PROJECT REPORT

FACIAL IMAGE GENERATION BY GENERATIVE ADVERSARIAL NETWORKS (GANS)

YU JING (YUKI)

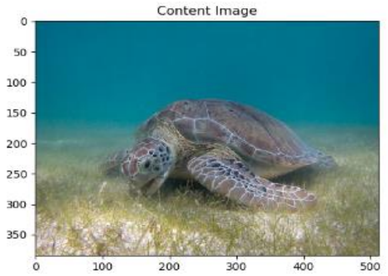
JULY 2019

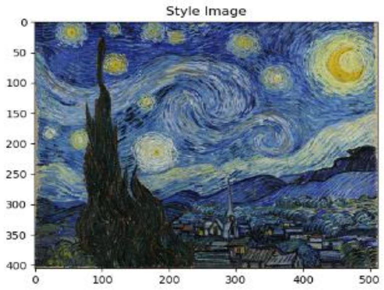
OUTLINE

- Part 1: Preparation Work
- Part 2: Facial Image Generation
- Part 3: Game Application Development
- Part 4: Further Work

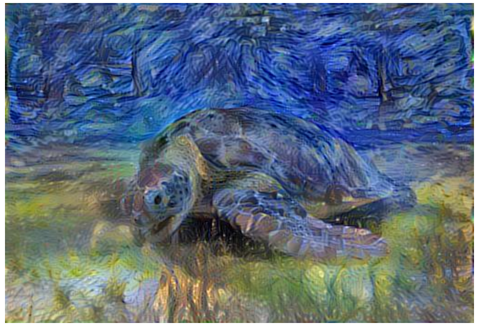
PART 1: PREPARATION WORK

- 1.1: Learn Theory
  - Coursera: Deep Learning Specialization (Andrew Ng)
  - Stanford CS231n class: Convolutional Neural Networks for Visual Recognition (Fei-Fei Li)
  - YouTube: Tutorial for Generative Adversarial Network (Hung-yi Lee)
  - Programming: TensorFlow Tutorials and Guide





- 1.2: Neural Style Transfer
  - Networks: VGG19
  - Calculate style
  - Results



Ref: https://www.tensorflow.org/beta/tutorials/generative/style_transfer

PART 2: PROJECT PROGRESS

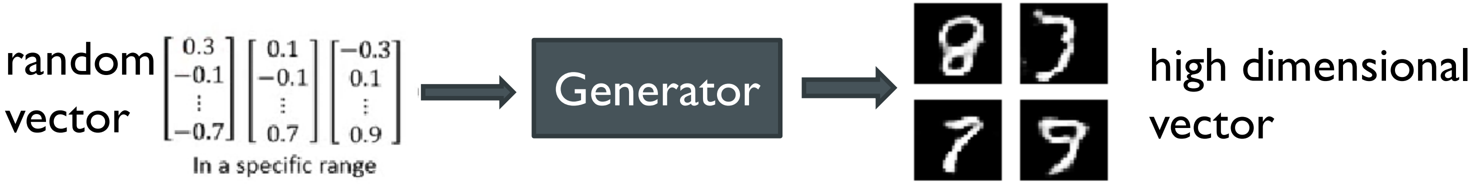
- 2.1: What are GANs ?
  - Proposed by Ian Goodfellow in 2014
  - “the most interesting idea in the last 10 years in ML” ——Yann LeCun

  - Basic idea (e.g. Hand-written Digit Images Generation)
    - Dataset: MNIST
    - 2 Networks: Generator(G) & Discriminator(D)

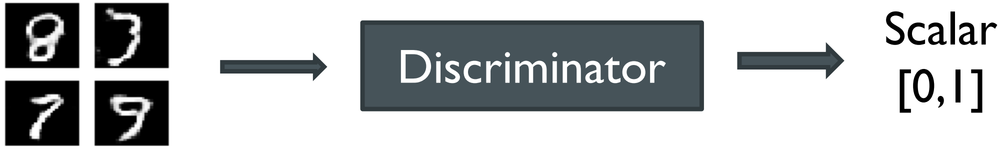
Ref: https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning

