

- 1 Introducción
- 2 Videoxogo baseado en axentes
- 3 Análise de requisitos
- 4 Xestión do proxecto
- 5 Arquitectura
- 6 Validación e probas
- 7 Conclusións

└ Táboa de contidos

Durante esta presentación seguiremos unha estrutura similar á memoria, centrándonos máis en algúns aspectos concretos do proxecto que expliquen en que consistiu o traballo realizado.

- 1 Introducción
- 2 Videoxogo baseado en axentes
- 3 Análise de requisitos
- 4 Xestión do proxecto
- 5 Arquitectura
- 6 Validación e probas
- 7 Conclusións

- Proxecto que aborda a creación dun **videoxogo** con necesidades de **comportamento complexo** por parte do inimigo.

Videoxogo

Loita 1 contra 1, Top-Down en dúas dimensións

Axente

Capaz de percibir e actuar sobre o **entorno competitivo** do
videoxogo mediante **sensores** e **actuadores**

De forma xeral, búscase a creación dun videoxogo que requira un inimigo con comportamento complexo. O axente que representará o inimigo necesita ser un competidor capaz, para o que se realizou unha etapa de entrenamento na que optivo a información que necesitaba.

Loita 1 contra 1 significa que soamente dous personaxes competirán entre eles contando ambos coas mesmas capacidades, accións posibles e atributos. **Top-Down** refírese ó plano picado utilizado para visualizar o combate. Por outra parte que sea en **dúas dimensións** implica que todo o contido do videoxogo son imaxes planas debuxadas unha a unha, sen que existan modelos en tres dimensións.

Un axente é aquilo capaz de percibir o entorno mediante **sensores** e actuar sobre o mesmo en consecuencia mediante **actuadores**, ambos son proporcionados pola súa interface co videoxogo. Ademáis atoparase nun entorno competitivo o que implica que buscará maximizar o seu **rendemento** mentres se minimiza o do contrincante.

- Proyecto que aborda a creación dun **videoxogo** con necesidades de **comportamento complexo** por parte do inimigo.

Videocage

Loita 1 contra 1, Top-Down en dúas dimensións

Appendix

Capaz de perceber e actuar sobre o **entorno competitivo** do videoxogo mediante **sensores e actuadores**

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entrenamento do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entrenamento.
- Obter datos sobre as capacidades do axente: Obteráanse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

- Implementación do videoxogo

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entreno do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entreno.
- Obter datos sobre as capacidades do axente: Obteráanse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

- Implementación do videoxogo
- Implementación do axente

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entrenamento do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entrenamento.
- Obter datos sobre as capacidades do axente: Obteráanse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

- Implementación do videoxogo
- Implementación do axente
- Realizar o adestramento do axente

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entrenamento do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entrenamento.
- Obter datos sobre as capacidades do axente: Obteráanse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

- Implementación do videoxogo
- Implementación do axente
- Realizar o adestramento do axente
- Obter datos sobre as capacidades do axente

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entrenamento do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entrenamento.
- Obter datos sobre as capacidades do axente: Obteráanse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

- Implementación do videoxogo
- Implementación do axente
- Realizar o adestramento do axente
- Obter datos sobre as capacidades do axente
- Analizar os resultados obtidos

- Implementación do videoxogo: Xa que necesitaremos unha plataforma que nos permita que o axente e o xogador interactúen seguindo unha serie de regras comúns para competir entre eles.
- Implementación do axente: Necesitarase implementar o axente capaz de desenvolverse correctamente durante a competición.
- Realizar o entrenamento do axente: O axente necesitará obter a información necesaria para logo comportarse adecuadamente gracias ó aprendido durante a etapa de entrenamento.
- Obter datos sobre as capacidades do axente: Obteránse datos sobre o rendemento do axente contra outras implementacións máis sinxelas.
- Analizar os resultados obtidos: Coa información obtida durante todo o proxecto, e especialmente na etapa anterior, realizarase un análise que describa o que conseguiu o axente.

Movimento

Movimiento libre nunha habitación rectangular.

Ataque

Permítese atacar a zona que se atopa cada onde o personaxe está mirando.

Defensa

Posibilidade de defenderse dun ataque permitindo atacar se a defensa ten éxito

Movemento libre nunha habitación rectangular que suma a complexidade de evitar situacións nas que non se poida escapar do contrincante por estar ó lado dunha parede ou unha esquina. Ademáis a única maneira de mirar cara unha dirección é mirar cara ela.

