



Reading the rules of Baba Is You

Arvi Teikari
Game developer (Hempuli)

GAME DEVELOPERS CONFERENCE
MARCH 16–20, 2020 | #GDC20

Topics of this talk

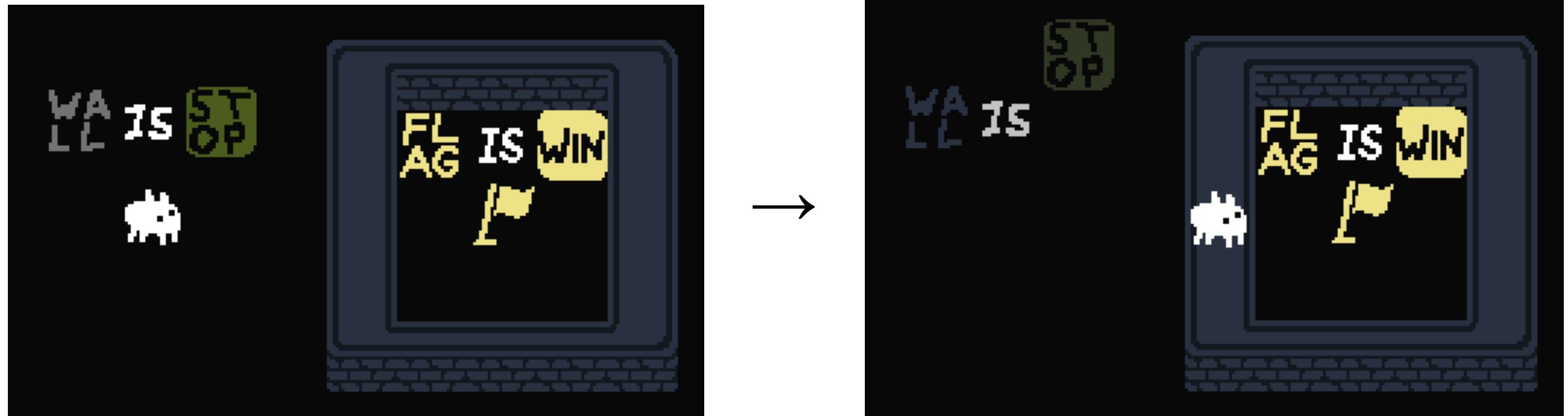
- Introduction
 - Who Am I?
 - What Is Baba Is You?
- The rule system in Baba Is You
 - Introduction
 - How it works
 - Development of the rule system
 - Problems
- Conclusions

Introduction

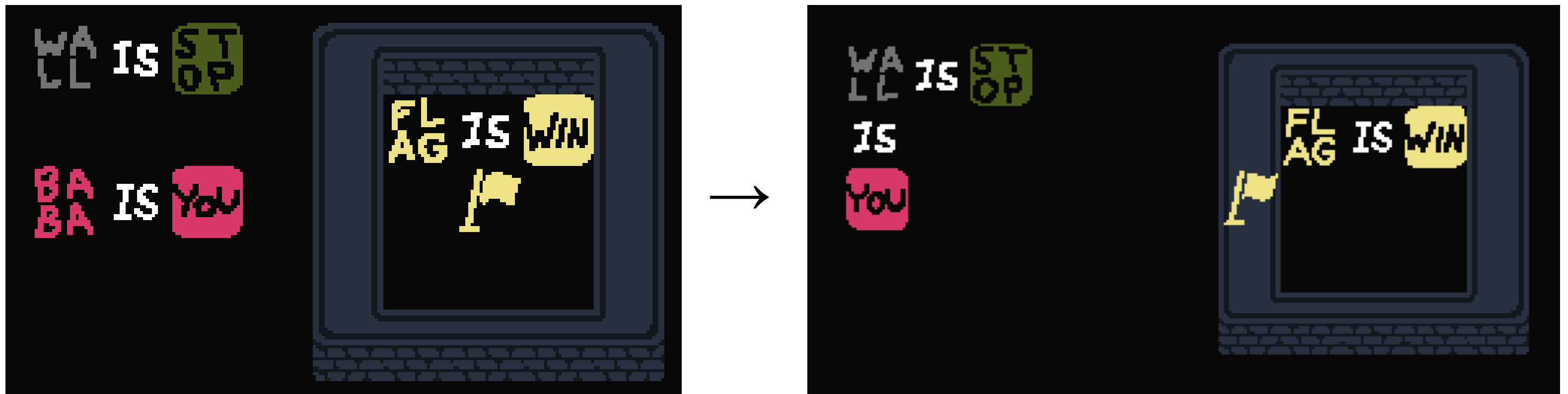
- Who Am I?
 - Game developer from Helsinki, Finland
 - Started as a hobby in primary school (though interest in kindergarten)
 - Trouble learning programming → game-creation tools
 - *Game Maker, then The Games Factory, finally Multimedia Fusion*
 - First game jam in ~2010, quickly a very important part of the hobby
 - *No More Sweden, Nordic Game Jam*
 - Company, first commercial release in 2015
 - *Environmental Station Alpha (2015)*
 - *Baba Is You (2019)*

