기계 학습 기반 감정 인식 게임 개발

2016.06.03

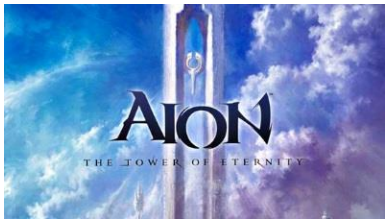
홍익대학교 게임학부 / 조교수

강신진

소개: 강신진

- **경력**
  - 홍익대학교 게임학부, 조교수 (2008-현재)
  - 엔씨소프트(NCsoft) (2006-2008)
  - 소니 컴퓨터 엔터테인먼트 코리아(Sony Computer Entertainment Korea) (2003-2006)

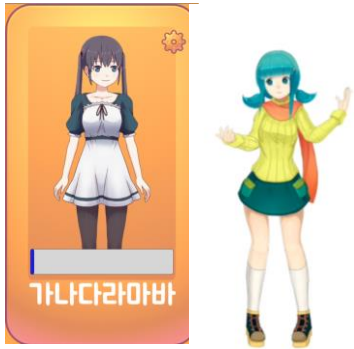
- **상용 게임**
  - 아이온 (AION), 와일드스타 (WildStar), 블레이드 앤 소울 (Blade and Soul) 외 10여 개의 PC, 모바일, 콘솔 게임 프로젝트 참여




Evolutionary Game Lab

EGLAB Research

ECA Content



Embodied Conversational Agent

+

Emotion Recognition

Multimodal Emotion Recognition

+

Emotional Model

Rule-based Model / DQN

Index

- Machine Learning in Computer Games

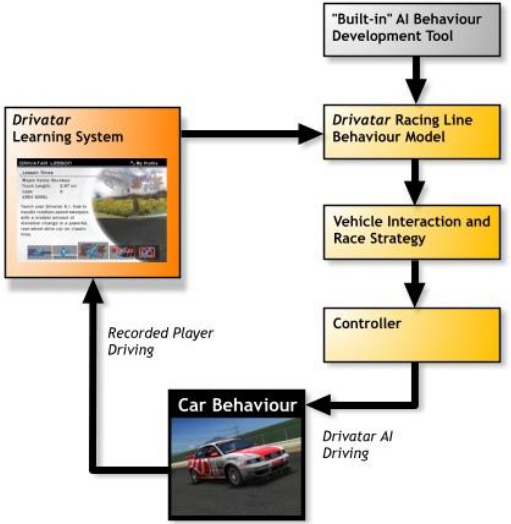
- Affective Computing in Computer Games

- [ML + Affective Computing] Game Development

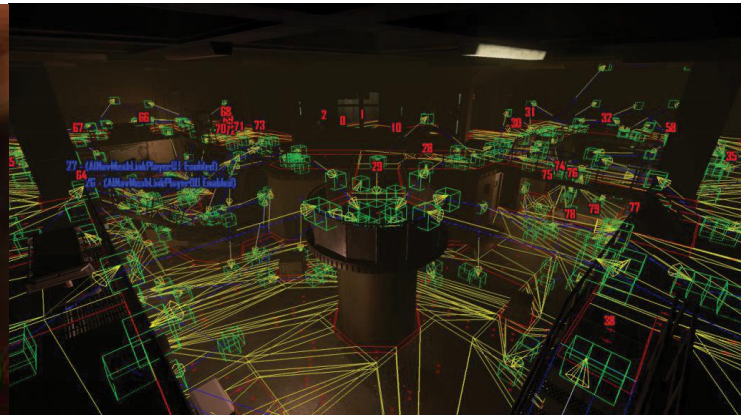
Machine Learning in Commercial Games

Black and White (Lionhead, 2001)

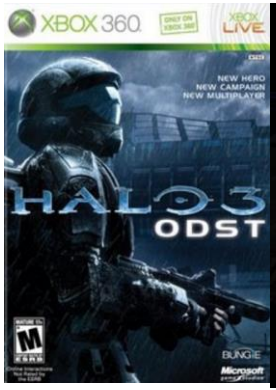
Forza Motorsport (Microsoft, 2005)



F.E.A.R. (Monolith Production, 2005)



HALO3 (Microsoft, 2007)



Virtua Fighter Ghost System (SEGA, 2015)

Tekken 5: Dark Resurrection (Namco, 2016)

Blade & Soul (NCsoft, 2016)

ML in Computer Games

Strengths

- Re-playability
- Emergent Game Play

Weakness

- Limitation of CPU resources for AI
- Low Development Priority
- Absence of Evaluation Function
- ML Content is not Fun
- ML Content is Unpredictable

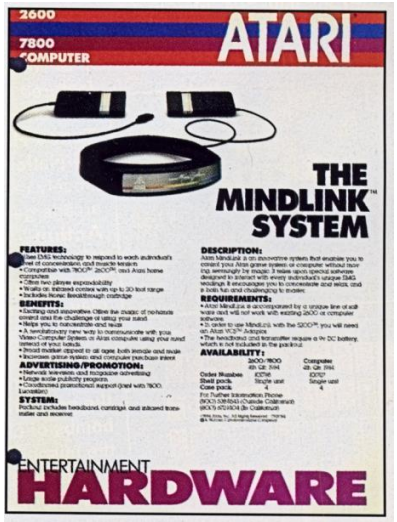
Index

- Machine Learning in Computer Games

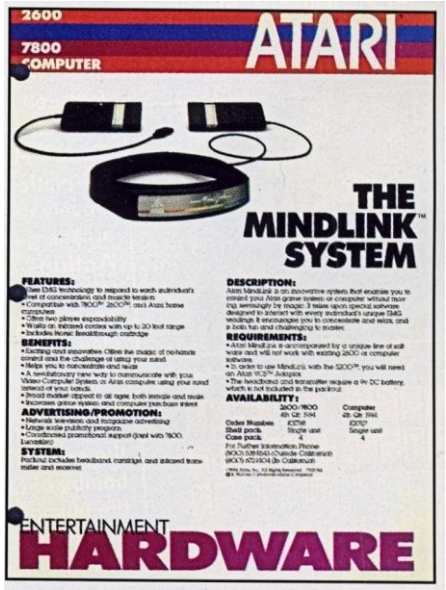
- Affective Computing in Computer Games

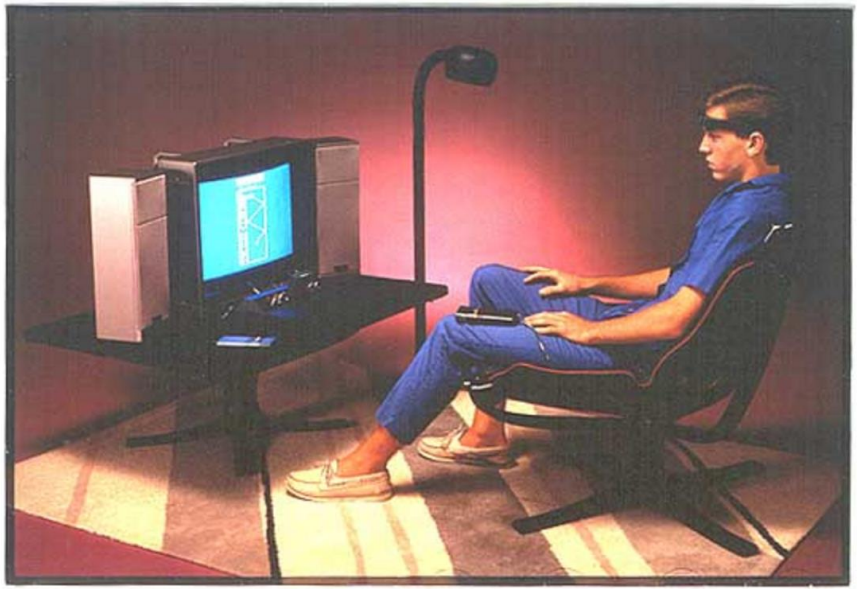
- [ML + Affective Computing] Game Development

Affective Computing in Commercial Games



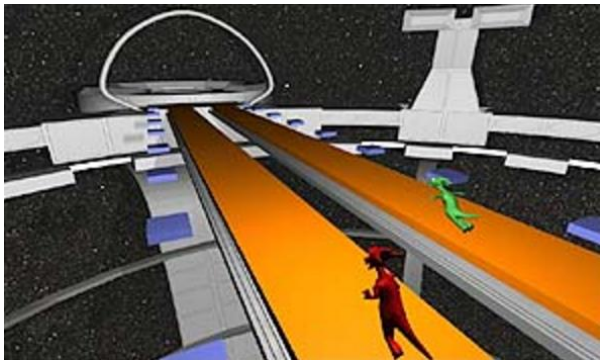
Mindlink (ATARI, 1984)