Como solo se permite atacar a zona directamente enfrente do personaxe é importante ter en conta cada donde se está mirando. Isto favorece unha actitude agresiva pois hai que moverse na dirección do enemigo antes de atacalo.

Pódese realizar unha manobra defensiva de alto risco e alta recompensa que permite evitar un ataque. Se se evita con éxito poderase realizar un ataque propio pero se non serase vulnerable durante uns instantes.

Esto fai que non exista unha estratexia idónea pois un estilo agresivo perde contra un defensivo que á súa vez perde contra xogadores que busquen a contra do movemento defensivo, este último ademáis perde contra o xogador agresivo. Esta fórmula de **pedra, papel, tesoiras** demostrou ser amplamente utilizada en diseño de videoxogos.

Primeira implementación realizada con **Unity3D**, estándar de facto para videoxogos de este tamaño.

Problemas de simulación

Imposibilidade de escalar o tempo sen romper o funcionamento do videoxogo.

2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

- └ Videoxogo baseado en axentes
 - └ Prototipo de Unity
 - └ Prototipo de Unity

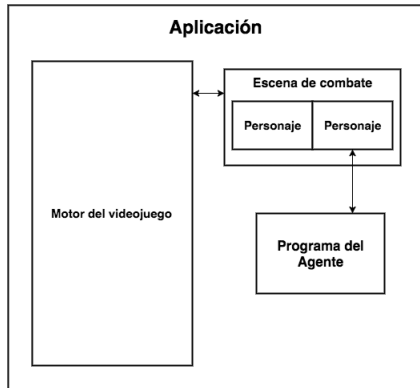
Primeira implementación realizada con **Unity3D**, estándar de facto para videoxogos de este tamaño.

Problemas de simulación
Imposibilidade de escalar o tempo sen romper o funcionamento do videoxogo.

Ensinar vídeo

Segunda aplicación

Implementación de un motor desde cero en C++



2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

- Videoxogo baseado en axentes

Segunda aplicación

Segunda aplicación

Segunda aplicación

Implementación de un motor desde cero en C++



```

1 while agent is running do
2   lastState ← currentState;
3   currentState ← getCurrentState();
4   deltaFitness ←
     calculateFitness(currentState) - calculateFitness(lastState);
5   if lastState ∈ stateActionData then
6     stateActionData.updateWith(lastState,selectedAction,deltaFitness);
7   else
8     stateActionData.insert(lastState,selectedAction,deltaFitness);
9   if currentState ∈ stateActionData then
10     if randomBetween(0,1) <  $\epsilon$  then
11       selectedAction ← randomAction ∈ allPossibleActions;
12     else
13       selectedAction ← action ∈ allPossibleActions |
         bestWeightedAction(stateActionData,currentState) = action;
14   else
15     selectedAction ← randomAction ∈ allPossibleActions;

```

2017-07-19

Inteligencia Artificial aplicada a Videoxogos Top-Down

└ Videoxogo baseado en axentes

└ Algoritmo

└ Algoritmo

```

1 while agent is running do
2   lastState ← currentState;
3   currentState ← getCurrentState();
4   deltaFitness ←
     calculateFitness(currentState) - calculateFitness(lastState);
5   if lastState ∈ stateActionData then
6     stateActionData.updateWith(lastState,selectedAction,deltaFitness);
7   else
8     stateActionData.insert(lastState,selectedAction,deltaFitness);
9   if currentState ∈ stateActionData then
10     if randomBetween(0,1) <  $\epsilon$  then
11       selectedAction ← randomAction ∈ allPossibleActions;
12     else
13       selectedAction ← action ∈ allPossibleActions |
         bestWeightedAction(stateActionData,currentState) = action;
14   else
15     selectedAction ← randomAction ∈ allPossibleActions;

```

2017-07-19

Inteligencia Artificial aplicada a Videoxogos Top-Down

└ Videoxogo baseado en axentes

└ Algoritmo

└ *Fitness*

Parámetro	Valor
INITIAL_FITNESS_VALUE	1000
MY_HEALTH_MULTIPLIER	100
ENEMY_HEALTH_MULTIPLIER	100
DISTANCE_MULTIPLIER	3
LOOKING_BONUS	200
WALL_BONUS	50

```

input : playerHealth, enemyHealth, distance,
        lookingAtEnemy, noWallsNear
output: fitness
1 fitness ← INITIAL_FITNESS_VALUE;
2 fitness ← fitness +(playerHealth *
  MY_HEALTH_MULTIPLIER);
3 fitness ← fitness -(enemyHealth *
  ENEMY_HEALTH_MULTIPLIER);
4 fitness ← fitness -(distance *
  DISTANCE_MULTIPLIER);
5 if lookingAtEnemy then
6   | fitness ← fitness + LOOKING_BONUS;
7 if noWallsNear then
8   | fitness ← fitness + WALL_BONUS;

