Game AI technologies for online games

Youichiro Miyake
y_miyake@fromsoftware.co.jp

2007/11/9

FROM SOFTWARE
All autonomous agents think and act by themselves
Game AI technologies for online games

Historical Reviews of Game AI technologies and the future

Youichiro Miyake
y_miyake@fromsoftware.co.jp

2007/11/9

FROM SOFTWARE
Address of thanks

I feel honored the invitation to this famous conference.

This is my second visit to Korea.
For the master’s degree,
I studied physics with three Korean students of Soul University.

Now I’m working with two Korean programmers in Tokyo, Japan,
and everyday I have lunch with a Korean friend.

All Korean friends are kind, polite, and wise!

Youichiro Miyake
On Monday ~ Friday, I’m working in Sibuya, Tokyo, Japan
On Saturday~Sunday, I’m playing in Akihabara and so on…
Contents

(Chapter.1)   Game AI Overview
(Chapter.2)   Game AI Technologies
(Chapter.3)   Summary
Chapter 1.

Game AI Overview
Game AI technologies will grow in this game age (2005~), and it will mature in the next age.
Game design & AI technologies

AI technologies expand the possibilities of game design!
Perspective of game AI

When you stand on fundamental game AI technologies, you can see further more…
Game AI Seminars in Japan

Many developers in Japan gather in the seminars
International Game Developer Association Japan!

<table>
<thead>
<tr>
<th>5th</th>
<th>Neural Net &amp; GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>4th</td>
<td>HFSM</td>
</tr>
<tr>
<td>3rd</td>
<td>multiagent</td>
</tr>
<tr>
<td>2nd</td>
<td>Planning</td>
</tr>
<tr>
<td>1st</td>
<td>Knowledge representation</td>
</tr>
</tbody>
</table>

In our company (once a week/〜 90 times)

IGDA Game AI seminars (once a two month/6 series)
How? Two categories in game AI

Game AI

AI in game system

For control of game system

- Level control
- Auto-generation of terrain

Difficult to be unified

Character AI

For control of NPC

Agent architecture

- C4 architecture
- Knowledge representation
- Planning
- FSM
- Genetic algorithm
Recent flow of game AI technologies

- **2000**
  - PS2, GC, Xbox
  - DC
  - C4 architecture (MIT Media Lab.)

- **2001**
  - Killzone (Guerrilla)

- **2002**
  - Halo (Bungie Studio)

- **2004**
  - Halo 2 (Bungie Studio)

- **2005**
  - Xbox360, PS3, Wii
  - F.E.A.R. (Monolith Production)

- **2006**
  - Chromehounds (FromSoftware)
Recent flow of game AI technologies

Game AI techniques are developed on the basis of previous works!
New game AI techniques are developed on the basis of previous works!
What you can do to develop more intelligent game AI for game designers and programmers

Research papers published in GDC and scientific journals! (in these time you can get most papers on internet.)

On the basis of these technologies explained in papers, develop techniques for game AI!

For a game you are developing, apply these technologies!

Think new game design
With new AI techniques!
Think how to implement with programming language you use!

In this lecture, game AI technologies are explained along these three steps!
For explaining all these technologies, I need $2.5 \times 6 = 15$ hours.
(In IGDA seminars in Japan, I had 15 hours lecture…)

But today I have only 1 hours,
I must speak as rapidly as I can for introducing all topics…

You should understand how to learn game AI techniques, rather than what I explained exactly!
Chapter 2.

Game AI technologies
Contents

1. **Agent architecture**  C4 Architecture  (MIT Media Lab.)
2. **Knowledge representation**  Killzone  (Guerilla)
3. **Event analysis**  Halo (Bungie Studio)
4. **Heuristic FSM**  Halo2 (Bungie Studio)
5. **Goal-oriented Action Planning**  F.E.A.R.  (Monolith)
6. **Multiagent system**  Chromehounds (FromSoftware)
Recent flow of game AI technologies

New game AI techniques are developed on the basis of previous works!
1. Agent architecture
In this lecture, game AI technologies are explained along these three steps:

1. **Research**
   - Research papers published in GDC and scientific journals!
   - (In these time you can get most papers on internet.)

2. **Study**
   - On the basis of these technologies explained in papers, develop techniques for game AI!