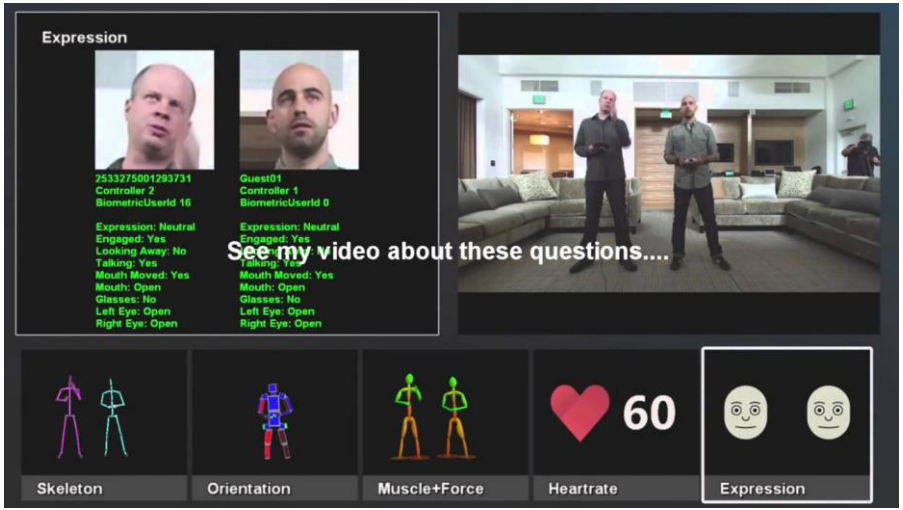


Tokimeki Memorial (Konami, 1997)

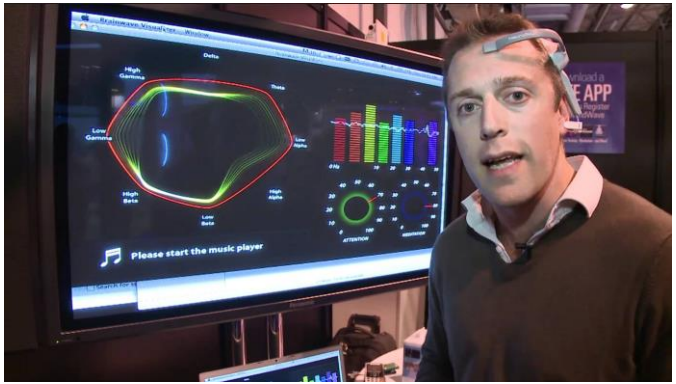
Relax to Win Game (McDarby, 2002)

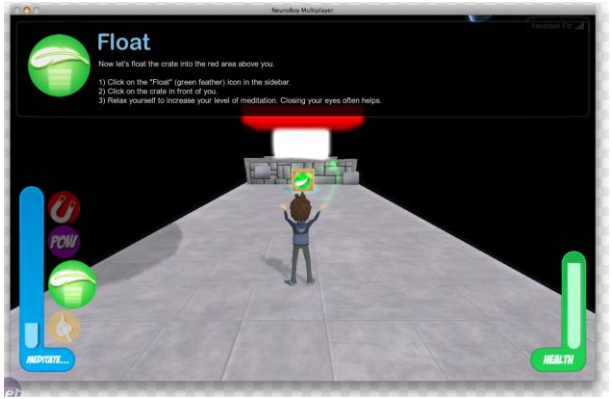


PlayStation Eyetoy / Microsoft Kinect (2003)



Mindwave (Neurosky, 2012)





Affective Computing in Computer Games

Strengths

- Providing New Experience
- Market Share Expansion

Weakness

- Intrusive Additional Hardware
- Low Recognition Rate of Player's Emotion

EGLAB
EVOLUTIONARY GAME

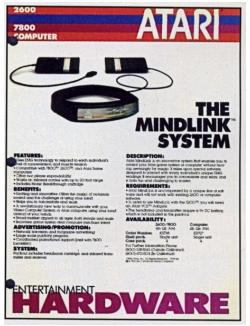
Index

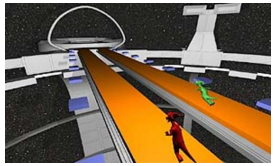
- Machine Learning in Computer Games

- Affective Computing in Computer Games

- [ML + Affective Computing] Game Development

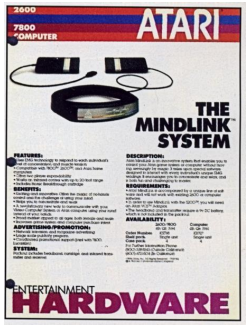
[ML + AC] in Computer Games

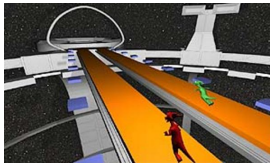




Commercially Successful Games

[ML + AC] in Computer Games





[ML + AC] in Computer Games

Strengths

- Re-playability
- Emergent Game Play

Strengths

- Providing New Experience
- Market Share Expansion

Weakness

- Limitation of CPU resources for AI
- Low Development Priority
- Absence of Evaluation Function
- ML Content is not Fun
- ML Content is Unpredictable

Weakness

- Intrusive Additional Hardware
- Low Recognition Rate of Player's Emotion

[ML + AC] in Computer Games

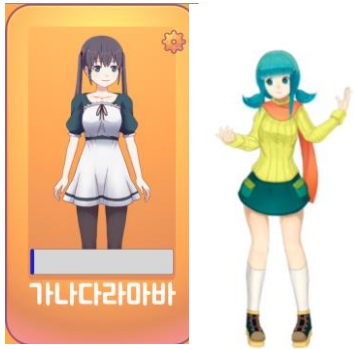
- Limitation of CPU resources for AI
- Low Development Priority
- Content is not Fun
- Content is Unpredictable
- Intrusive Additional Hardware
- Low Recognition Rate of Player’s Emotion
- Absence of Evaluation Function

Solutions

- Limitation of CPU resources for AI ➔ **AI Oriented Content**
- Low Development Priority ➔ **Long Term Project**
- Content is not Fun ➔ **Appropriate Genre**
- Content is Unpredictable ➔ **Non-Competitive Game**
- Intrusive Additional Hardware ➔ **Non-Intrusive/Default Hardware**
- Low Recognition Rate of Player's Emotion ➔ **ML with Big Data**
- Absence of Evaluation Function ➔ **Robust Emotional Model**