- What Is Baba Is You?
 - Block-pushing puzzle game
 - Main gimmick: rules present as pushable blocks → rule-changing
 - Initially created in 2017 for Nordic Game Jam in Copenhagen
 - *Theme: "Not There" - inspiration from how "Not" behaves in logic → "Ice Is Not Melt"*
 - Heavy inspiration from other puzzle games
 - *Stephen's Sausage Roll, Snakebird, A Good Snowman Is Hard To Build, Braid, Corrypt*
 - Won the jam, later other awards (e.g. IGF 2018)
 - Release in 2019, fairly successful (yay)

- Basic gameplay of Baba Is You:



- Basic gameplay of Baba Is You:



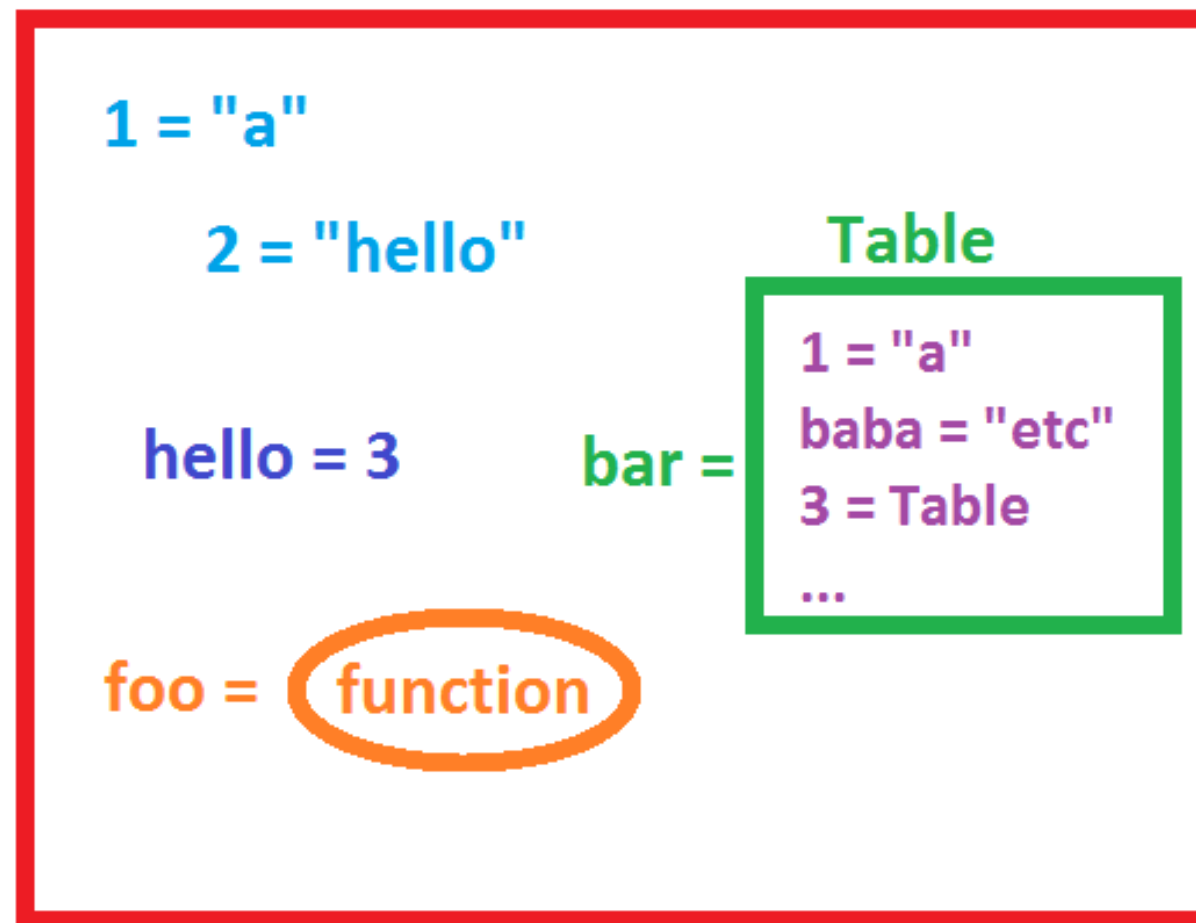
The rule system

• Introduction

- A novel system → novel solutions needed
 - *Needed to figure out something fast for the jam; first version very simplistic*
- The system evolved a lot during the 2 years of development
- At its basis, every rule consists of 3 fundamental components:
 - Object: **Baba** (= *Which object the rule is applied to*)
 - Verb: **Is** (= *How the quality is applied to the object*)
 - Quality: **You** (= *The effect applied to the object*) (can also be another object word)
 - These are needed for every rule, but additional components may be around
- During gameplay: various systems ask the engine which rules are present
 - *"What objects have this rule?" or "Which objects does this rule apply to?"*
 - *"Does this rule apply to this specific object?"*
 - *"What qualities does this verb give to the object?"*