3. **Implement**
   - For a game you are developing, apply these technologies!
   - Think new game design
   - With new AI techniques!
   - Think how to implement with programming language you use!
References for C4 Architecture

(1) MIT Media Lab Synthetic Characters Group, http://characters.media.mit.edu/


Figures on following pages are from these references.
Three Steps

Research papers published in GDC and scientific journals! (in these time you can get most papers on internet.)

On the basis of these technologies explained in papers, develop techniques for game AI!

For a game you are developing, apply these technologies!

Think new game design With new AI techniques! Think how to implement with programming language you use!

In this lecture, game AI technologies are explained along these three steps!
What is Agent?

Three conditions for “agent”

- **Sensor** to get information of environment
- **Decision-making** to decide its will
- **Effecter** to action in the world

Sensor → Decision-making → Effect on the world
Agent of game

Game world

Sense the game world around himself

Effect on the game world through his body
Flow of information between an agent and a game world

- Sense the game world around himself
- Effect on the game world through his body
Agent architecture

NPC’s intelligence

Cognition Process → Decision-making → behavior-generation process

Memory

Status

Sensor

Effector

Knowledge representation
World representation
(data extracted and abstracted from game world)

Game World

Body

From a view of NPC

sense

action

modeling

interaction

time
Agent architecture

NPC’s intelligence

Cognition Process → Decision-making → behavior-generation process

Memory

Sensor

Effector

Knowledge representation
World representation
(data extracted and abstracted from game world)

Game World

Body

From a view of NPC

(modeling)

sense

action

interaction

time
Agent architecture

NPC’s intelligence

Technology
(implemented in each module)

Sensor

Effecter

Status

Body

effect

interaction

Knowledge representation
World representation
(data extracted and abstracted for)

From a view of NPC

sense

action

time
Three Steps

Research papers published in GDC and scientific journals! (in these time you can get most papers on internet.)

On the basis of these technologies explained in papers, develop techniques for game AI!

For a game you are developing, apply these technologies!

Think new game design With new AI techniques! Think how to implement with programming language you use!

In this lecture, game AI technologies are explained along these three steps!
C4 Architecture (2001)

AI for virtual creature

MIT Media Lab.
Synthetic Characters Group

D. Isla, R. Burke, M. Downie, B. Blumberg (2001),
“A Layered Brain Architecture for Synthetic Creatures”,
http://characters.media.mit.edu/Papers/ijcai01.pdf
(One of the most important papers for game AI)

Real AI of virtual dog who lives in virtual world was developed,
and this architecture was applied to F.E.A.R.
C4 Architecture

Real AI for virtual creature in virtual world

Example) Duncan

“Duncan” have intelligence and memories and a player can teach him through his voice.
MIT Media Lab. C4 Architecture

NPC’s intelligence

Cognitive tree
Action Selection
Motor system

PercepMemoryObjects
Working Memory

Blackboard

World Representation

Game world
Body

Sensor

Motion

Interaction

Time
Information extracted and classified through percept tree is stored in Working Memory sequentially.
What do memories enable?
= Prediction & surprise

Even if an object becomes behind a wall, he can follow its orbit by using his memories!

**But if an ball is disappeared against his prediction , Duncan becomes surprised!**
MIT Media Lab. C4 Architecture

- Percept tree
- Memories

NPC’s intelligence

Action selection

Motor system

Blackboard

Sensor

Motion

World

Interaction

- Prediction & Surprise
- Human-like Mistake
- Learning
- Proper perception

Time
Recent flow of game AI technologies

PS2, GC, Xbox

DC

2005

Xbox360, PS3, Wii

(Next gen.)

Time

Agent architecture
(MIT Media Lab.)