EGLAB Research

ECA Content



Embodied
Conversational Agent

+

Emotion Recognition

Multimodal Emotion
Recognition

+

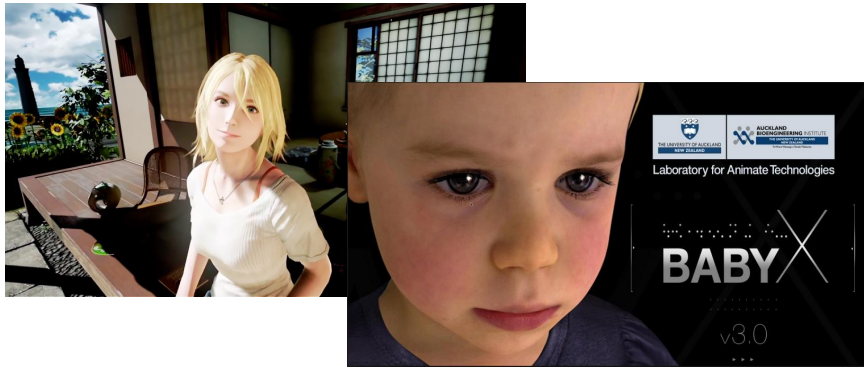
Emotional Model

Rule-based Model / DQN

EGLAB
EVOLUTIONARY GAME

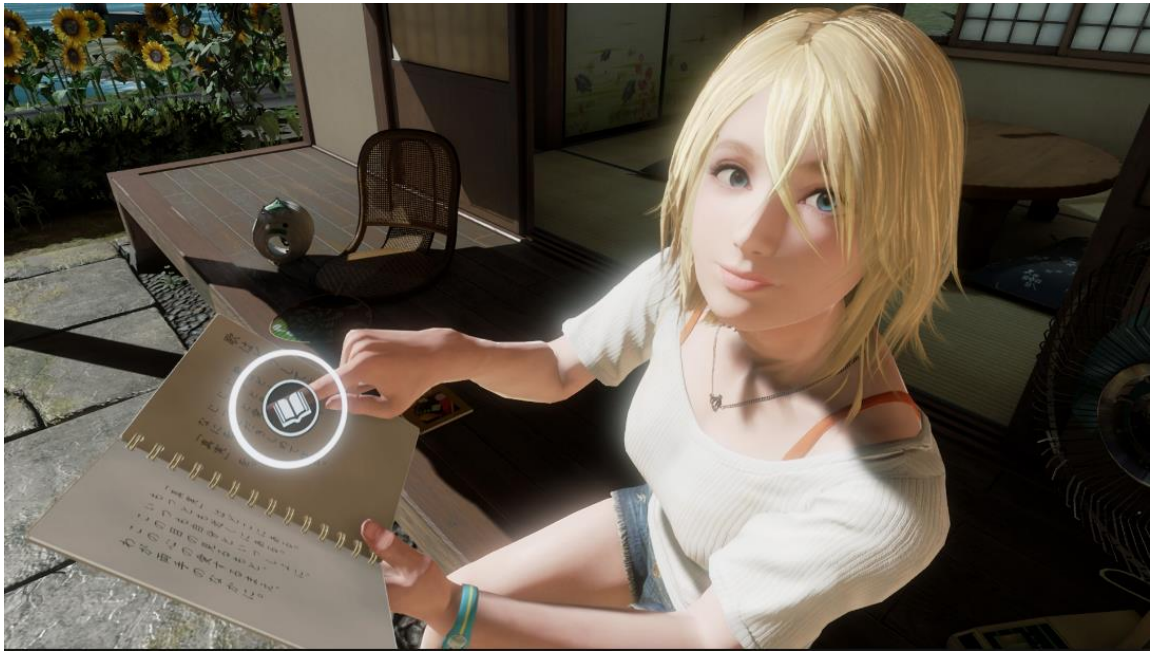
Embodied Conversational Agent (ECA)

GRETA, SEMAINE, HUMAINE (2007)



Summer Lesson, BabyX (2016)

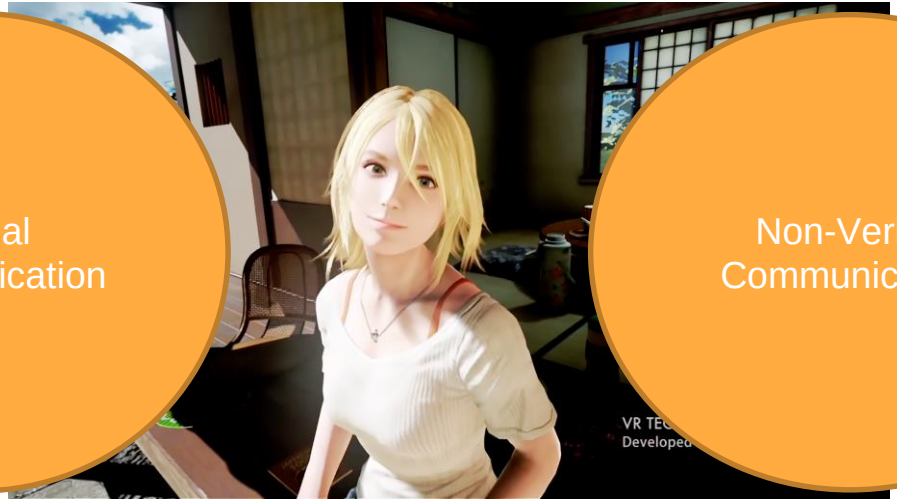
Summer Lesson (Namco, 2015)



BabyX (Sagar, 2016)

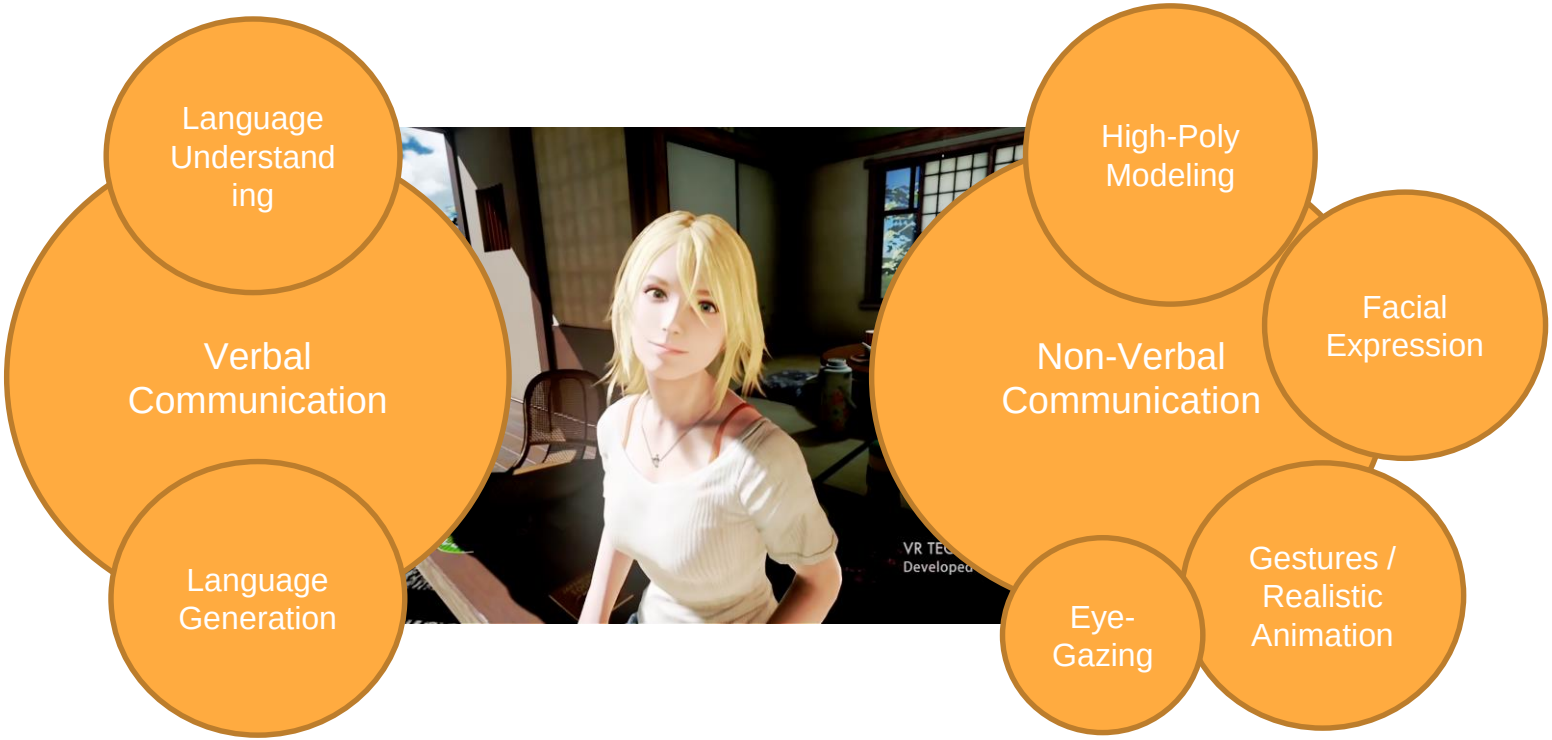
ECA Core Technologies

Verbal Communication

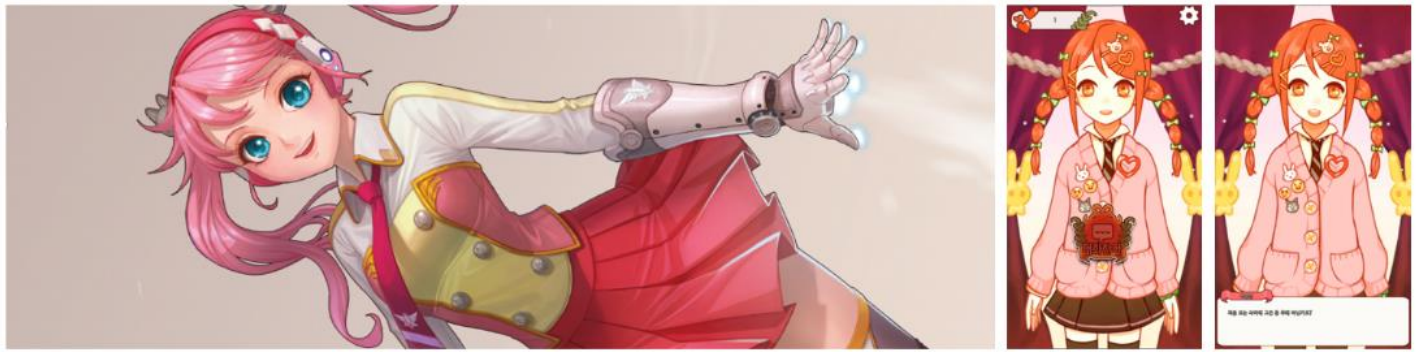


Non-Verbal Communication

ECA Core Technologies



Implementation Results



“Love Senor”

