Geometry Friends: um jogo de cooperação

Relatório Intercalar

Agentes e Inteligência Artificial Distribuída

4º ano do Mestrado Integrado de Engenharia Informática e Computação

Grupo T12_1
Catarina Correia - up201405765 - up201405765@fe.up.pt
José Aleixo Cruz - up201403526 - up201403526@fe.up.pt
José Carlos Coutinho - up201404293 - up201404293@fe.up.pt

Novembro 2017

Enunciado

Descrição do cenário

Geometry Friends é um jogo de plataformas num ambiente 2D com um motor de simulação física (com gravidade e fricção). O objetivo dos jogadores é colecionar um conjunto de diamantes no menor tempo possível. Cada jogador (agente) controla uma das duas personagens (formas geométricas): um círculo amarelo e um retângulo verde.

Os agentes controlam a sua personagem, que dispõe de um número restrito de movimentos. O retângulo é capaz de mudar a sua forma para horizontal ou vertical (com a mesma área), andar para o lado direito ou esquerdo. O círculo pode andar para os lados, saltar ou crescer.

Num nível podem também existir obstáculos. Os obstáculos podem obstruir o caminho de apenas um dos agentes ou de ambos. Os obstáculos com a cor preta bloqueiam tanto o círculo como o retângulo. Os obstáculos com a cor verde apenas permitem que o retângulo os atravesse. Os obstáculos com a cor amarela apenas permitem que o círculo os atravesse.

