Anakin Breaking Bad

Anirudh Belaguppa Manjunath, Manikantan Srinivasan, Sri Harsha Patallapalli, Tejus Dinesh
CS5100 Foundations of Artificial Intelligence, Khoury College of Computer Sciences

Abstract

The game "ANAKIN BREAKING BAD" is a grid-based game in which a bot, named Anakin, navigates through a NxN environment populated with obstacles, enemies, and two distinct goal states: the Light Side and the Dark Side. Anakin's decision-making is influenced by a set of intrinsic parameters - Trust, Health, and Compassion. Initially, Anakin is oriented towards the Light Side. The primary objective of the player is to strategically guide Anakin toward the Dark Side using assertion statements such that he goes to the dark side while killing the maximum number of enemies possible before his health runs out. These statements can be either affirmative, such as "You are a good person," or negative, like "You are a ruthless person." The simulation concludes when one of the three end states is achieved: Anakin turns toward the dark side, aligns with the light, or perishes.

1. Introduction

In this project, we delve into the creation of a captivating grid-based game centered around Anakin, an autonomous bot navigating through a dynamically evolving environment. The game intricately combines elements of strategy, sentiment analysis, and artificial intelligence to present players with a unique and immersive experience. Anakin's journey unfolds within a maze-like grid, featuring obstacles, enemies, and two divergent goals—the Light Side and the Dark Side. Players are entrusted with the strategic guidance of Anakin, whose decision-making process is influenced by intrinsic parameters like Trust, Health, and Compassion. Trust and health are parameters that decrease at a constant rate as the game progresses. Compassion is modified based on the statements given by the player. What sets this game apart is the incorporation of user-provided assertion statements, allowing players to shape Anakin's moral alignment by manipulating the Compassion parameter. As Anakin progresses, players witness the consequences of their choices, leading to one of three potential end states: alignment with the Light Side, drawn towards the Dark Side, or a perilous demise.

Our project encompasses a multifaceted approach, combining Python programming, game development principles, and artificial intelligence techniques. From grid generation and path-finding algorithms to reinforcement learning with MDP and value iterations, the technical foundation is robust. Integrating sentiment analysis through a sentiment classifier introduces a dynamic element, where user-provided statements directly impact the bot's moral compass. With the Pygame library facilitating graphical interactions and visualization, our game not only challenges players strategically but also immerses them in a narrative-driven environment where choices matter. This report provides an in-depth exploration of our game's architecture, algorithms employed, and the seamless fusion of gaming, AI, and sentiment analysis.

2. Methods

The project encompasses several methods and techniques, integrating elements of game development, artificial intelligence, natural language processing, and reinforcement learning. Here's an overview of the key methods employed.

2.1. Grid Generation:

This sets the stage for the grid-based game, defining the environment's structure, obstacles, goals, and enemies. The grid which is NxN is defined before the start of the game. After the grid is generated, we set Anakin's start state, light state, and dark state. Anakin starts at (N/2,0), the light state is at (N-1, N-1) and the dark state is at (0, N-1). With a small probability of 0.2, obstacles are placed all over the grid. At this juncture, we employ **Breadth First Search (BFS)** to check if the light side and dark side are reachable. If any of the goal states are found unreachable, we restart the whole process of grid generation.

Once a grid is made with all states defined and the reachability of the states is checked, we find the shortest path to the light side using **A*.** We then place enemies (younglings in the context of the game), with a probability of 0.2 all over the grid except on the light state, dark state, start state, obstacles, and the shortest path to the light state. This is to ensure that there is a path to the light side at the beginning of the game for Anakin to follow without killing anyone as killing enemies will modify the compassion parameter and will make Anakin move towards the dark side. Once all of the younglings are set, the grid is ready.

2.2. Bot Navigation and MDP:

Anakin navigates the grid-based game environment using Markov Decision Process (MDP) and Value Iteration approach to determine the optimal strategy for decision-making. MDP is the best-suited approach for a grid-based game. Value Iteration is a method used in reinforcement learning to find the optimal policy for a Markov Decision Process (MDP). An MDP provides a mathematical framework for modeling decision-making situations where outcomes are partly random and partly under the control of a decision-maker. MDPs are characterized by their states, actions, transition model, and reward function. The value function, V(s), for a state s, is updated iteratively using the Bellman Equation as:

$$V_{k+1}(s) = max\sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma V_k s'\right]$$

Here, $V_{k+1}(s)$ is the value of state (s) at iteration (k+1), (max) represents the maximum value over all possible actions (a), (P(s' | s, a)) is the probability of transitioning to state (s') from state (s) by taking action (a), (R(s, a, s')) is the reward received after transitioning from state (s) to state (s'), and γ is the discount factor which balances immediate and future rewards.

The process repeats until the value function converges, meaning the change in the value function between iterations is below a certain threshold. Once convergence is achieved, the optimal policy can be derived from the value function. This policy dictates the best action to take in each state to maximize the expected reward. In our game, the probability of transitioning into a state is always one. Anakin takes based on the optimal value given by value iteration.