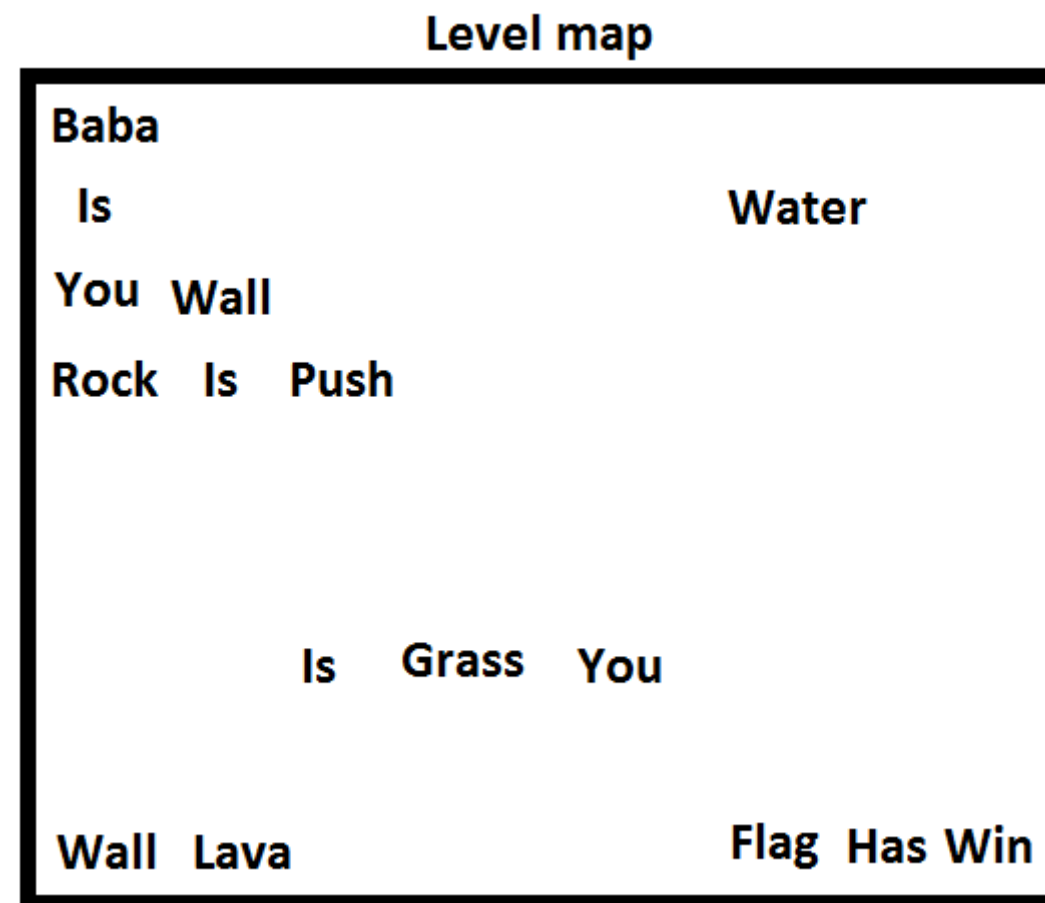
- Mostly implemented in the lua scripting language
- Tables a very important concept
 - Databases of key-value pairs; key can be a *number* or a *string*, value can be a *number*, *string*, *function*, *another table* etc.
 - Very flexible: can represent lists, arrays, or more complex databases