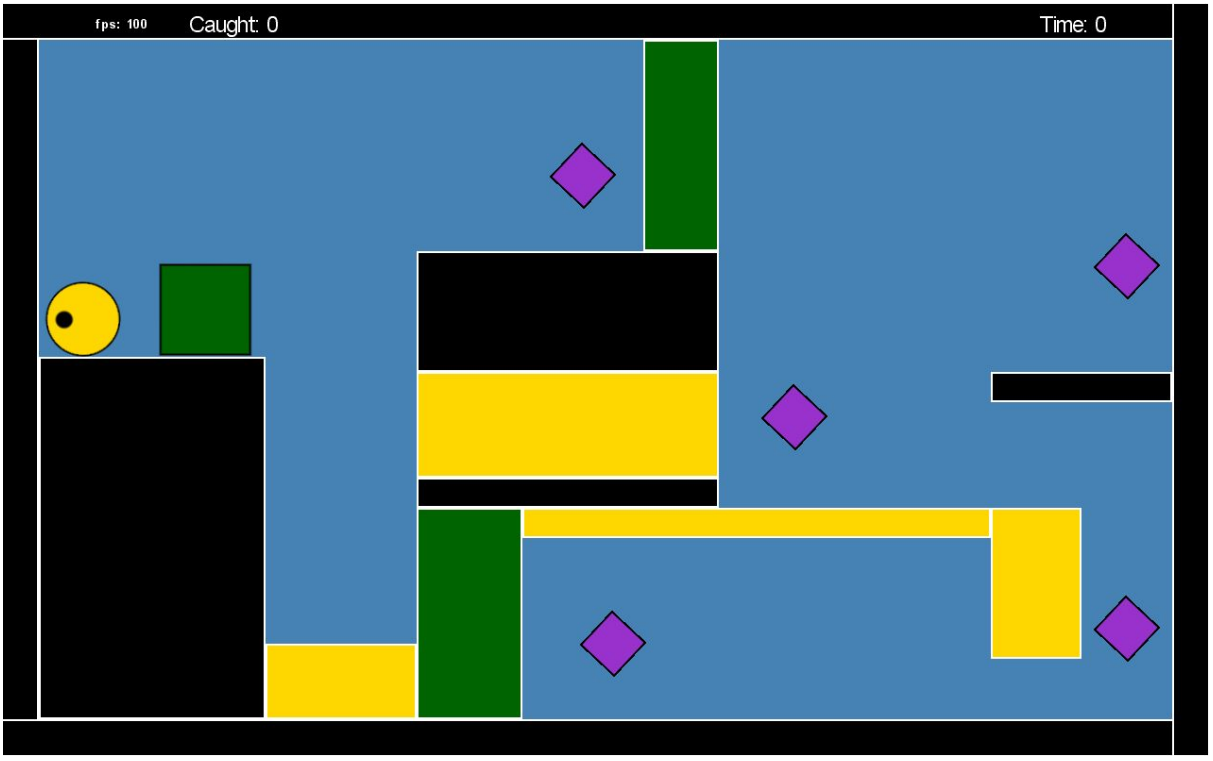

Figura 1 - Exemplo de um nível e dos vários tipos de obstáculos

No jogo existem dois tipos de níveis: cooperativos e individuais. Nos níveis cooperativos, os agentes devem utilizar os seus movimentos característicos e interagirem de forma a apanhar os diamantes. Nos níveis individuais apenas uma das personagens é posta à prova. Para este projeto serão focados apenas os níveis cooperativos.

Objetivos do trabalho

O objetivo deste trabalho consiste na implementação de um algoritmo e de um protocolo de comunicação que permita aos dois agentes cooperar entre si. Esta cooperação é fundamental para que os agentes consigam apanhar todos os diamantes dos diferentes níveis cooperativos.

Visto que uma das principais metas da competição é conseguir passar um nível no menor tempo possível, um dos propósitos do grupo, e o maior desafio do projeto, é conseguir implementar um algoritmo eficiente que tenha em conta a posição dos agentes, dos obstáculos e dos vários diamantes do nível e determine qual a ordem pela qual estes devem ser apanhados, bem como, para cada um dos diamantes, qual o agente que mais facilmente o consegue apanhar e se é necessário que ambos utilizem movimentos em conjunto para apanhar um diamante.

Resultados esperados e formas de avaliação

A análise de resultados será uma componente bastante importante neste projeto, pois permitirá perceber se, de facto, os agentes estão a comunicar entre si e o quão eficiente é a estratégia (algoritmo) desenvolvida pelo grupo.

Como já foi referido, um dos intuitos do jogo é conseguir apanhar todos os diamantes de um determinado nível. Os dois agentes devem conseguir comunicar entre si para uma resolução eficiente do problema.

É esperado que os agentes consigam perceber se são ou não capazes de resolver o problema de forma independente. Caso tal não seja possível, o(s) agente(s) deve(m) poder mandar uma mensagem com a identificação do problema que os impossibilita de alcançar o objetivo. Por exemplo, se um diamante se encontrar numa posição mais elevada relativamente à distância que o círculo consegue saltar, este deve mandar uma mensagem ao retângulo a dizer que a distância é demasiado elevada, de forma a que o retângulo possa servir de plataforma para o seu salto.

No caso de ambos os agentes conseguirem alcançar um determinado diamante individualmente e a escolha tomada não alterar a possibilidade de alcançar outros diamantes,

o agente que estiver mais perto (distância calculada pelo algoritmo a implementar) é o que toma a iniciativa de apanhar o diamante.

Em termos de jogo, a pontuação é calculada da seguinte forma:

$$Pontuação = Nível\ Completo * \frac{(Tempo\ Limite - Tempo)}{Tempo\ Limite} + (Diamantes\ Apanhados * Pontuação\ do\ Diamante)$$

Desta forma, o maior objetivo é apanhar todos os diamantes dentro do tempo limite da forma mais rápida possível. Caso isso não seja possível, o objetivo passa a ser apanhar o maior número de diamantes.