2.3. Simulation Environment:

The simulation environment is fully observable and deterministic. It represents a single-agent game where the player's task is to guide Anakin, the bot, through the maze. The agent parameters are as follows

2.3.1. Compassion:

Compassion is a score between 0 to 100. Compassion gets modified whenever the user gives input or Anakin kills an enemy. Compassion is modified as follows:

$$compassion = compassion + \begin{bmatrix} (\,sentiment\ intensity) \times (\,scaling\ factor) \\ \times (\,1 - similarity\ score) \times trust\ factor \end{bmatrix}$$

The scaling factor is a hyperparameter obtained through trial and error. The vector embedding score is a similarity value. The sentiment intensity is a score between -1 to 1, indicating the intensity and the emotion of the statement. Based on this, we modify the existing compassion, either adding to subtracting from it based on the emotion of the sentiment. When Anakin kills an enemy, the compassion gets reduced by 4 times the number of people Anakin has killed.

2.3.2. Trust:

Trust is initially set to 1 and decreases at a constant rate as the game progresses. Trust has a minimum value of 0.5. As the game progresses, the impact of the user input reduces as the trust factor decreases. This is done to increase the difficulty of the game and to make the player come up with more creative ways of manipulating Anakin.

2.3.3. Health:

The health of Anakin initially is 100 and decreases by 4 with every move of the bot. The goal of the player is to guide Anakin to the dark side before his health becomes 0. If Anakin's health becomes 0 before reaching the dark side, the player loses the game.

2.4. Agent Actions:

Anakin, as the primary agent, can perform the following actions within the grid:

ACTION SPACE = {UP, DOWN, LEFT, RIGHT}

2.5. User Interaction:

The user interacts with the simulation by providing input in the form of statements. These statements influence Anakin's internal parameters and consequently his decisions.

2.6. Sentence Similarity:

We have used sentence transformers to limit the user from entering the same statements over and over again. This forces the user to come up with creative statements and makes the game more difficult. This is done by storing the vector embeddings for each input statement. When a new input statement is given, we calculate its vector embedding and try to find the similarity with the existing statements and use them while modifying compassion. If a sentence that is already in the history is given again, there is no modification to the compassion score, effectively discouraging redundancy and encouraging diverse and strategic gameplay

2.7. Sentiment Analysis:

In this project, we have used VADER (Valence Aware Dictionary For Sentiment Reasoning) for sentiment analysis. VADER is a lexicon and rule-based sentiment analysis tool for the analysis of text. We pass the input statements given by the user to the VADER model which gives us a dictionary containing the sentiment scores for the text. The negative sentiment score ('neg'), the neutral sentiment score ('neu'), the positive sentiment score ('pos'), and the overall sentiment score ('compound'). The compound score gives a value between -1 to 1, indicating whether the statement was positive or negative and also the intensity of the statement. VADER not only labels the sentiment behind the text but also gives the intensity of the sentiment, which in our game is used tp modify the agents parameters.

2.8. Game Visualization:

We utilized the Pygame library to construct the user interface (UI) for our game. In this interface, bricks symbolize obstacles, representing unreachable states for the agent. The light goal is embodied by a picture of Yoda, while the dark goal is represented by an image of Darth Sidious. The agent, which is represented by a picture of Anakin with a blue sword, interacts within this framework, while younglings are portrayed as adversaries. A text box facilitates user inputs, and adjacent to it, game statistics are displayed. Upon user input, the data is transmitted to the sentiment analyzer, details of which are previously outlined. Notably, the game stats feature a parameter termed "Dark Pull," calculated as 100 minus the compassion value. The game stats are the number of enemies killed and the game score which is nothing but 10 times the number of enemies killed.

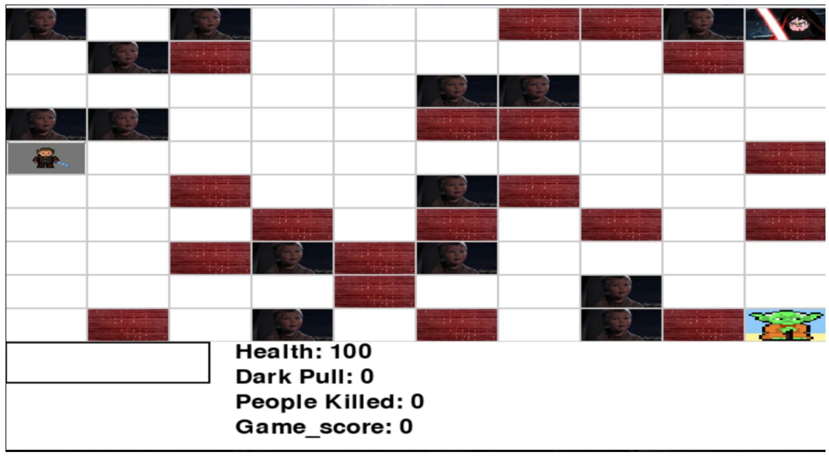

Figure 1. UI screen of the game

2.9. Rewards:

The initial rewards of the game are as follows: dark state reward is -50, light state reward is 50, killing enemies reward is -50, obstacles reward is negative infinity, and the empty cells are -1. The light reward is highest in the beginning as we want Anakin to go towards the light side if the user does not provide any input and the player loses the game. The reward is based on the compassion factor which has a range of 0 to 100.

