BetaSplendor Zero

ETSU

Computer Science

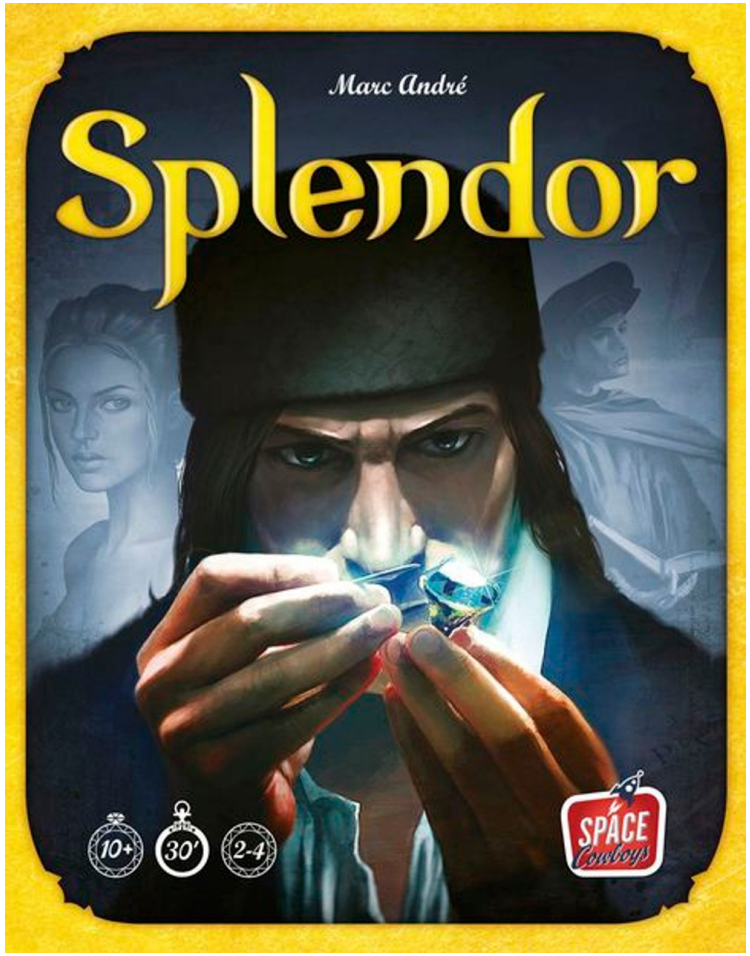
Cary FitzHugh

2023-03-08

Contents

1. Prologue

- Why Splendor?
- Fun family game. Tractable strategy, with non-trivial complexity requiring other approaches than simple exhaustive searching.
- I am not very good at it, so I can try approaches which don't require a human genius.

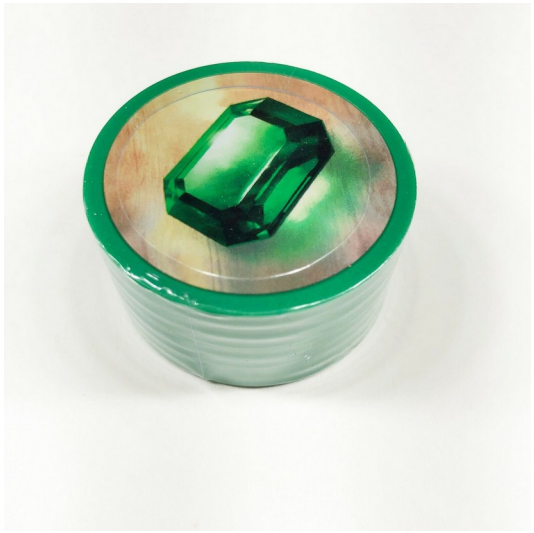


2. Splendor

- Players 2 - 4
- Engine-Building / Resource Management
- Use Gems to Buy… More Gems
- Gain enough prestige to win the game

