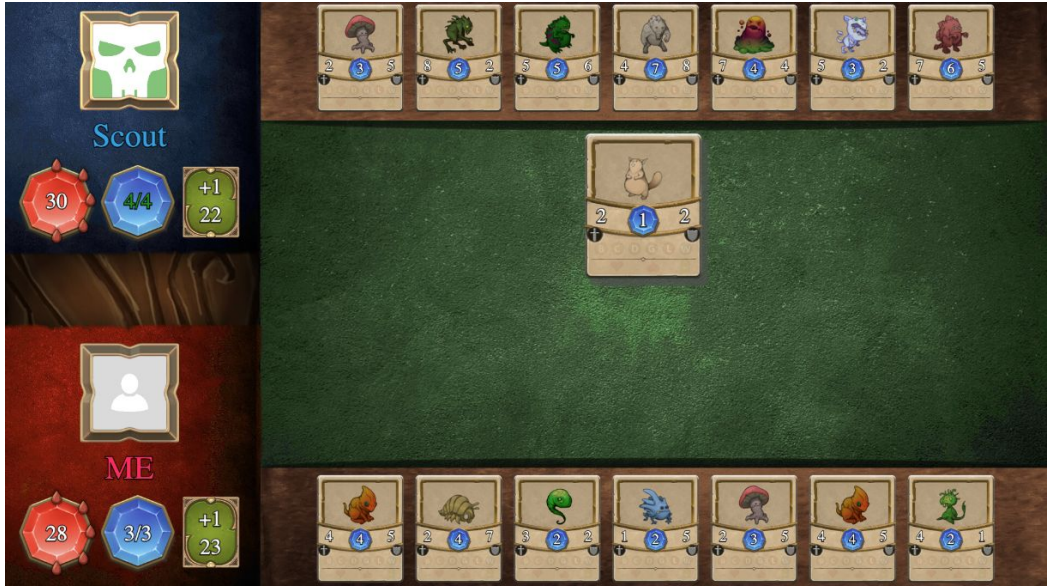
AI for Legends of Code & Magic

Jan Plot, Adam Šebesta

Game recap

- Two-player card game
- Each player drafts cards, plays them, attacks and tries to get the opponent's health to 0
- Game has two phases
  - **Draft phase**
  - **Battle phase**

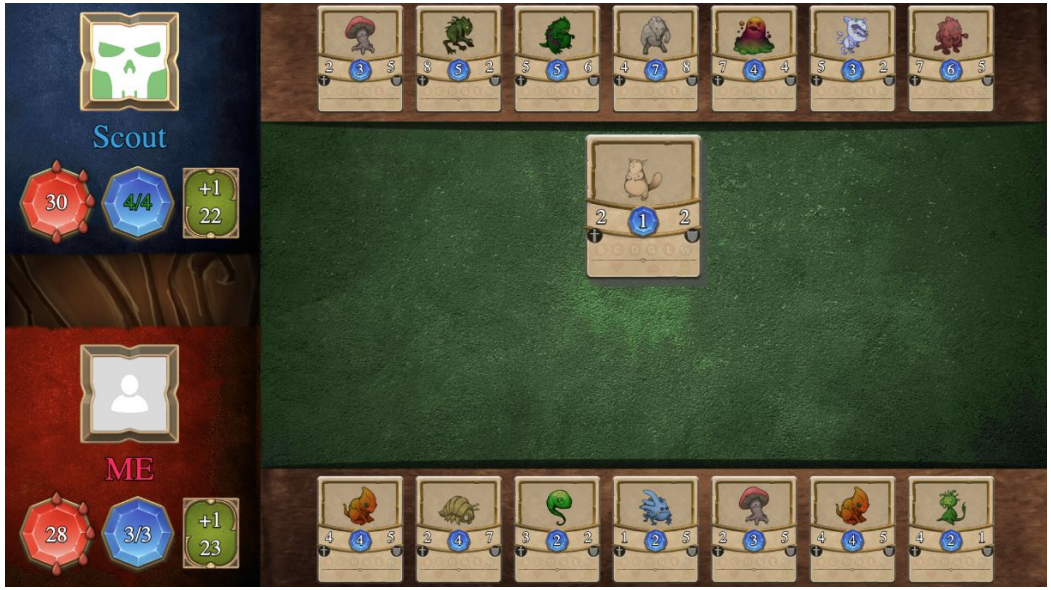


Draft Phase

- Both players must create a deck of 30 cards
- For 30 turns, both players are given a choice between 3 cards same for both players
- Both players can select the same card, they will each receive a copy
- They see what the other player picked
- Then both decks are shuffled

Battle Phase

- Players switch turns. Each player always draws a card at the start of their turn
- Spend mana to summon creatures/items
- Summoned creature can attack once per turn
- Each player can spend up to their max mana per turn
- Each player is granted one additional max mana per turn until they have 12 max mana



Progress so far

- 2 phases -> we split the work between us
- Iteratively improve based on each other
- Right now only simple approaches
- Wood -> Bronze -> Silver -> Gold -> Legend
- Last time (midterm) we were 179th out of 409 in Silver and 600 out of 3127 overall
- Currently (end of semester) we are 140th out of 279 in Gold and 305 out of 3127 overall

Draft Phase - Current Approach

- Weight for each property of a card (attack, defense, cost,...)