Table A



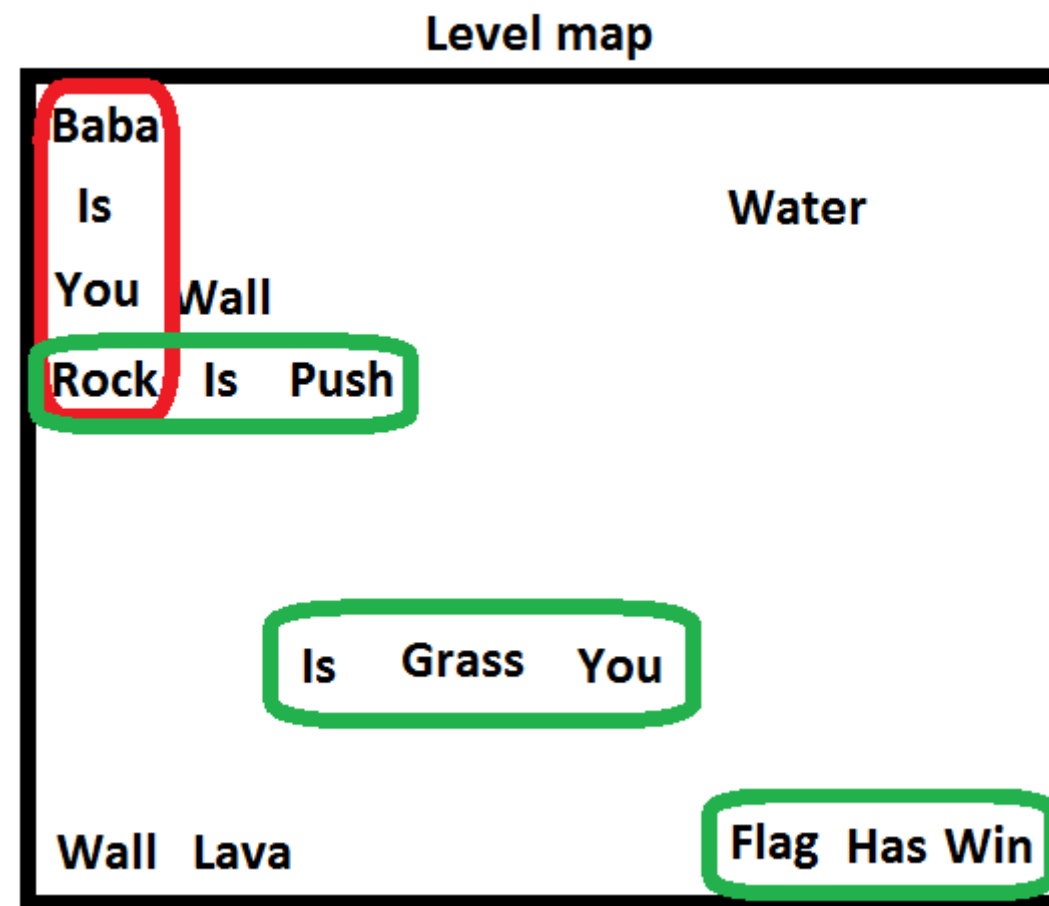
- How it works

- Problem: words can lie anywhere in the level
 - Some might form sentences, some might form nonsense, some might be alone, etc.
- Multiple passes