2.1: GANS - BASIC IDEA

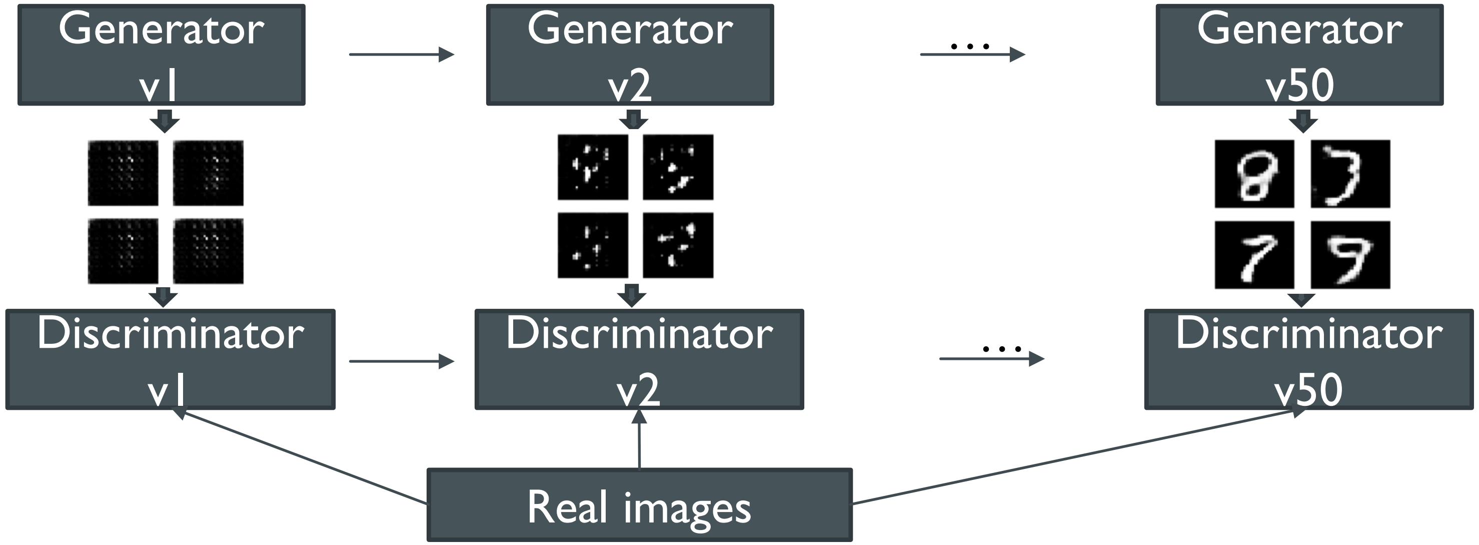
- Generator:



- Discriminator:



- Adversarial:

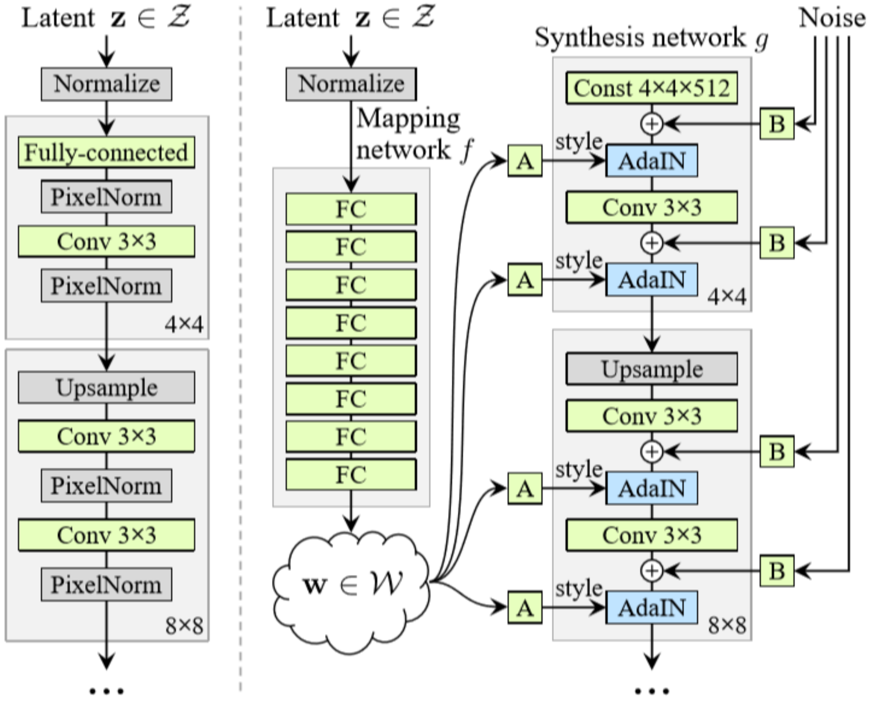


results

Ref: https://www.youtube.com/watch?v=DQNNMiAP5lw&list=PLJV el3uVTsMq6JEFPW35BCiOQTsoqwNw&index=1