Implementation Results

Solutions

- Limitation of CPU resources for AI ➔ **AI Oriented Content**
- Low Development Priority ➔ **Long Term Project**
- Content is not Fun ➔ **Appropriate Genre**
- Content is Unpredictable ➔ **Non-Competitive Game**
- Intrusive Additional Hardware ➔ **Non-Intrusive/Default Hardware**
- Low Recognition Rate of Player's Emotion ➔ **ML with Big Data**
- Absence of Evaluation Function ➔ **Robust Emotional Model**

Hardware for Affective Computing

Intrusive H/W

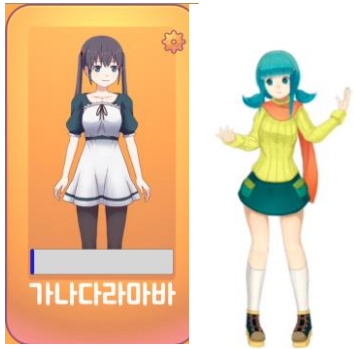
- EEG
- Heart Rate
- Depth Camera
- Brain Wave

Non-Intrusive H/W

- Multimodal Interface
- Keyboard
- Mouse
- Webcam

EGLAB Research

ECA Content



Embodied
Conversational Agent

+

Emotion Recognition

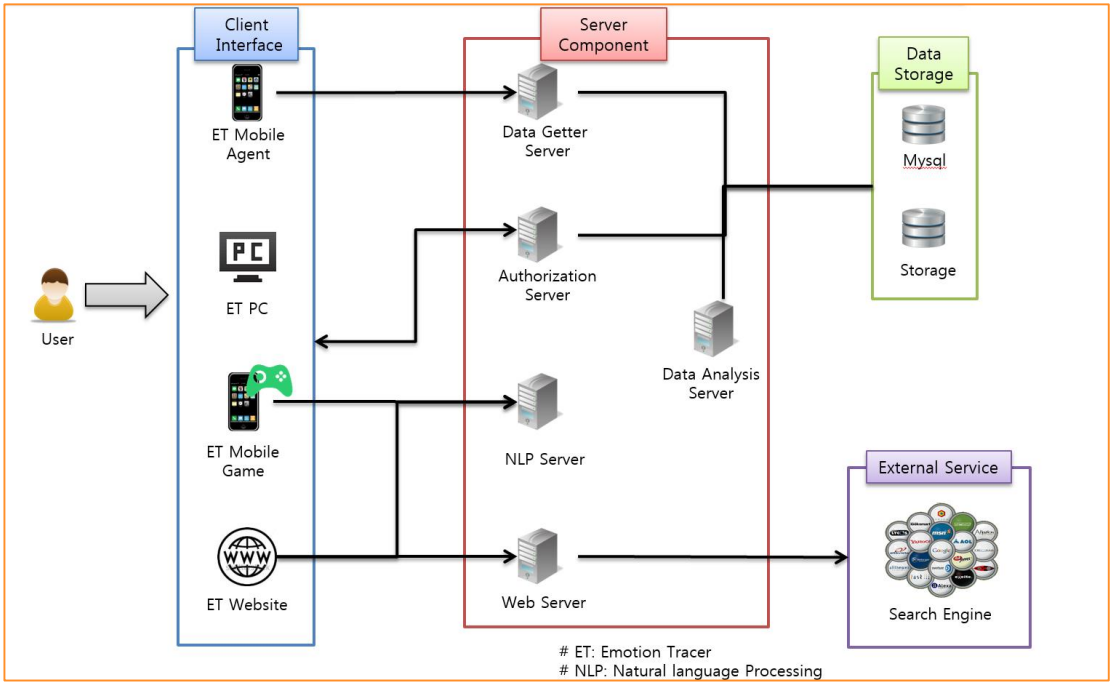
Multimodal Emotion
Recognition

+

Emotional Model

Rule-based Model / DQN

Implementation Result



"Emotion Recognition System with Multimodal Interface"

Implementation Result

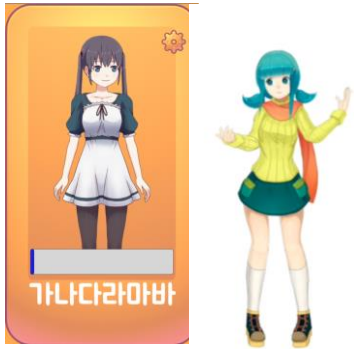
"Emotion Tracer Client"

Solutions

- Limitation of CPU resources for AI ➔ **AI Oriented Content**
- Low Development Priority ➔ **Long Term Project**
- Content is not Fun ➔ **Appropriate Genre**
- Content is Unpredictable ➔ **Non-Competitive Game**
- Intrusive Additional Hardware ➔ **Non-Intrusive/Default Hardware**
- Low Recognition Rate of Player's Emotion ➔ **ML with Big Data**
- Absence of Evaluation Function ➔ **Robust Emotional Model**