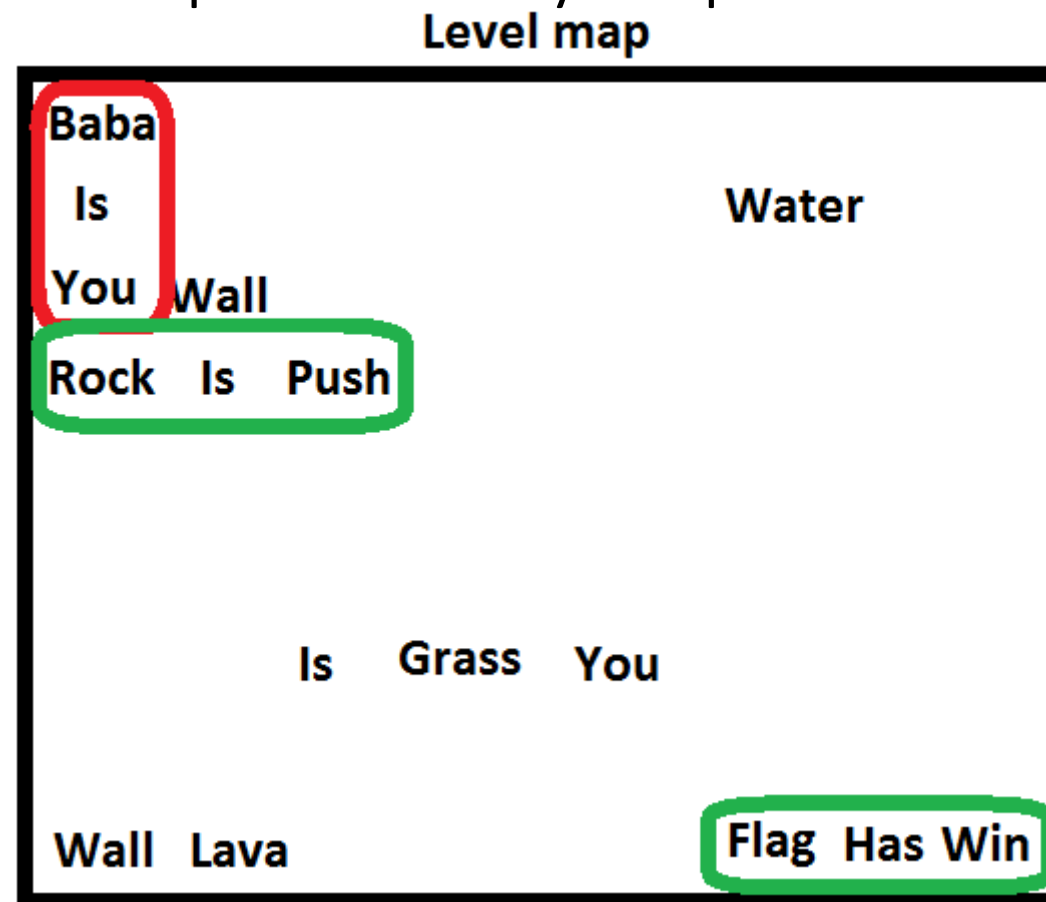
- First pass

- Find potential first words of sentences and check how long potential sentences they'd form
- For simplicity & design reasons: only **left-to-right** and **top-to-bottom** is accepted & rules must be in straight lines
- Since the shortest possible rule is 3 words, any 1- or 2-word lines can be discarded right away



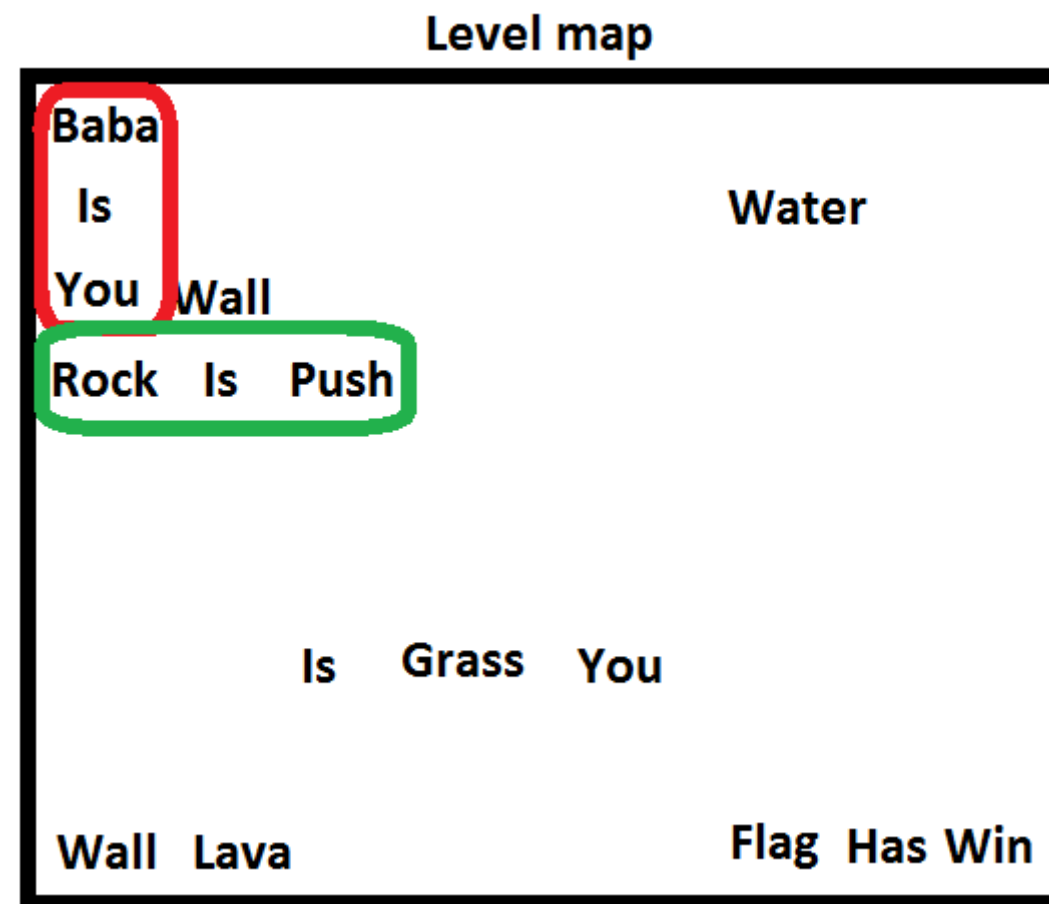
- Second pass

- Go through the potential sentences and see if their word types are in an order that's seemingly valid
 - (e.g. "Baba Is You Rock" → "You Rock" doesn't parse, disregard "Rock")
- If a word is discarded, it's checked again later to see if it could form a valid sentence otherwise
 - (e.g. "Baba Is You Rock Is Push" → "You Rock" doesn't parse, but "Rock Is Push" does)
- Later, more complex word types make this pass extremely complicated



- Third pass

- Go through the sentences left and see if the syntax makes sense; if not, discard
 - (e.g. "Flag Has Win" follows the correct word order (Object-Verb-Quality), but "Has" doesn't accept "Win" as a following word in the sentence ("Flag Has Baba" would be valid))
- If the syntax works, transform the sentence to a format the game logic can understand and add it to various rule lists (e.g. a global list, an object-specific list, etc.)

