Personalized Mario

1 Gradient Ascent Optimization

The algorithm I implemented is a variation of the gradient ascent algorithm. What is being optimized is the difficulty settings of the game at each segment, by calculating the gradient of emotions at the present and previous segment.

Before presenting the algorithm, there are some fundamentals to be discussed.

- Emotions are tracked while the game is being played, but they are processed and taken into account at the event of a segment being finished or death of Mario (offline).
- Emotions are calculated for each section separately, thus providing us with 5 vectors of emotions at each game segment. The calculations are done for each segment independently.
- Segments are considered independent of each other. Performance in segment t does not affect the difficulty of segment $t + 1$
- Emotions affecting the calculations are: Neutral, Happy & Angry.

The algorithm is provided step by step below:

Algorithm 1 Gradient Ascent Optimization for Personalized Mario

1: **procedure** GAOptimize(e_t, e_{t-1}) ▷ Emotion vectors of current and previous segment
2: $\alpha \leftarrow variance(e_1)$ ▷ compute alpha (first segment)
3: **for** $each : Section$ **do**
4: $\epsilon \leftarrow argmax|e_t - e_{t-1}|$ ▷ Get the highest emotion difference
5: $newDiff \leftarrow \epsilon * 5 * \alpha * (\pm 1)$ ▷ Calculate next difficulty (t+1)
6: **return** $newDiff$

There are a few features that aim towards implementing the algorithm, that do not belong to the gradient ascent implementation but are mostly of heuristic nature.

- Post-death emotions can directly change the difficulty (before segment is finished)
- High average of neutral emotion will result in a minimum increase in difficulty

2 Evaluation Hypotheses

3 Experimentation

- introduce a factor α on how "steep" the action decision can be
- test on the same person for different $\alpha = [0..1]$
- test for different initial parameters
- test for different decision making functions
- add convergence threshold (when changes in actions are small enough to not really affect the game difficulty)

4 Evaluation

- Maximize happiness: happiness curve should incline after N segments.
- Decrease neutral as much as possible. We want the user to be as emotional as possible, to boost our decision making.
- Evaluate the stability of neutral. If neutral percentages are high, this could also mean the user is satisfied. For example, if the initial angry levels are high and then start to decline.
- Test on different kinds of persons, (not expressive / really expressive) and evaluate the speed of convergence to some particular difficulty levels.