EGLAB Research

ECA Content



Embodied
Conversational Agent

+

Emotion Recognition

Multimodal Emotion
Recognition

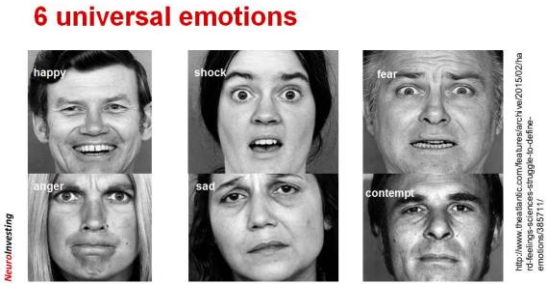
+

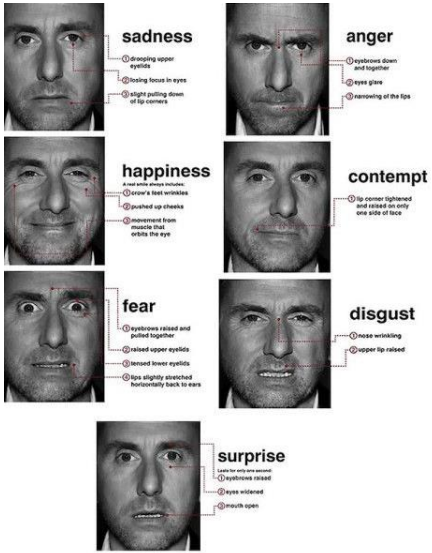
Emotional Model

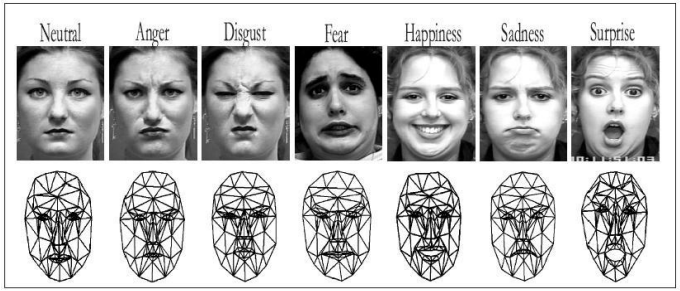
Rule-based Model / DQN

EGLAB
EVOLUTIONARY GAME

Discrete Emotion Model

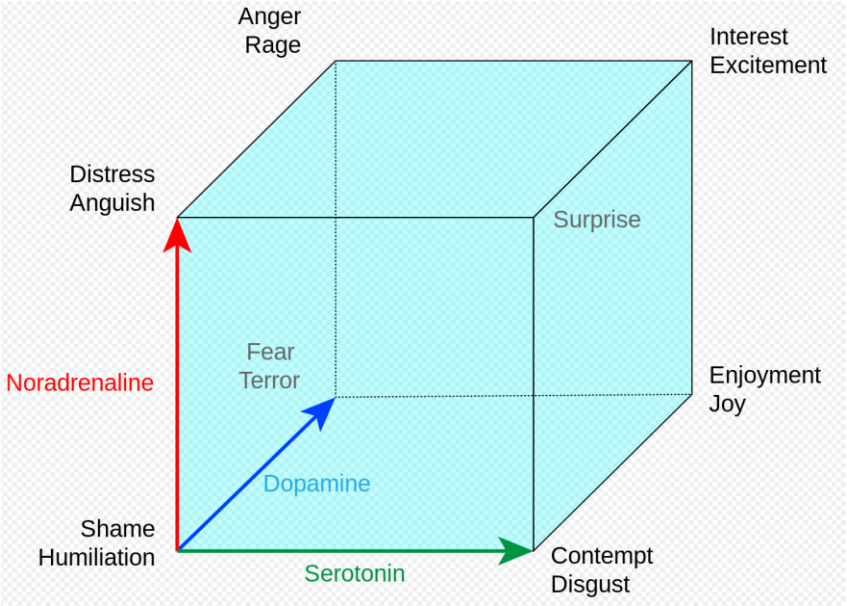


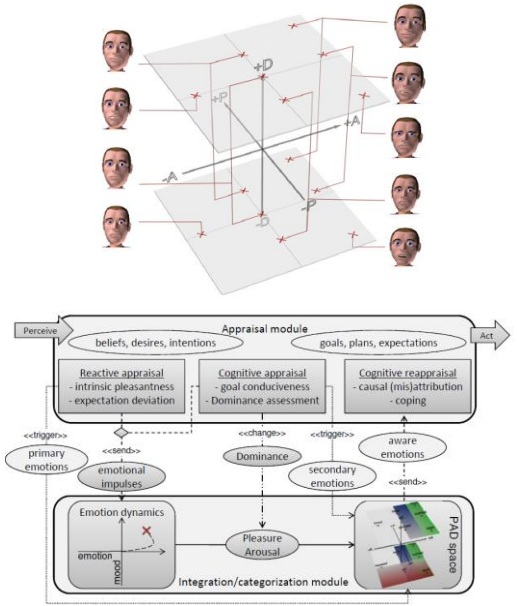




- Ekman’s 6 Universal Emotions

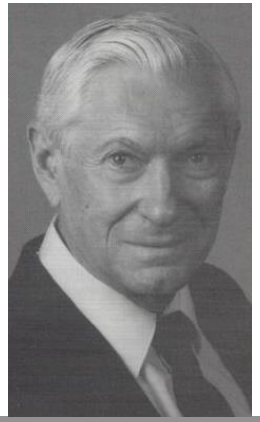
Dimensional Theory

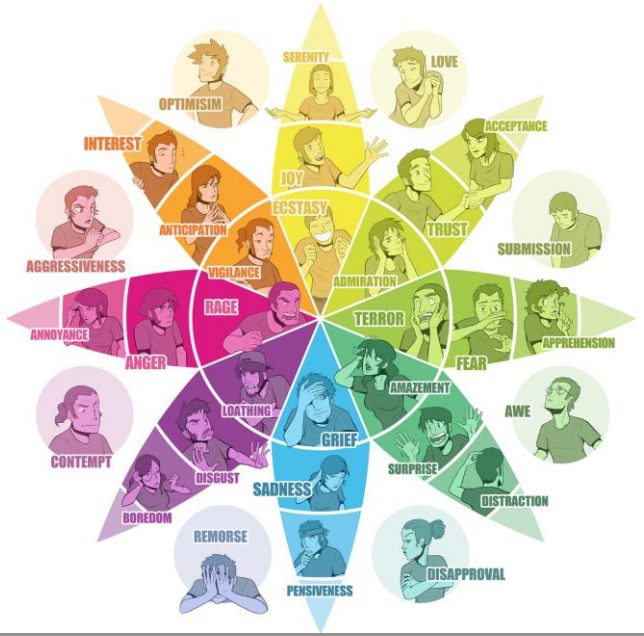


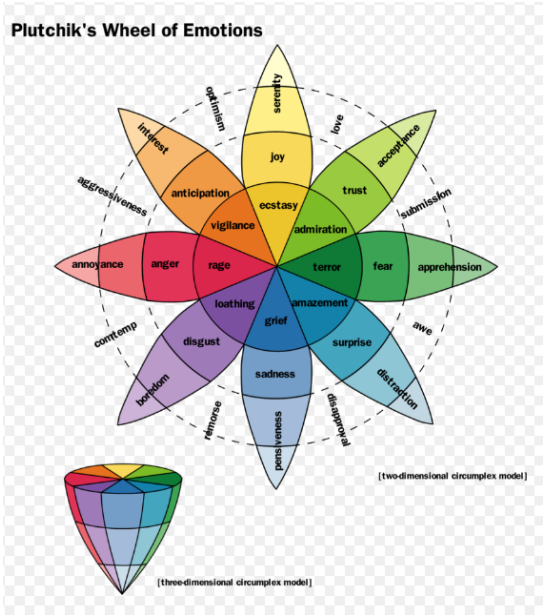


- The PAD (*Pleasure*, *Arousal*, *Dominance*) model, Albert Mehrabian and James A. Russell

Evolutionary Model

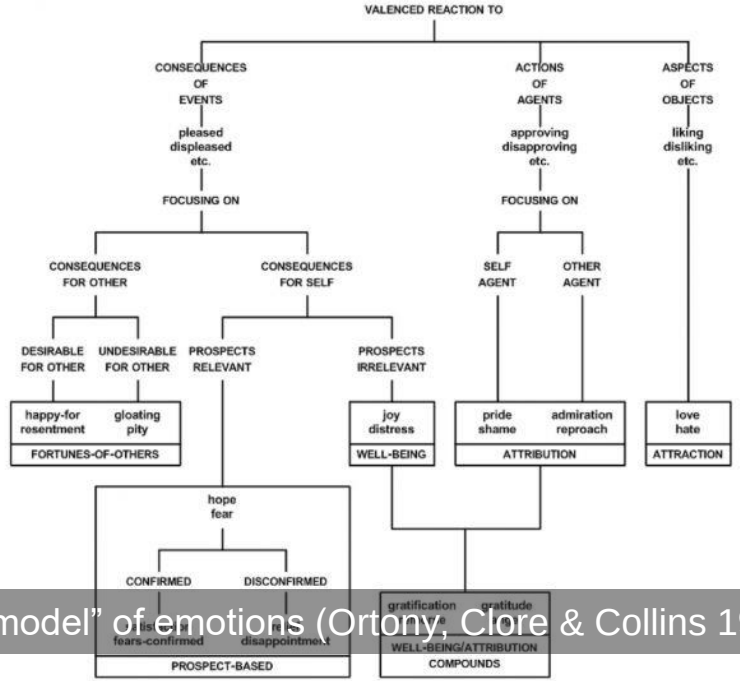


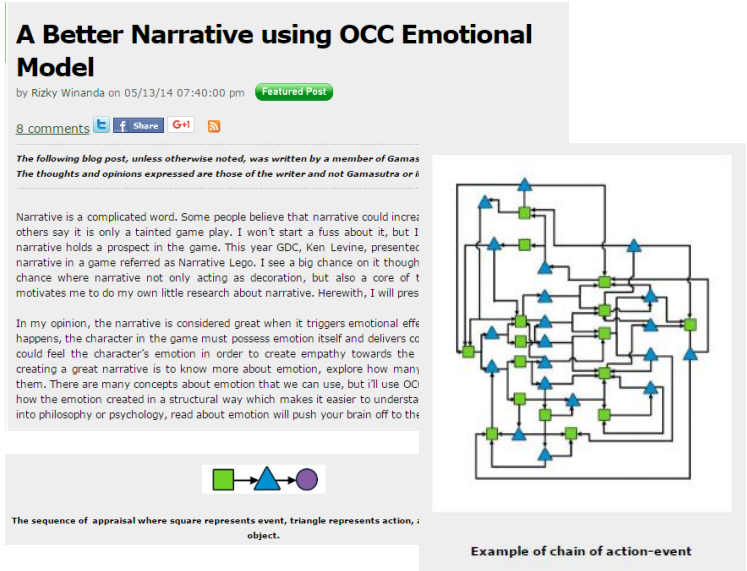




- Plutchik's wheel of emotions

OCC Model





- "OCC-model" of emotions (Ortony, Clore & Collins 1988)

Circuit Model (Neural Darwinism)