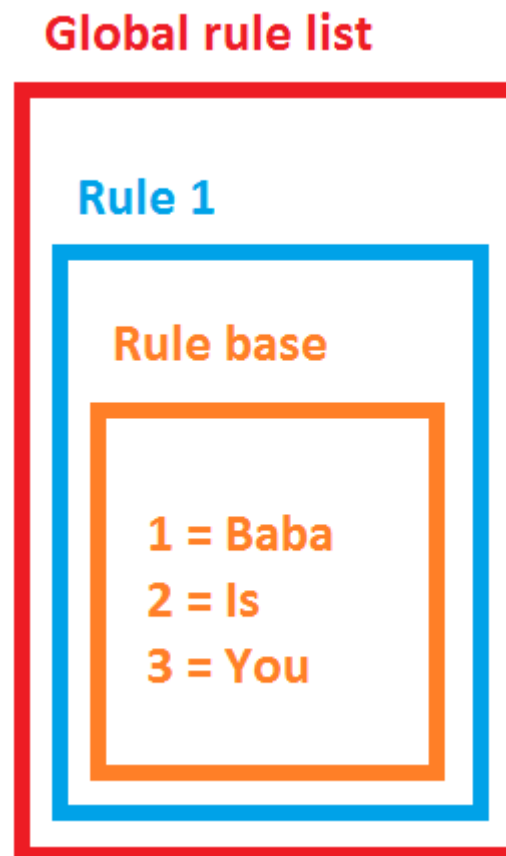


- Finally

- Do some complicated word shuffling to ensure that the words are grouped right
 - Mainly related to conditionals with parameters (more about them later)
- Handle rules that create other rules
 - "All Is You" → "Baba Is You", "Rock Is You", "Wall Is You" ...
 - "Not Baba Is You" → "Rock Is You", "Wall Is You", ...
- Handle rules that disable other rules
 - "Baba Is Not You" disables all "Baba Is You" rules
- Mark active rules as such, play sounds & sprinkle particles
- Mark disabled rules as such

- Development of the rule system

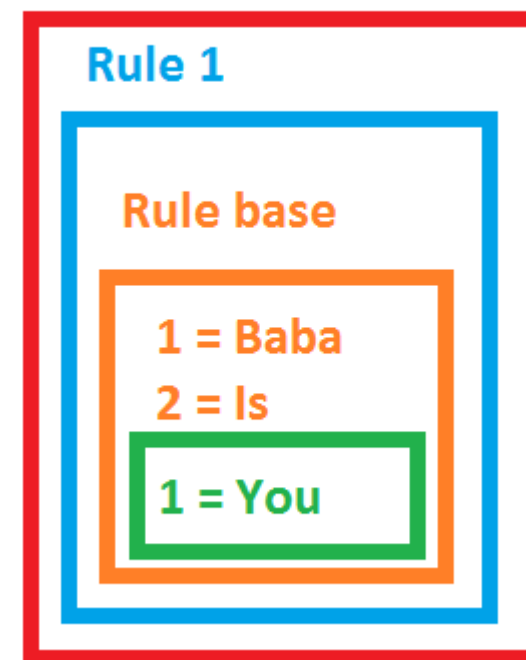
- At least 5 iterations
- Augmented as needed, until problems grew too big
→ rewrite & new iteration
- First iteration:
 - Only 3-word-long rules
 - Parsing very simple
 - Same for rule formatting



• Second iteration: conditional words & “And”