Event driven
(Bungie Studio)

Heuristic FSM
(Finite State Machine)
(Bungie Studio)

Goal oriented
Action Planning
(Monolith Production)

Hierarchical planning
(FromSoftware)

World representation
(Guerrilla)

New game AI techniques are developed on the basis of previous works!
Store & Extract

Store

World information

PercepMemoryObjects
Working Memory

Extract

New information
Contents

1. **Agent architecture**  C4 Architecture (MIT Media Lab.)
2. **Knowledge representation**  Killzone (Guerilla)
3. **Event analysis**  Halo (Bungie Studio)
4. **Heuristic FSM**  Halo2 (Bungie Studio)
5. **Goal-oriented Action Planning**  F.E.A.R. (Monolith)
6. **Multiagent system**  Chromehounds (FromSoftware)
2. Knowledge representation
Killzone

◆ 内容 : FPS  
◆ 開発元 : Guerrilla  
◆ Publisher: SCEE (Europe) , SEGA (Japan)  
◆ Hardware: PS2
◆ 出版年 ： 2004-2005 (Million seller)  
◆ Online (not for Japanese version)
References for Killzone


*Figures on following pages are from these references.*
Killzone NPC

Strategic movement in a large map

- Study

In battle field
Max 7 vs 7
(one of 7 is player)

- My team
- Human
- Strong Human
- AI Enemies
Agent architecture

NPC’s intelligence

Sensor

Knowledge representation
World representation

Game World

Effector

Body

interaction

time
Knowledge representation

Game world → Knowledge Representation (data) → Smiley

It should be prepared by a developer
World representation

Knowledge representation of global information of the game world
(Example) Waypoint data

Terrain polygon data

Waypoint data

I can find a way!
(Example) Waypoint data
point coordinate + **intensity of lighting**

Terrain polygon data  \( \rightarrow \)  Waypoint data + **Intensity of lighting**

I can find a light way!
Enemies come from east!
I can find a dark way which cannot be seen from east!

(Example) Waypoint data
point coordinate + intensity of lighting + LOS(line of sight) from east

Terrain polygon data
Waypoint data
+ Intensity of lighting
+ LOS(line of sight) from east
Killzone: Look-up table for waypoint visibility

LOS (line of sight) with a quick calculation
How to use world representation

Tactical position picking
Step 1 selected nearby waypoint, annotated with proximity
Step2 Annotations for positions with a line-of-fire to primary threat

40 … LOF only to crouched stance
20 … LOF both to standing and crouched stance
Step3 Annotations for positions with cover from the secondary threats
Step4  Annotations for positions inside the preferred fighting range
Step 5: Adding up Step 1~4 annotations yields the best attack position.

Dynamic tactical position picking
Dynamic tactical position picking

Look-up table for waypoint visibility

Knowledge representation
World representation

Game World

Body

tactical position picking
- tactical path finding
- suppression fire

interaction

time

time
New game AI techniques are developed on the basis of previous works!
1. Agent architecture  C4 Architecture (MIT Media Lab.)
2. Knowledge representation  Killzone (Guerilla)
3. Event analysis  Halo (Bungie Studio)
4. Heuristic FSM  Halo2 (Bungie Studio)
5. Goal-oriented Action Planning  F.E.A.R. (Monolith)
6. Multiagent system  Chromehounds (FromSoftware)
Reference for Halo & Halo2

- Damian Isla (2005), “Dude, where’s my Warthog? From Pathfinding to General Spatial Competence”,
  http://www.aiide.org/aiide2005/talks/isla.ppt

- Damian Isla (2005), Handling Complexity in the Halo 2 AI, Game Developer's Conference Proceedings.,

- Jaime Griesemer(2002), The Illusion of Intelligence: The Integration of AI and Level Design in Halo,

  http://www.stuffo.com/halo2-ai.htm (現在はclosed)

Figures on following pages are from these references.
Halo (2002)

- Game: SF/FPS
- Developer: BUNGIE Studio
- Publisher: Microsoft
- Hardware: Xbox, Windows, Mac
- Publishing year: 2002
Halo NPC Architecture

**NPC’s intelligence**

- **Situation Analysis**
- **Decision-making**
- **Motion Control**

- **Emotion (only for dramaturgy)**
- **Working Memory**

---

- **Sensor**
- **Motion**

---

**Game world**

- **Interaction**

---

**Time**
Halo AI decision-making

Simple logic for selecting an action for each character from a situation.

Level designers’ work

Each behavior has a trigger to cause. An event causes some numbers of behaviors.
Halo NPC Architecture

- Situation analysis
- Decision-making logic
- Emotion

NPC’s intelligence

Emotion (only for dramaturgy)

Working Memory

Sensor

Motion

- Response to an event
- Proper reaction of each character
- Emotion & Dialogues

Event
“Enemy found!”