2.2: HIGH-RESOLUTION IMAGES GENERATION

- Paper: Karras, T., Laine, S. and Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks*.
- Dataset: Flickr-Faces-HQ (FFHQ): 70,000 PNG images at 1024×1024 resolution
- Networks: Style-Based GANs
  - Generator
    - Mapping network f: 8 fully-connected layers
    - Synthesis network g: 18 layers—two for each resolution (4^2 −1024^2)
    - Output: images in 1024x1024 resolution
  - Discriminator
    - A CNN-based image classifier
    - Convolutional layers

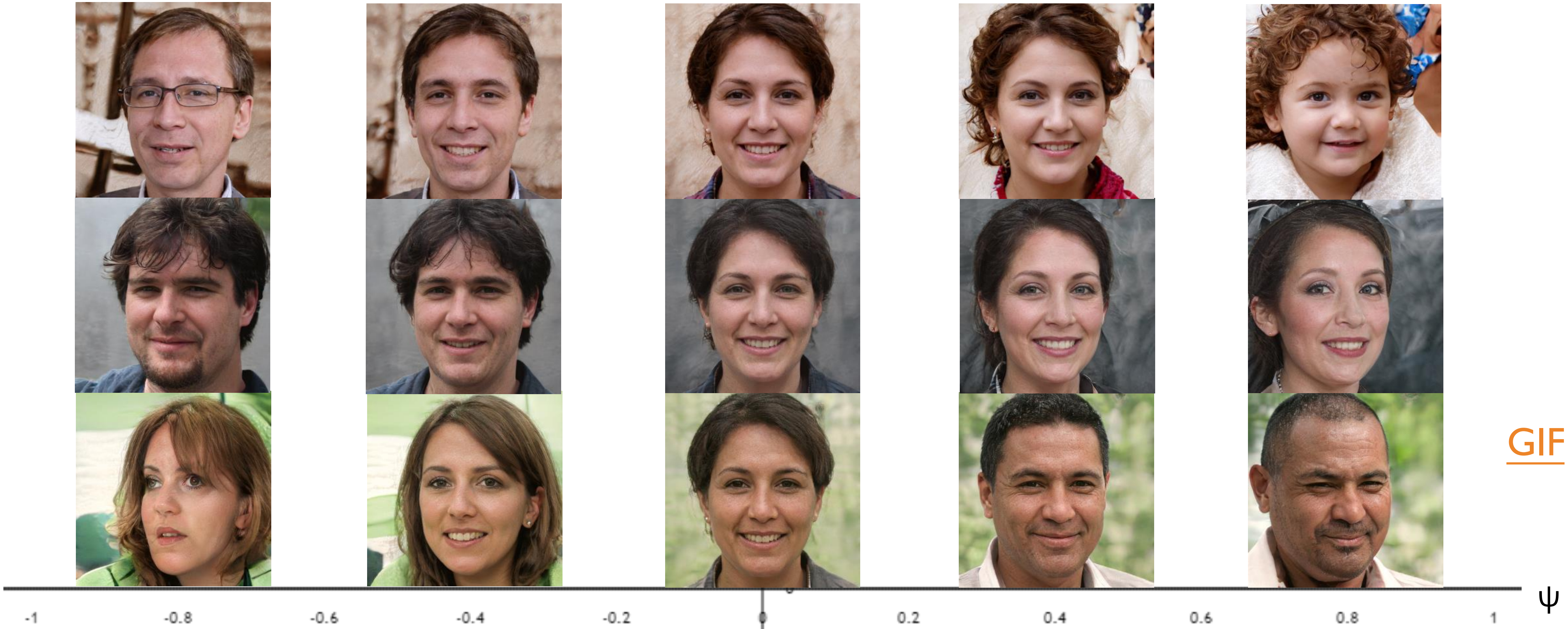


(a) Traditional (b) Style-based generator

Ref: https://arxiv.org/abs/1812.04948

2.2: HIGH-RESOLUTION FACE IMAGE GENERATION

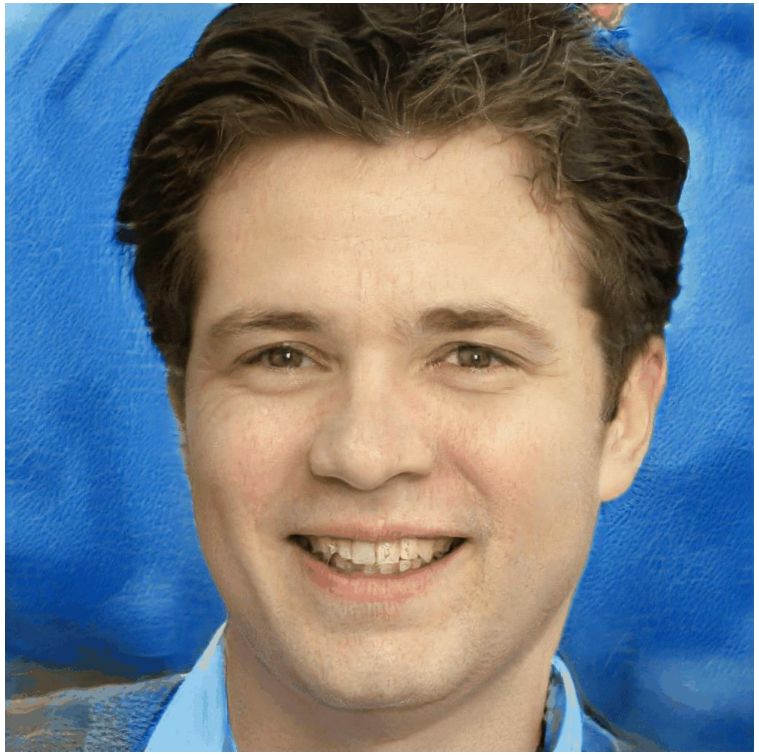
- Truncation trick ψ



GIF

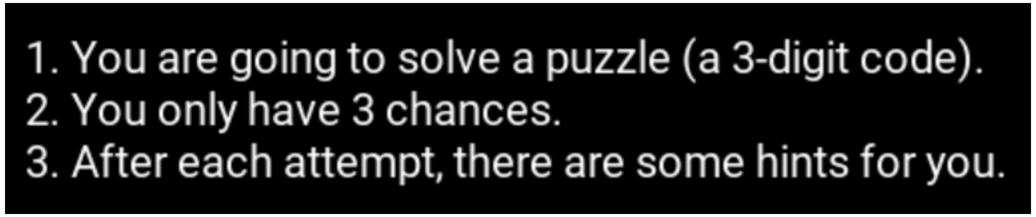
2.2: HIGH-RESOLUTION FACE IMAGE GENERATION

- GIF Demo

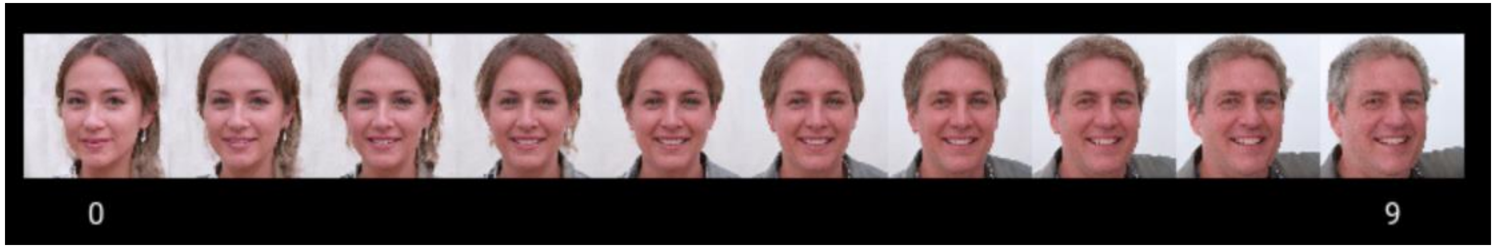


PART 3: GAME APPLICATION DEVELOPMENT

- Developed by Kivy library in Python
- Support on both Windows and Android Systems
- Game Rules:



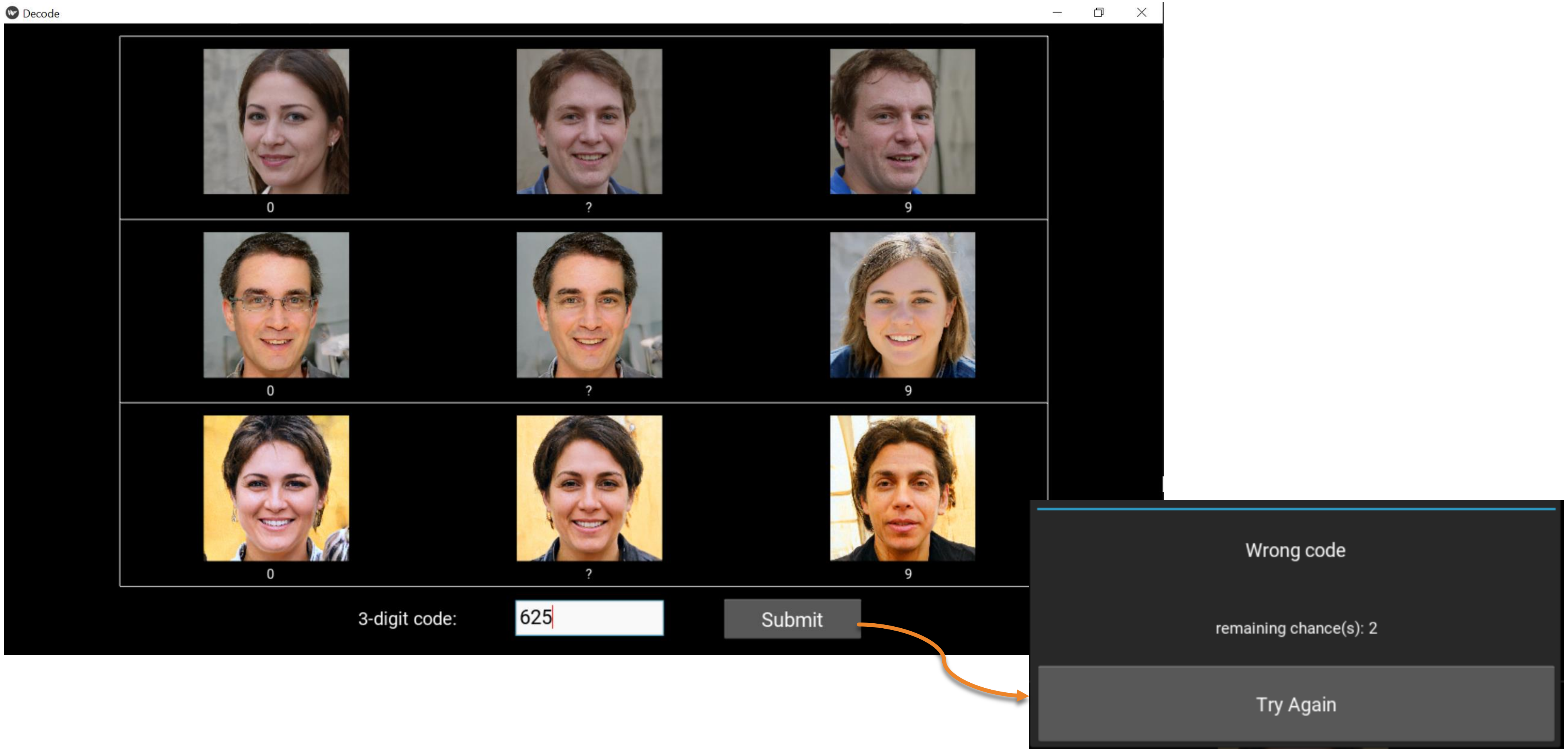
1. You are going to solve a puzzle (a 3-digit code).
2. You only have 3 chances.
3. After each attempt, there are some hints for you.