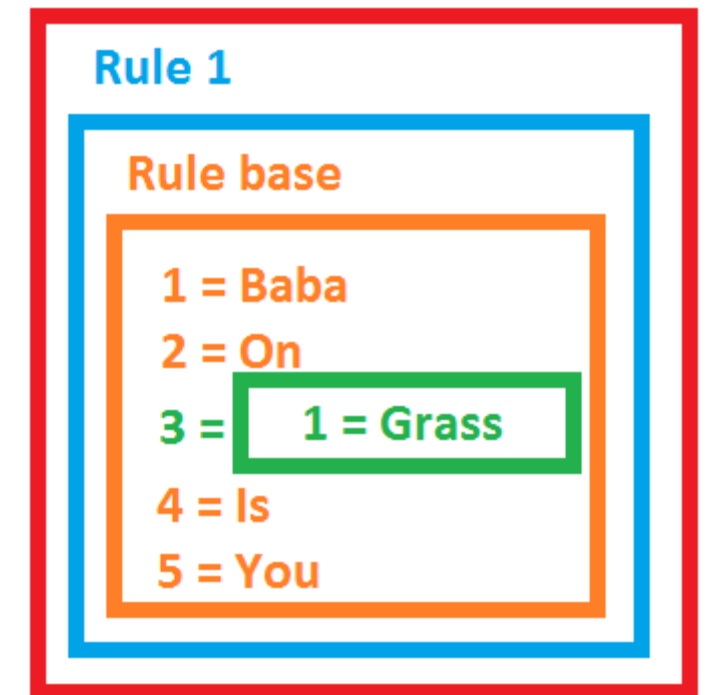
- “And” allows bundling multiple objects or qualities together
 - (e.g. “Baba And Rock Is You” or “Baba Is You And Move”)
- In this version, only one conditional word: “On”
 - (“Baba On Grass Is You” → If Baba is on a Grass object, Baba is You)
 - (“And” may also apply: “Baba On Grass And Flower Is You”)
- Rules may apply differently to different objects depending on their surroundings → new checks
- A very naïve implementation:
 - Bundle objects and qualities into their own groups
 - If rule has 3 entries, assume normal structure
 - If rule has 5 entries, assume conditional structure
 - *This feels extremely silly in hindsight*