Time
Recent flow of game AI technologies

New game AI techniques are developed on the basis of previous works!
Contents

1. Agent architecture  C4 Architecture  (MIT Media Lab.)
2. Knowledge representation  Killzone  (Guerilla)
3. Event analysis  Halo (Bungie Studio)
4. Heuristic FSM  Halo2 (Bungie Studio)
5. Goal-oriented Action Planning  F.E.A.R.  (Monolith)
6. Multiagent system  Chromehounds (FromSoftware)

- Game: SF/FPS
- Developer: BUNGIE Studio
- Publisher: Microsoft
- Hardware: Xbox, Windows, Mac
- Publishing year: 2004
Halo2 NPC Architecture

NPC’s intelligence

Situation Analysis

DAG HFSM

Emotion (only for dramaturgy)

Working Memory

Sensor

Motion

Time

Event

“Enemy found!”
“Friend dead”
“I was shot!”
“Player shot!”

Knowledge representation/
World representation

Game world

Body

Interaction

Time
Decision-making by HFSM (Hierarchical FSM)

DAG (Directed acyclic graph) = behavior tree

- **Root**
- **Engage**
  - Retreat
  - Self-preservation
    - Cover
    - Guard
  - Engage
    - Charge
    - Vehicle fight
    - Fight
    - Search
    - Guard
    - Vehicle strafe
      - Melee
      - Presearch
      - Uncover
      - Investigate
  - Uncover
- **Idle**
  - Guard

0th layer | 1st layer | 2nd layer | 3rd layer
How to use HFSM

Order & Style

Define conditions for area transition

Behavior lists that NPC can execute

Area

Zone

Position
**Halo2 NPC Architecture**

- **Situation Analysis**
  - HFSM
  - Emotion (only for dramaturgy)
  - Working Memory

- **Sensor**
  - 3-layered World representation

- **Motion**
  - for each character
  - for each situation
  - for each area
  - unified method (=HFSM)
    - (-> Scalable AI)

Event:
- “Enemy found!”
- “Friend dead”
- “I was shot!”
- “Player shot!”

Time flow:

- Input
- Output

- Scalable AI
New game AI techniques are developed on the basis of previous works!
Contents

1. **Agent architecture**  C4 Architecture (MIT Media Lab.)
2. **Knowledge representation**  Killzone (Guerilla)
3. **Event analysis**  Halo (Bungie Studio)
4. **Heuristic FSM**  Halo2 (Bungie Studio)
5. **Goal-oriented Action Planning**  F.E.A.R. (Monolith)
6. **Multiagent system**  Chromehounds (FromSoftware)

Goal-Oriented Planning
What is Goal-Oriented Planning?
What is “goal-oriented”?
AI who can find his purpose from its environment!

For video games, developer must prepare some goals for AI, And AI select suitable one of them
Goal-oriented agent in video games

(1) Developer must prepare some goals for AI in development.

(2) AI select one of them, and act for its goal.

I select one goal, and achieve it!
What is planning?

Fundamental Concept: Initial State, Goal, Planner
Goal oriented planning Agent

Goals □ Planning

What?  How?

= an agent who decides what he should do and how he should do it.

Planning

Goals

Attack player

Protect friend

Protect base
Planning has two methods

1. **chaining**
   Planner chains actions automatically.

2. **Hierarchical**
   A big goal calls some goals, and its goal calls smaller goals…
Contents

1. Agent architecture  C4 Architecture  (MIT Media Lab.)
2. Knowledge representation  Killzone  (Guerilla)
3. Event analysis  Halo  (Bungie Studio)
4. Heuristic FSM  Halo2  (Bungie Studio)
5. Goal-oriented Action Planning  F.E.A.R.  (Monolith)
6. Multiagent system  Chromehounds  (FromSoftware)

Goal-Oriented Planning

Game: Horror FPS
Developer: Monolith Production
Publisher: SIERRA
Hardware: Windows, PS3
Publishing year: 2004
References for F.E.A.R.

- Mat Buckland, “Programming Game AI by Example”, Chapter 9, WORDWARE publishing

*Figures on following pages are from these references.*
F.E.A.R NPC AI architecture

NPC’s intelligence

Cognitive Process
Goal-oriented Action Planning
Motion-generation

Blackboard

Sensor

Knowledge representation/
World representation

Game world

Body

Interaction

Time
Representation form of behavior for planning by chaining

Precondition = necessary condition for executing the behavior
Effect = effect caused by the behavior
Planning system for F.E.A.R.