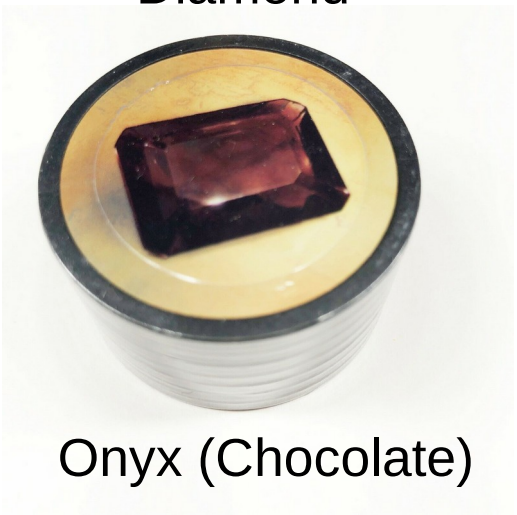
2.1 Gems

Diamond



Emerald

Sapphire (Blueberry)



Onyx (Chocolate)

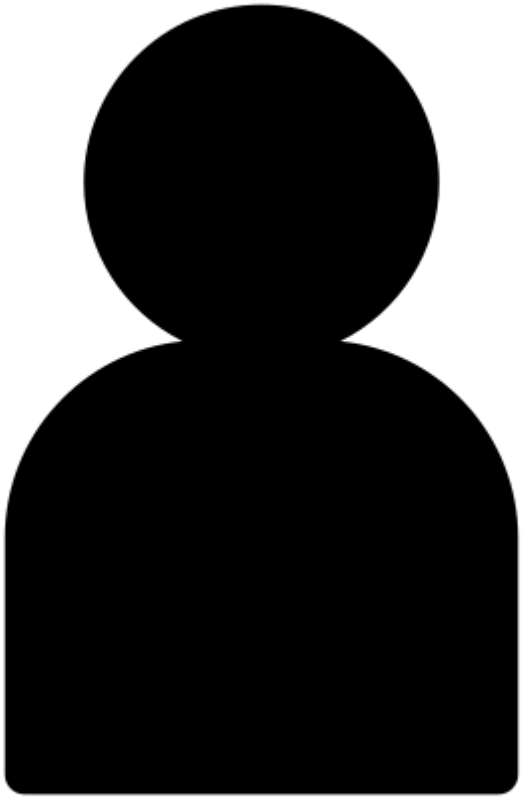
Ruby (Strawberry)

Gold (Wildcard)

2.2 Development Cards

2.3 Nobles

3. Turn



Pickup Gems

Buy a Card

Reserve a Card

3.1 Pickup Gems

Pickup 3 Gems

(Must all be unique)

Pickup 2 of the same kind

(only if there are > 4 left)

3.2 Buy Development Cards

Buy one face up from the table

Buy one that you reserved

Paying with gems + your bonus

Gold is “wild”

3.3 Reserve Card

You can reserve any card on the table.