Global rule list



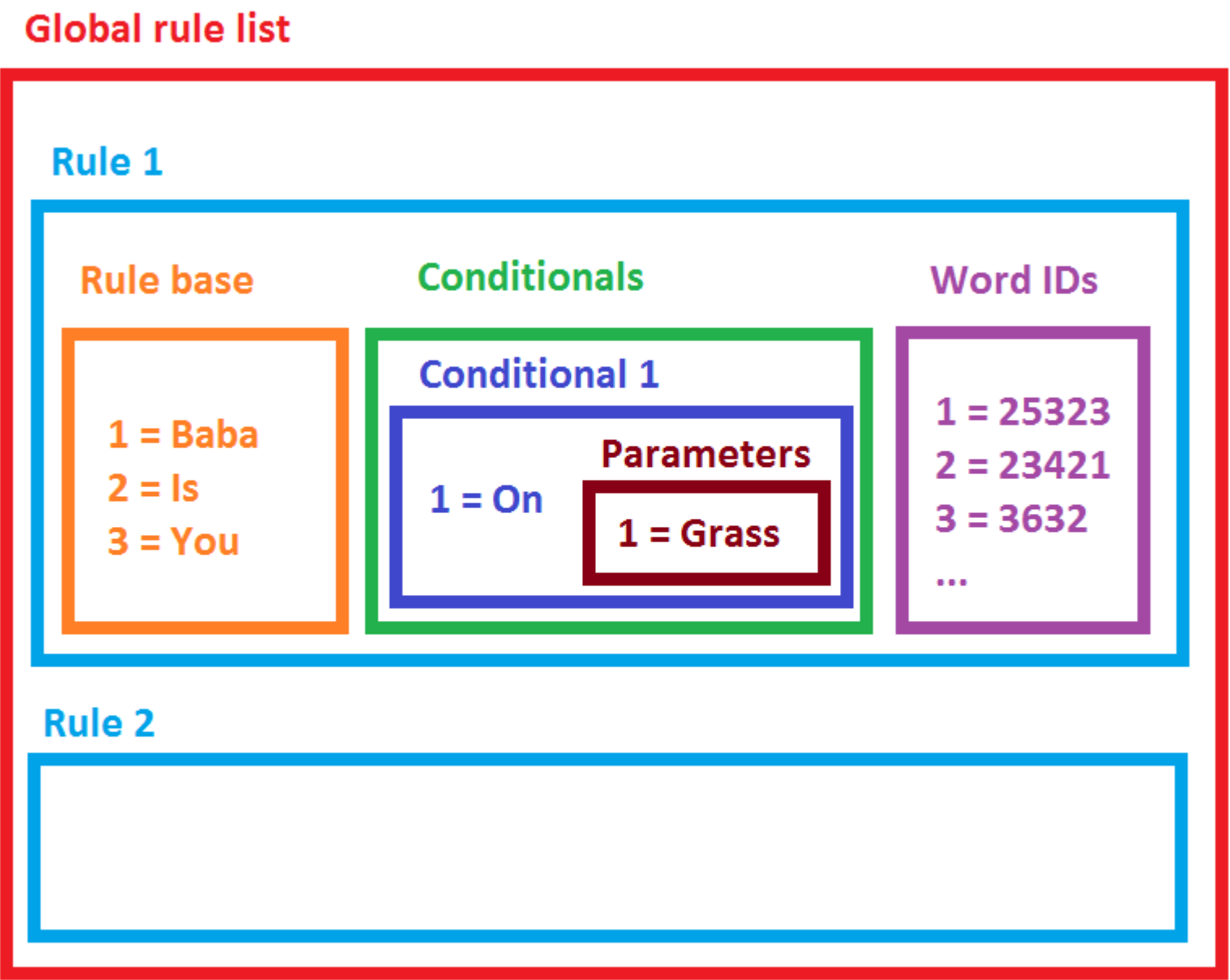
OR

Global rule list



- Third iteration: more conditionals
 - New conditionals that use parameters: "Near", "Facing"
 - New conditional types: "Lonely"
 - *(Doesn't use parameters but acts like a "prefix": "Lonely Baba Is You")*
 - Old system way too rigid for this → a very large rework
 - After this, the rule format stayed pretty much the same
- Later: "Not", very exotic special rules
 - At least one larger rework to accommodate new structures; basic system stayed mostly the same

- What a rule looks like in the rule list in the current iteration:
 - I'm sure this needs no explanation



• Problems & complications

- Increasing complexity & word combinations
- New mechanics were cool but complicated matters further

- Some examples:

- "Not"

- "Not Baba Not On Not Grass Is Not You", "Not Not Not Baba Is You"

- Chaining conditionals

- "Baba On Grass And Near Wall Is You"

- Letters

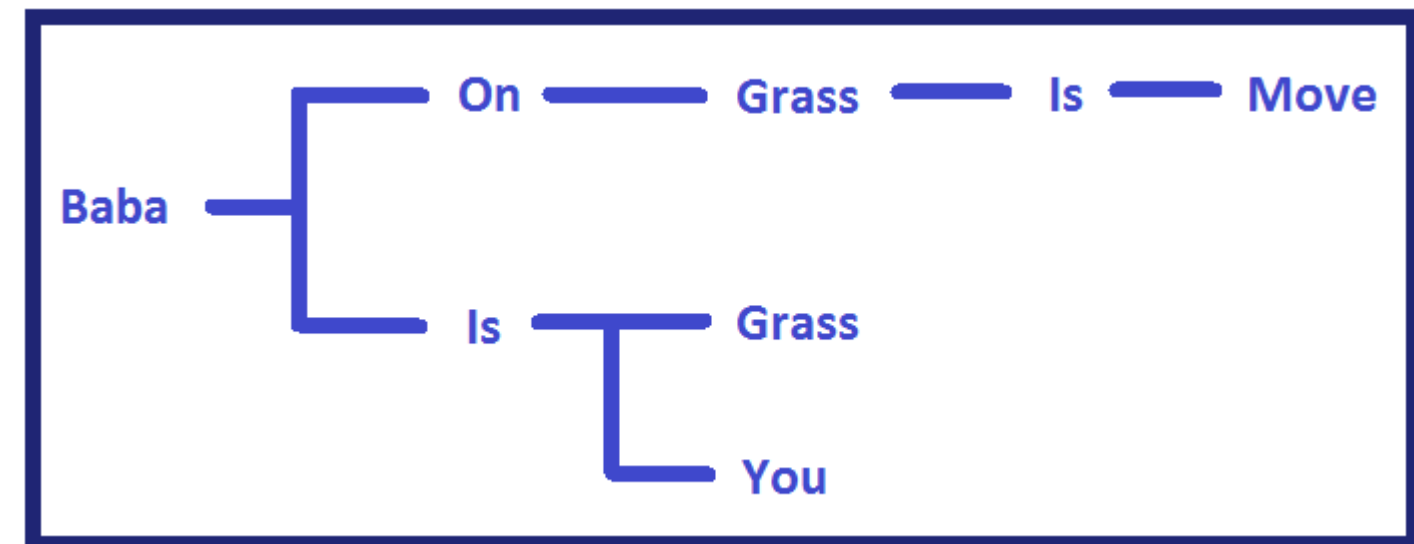
- "B-A-B-A Is You", "W-A-L-L Is Stop", "B-A-B-A-I-S-Y-O-U"

- Stacked rules

- "Baba/Keke Is You", "Baba On/Is Grass/You Is Move"

- Rule combos

- "Baba Is Grass And Rock Is Push"



Conclusions

- Futile to attempt to make a perfectly parsing system without removing something
 - Remove most unstable elements vs. accept some quirks
 - Including words that felt "fitting" took priority over the stability of the rule system
 - *Especially letters!!*
- Possibility of limiting player access to hide the worst issues
 - However: releasing a Baba Is You level editor in 2020
- Some words didn't feel worth the pain (so far): "If"
 - Lonely Baba Near Grass Is You If Rock Is On Flag And Facing Right
 - *Ugggggghhh*

Thank you very much!