Simple representation of agent’s world

20 symbols for representing agent’s world

Symbol

- kSymbol_AtNode
- kSymbol_TargetIsAimingAtMe
- kSymbol_WeaponLoaded
- kSymbol_WeaponArmed
- kSymbol_TargetIsSuppressed
- kSymbol_UsingObject
- kSymbol_TargetIsDead
- kSymbol_RidingVehicle
- kSymbol_AtNodeType
- kSymbol_TargetIsDead

Agent-centric representation
Planning system for F.E.A.R.
Chaining based on symbol representation

Planning

Chaining

GOAL

kSymbol_Waparamed

kSymbol_TargetIsDead

kSymbol_WeaponLoaded

kSymbol_WeaponArmed

NoCondition

Draw weapon

Load weapon

Attack

Planner
Recent flow of game AI technologies

New game AI techniques are developed on the basis of previous works!
Contents

1. **Agent architecture**  C4 Architecture  (MIT Media Lab.)
2. **Knowledge representation**  Killzone  (Guerilla)
3. **Event analysis**  Halo (Bungie Studio)
4. **Heuristic FSM**  Halo2 (Bungie Studio)
5. **Goal-oriented Action Planning**  F.E.A.R.  (Monolith)
6. **Multiagent system**  Chromehounds (FromSoftware)
References for this chapter

(1) AOGC2007 (Asia Online Game Conference) (invited lecture),
    Youichiro Miyake, "The future of online games with game AI technology"
    (Material) http://www.bba.or.jp/AOGC2007/pdf/miyake_PPT.pdf
    (online article) http://www.rbbtoday.com/news/20070226/38939.html

(2) CEDEC 2007 (invited lecture),
    Youichiro Miyake, "How to make game AI on Agent architecture"
    (CEDEC2007) http://cedec.cesa.or.jp/contents/r22.html
    (Material) http://www.geocities.jp/mnagaku2000/igda/cedec07r22t07pre.zip

(3) CEDEC 2006,
    Youichiro Miyake, "Perspective of game AI from the making of ChromeHounds"
    (CEDEC2006) (To R41) http://cedec.cesa.or.jp/2006/rs04.html
    (Material) http://www.geocities.co.jp/mnagaku2000/igda/cedec2006r41.html
    (Online article) http://www.famitsu.com/game/news/2006/09/01/103,1157106762,59438,0,0.htm

(4) IGDA Japan chapter, Game AI seminars No.3,
    Youichiro Miyake, "Multiagent system for game AI"
    (IGDAJ) http://www.igda.jp/modules/eguide/event.php?eid=41
    (online article) http://www.rbbtoday.com/news/20070513/41655.html
Chromehounds

In a large map, two player teams fight each other in 15min. in 3D online world, and player team can also fight with AI team.

Game : robot action online-game
Developer : FromSoftware
Publisher: SEGA
Hardware: Xbox360
Publishing year : 2006
Mission of Chromehounds AI Team

AI Team must fight against player team on the equal condition.
The abilities needed for NPC

- Sense the situation around himself
- Decide his action corresponding his situation
- Move and attack

(1) autonomous agent

Individual

(2) Multiagent system

Team
For building an autonomous agent

- goal-oriented hierarchical planning system in agent architecture
- real-time path planning
  (makes NPC move between any two points on a map)

\[ = \text{autonomous agent} \]
goal-oriented hierarchical planning system in agent architecture
AI architecture of ChromeHounds NPC

NPC’s intelligence

- Selecting an goal
  By Evaluation Function
  + Hierarchical
  Goal-Oriented Planning

Behavior-making Process

Reactive mode

Working memory

Cognition

Sensor

World Representation

Game World

Motion

Body

interaction

Time
Hierarchical goal-oriented planning

A goal is a combination of small goals
Hierarchical goal-oriented planning

A small goal is a combination of smaller goals
**In Chromehounds (Xbox360)**

*real-time* Hierarchical Goal-Oriented Planning

- Go to the tower
- Find the tower
- Move along the path
- Find path to the tower
- Wait 10 seconds

