JavaScript file Gamygdala.js, and our own procedures interfacing Gamygdala (such as agent goals setup and event generation) are stored in an additional file GamygdalaUfe.js. Thanks to Jurassic, we can load, execute and communicate with these files directly from C# code:

```
// a fragment of BattleGUI.cs
public class BattleGUI : UFEScreen {
    protected   ScriptEngine   engine   =   new
ScriptEngine();
    protected virtual void
        OnGameBegin(CharacterInfo player1,
                    CharacterInfo player2,
                    StageOptions stage) {
            BattlePrepare();
        }


    protected void BattlePrepare()
    {
        engine.SetGlobalValue("humanlife",
            (int)player1.totalLife);
        engine.SetGlobalValue("npclife",
            (int)player2.totalLife);
        engine.ExecuteFile("Gamygdala.js");
        engine.ExecuteFile("GamygdalaUfe.js");
    }
    ...
}
```

To make Gamygdala aware of changes in the fighting game world, we translate relevant Unity events into JavaScript code via global functions. For example, Gamygdala must be aware of damage caused or received by the players:

```
// GamygdalaUfe.js implements
// global functions OnHit() and getDamage()

// C# code
protected     virtual     void     OnHit(HitBox
strokeHitBox,
            MoveInfo    move,    CharacterInfo
player) {
    if (player.playerNum == 2) {
        int life1 = (int)player1.Life;
        engine.CallGlobalFunction("OnHit",
life1);
    }
    else if (player.playerNum == 1) {
        int life2 = (int)player2.Life;
        engine.CallGlobalFunction("getDamage",
life2);
    }
}
```

By calling Gamygdala emotional appraisal functionality, we obtain emotional state of the NPC character and store it in a global variable as a stringified JSON object:

```
// GamygdalaUfe.js
emolen = emotionAgent.internalState.length;
for (var i=0;i<emolen; i++) {
    emo[i] =
JSON.stringify(emotionAgent.internalState[i].na
me);
    intensity[i] =
emotionAgent.internalState[i].intensity;
}

emoall = JSON.stringify(emo);
intensityall = JSON.stringify(intensity);
```

On the C# side, we read the global variable and convert it from JSON to a conventional List object:

```
// C# code
int length =
(int)engine.GetGlobalValue("emolen");
string emoall =
(string)engine.GetGlobalValue("emoall");
string intensall =
(string)engine.GetGlobalValue("intensityall");
List<string> emotion =
JsonConvert.DeserializeObject<List<string>>(emo
all);
List<float> intensity =
JsonConvert.DeserializeObject<List<float>>(inte
nsall);
```

2.3 Goals and Beliefs of UFE Agents

In the current implementation of our system, there are only six goals — all associated with the NPC:

1. Win by KO (utility = 1). The agent wins when the opponent's health level reaches zero.
2. Lose by KO (utility = -1). The agent loses when the agent's health level reaches zero.
3. Win by Points (utility = 0.7). The agent wins by points when the round is over, and the agent's health level is higher than the opponent's health level.
4. Lose by Points (utility = -1). The agent loses by points when the round is over, and the agent's health level is lower than the opponent's health level.
5. Keep High Morale (utility = 0.6). The agent's morale is affected by several ad-hoc events.
6. Keep Low Morale (utility = -0.6). This negative goal is handled analogously to the previous one.

Belief name (causal agent) ***Event trigger***	**Goals affected (+/–)**
Caused damage (NPC). *Occurs when NPC hits the opponent, reducing its health level.*	Win by KO (+) Win by Points (+) Lose by Points (–)
Received damage (Opponent). *Occurs when the opponent hits NPC, reducing its health level.*	Lose by KO (+) Lose by Points (+) Win by Points (–)
Spent time winning (Empty). *Occurs every second as long as NPC's health level is higher than the opponent's health level.*	Win by Points (+) Lose by Points (–)
Spent time losing (Empty). *Occurs every second as long as NPC's health level is lower than the opponent's health level.*	Lose by Points (+) Win by Points (–)
About to win by KO (NPC). *Occurs when the opponent's health is very low.*	High Morale (+) Low Morale (–)
About to win by points (NPC).	High Morale (+) Low Morale (–)

Belief name (causal agent) ***Event trigger***	**Goals affected (+/–)**
Occurs when time is running out while the agent has more health than the opponent.	
About to lose by KO (Opponent). *Occurs when the agent has very low health.*	Low Morale (+) High Morale (–)
About to lose by points (Opponent). *Occurs when time is running out while the opponent has more health than the agent.*	Low Morale (+) High Morale (–)
Made 3 successful attacks (NPC). *Three consecutive attacking moves of the agent were successful.*	High Morale (+) Low Morale (–)
Failed to attack 5 times (Opponent). *Five consecutive attacking moves of the agent were unsuccessful.*	Low Morale (+) High Morale (–)
Opponent is evasive (Opponent). *The agent failed to cause any damage for 10 seconds while receiving no damage.*	Low Morale (+) High Morale (–)
Opponent is very resilient (Opponent). *The agent received damage five times consecutively without being able to cause any damage.*	Low Morale (+) High Morale (–)

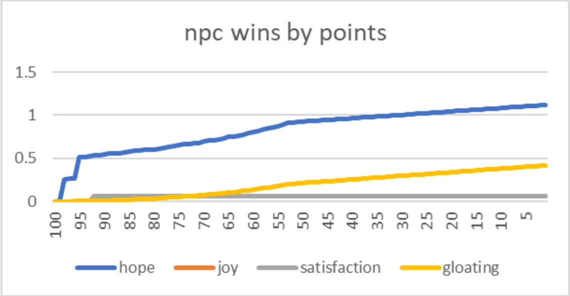
Choosing belief/goal congruence values can be tricky since one has to decide to what extent a certain belief blocks or facilitates a given goal on a scale [-1, +1].