  Weights for combinations of properties

- 4 pools of cards based on mana cost (0-2, 3-4, 5-6, 7+)
- Trying to fit a curve by favoring cards from pools with low card counts
- Pick best card based on weight value + curve bonus
- Numbers are handmade

```
IDEAL_COUNTS = [
    9,  # ideally pick 9 cards from cost 0-2
    9,  # ideally pick 9 cards from cost 3-4
    7,  # ideally pick 7 cards from cost 5-6
    5   # ideally pick 5 cards from cost 7+
]
```

```
WEIGHTS = {
    'cost': -1.5,
    'attack': 0.7,
    'defense': 0.7,
    'myHealth': 0.5,
    'oppHealth': -0.5,
    'draw': 1.0,
    'drain': 0.3,
    'ward': 0.9,
    'charge': 2,
    'break': 0.3,
    'guard': 1,
    'lethal': 3,
    'creature': 1,
```

```
'charge_and_lethal': 2.5,
'lethal_and_ward': 2.0,
'guard_and_high_def': 1.3,
'break_and_high_atk': 0.7,
'draw_and_cheap': 2.0,
'drain_and_high_atk': 0.6,
```

Battle Phase - Past Approach

- Simple “do all actions” agent - got us to bronze
- It does all the actions
- Tried to improve with some basic rules
  - Kill only if safe
  - Kill opponent if you can
- More rules didn’t get us to silver

```
# very simple bot
for a in cards:
    actions.append(f"SUMMON {a['instance']}")
    for b in cards:
        actions.append(f"ATTACK {a['instance']} {b['instance']}")
    actions.append(f"ATTACK {a['instance']} -1")
```

```
def check_if_lethal(cards, enemy_cards, enemy_health):
    combined_attack = 0
    guard_defenders = [c for c in enemy_cards if c.get('guard', False)]

    if guard_defenders:
        return

    for card in cards:
        combined_attack += card['attack']

    if combined_attack >= enemy_health:
        for card in cards:
            actions.append(f"ATTACK {card['instance']} -1 face")
```

Battle Phase - Past Approach

Silver

- Flat Monte Carlo rollout
- We randomly select one of possible actions until we can't
- We try this while we still have time
- Then we select the best sequence of actions and play it

```
while time.perf_counter() - start < 0.07 and stop != True:
    result = simulation(copy.deepcopy(my_board), copy.deepcopy(opp_board),
                        player1_health, player2_health, cut_prob: 0.1)
    score = evaluate_state(result[1])
    result[2] = score

    if score > best[2]:
        best = result

    counter += 1
    pass
```

```
def simulation(ally_board, enemy_board, ally_hp, enemy_hp, cut_prob):
    poss_attacks = get_all_attacks(ally_board, enemy_board)
    acts = []
    while len(poss_attacks) != 0 and random.uniform( a: 0,  b: 1) > cut_prob:
        act = random.choice(poss_attacks)
        acts.append(act)
        # ...
```

Battle Phase - Improvements from last time

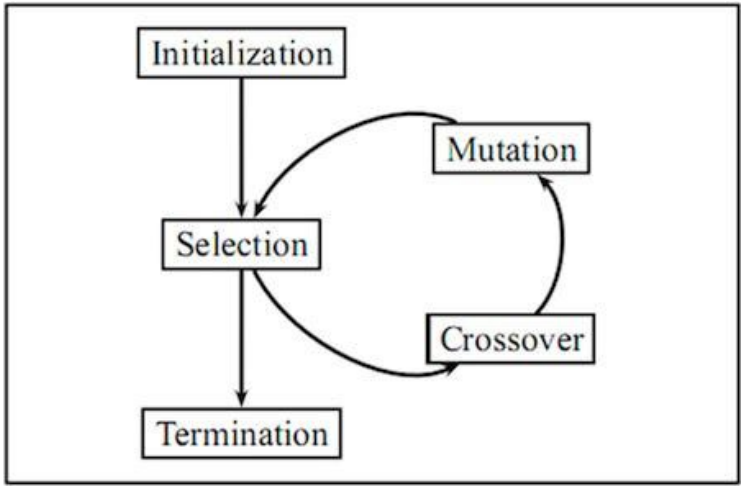
- Better function for checking if we can kill our opponent this turn
  - kill all guards first with minimal overkill, then attack enemy player with everything
- Improved evaluate_state function
  - add punishment for having full board
- Improved simulation function
  - now correctly simulates one round after attacking with units
  -> improved fights
- Improved summoning logic
  - before we just tried to summon all cards in our hand
  - now we summon the cards with best total value
- Preventing enemy lethal
  - If the enemy could kill us in his next turn, we try to prevent it by killing his monsters

What we tried - items

- Until now we only played with creatures and ignored items
- Items can help you upgrade your creatures, destroy/weaken enemy creatures or deal damage
- Many fights we lose because the enemy is properly utilizing items
- We tried to utilize items but they need to be played almost perfectly to get value from them
    - Otherwise they are just taking a card slot from us and we would be stronger with a creature card instead

What we tried - EA for better draft phase

- We wanted to get the weights used to score cards using evolutionary algorithm
- The simulator is written in Java
- We spent many hours trying to make it work (too many)
- We also tried to recreate the simulator in python, didn't manage to finish that either
- No simulator -> no fitness function -> no EA



Thank you for
your attention!