You can reserve an unseen card from the top of one of the tier decks

You can only reserve 3 cards at a time

When reserving you get 1 gold

3.4 TMG

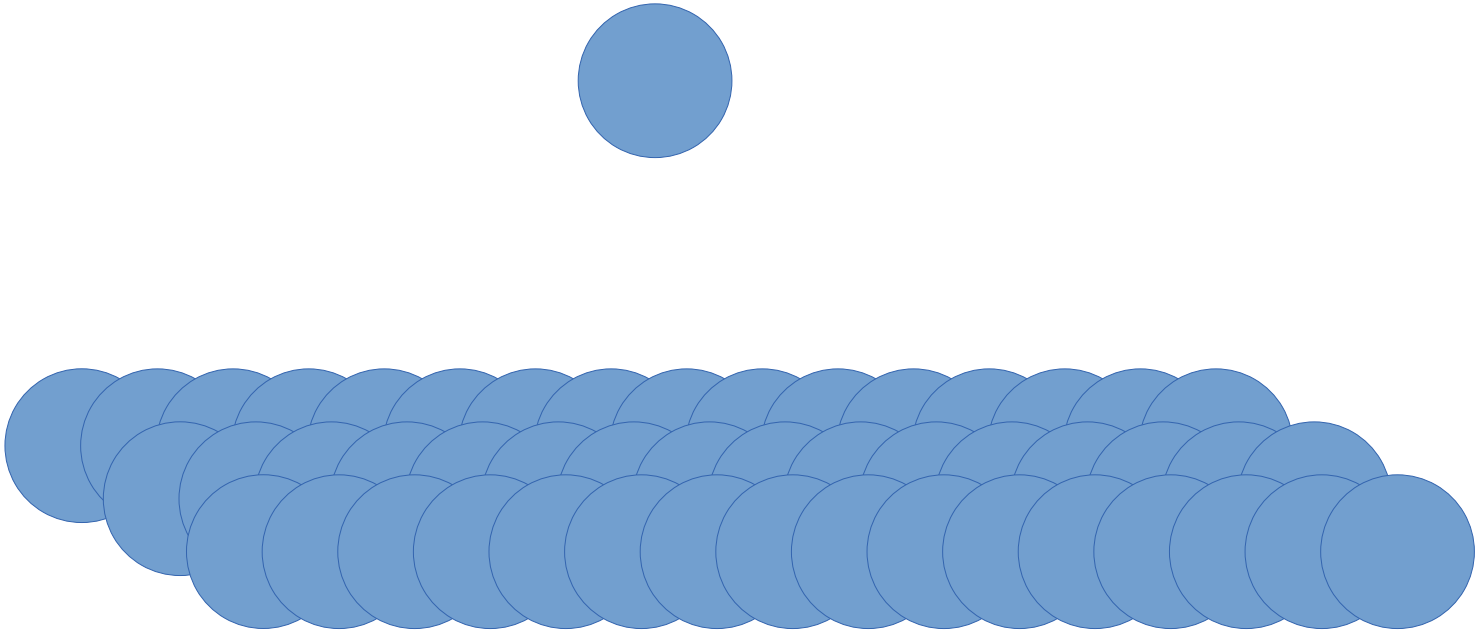
If you have > 10 gems + gold

You need to return them until you have a valid number.

4. The ending

- Play to 15 prestige points
- Everyone gets the same number of turns

5. Complexity



~ 40 Available Actions

In one 4 turn round, that is

40^4 possible states

2,560,000

5.1 Actions

Pick Gem Options	Reserve Cards	Buy Card Options
PICK_DSE	RESERVE_TIER_0	BUY_TIER_0_0
PICK_DSR	RESERVE_TIER_0_0	BUY_TIER_0_1
PICK_DSO	RESERVE_TIER_0_1	BUY_TIER_0_2
PICK_DER	RESERVE_TIER_0_2	BUY_TIER_0_3
PICK_DEO	RESERVE_TIER_0_3	BUY_TIER_1_0
PICK_DRO	RESERVE_TIER_1	BUY_TIER_1_1
PICK_SER	RESERVE_TIER_1_0	BUY_TIER_1_2
PICK_SEO	RESERVE_TIER_1_1	BUY_TIER_1_3
PICK_SRO	RESERVE_TIER_1_2	BUY_TIER_2_0
PICK_ERO	RESERVE_TIER_1_3	BUY_TIER_2_1
PICK_DD	RESERVE_TIER_2	BUY_TIER_2_2
PICK_SS	RESERVE_TIER_2_0	BUY_TIER_2_3
PICK_EE	RESERVE_TIER_2_1	BUY_RESERVED_0
PICK_RR	RESERVE_TIER_2_2	BUY_RESERVED_1
PICK_OO	RESERVE_TIER_2_3	BUY_RESERVED_2
Total(15)	Total(15)	Total(15)

6. AlphaGo Zero

- Only uses one NN to calculate

  P_{win}, $P_{vec}AllMoves$

- Inputs to NN are each position on the go board (19x19 inputs)

- Plays games against itself using MCTS, informed by the player's current NN

  The utility of a node for MCTS is calculated by the current NN (or 1 or -1 if terminal leaf)

- After each finished game, the networks are trained to reinforce or diminish their affinity for their move history

- Once the games have been played and a winner is found, that winning NN is played against the "current best NN" – if it is a substantial winner, it replaces the old best

Do this a lot.

7. BetaSplendor Zero

- Uses a single NN
- Inputs are each discrete board entity (400+ inputs)
- Ouputs are the same as AGZ
- Play games against itself like AGZ
- Training may benefit from scaling the ‘diminishment’ of a network depending on how badly it lost
- Hoping to train 2, 3, and 4 player Networks
- I want to add a phase for 3+ player games where the “best current” network is played against each of the latest networks, to try to consider multi-player interactions