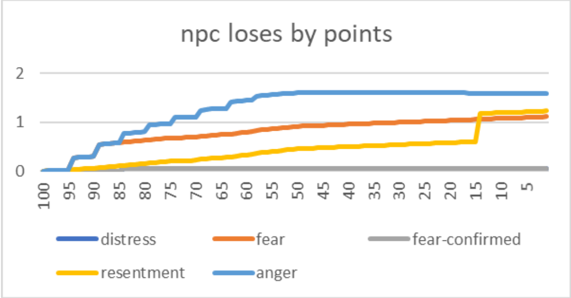
2.4 Evaluation

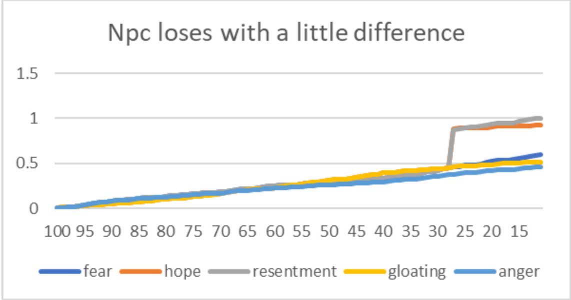
To show the change of emotion output by Gamygdala, graphs are the most obvious way. Since several graphs would be needed for tests, we used node.js, JavaScript runtime environment which let us simulate and store the numbers without manually playing the game. Then, generate graph using Excel out of those numbers.

3 Results

Since we only take into account those affects that relate to the overall goal of winning against a disliked opponent, the generated emotions by Gamygdala is limited to 9 such as hope, joy, satisfaction, gloating, distress, fear, fear-confirmed, resentment and anger. Here are three examples of graphs generated over game:







In the games of “NPC wins/loses by point”, NPC/the opponent’s health level decrease until it will be less than 50 percent and stop reducing health level until game will be time limited. You can see the angle of graphs getting smaller around middle. However, resentment on “NPC loses by point” getting high in the ending. It is because of our belief and goal of Morale is accepted. On “NPC loses with a little difference”, both NPC and the opponent’s health reduced from beginning to end and NPC won with a small difference. This graph is relatively smooth because it is not satisfied our goals of winning or losing but probably moral at the ending again. The meaning of each emotions of Gamygdala are as follows:

Emotion	**Eliciting condition**
hope	a desirable uncertain goal increases in likelihood of success or an undesirable uncertain goal decreases in likelihood of success
fear	an undesirable uncertain goal increase in likelihood of success or a desirable uncertain goal decreases in likelihood of success
joy	a desirable goal succeeds or an undesirable goal fails
distress	an undesirable goal succeeds, or a desirable goal fails
satisfaction	a desirable goal was expected to succeed and it actually does succeed
fearsConfirmed	an undesirable goal was expected to succeed, and it actually does succeed
anger	an undesirable event is caused by another NPC
gloating	an undesirable event happens to a disliked NPC
resentment	a desirable event happens to a disliked NPC

4 Conclusion

In this paper, we show how to connect JavaScript-based emotion engine Gamygdala with a Unity-based Universal Fighting Engine environment. This approach was used to introduce affective behavior into UFE AI system, controlling NPCs. We believe emotions will improve user experience and will make playing against AI characters more fun. Our experience demonstrates that Gamygdala can be integrated quite easily into a Unity project. This result can be of interest to a wider community of game makers, given high popularity of Unity as a game development instrument.

References

[1] I. Umarov, M. Mozgovoy, and P.C.Rogers, "Believable and Effective AI Agents in Virtual Worlds: Current State and Fudture Perspectives," International Journal of Gaming and Computer-Mediated Simulations. 2012. Vol.4, No.2, P.37-59.

[2] Universal Fighting Engine [internet resource] URL: http://www.ufe3d.com

[3] A. Popescu, J. Broekens, and M. van Someren, "Gamygdala: An emotion engine for games," IEEE Transactions on Affective Computing, vol. 5, no. 1, pp. 32–44, 2014.

[4] K. Yuda, M. Mozgovoy, A. Danielewicz-Betz, "Creating an Affective Fighting Game AI System with Gamygdala," Proceedings of the 2019 IEEE Conference on Games, London, UK, 2019, pp. 262-265.

[5] A. Ortony, GL. Clore, and A. Collins, "The Cognitive Structure of Emotions" Cambridge University Press, 1990.

[6] Jurassic [internet resource]
URL: https://github.com/paulbartrum/jurassic