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 starting difficulty is "1" for each section.

The algorithm is provided step by step below:

Algorithm 1 Gradient Ascent Optimization for Personalized Mario

procedure GAOPTIMIZE(e_t, e_{t-1}) ▷ Emotion vectors of current and previous segment
  $\alpha \leftarrow 1 - variance(e_1)$ ▷ compute alpha (first segment)
  $\delta \leftarrow \alpha * 5$ ▷ scale to difficulty space [-5...5]
  for $each : Section$ **do**
    $\epsilon \leftarrow argmax|e_t - e_{t-1}|$ ▷ Get the highest emotion difference
    $nextAction \leftarrow \epsilon * \delta$ ▷ Calculate next action (raise/lower difficulty)
    if $\epsilon \in \{angry, neutral\}$ **then** ▷ If the emotion change is "negative"
      $nextAction \leftarrow -nextAction$
    $nextDifficulty \leftarrow previousDifficulty + nextAction$ ▷ Calculate next difficulty
    return $newDifficulty$

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 (The threshold should be relevant to α)

2 Evaluation Hypotheses

This implementation will be tested in comparison with an existing baseline implementation of Infinite Mario, which is not using facial expression recognition.

I expect this implementation to perform at least equally good with the baseline algorithm, although due to the high probability of neutral expressions while playing, some players may find it quite challenging, or even hard to play.

The way the algorithm is designed, I expect it to converge to an optimal level of difficulty, where the emotions of the user will be stable and not dominated by neutral. However, given the fact that the emotion recognition software is not perfect, and could produce false predictions, in some cases I expect user disappointment. This is the reason why I have added some heuristic features to the algorithm, to tackle problems like the one mentioned previously.

3 Experimentation

Since the α factor is computed automatically, this feature will not be tuned in the experimentation.

One possible experiment could be a variety of starting difficulties, to test the convergence speed of the algorithm, or maybe also test its responsiveness to user dissatisfaction (when the game starts at really high levels of difficulty).

The baseline experiment will ask the user to play a number of segments which will be produced by either the baseline implementation or the one I'm presenting. In fact, the user should not be told when his expressions are recorded in order for him to provide me with unbiased feedback.

The results I'm looking to reach is user satisfaction, a feeling that the game is adapting to the user's requirements and minimum game abandonment.

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.