AI Tournament v6

AI Tournament v6

Expression

What is an AI Tournament?

- Someone makes a game that can be played by simple AI
- The game is presented
- As a participant you will:
  - Install the game
  - Work from a template to make their own AI
  - Submit your AI to the game master
- At the end of the day the AI's compete in a tournament
- Fun and games, don't take it too seriously even though that can be hard

Todays game

- Scrappy hobby project
    - Pyglet based
    - About 2500 lines of code
    - 0 unit tests

    - Basic steps have been taken to avoid cheating, but it will be possible
    - Instead we rely on good sportsmanship

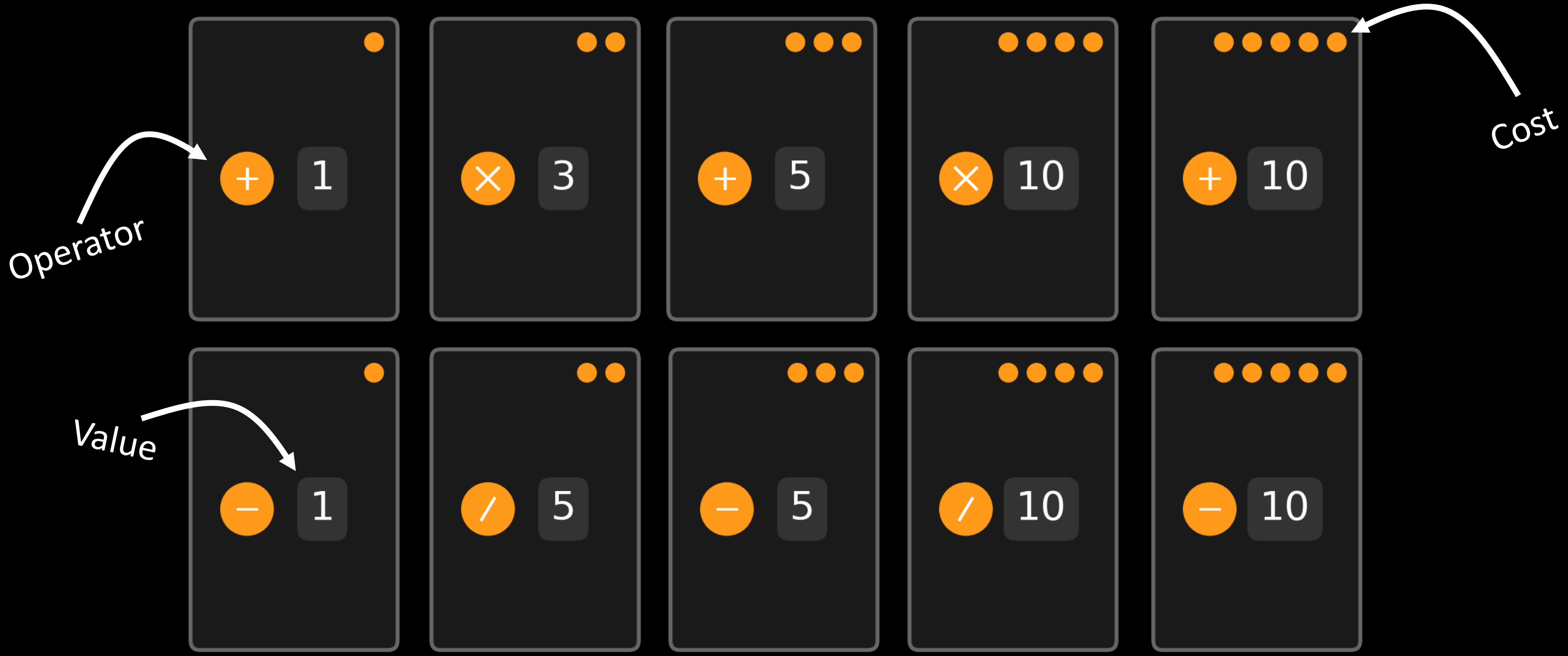
Expression

Expression

- Two players
  - Positive player / Negative player
- Together they build a mathematical expression
- The positive players goal is for it to evaluate to 100 or more
- The negative players goal is for it to evaluate to -100 or less