PART 3: GAME APPLICATION DEVELOPMENT

- Example
  - I st guess:

    625



3-digit code: 625

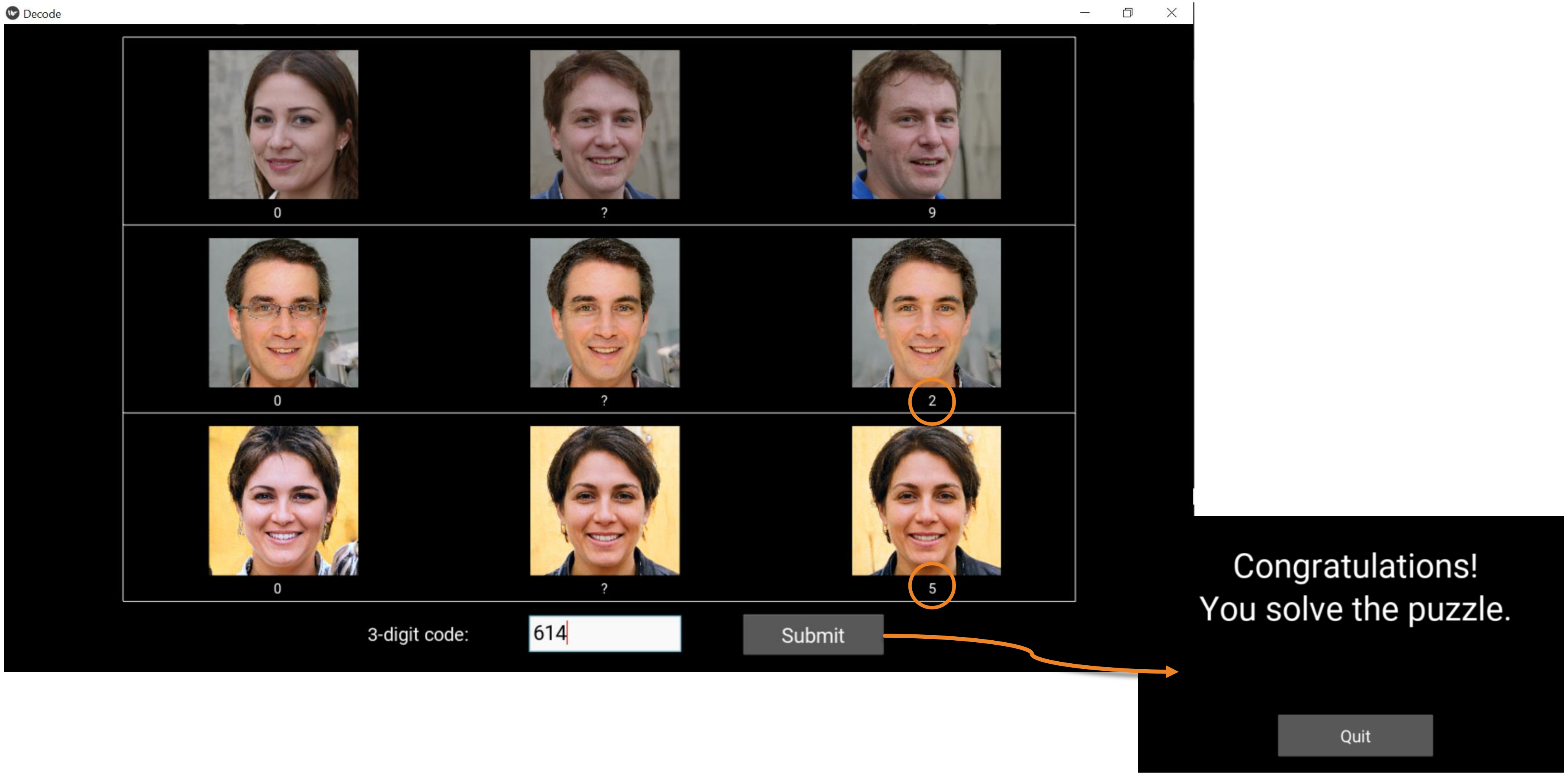
Wrong code

remaining chance(s): 2

PART 3: GAME APPLICATION DEVELOPMENT

- Example
  - 2nd guess:

    614



3-digit code: 614

Congratulations!
You solve the puzzle.

PART 4: FURTHER WORK

- Evaluation and Model Improvement
- Conditional GANs

THANK YOU !