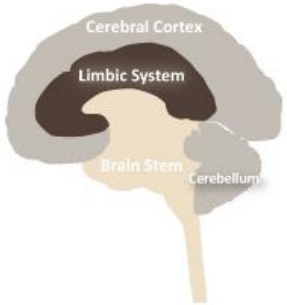
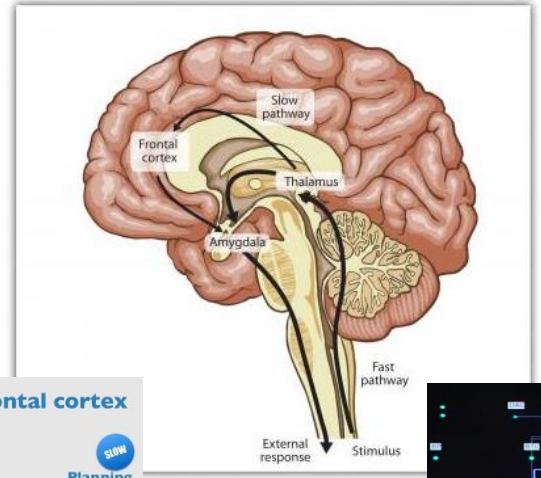
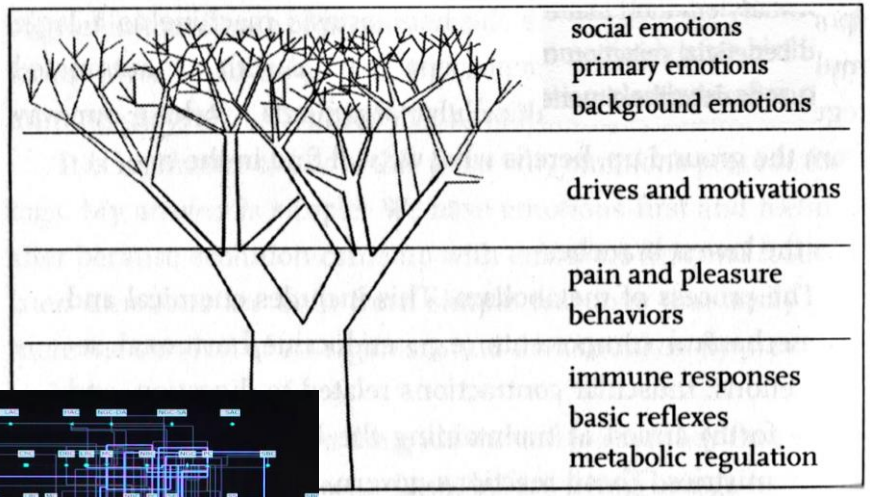
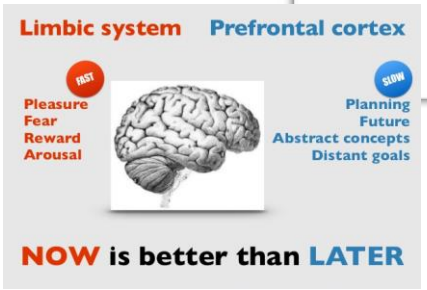
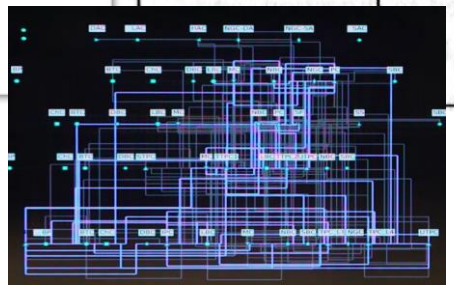


Figure10.3.Slow and Fast Emotional Pathways

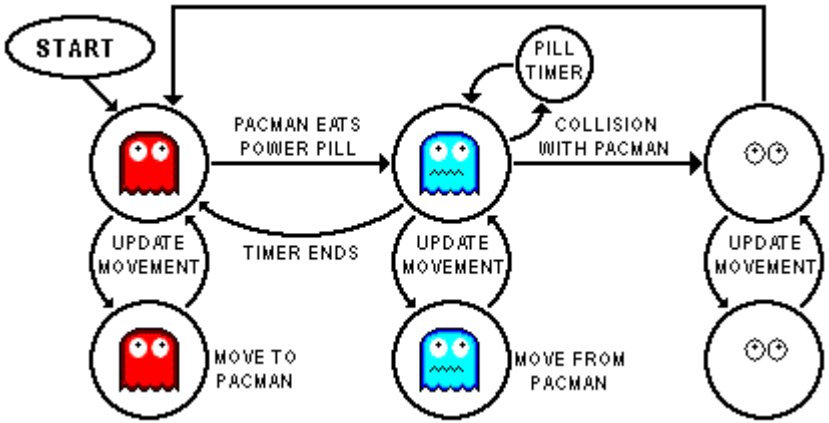




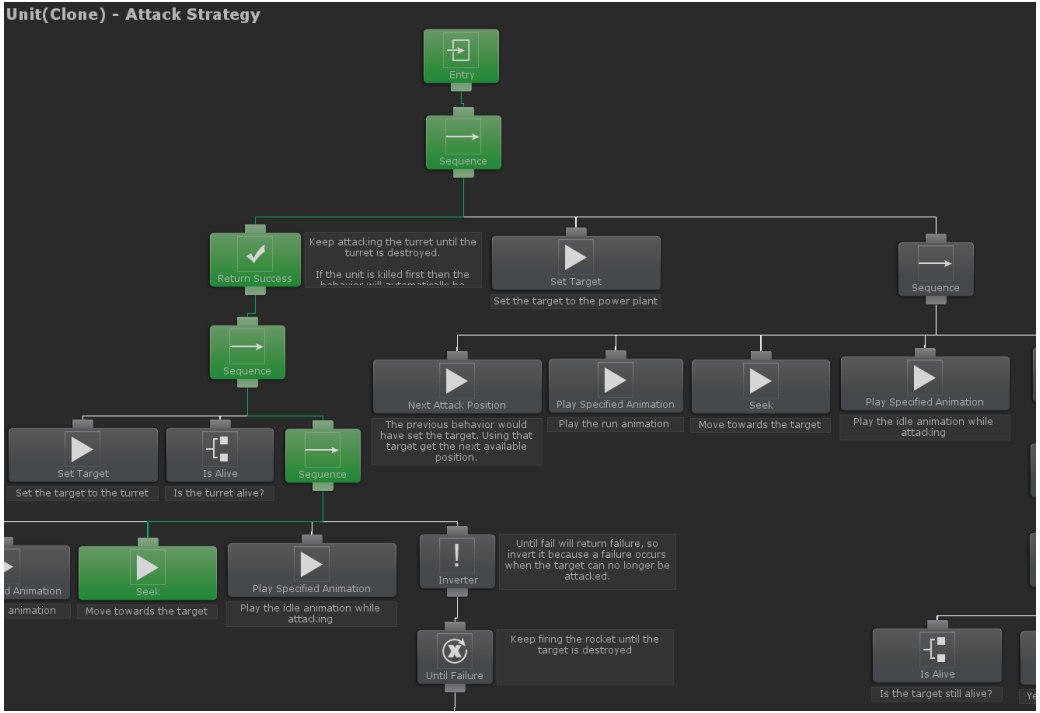




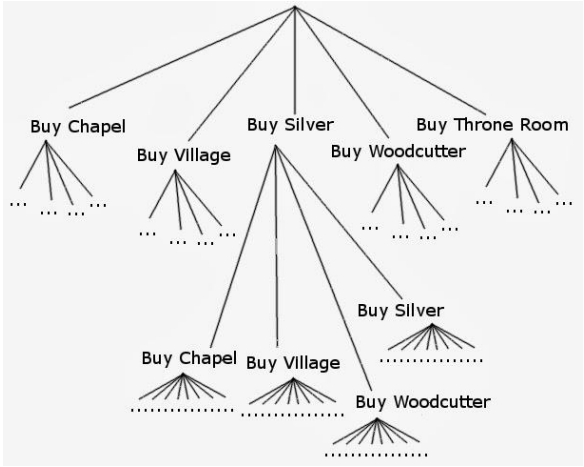
Finite State Machine (FSM)



Behavior Tree

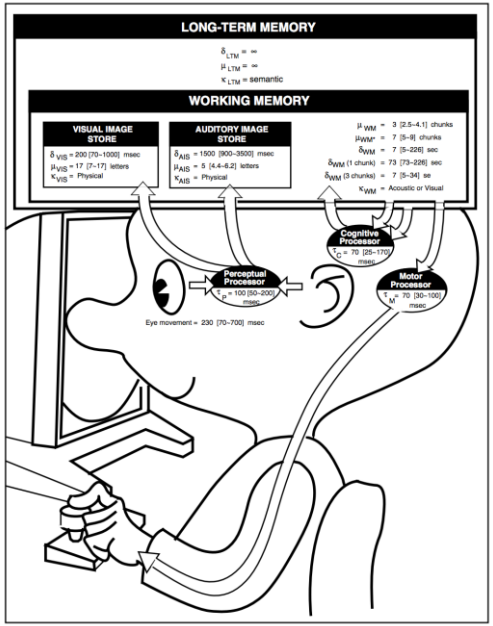


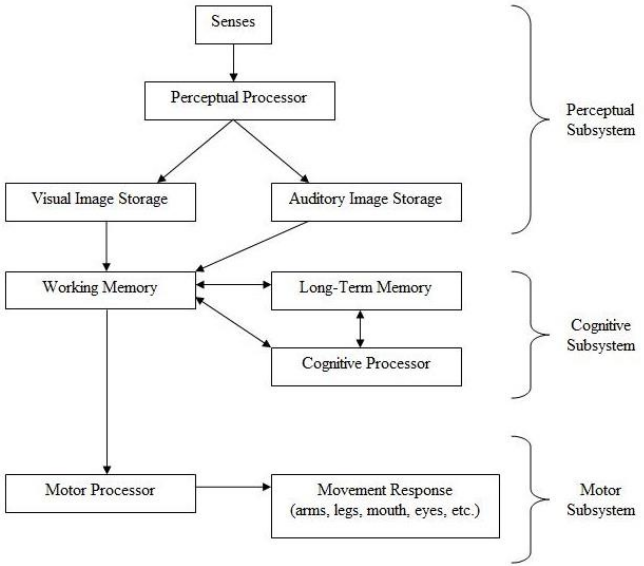
Decision Tree



MHP (Model Human Process)

Figure 1. Model Human Processor. Based on Card, Moran, and Newell (1983).





