

The Game



Team 5

- 4 artists
- 5 programmers
- 2 producers
- 2 designers
- ...and a sound guy



Problem space

- Rules are objects in the game. Game is based on making new and breaking/modifying rules
- System need to handle insane amount of iteration. Blizzard factor, as well as balancing for a card game
- Small team, limited time, very small AI CPU budget

Considerations

- Design driven
 - Easy Creation/editing of cards
 - How much Al
 - Difficulty level of Al
- To Data or not to Data. ARGUMENT PRIMO
 - Cards are data
 - Al was not

Gameplay System

Entities

- Has attributes called Tags
- Can only do things with Powers
 - Modifier power
 - Trigger power
 - Action power
 - Activate power

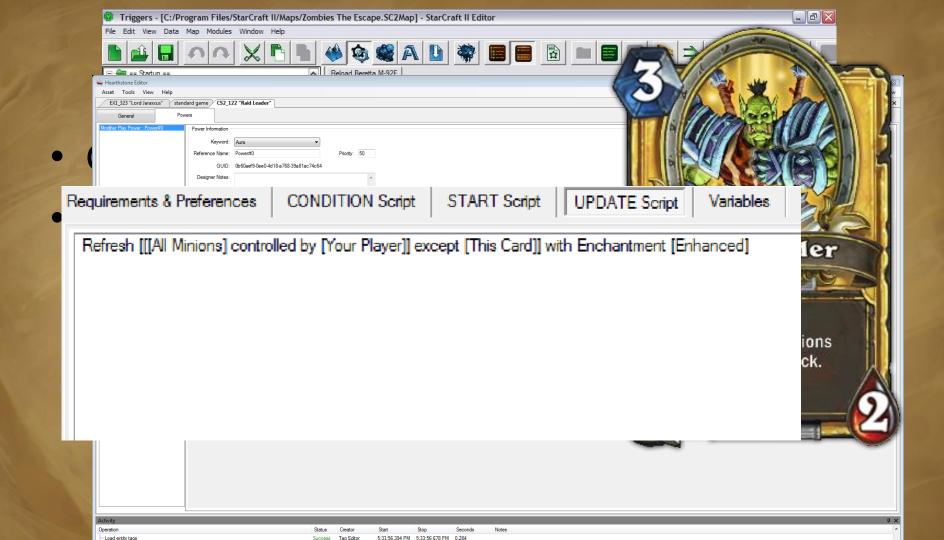
Update State

- Clear all modifiers and rerun them
- Check for triggers, if anything then Update State
- Check for deaths, if anything then Update State

Main Loop

- Get current options and send to player/AI
- Run power associated with option taken
- Update state





Al system

- Designed to imitate intermediate player
 - Interviewed many different levels of players to identify the play style of an intermediate
- Piggybacks player options
- Very simple scoring system
- Two card kills are only preplanned comb
- Very few holdbacks
- Difficulty leveling done a few ways
 - Metering of ai responses
 - Metering of ai mulligan
 - Adjustment of ai decks
 - Slight adjustment to a few ai percepts (momentum, target class, holdbacks)
- NO CHEATING, OMNISCIENCE, OR MUCKING WITH RNG

```
//-----
int GameAlPerceptions::CS2_122()
{//Raid Leader
// minion
return EF_Aura(1,0,0,TT_FRIENDLY_MINIONS);
}
```

Things that could have been better

- Slowish start code wise
- Tool maturity
- Bringing design behind curtain sooner

Things that worked out well

- Over a year of design iteration
- Flash Prototype
- Extended alpha and beta periods
- Al is crazy performant
- Game is pretty damn good