```

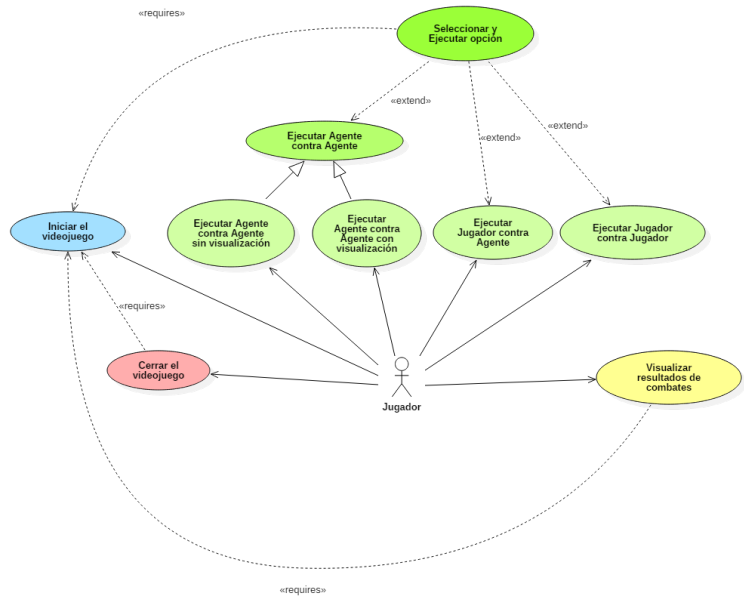
Parámetro	Valor
INITIAL_FITNESS_VALUE	1000
MY_HEALTH_MULTIPLIER	100
ENEMY_HEALTH_MULTIPLIER	100
DISTANCE_MULTIPLIER	3
LOOKING_BONUS	200
WALL_BONUS	50

Contra axente baseado en regras

Pretende simular un aprendizaxe contra xogadores reais.

Contra él mesmo

Buscando unha exploración mais extensa de estados que o axente baseado en regras non pode aportar.



2017-07-19

Inteligencia Artificial aplicada a Videoxogos Top-Down

- └ Análise de requisitos
 - └ Casos de uso
 - └ Casos de uso



- **RF-1/2/3:** Funcionalidades do menú
- **RF-4/15:** Consola con comandos/resultados
- **RF-5:** Saír da aplicación
- **RF-6:** Entrar na escena de combate
- **RF-7/8/9:** Moverse/Atacar/Defender
- **RF-10:** Gañar/Perder partida
- **RF-11:** Esgotar o tempo de combate
- **RF-12:** Volver ó menú
- **RF-13:** Visualizar combate entre axentes
- **RF-14:** Simular múltiples combates
- **RNF-1:** Rendemento da aplicación
- **RNF-2: Velocidade das simulacións**
- **RNF-3:** Extensibilidade do motor
- **RNF-4:** Facilitade para depurar
- **RNF-5:** Aplicación autocontida
- **RNF-6:** Extensibilidade de escenas
- **RNF-7:** Documentación
- **RNF-8:** Usabilidade da interface

2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

- └─ Análise de requisitos
 - └─ Requisitos
 - └─ Requisitos

Requisitos	
<ul style="list-style-type: none">● RF-1/2/3: Funcionalidades do menú● RF-4/15: Consola con comandos/resultados● RF-5: Saír da aplicación● RF-6: Entrar na escena de combate● RF-7/8/9: Moverse/Atacar/Defender● RF-10: Gañar/Perder partida● RF-11: Esgotar o tempo de combate● RF-12: Volver ó menú● RF-13: Visualizar combate entre axentes● RF-14: Simular múltiples combates	<ul style="list-style-type: none">● RNF-1: Rendemento da aplicación● RNF-2: Velocidade das simulacións● RNF-3: Extensibilidade do motor● RNF-4: Facilitade para depurar● RNF-5: Aplicación autocontida● RNF-6: Extensibilidade de escenas● RNF-7: Documentación● RNF-8: Usabilidade da interface

Contexto do proxecto

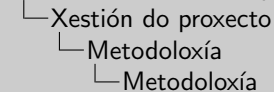
- Traballador único
- Duración relativamente corta
- Necesidade de avanzar rapidamente nas etapas iniciais

Programación Extrema

- Flexibilidade ante cambios
- Evitase utilizar demasiado tempo en tarefas de xestión
- Rápida iteración
- Reunións entre *sprints*