**Strategy**  **Tactics**  **Behavior**  **Command**

**NPC**
Demo movie for real-time Hierarchical Goal-Oriented Planning

Players or AI control this type of machine from this viewpoint.
Demo movie for real-time Hierarchical Goal-Oriented Planning

NPC’s Goal-oriented planning

Strategic goal

Tactics goal

Behavior goal

Demo movie
Hierarchical goal structures for NPC in Chromehounds

**Strategic goals**
- Attack Enemy
- Conquer The tower
- Protect Friend
- Protect Base
- Attack Enemies’ Base
- Rescue Friend
- Patrol
- Scout

**Tactics goals**
- Follow Path
- Close
- Attack
- Go to a point
- gather
- Run about
- Escape

**Behavior goals**
- Move Between two point
- Stop & Go
- Stop
- Go back
- Go forward
- Go to the side of enemy

**Command goals**
- Walk
- Shot
- Stop
Programming for goal-oriented

*Design Pattern: Composite structure*

Class Goal

Condition for clear

Activate()

Process()

Terminate()

Activate … initial setting

Process … behavior in active

Terminate … process for finish this goal

*Function are written by script language*
Reference for programming of goal-oriented programming

Mat Buckland, “Programming Game AI by Example”, WORDWARE publishing

http://www.ai-junkie.com/ai-junkie.html
Decision making for Hierarchical goal structures

Agent must select one adequate goal to present situation.

**Strategic goals**
- Attack Enemy
- Conquer The tower
- Protect Friend
- Protect Base
- Attack Enemies’ Base
- Rescue Friend
- Patrol
- Scout

**Tactics goals**
- Follow Path
- Close
- Attack
- Go to a point
- gather
- Run about
- Escape

**Behavior goals**
- Move Between two point
- Stop & Go
- Stop
- Go back
- Go forward
- Go to the side of enemy

**Command goals**
- Walk
- Shot
- Stop
Evaluation value for strategic goals that changes to a situation in game

Decision making = select one goal

Select best high-score goal

Strategic goals
AI architecture of Chromehounds NPC

- **NPC’s intelligence**
  - Hierarchical Goal-oriented planning

NPC decides one goal by himself

NPC makes plans to accomplish the goal NPC he selected.
Real-time path planning system
(makes NPC move between any two points on a map)
Data structure for path planning

(1) Waypoint
   many points should be put on a map

(2) Navigation mesh
   Mesh data can include information of surface of terrain
What is Navigation Mesh?

NPC can move over □-polygons which a whole 2D map is covered with.

Steps
(1) Divide a map into triangles which do not include obstructions (in developing a game).
(2) Find a path to a goal point from a start point (in game).
(3) Move along a path.
Navigation mesh for Chromehounds

(1) 30000 – 80000 meshes
(2) 80 maps
(3) All meshes are **auto-generated from collision models of stages.**
(4) Information of terrain’s surfaces (snow, sand, water, edge, near edge) can be implemented in mesh data.

80 maps of terrains such as mountains, cities, and lakes can be represented as a unique navigation mesh data format.
Navigation mesh information of surfaces are implemented

(1) Sand and water reduces NPC’s speed.
These surface information are implemented in Navigation mesh data.

Finding not a shortest path but a minimum-time path
**NPC can move while considering surface of terrain.**

(2) When any object is destructed, navigation mesh data is updated.

**NPC can move corresponding a change of its environment.**
Demo movie
Contents

1. Game AI overview

2. Game AI technologies in Chrome Hounds
   (1) Autonomous agent
       - goal-oriented hierarchical planning system in agent architecture
       - real-time path planning system
   (2) Multiagent system

3. Developing Character AI based on agent architecture
For building an autonomous agent

- goal-oriented hierarchical planning system in agent architecture

- real-time path planning

autonomous agent
For building a strong NPC team

Cooperation and teamwork are needed!

multiagent system
Multiagent system in Chrome Hounds
Five techniques for cooperation of goal-oriented planning agents

To help each other
- Protect
- Rescue

Cooperation by strategic goal

To judge the situation
- Circumstantial judgement
- Concentrated attack

Cooperation by algorithm

Teamwork to win
- TeamAI

Cooperation by TeamAI

There are 3 categories and 5 techniques.
Multiagent technology  ⌾  Protect

A strong agent follows a weak agent to protect him.

