# Single player blackjack

S. Arun Karthick U2323073C || Kavya Sethi U2323682C || Garv Sachdev U2323262H

**Introduction:**

In today's rapidly evolving digital world, the integration of artificial intelligence (AI) in our lives has increased exponentially, particularly across various industries and applications. AI has a wide array of real-world applications and has revolutionized many industries due to its immense power. However, in the spectrum of gaming and entertainment, there remains a need for sophisticated AI-driven solutions.

Blackjack, a classic card game beloved by millions worldwide, provides a great opportunity to harness the power of AI in creating a dynamic and immersive gaming experience. While traditional implementations of blackjack rely on pre-programmed algorithms or random number generators to simulate gameplay, AI-driven blackjack takes it to whole a new level of sophistication and realism. Based on advanced machine learning algorithms and predictive analytics, AI can enhance the gameplay of blackjack by providing intelligent decision-making capabilities, adaptive strategies, and personalized experiences tailored to individual players' preferences and skill levels.

Our team has decided to seize the opportunity and create an AI agent which can solve or play Blackjack game through algorithms based on machine learning. We aim to model our agent in a way that it can play a single player blackjack game on our behalf, against another AI agent which is already provided in the game environment

**Type of agent:** The agent we aim to build is a Non-Autonomous, Goal Based, Reflex Agent with State.

* Non-Autonomous: All the necessary information and rules are given to the agent beforehand. Then, the agent will analyze the present state based on the information provided such as, deciding whether to *hit* (asking for another card) or *stand* (decline additional cards and keep current hand).
* Goal based agent: The ultimate goal is to reach a *card-sum of 21* or get *as close as possible to 21*.
* Reflex agent with state: This means that in every state, the agent will take a suitable action based on the current card-sum at every state. The agent moves on the next state every time a card is dealt, and the card-sum is stored in memory.

**Types of environments:**

|  |  |
| --- | --- |
| **Properties** | **Elaboration** |
| Accessible | All the information required (*card-sum*, conditions whether to *hit* or *stand*) is available to the agent to make the appropriate decision. There is no hidden or unknown information which may alter the environment |
| Deterministic | The outcome of the next state is completely determined by the current state (*card-sum* at hand) and the action it takes, deciding whether to *hit* or *stand* |
| Sequential | The environment of the problem is sequential because the next state is influenced by the current state |
| Static | Since the game is single player, the environment does not change while the agent is analyzing or making its decision |
| Discrete | There is a finite number of states to approach the goal. For instance, after the initial 2 cards are dealt, there are only 49 more card or states to come. The number of cards or states decrements as the round goes on |

**Problem formulation**

|  |  |
| --- | --- |
| **Properties** | **Elaboration** |
| Problem Definition | 2 cards will be dealt first. The aim is to achieve a *card-sum* of 21 or to approach 21 as close as possible and also to ensure not to *bust* (to exceed a card-sum of 21).  For e.g. the figure on the left shows that the *card-sum* at hand is 7+5=12.  It would move on to the next state by choosing the 2 following actions:   * *Hit* (Draw another card) * *Stand* (decline additional cards and keep current hand) |
| Initial State | The initial state is being dealt with 2 cards. |
| Action Sets | Analysing the current card-sum and then deciding whether to *hit* or *stand*. |
| Goal Test Predicate | To successfully achieve the goal, the agent should (without time constraint):   * Attain a card-sum of 21. * If not exactly 21, achieve a card-sum as close as possible to 21 when declared *stand*. * The card-sum should not *bust.* |
| Solution | Configure the agent to reach the goal successfully from the initial state by taking actions based on every current state. The agent will try to reach as close as possible to 21 and also ensure that it does not *bust.* |

**References and citations:**

* 24/7 Games LLC .*24.7BLACKJACK*.

https://www.247blackjack.com/