2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

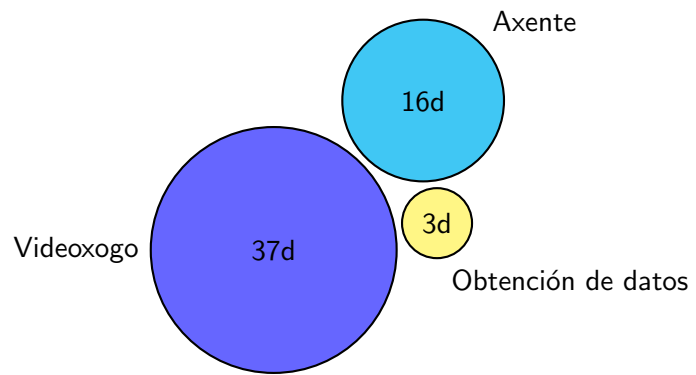


Contexto do proxecto

- ▼ Traballador único
- ▼ Duración relativamente curta
- ▼ Necesidade de avanzar rapidamente nas etapas iniciais

Programación Extrema

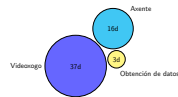
- Flexibilidad ante cambios
- Evitase utilizar demasiado tiempo en tareas de gestión
- Rápida iteración
- Reuniones entre sprints

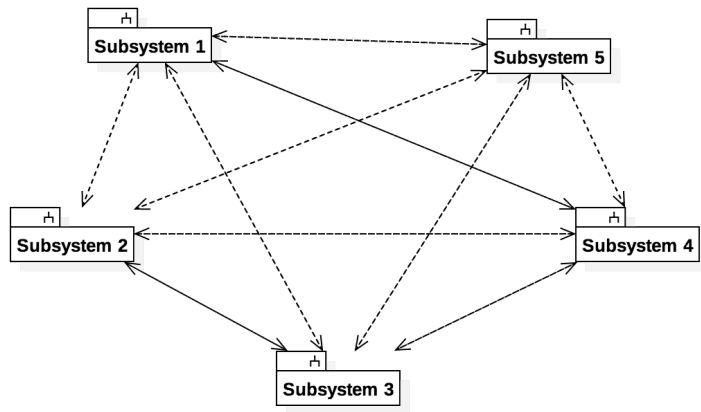


2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

- └ Xestión do proxecto
 - └ Planificación temporal
 - └ Planificación temporal

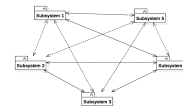


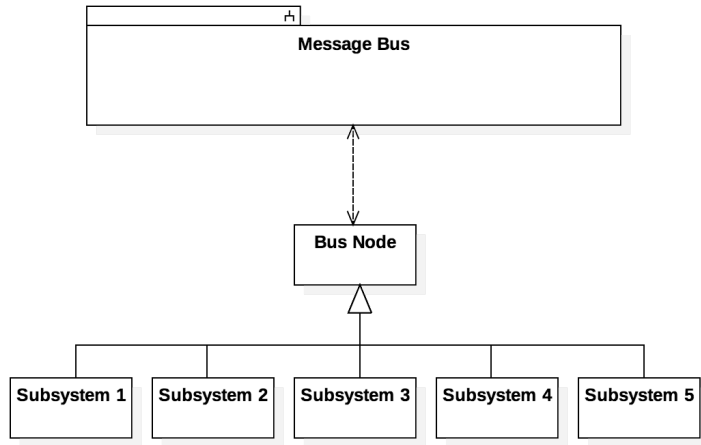


2017-07-19

Inteligencia Artificial aplicada a Videoxogos Top-Down

- Arquitectura
 - Arquitectura do sistema
 - Subsistemas conectados





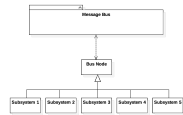
2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

└─Arquitectura

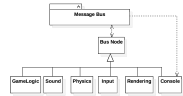
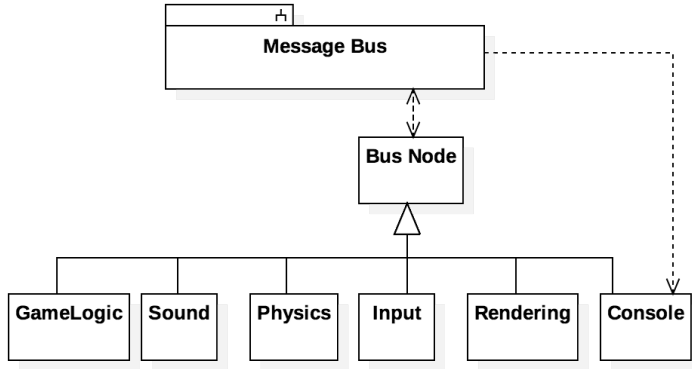
└─Arquitectura do sistema

└─Bus de mensaxes



2017-07-19

- Arquitectura
 - Arquitectura do sistema
 - Arquitectura final



Probas unitarias

Unha ou mais probas por cada requisito tanto funcional como non funcional superadas na súa totalidade.

