



# RUMMY.BUDDY

## AN A.I. VERSUS HUMAN CARD GAME



### 1. BACKGROUND

- ♣ We asked ourselves: what if Siri can play cards with you and make jokes when you lose?
- ♣ RummyBuddy is a prototype of such idea. It is a player versus A.I. video game application of the **gin rummy** card game.
- ♣ The development focus is on the A.I. which implements the **Monte Carlo** tree search method.
- ♣ Its final version is planned to include a voiced A.I. with a sense of humor and personality.
- ♣ The UX theme is **retrofuturism**.

### 2. MONTE CARLO A.I. METHOD

- ♠ **Monte Carlo** uses tree:
  - ♦ Root is our current game stage
  - ♦ Each child node: a new game stage obtained by making a single move.
- ♠ Instead of building every possible branches, the A.I. will:
  - ♦ Only look at the most **interesting** nodes.
  - ♦ Run **simulations** by making random moves until it reaches end game.
  - ♦ Use the result of the simulation (win or loss) to score the promising value of each node.
- ♠ This process will be repeated until we say stop.
- ♠ The branch with the most number of simulations, as the A.I.'s move.

	A	2	3	4	5	6	7	8	9	10	J	Q	K
H	0		0		0		0				1		
D		1	0				1		0	1		1	
S	3		1			1	3						
C				0		1	0		0	1			3

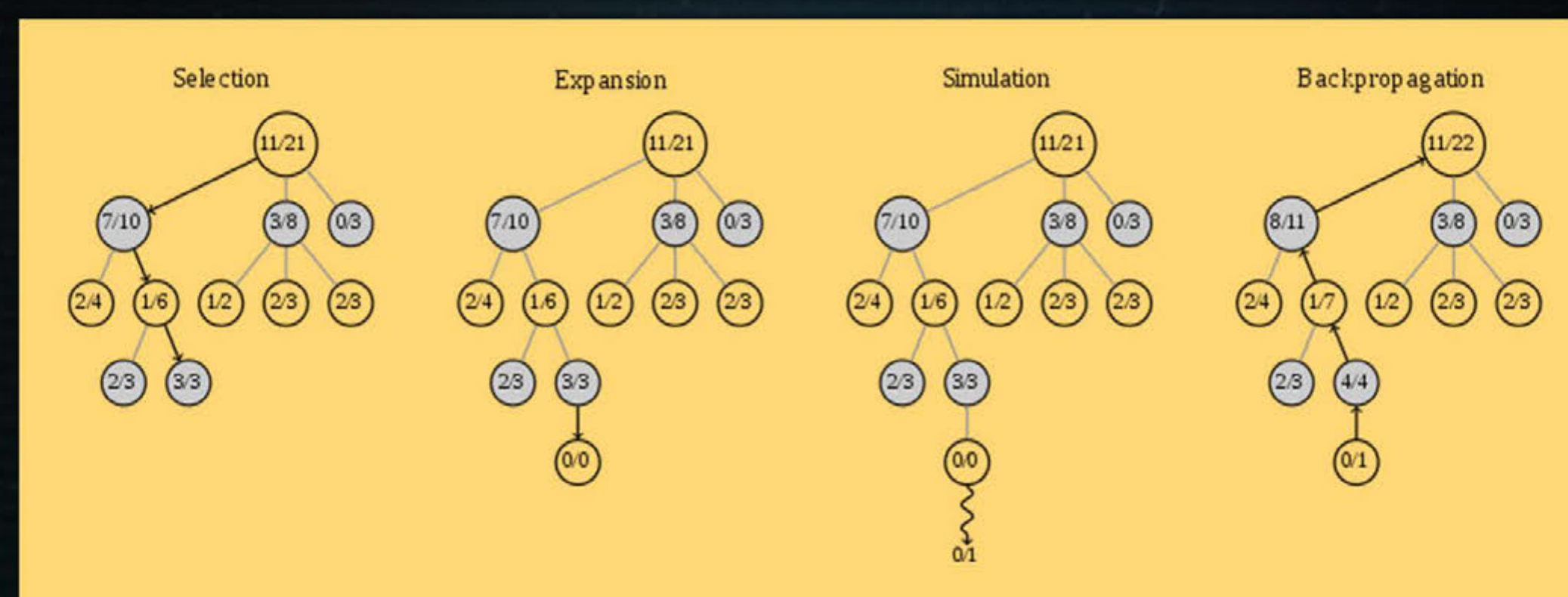
Game state matrix (empty = -1)

### 3. GAME STATE

- ♥ The representation of the world in Rummy Buddy is a **4 x 13 matrix**:
  - ♦ Row: 4 suits in the deck
  - ♦ Column: values of the cards in a suit, A to K.
- ♥ There are four states each card can be in:
  - 1 = unknown, 0 = Player Hand, 1 = AI Hand, 3 Discard Stack
- ♥ This matrix is fed into the Monte Carlo method to run simulations.
- ♥ It is also used to calculate **probabilities** to add another layer of accuracy for the A.I.



RummyBuddy's UI (not final)



Monte Carlo tree

### 4. TOOLS