Parameter	Mean	Range
Eye movement time	230 ms	70-700 ms
Decay half-life of visual image storage	200 ms	90-1000 ms
Visual Capacity	17 letters	7-17 letters
Decay half-life of auditory storage	1500 ms	90-3500 ms
Auditory Capacity	5 letters	4.4-6.2 letters
Perceptual processor cycle time	100 ms	50-200 ms
Cognitive processor cycle time	70 ms	25-170 ms
Motor processor cycle time	70 ms	30-100 ms
Effective working memory capacity	7 chunks	5-9 chunks
Pure working memory capacity	3 chunks	2.5-4.2 chunks
Decay half-life of working memory	7 sec	5-226 sec
Decay half-life of 1 chunk working memory	73 sec	73-226 sec
Decay half-life of 3 chunks working memory	7 sec	5-34 sec

- Cognitive Model (Card, Moran, and Newell)

ACT-R (Adaptive Control of Thought)

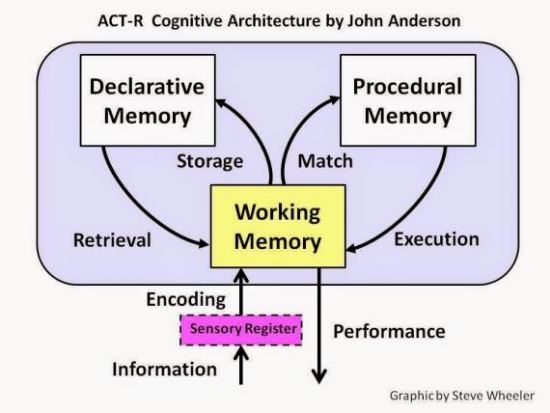
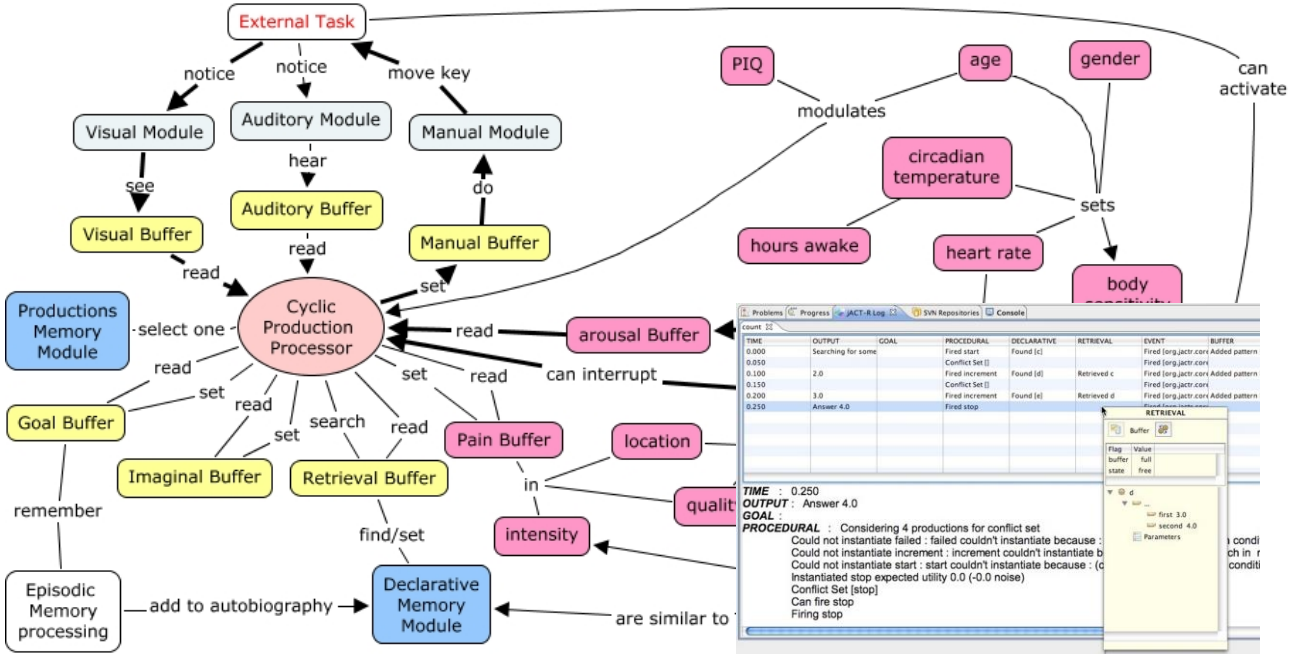
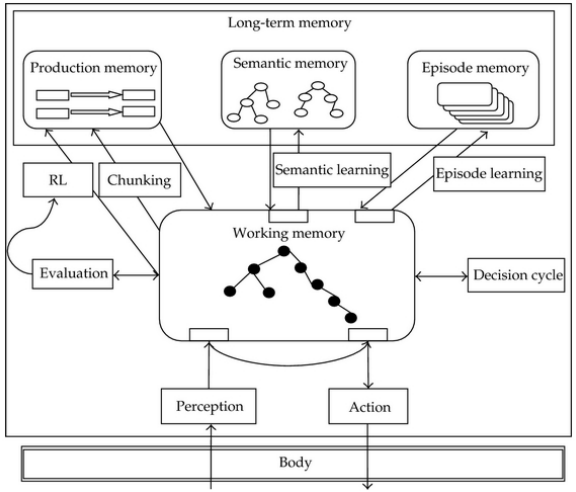


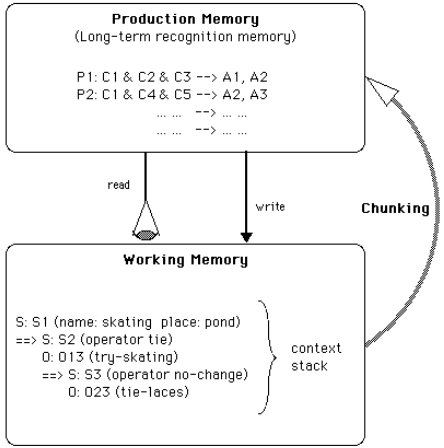
Figure 1. Sample knowledge tree where information becomes more structured as it reaches progressively higher stages of the pyramid.



Cognitive Architecture (Anderson & Lebiere)

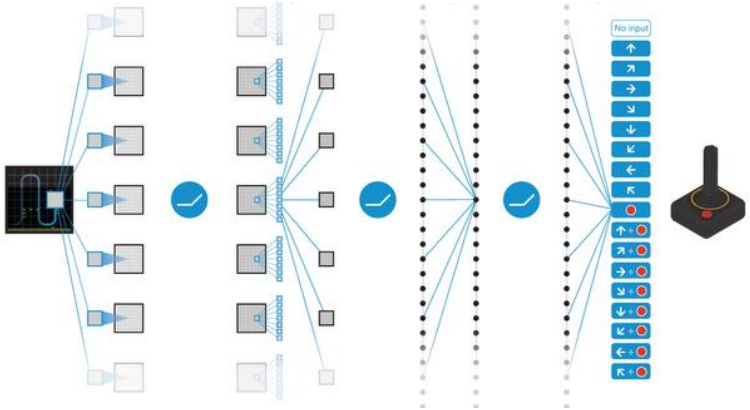
SOAR (State, Operator And Result)

