I had some notes from my first scan though it, feel free to take or leave any of these. Just general observations.

- *Quota* in the BetMode/GameConfig is used to set the ratio of criteria simulations to the total sims for that mode. Quota = 0.5 for the criteria=0 means that out of 10k sims 5k would be non winning etc… The actual hit-rates for handled exclusively in the optimization setup
- Set the number of scatters to be only 3 and only 5 within the BetMode config for bonus1/bonus2 respectively. Optionally, if you want to control the hit-rate of bonus1 and bonus2 explicitly, these can be their own criteria.
- I added another reelstrip in the wincap criteria and made is more likely to use this reelstrip on the initial reveal. Also increase chances of high multipliers, so for example with bonus1:

```
conditions={
    "reel_weights": {
        self.basegame_type: {"BR0": 1},
        self.freegame_type: {"FR0": 1, "WCAP": 10},
    },
    "wr_mult_values": {2: 10, 3: 80, 4: 100, 5: 200, 10: 400},
    "wc_mult_values": {2: 10, 3: 20, 4: 30, 5: 50, 10: 80, 15: 90, 20: 100},
    "landing_wilds": {0: 10, 1: 20, 2: 50, 3: 80},
    "scatter_triggers": {3: 2, 5: 10},
    "force_wincap": True,
    "force_freegame": True,
},
```

- General suggestion: I tend to use a lot of self.record() calls for any special events. Such as recording simulation wher the WR/WC symbols are triggered. How many were on the board etc… This is helpful for passing keys to the “run_analysis” section of the run.py file to see the frequency of many events.
- Might need events within expand_rabbits to let the FE know what symbols and changing and pass updated multiplier values?
- No 0-wins in superspin. This can be set with a criteria (like 1 in 10 spins do not win etc..) Or modify the self.check_repeat() function to accept 0-wins at the natural rate.
- Can either explicitly set the multiplier values for special symbols such as done in the reveal_multiplers executable. Or use assign_special_sym_function (in game_override.py) to do this automatically when the symbol is created.
- The hit-rate of any wins in the bonus is very low. Mostly seem to get a non-winning board for most reveals
- If the optimization algorithm stalls - run the program through the terminal. Sometimes the RTP is too low or high and the Rust script will throw a warning, but this isn’t piped back to the python script.