Items	Initial Rewards
Kill	-50
Light Goal	50
Dark Goal	-50
Empty Cell	-1
Obstacles	-inf

Figure 2. Table for initial rewards

Designing the rewards was challenging because the game had to allow three scenarios where only the light side is rewarding while the dark and killing have negative rewards. The second scenario is where the dark side is more rewarding compared to killing and the light side has negative rewards. The third scenario is where killing enemies gives the highest reward compared to the dark side and Anakin goes on a killing spree across the maze. The following are the reward mechanics:

$$Light\ goal\ rewards = 50 - compassion$$

$$Dark\ goal\ reward = -50 + compassion$$

$$killing\ rewards = \left(\frac{compassion - 75}{-34.5}\right)^{\frac{1}{0.13}}$$

When the game begins, the compassion is set to 100. Until the compassion drops below 50, the reward for killing enemies is a constant -50, after which the killing rewards start increasing. If the compassion factor drops too low, Anakin will perceive that killing enemies is much more rewarding than reaching the dark side and enters a killing spree. The killing reward computation is exponential after compassion drops below 50, which also in turn reduces compassion further.

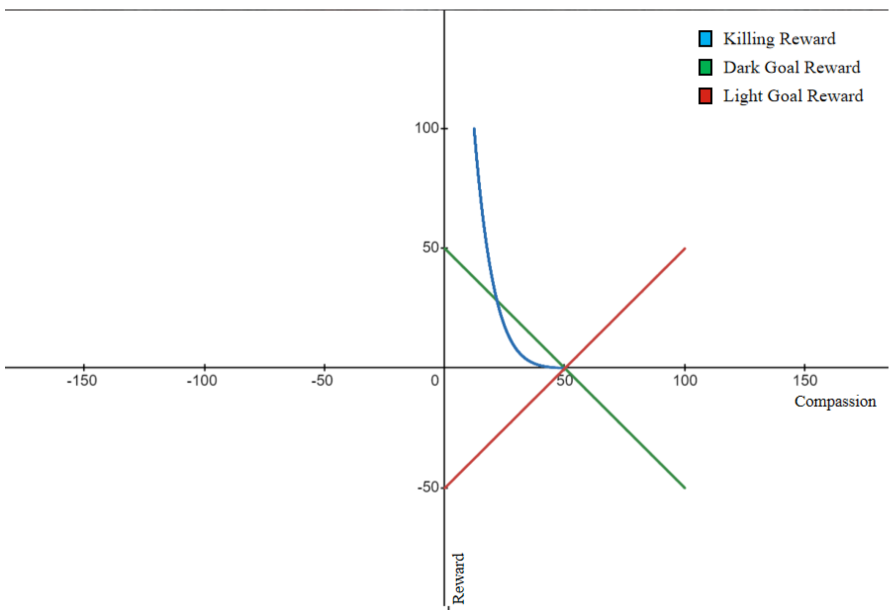

Figure 3. Plot for all reward distributions.

3. Experiments/Results:

3.1 Q-learning v/s Value iteration

During the development, we implemented Q-learning to determine the actions of the agent in the given environment. We observed that Q-learning was computationally intensive and thus inhibited user game experience. A great alternative to tackle this problem was using value iteration instead. Value iteration was efficient and provided the results as expected much faster than Q-learning.

3.2 Changing Rewards

The rewards were highly experimented with using the trial-and-error method. Starting with changing all the rewards linearly, the agent always went to the dark state never going on a killing spree. Thus we had to curve the rewards for killing to make it more desirable to the agent when the compassion is very low.

3.3 Final Results

We have a fully functional game with all the expected features implemented.

4. Conclusion

This project was a great way to strengthen the concepts of Artificial Intelligence such as the Markov Decision Process, Q-learning, Value-iteration, Natural Language Processing, text encoding, and search algorithms. Additionally, this is a great learning opportunity in the field of game development.

5. Team Contributions:

This was a collaborative undertaking characterized by equal contributions from all team members. Our joint efforts in meeting rooms resulted in the collective development and refinement of the project. However, for precision, the following list enumerates members and their major contributions to specific domains

- Game UI - Anirudh & Tejus
- Rewards and Sentiment Analysis - Sriharsha
- Value-iteration & MDP - Manikantan

6. Link to code repository:

https://github.com/Mani676901/ANAKIN_BREAKING_BAD

7. References

1. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

2. Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Eighth International Conference on Weblogs and Social Media* (pp. 216-225).

3. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv preprint arXiv:1908.10084.