Probas de integración

Comproban a integración entre subsistemas e do o axente ca aplicación.

2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

└─ Validación e probas

└─ Aplicación

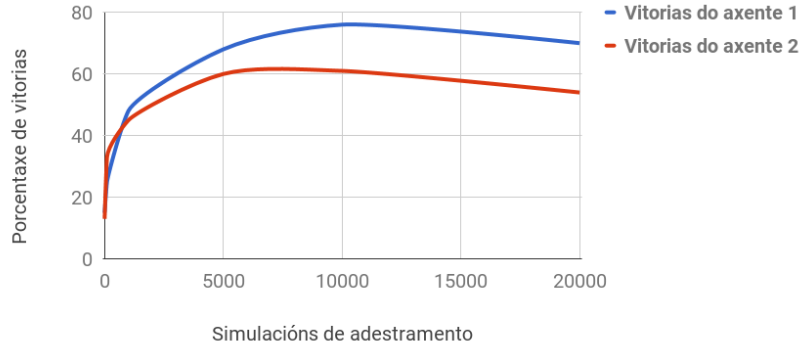
└─ Validación e probas da aplicación

Probas unitarias

Unha ou mais probas por cada requisito tanto funcional como non funcional superadas na súa totalidade.

Probas de integración

Comproban a integración entre subsistemas e do o axente ca aplicación.



2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

Validación e probas

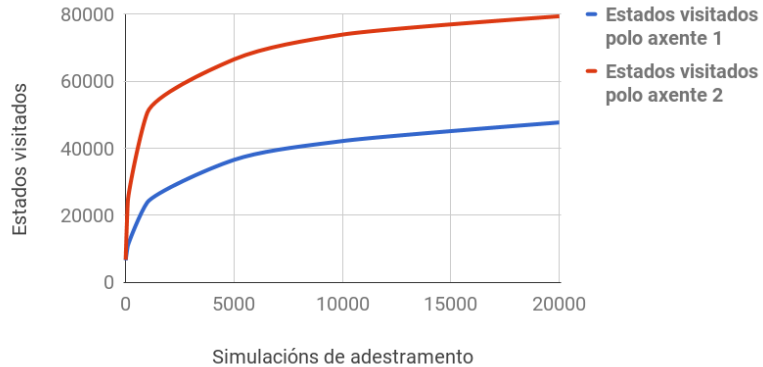
Validación do Axente

Comparativa de vitorias

Comparativa de vitorias



Comparativa de estados visitados



2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

Validación e probas

Validación do Axente

Comparativa de estados visitados



Combinación de ámbolos dous métodos de adestramento.

Contra o axente baseado en regras

Favorece un aprendizaxe moi rápido nas primeiras simulacións.

Contra él mesmo

Aporta unha exploración de estados superior.

Logros do proxecto

- O comportamento, aspecto e rendemento da aplicación cumpriu as expectativas.
- O axente é capaz de competir contra outras implementacións e contra xogadores humanos.

Leccións aprendidas

- Importancia de ter en conta os posibles riscos do proxecto o antes posible.
- Utilidade de un deseño flexible previo á implementación.
- Calidade dos resultados de implementacións sinxelas de Intelixencia Artificial.

2017-07-19

Intelixencia Artificial aplicada a Videoxogos Top-Down

Conclusións

Conclusións e leccións aprendidas

Conclusións e leccións aprendidas

Logros do proxecto

- O comportamento, aspecto e rendemento da aplicación cumpriu as expectativas.
- O axente é capaz de competir contra outras implementacións e contra xogadores humanos.

Leccións aprendidas

- Importancia de ter en conta os posibles riscos do proxecto o antes posible.
- Utilidade de un deseño flexible previo á implementación.
- Calidade dos resultados de implementacións sinxelas de Intelixencia Artificial.

- Melloras de compatibilidade.
- Ampliación do proceso de probas con xogadores humanos.
- Implementación de máis técnicas para o axente.
- Novas mecánicas para o videoxogo.