- The mathematical expression is built with cards
  - Each card has a cost, resources increase throughout the game
  - Each card can add something to the expression

Expression

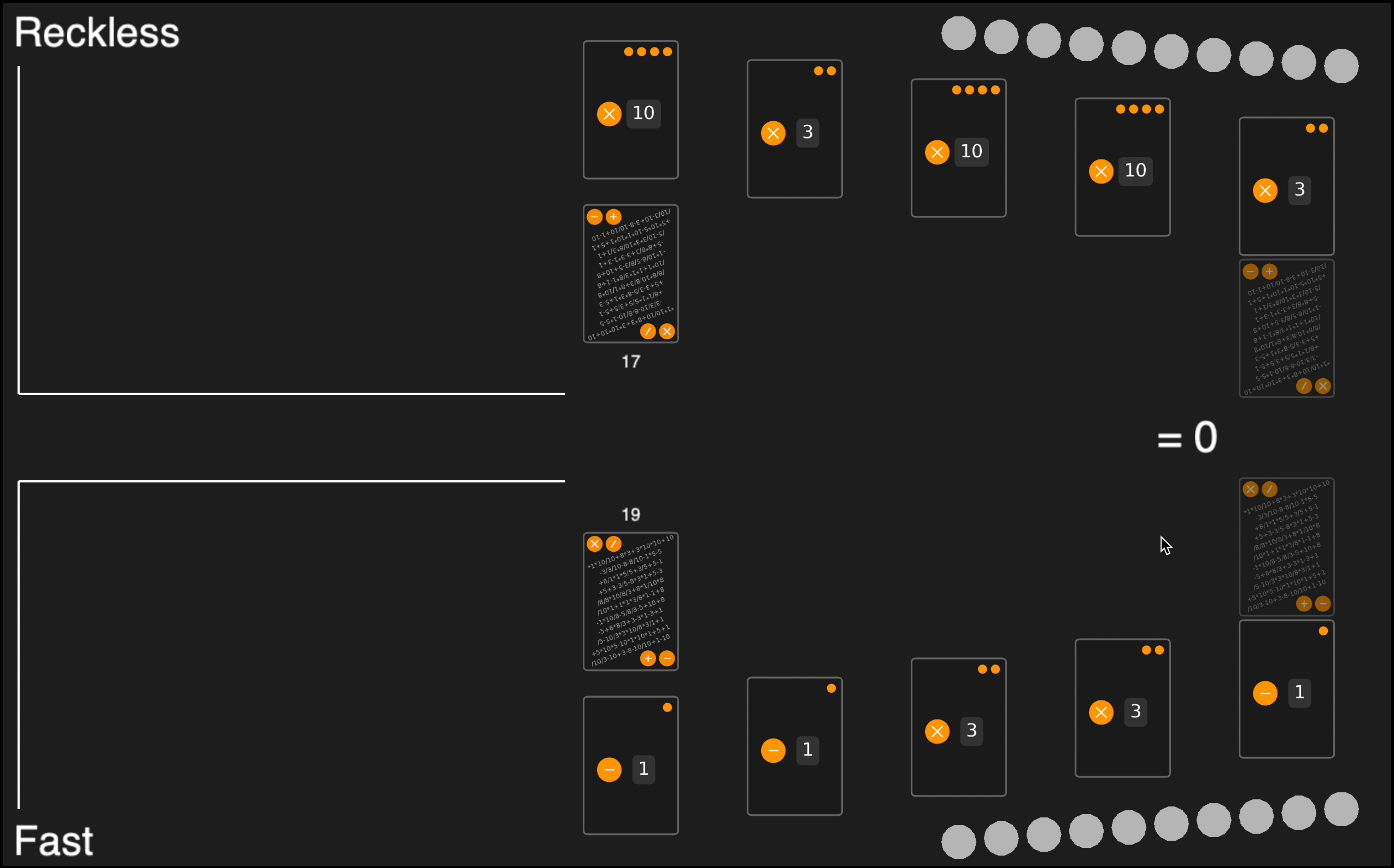


Expression

- Expression starts at 0
- Players take turn playing cards, building the expression
    - Negative player starts
    - Both start with 5 cards
    - Cards costs energy
- Game ends when: abs(value) >= 100
- Energy increases on these turns:

0	1	3	6	8/9	12	15	18	21	24

    - To a maximum of 10 energy
- Agents make their own deck, must have at least 15 cards


Reckless
10
3
10
10
3
17
= 0
19
1
1
3
3
1
Fast

AI interface

- Input:
  - hand: list of Card objects with operator, value and cost
  - energy: integer, the amount of energy available
  - game_info: dictionary with info on current game
  - locked_term: part of expression that can no longer be changed
  - current_term: part that can be changed

2+2*4-5*5+10

Be aware the current expression value is locked_term + current_term

AI interface

- game_info: dict
  - turn_number: integer, current turn for this player
  - n_opponent_cards: integer, number of cards in opponents hand
  - last_played_hand: list of cards, hand opponent played last round

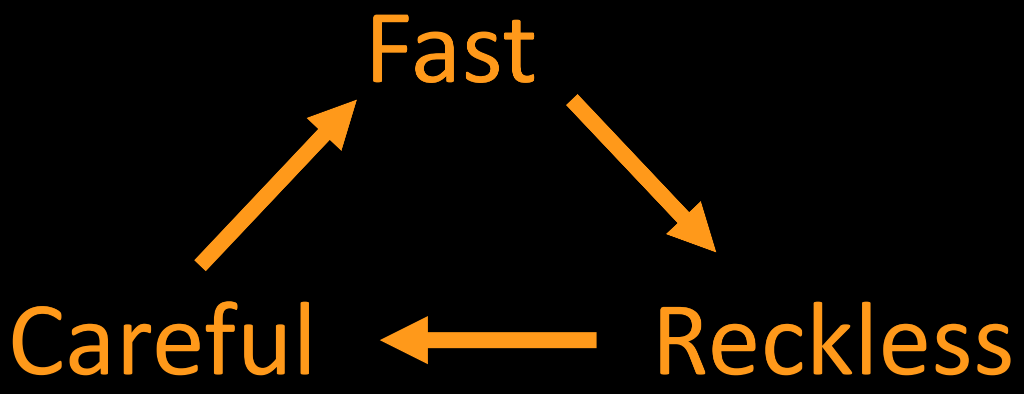
AI interface

- Helpers:
  - apply_card(card, locked_term, current_term):
    - Applies a card, returns new locked_term and current_term
  - can_be_played(hand, energy, expensive_first=True):
    - Returns list of cards you can play from hand with energy budget
    - Sorts before returning, choose either expensive_first or not

- Output:
  - List of card objects (that were actually in your hand)
  - List of indices matching cards in your given hand
  - Okay to not play anything!

Templates

- Three templates
  - Fast – Uses cheap cards to get an early advantage
  - Reckless – Uses expensive cards, hopes to get energy enough
  - Careful – Catious and uses division cards to be safe



Fast

Careful

Reckless

Templates

- Look at template code

Modes of playing

- Can run visualized with pyglet
  - 0, 1 or 2 human players (pick cards with mouse, commit with corner button)
  - Press spacebar to progress
  - Press enter for auto play
- Can run as function without pyglet
  - As terminal command
  - In jupyter notebook with analysis

What now?

- I make the github repo public (github account mads-bertelsen)
- Clone the github repository or download the code
- Install with: pip install -e .
- Test it works with: python test_expression.py
- Copy a template and rename it to your actual name
- Edit the template
- Use the jupyter notebook to check your progress
- At end of day, send your AI file to the game master