- Strategic goal “Protect friend” is prepared
- A weak agent or one who cannot move fast are protected by a strong agent.
- An agent who protect a friend attack enemies around it.
NPC’s Goal-oriented planning

Protect

Protect Friend

Attack enemies around his friend.

guardian

Agent who can move only slowly...

Demo movie
Protect

Demo movie
Multiagent technology ⊕ Rescue

One agent come to the agent who are attacked and fight with him.

- Strategic goal “Rescue friend” is prepared
- A agent who have little stamina could be rescued
- But because player could attract agents by attacking an agent, an agent who have too little stamina are not rescued.
Rescue

Demo movie
Multiagent technology □ circumstantial judgment

A group of agents judge whether their war potential is higher or lower than enemies. If it is lower, they escape, and if not, they fight.

All war potential > 1.4 x all war potential

Escape!

All war potential < 1.4 x all war potential

Attack!

Don’t fight with much stronger enemies, Fight with equal or weaker enemies!
By circumstantial judgment, they come back to the base, and join their friend.
Multiagent technology □ concentrated attack

Concentrate fire to one of enemies

Concentrate fires to one enemies that has lowest war potential.

Concentrate fires to one enemies that has lowest war potential, And destroy enemies step by step.
concentrated attack

Demo movie
Multiagent technology □ TeamAI
TeamAI control agents by goal
Agents share a common goal

TeamAI cast goals to agents
What is TeamAI?

TeamAI has team strategies, and decide one of them corresponding to the situation in game by evaluating them.
What is TeamAI?

TeamAI has team strategies, and decide one of them corresponding to the situation in game by evaluating them.

One team strategy is consist from some strategic goals to agents.
What is TeamAI?

TeamAI has team strategies, and decide one of them corresponding to the situation in game by evaluating them.

One team strategy is consist from some strategic goals to agents.
What is TeamAI?

TeamAI has team strategies, and decide one of them corresponding to the situation in game by evaluating them.

Team strategy is a combination of goals.

One team strategy is consist from some strategic goals to agents.
What is TeamAI?

TeamAI has team strategies, and decide one of them corresponding to the situation in game by evaluating them.

Team strategy is a combination of goals.

One team strategy is consist from some strategic goals to agents.

Planning
When a goal is given from team AI, how does NPC decide his goal?

COM 2

Destory All enemies
Protect My Base
Conquer All Towers
Destroy Enemies’ base

Team’s rationality

Attack Enemies’ Base

Conquer Towers

Individual’s rationality

Now executing…
When a goal is given from team AI, how does NPC decide his goal?

NPC select a goal which get higher score between two goals.

- Destroy All enemies
- Protect My Base
- Conquer All Towers
- Destroy Enemies’ base

Evaluation value: 64

Now executing…

Evaluation value: 53
When a goal is given from team AI, how does NPC decide his goal?

NPC select a goal which marks higher score between two goals.

- Destroy All enemies
- Protect My Base
- Conquer All Towers
- Destroy Enemies’ base

Evaluation value: 76

Evaluation value: 88

Now executing…
How team AI interfere individual AI

In the first part, individual NPC’s decision should be dominant.
In the last part, team AI’s decision should be dominant.

Final evaluation value of team AI’s goal = Goal evaluation value × comp_team
Final evaluation value of NPC’S goal = Goal evaluation value × comp_idv

Control game’s flow by team AI!

the first part (0~5 min., or before a robot is destroyed)
the middle part (the first part + 5min.)
the last part (the middle part + 5min.)
This agent is executing the goal “Protect” he has chosen.

Demo movie
A goal from a team strategy has interrupted agent’s goal.
Chapter 3.

Summary
Summary for this lecture

(1) In these time, developer can search many papers on internet.

   *In these 5 years!*

(2) From those documents, developers can study much knowledge and many techniques.

   *But most techniques are explained for the game the author developed!*

(3) To apply game AI techniques to your game, you should and you can arrange and apply them for your game design.
For opening the next generation game...

I hope new Korean games open the future of digital game for the future of digital games in the world!
Thank you very much for your attention!

Any question and idea

Today: In the front of ROOM 304(Speaker’s room) 17:30-17:45 : or

y_miyake@fromsoftware.co.jp

http://www.fromsoftware.jp

(as a member of IGDA Japan:  yoichi-m:pk9.so-net.ne.jp )

http://www.igda.jp