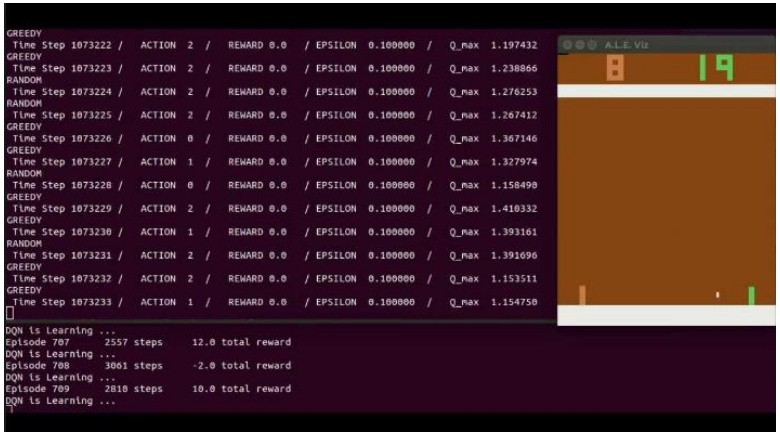




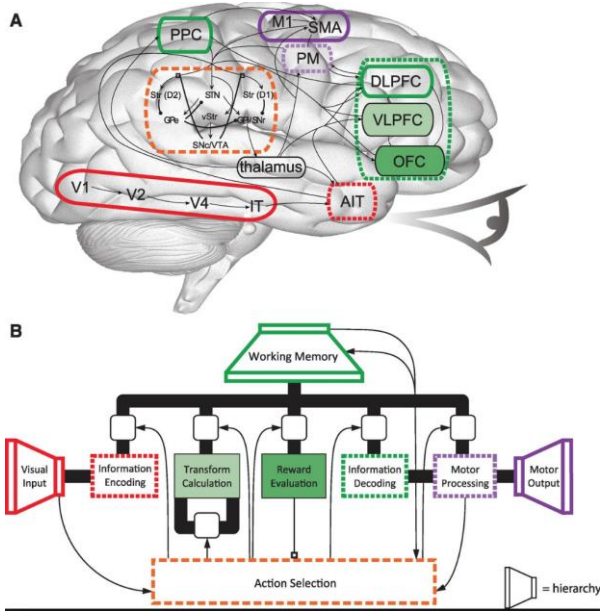
- Cognitive Architecture (Laird, 2008; Laird, Newell, & Rosenbloom, 1987; Newell, 1990)

Deep Q-Learning





Neural Engineering Framework



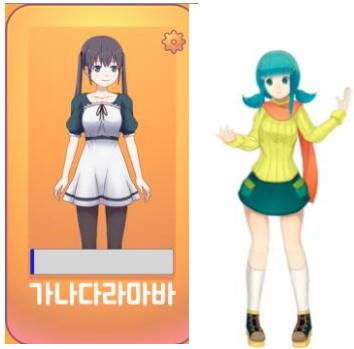

```
import nengo
model = nengo.Model("Communication Channel")
A = nengo.Ensemble(nengo.LIF(30), 1)
B = nengo.Ensemble(nengo.LIF(30), 1)
input = nengo.Node(0.5)
nengo.Connection(input, A)
nengo.Connection(A, B)
Bval = nengo.Probe(B,
    "decoded_output", filter=0.01)
sim = nengo.Simulator(model)
sim.run(1)
Bdata = sim.data(Bval)
```

```
import pynn.nest as pyNN
import numpy as np
pyNN.setup(timestep=1)
lif_params = {'tau_refrac': 2.0, 'tau_syn_E':100, 'tau_syn_I':100}
def encoders(n_neurons, dimensions):
    samples = np.random.randn(n_neurons, dimensions)
    norm = np.sum(samples * samples, axis=1)
    return np.sqrt(norm)[:, np.newaxis]
def gain_bias(max_rates, intercepts):
    x = 1.0 / (1 - np.exp((lif_params['tau_refrac'] - (1.0 / max_rates))
                          / pyNN.IF_cond_exp.default_parameters['tau_m']))
    gain = (1 - x) / (intercepts - 1.0)
    bias = 1 - self.gain * intercepts
    return gain, bias
A = pyNN.Population(30, pyNN.IF_cond_exp, lif_params)
Again, Abias = gain_bias(np.random.uniform(80, 100, 30), np.random.uniform(-1, 1, 30))
biasinputA = [pyNN.DCSource(amplitude=val) for val in Abias]
for i, pulse in enumerate(biasinputA):
    pulse.inject_into(A[i:i+1])
Aencoders = encoders(30, 1) * Again[:, np.newaxis]
B = pyNN.Population(30, pyNN.IF_cond_exp, lif_params)
Bgain, Bbias = gain_bias(np.random.uniform(80, 100, 30), np.random.uniform(-1, 1, 30))
biasinputB = [pyNN.DCSource(amplitude=val) for val in Bbias]
for i, pulse in enumerate(biasinputB):
    pulse.inject_into(B[i:i+1])
Bencoders = encoders(30, 1) * Bgain[:, np.newaxis]
inputnode = [pyNN.DCSource(amplitude=val) for val in 0.5 * Aencoders]
for i, pulse in enumerate(inputnode):
    pulse.inject_into(A[i:i+1])
# Decoder solving too long to include; assume we have Adecoder and Bdecoder
weights = []
for i in xrange(30):
    for j in xrange(30):
        weights.append((i, j, np.dot(Adecoder[i], Bencoder[j]), 1.0))
connection = pyNN.Projection(A, B, pyNN.FromListConnector(weights))
B.record('spikes')
pyNN.run(1000)
Bdata = numpy.zeros(1000)
for i in xrange(1000):
    Bspikes = B[i:i+1].getSpikes()[:,1].astype('int')
    Bdata[B_spikes] += Bdecoder[i]
decay = np.exp(-1.0 / 100)
Bdata[0, :] *= (1 - decay)
```

- Nengo: graphical and scripting based software package for simulating large-scale neural systems,
- SPAUN: 2.5 million simulated neurons

EGLAB Research

ECA Content



Embodied Conversational Agent

+

Emotion Recognition

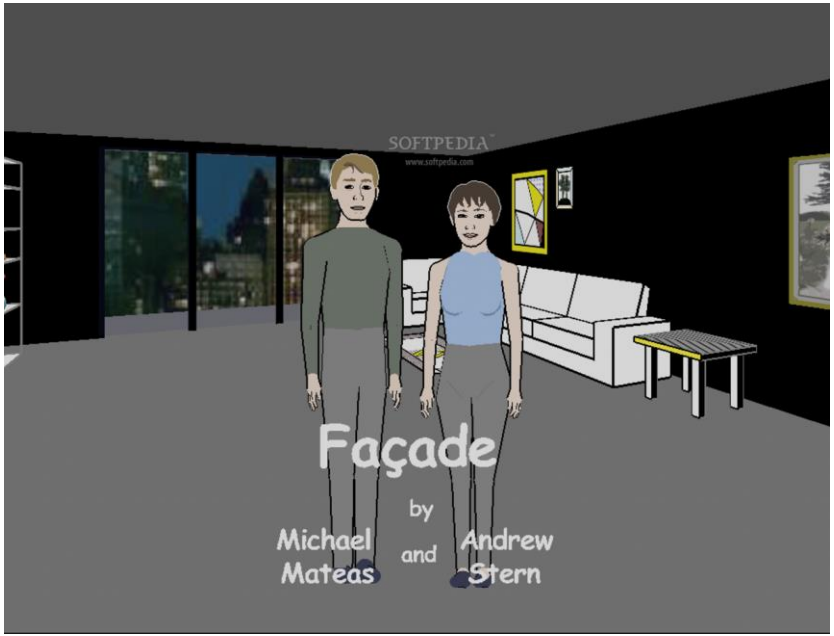
Multimodal Emotion Recognition

+

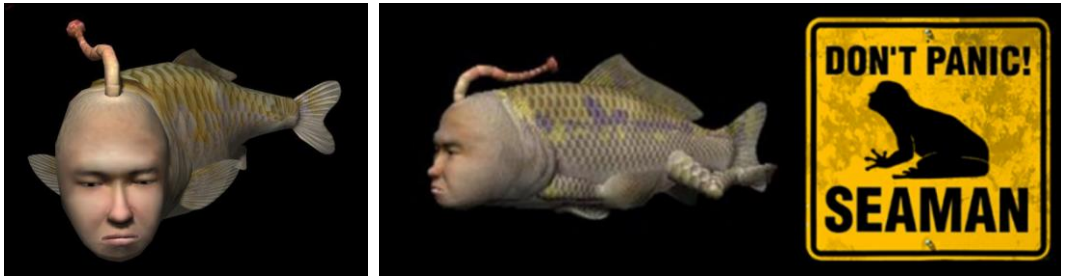
Emotional Model

Rule-based Model / DQN

Future Works: Façade (Mateas & Stern, 2004)



Future Works: Seaman (SEGA, 1999)



QnA

경청해 주셔서 감사합니다.

http://www.myeglab.com/