Plataforma/Ferramenta

O Geometry Friends é desenvolvido pelo laboratório GAIPS INESC-ID. A comunicação entre agentes já está estabelecida através de uma interface de mensagens. Como tal, este trabalho não requer a utilização de umas das plataformas propostas, pois é desenvolvido utilizando o software específico da competição, que requer programação em C#. O kit da competição inclui um projeto em Visual Studio, que permite adicionar a implementação de agentes

Cada agente possui uma lista de mensagens que são objetos da classe AgentMessage, que consiste numa string e num anexo opcional que pode ser qualquer objeto. Esta lista é verificada sempre que é feita uma atualização ao estado de jogo. Se um agente tiver adicionado uma nova mensagem à lista, a mensagem é transmitida ao outro agente. A cada atualização, um agente verifica se recebeu novas mensagens e invoca o método para lidar com elas.

Especificação

Identificação e caracterização dos agentes

Na competição Geometry Friends, existem dois agentes: o círculo e o retângulo. O círculo é um objeto da classe CircleAgent e o retângulo é um objeto da classe RectangleAgent. Ambos derivam de AbstractAgent e implementam os seus métodos.

As únicas ações que os agentes podem tomar reduzem-se ao seu movimento e estão descritas na classe Moves.

Movimentos do retângulo	**Movimentos do círculo**
Moves.MOVE_LEFT	Moves.ROLL_LEFT
Moves.MOVE_RIGHT	Moves.ROLL_RIGHT
Moves.MORPH_UP	Moves.JUMP
Moves.MORPH_DOWN	Moves.GROW

A solução para este projeto passa por executar em cadeia uma série de movimentos que leve à maior pontuação possível no nível.

Propriedades dos Agentes

Tanto o CircleAgent como o RectangleAgent possuem várias características em comum . Tal pode ser observado na tabela que se segue, que descreve as propriedades destes dois agentes.

Tabela 1 - Propriedades dos agentes CircleAgent e RectangleAgent

Propriedades	**Domínio de Valores**	**Justificação**
Tempo de Vida	Permanente	O tempo de vida dos agentes corresponde à totalidade do tempo de jogo.
Nível de Cognição	Deliberativo	Os agentes mantêm uma representação do ambiente, tomam decisões com base nessa representação e em informações transmitidas pelo outro agente.

Implementação	Procedimental	As informações de controlo necessárias ao uso do conhecimento estão embutidas no próprio conhecimento.
Mobilidade	Estacionário	Os agentes trocam informação entre eles através de mensagens. Não existe um agente que faça a recolha de informação.
Adaptabilidade	Fixo	As capacidades dos agentes são independentes do ambiente e respetivas interações.
Localização	Local	Os agentes não conseguem correr em máquinas diferentes.
Autonomia Social	CircleAgent: Independente RectangleAgent: Controlado	O círculo toma decisões por ele e pelo retângulo.
Sociabilidade	CircleAgent: Responsável RectangleAgent: Atento	O retângulo aguarda mensagens enviadas pelo círculo.
Colaboração	Cooperativos	Os agentes trocam entre si mensagens para atingir um objetivo comum.

Entre os dois agentes, o CircleAgent será o líder. Ele interpretará as suas informações e as informações do RectangleAgent para calcular a melhor estratégia para completar o nível e depois transmiti-la e executá-la.

Protocolos de interação

Como foi referido, a plataforma de comunicação já implementada faz uso de uma classe AgentMessage que permite que um objeto seja anexado aos seus objetos.

Para implementar o protocolo de interação foi criada uma classe Message, que possui um identificador próprio e que será transmitida como anexo de uma AgentMessage.

Da classe Message derivam as classes Request e Answer. Um objeto Request representa um pedido feito por um agente. Cada Request incluirá um objeto Command, que indica a ação que o agente invocador pede que o outro agente execute. Um objeto Answer representa a resposta de um agente a um determinado Request.

Para cada Request enviado por um agente, espera-se uma Answer enviada pelo outro, confirmando ou não a execução do Request.

Por exemplo, se o CircleAgent precisar que o RectangleAgent se mova para a esquerda, gerará um Request com um Command que invoque a função MoveLeft() do retângulo. De seguida, anexará o Request a uma AgentMessage e adiciona-a à sua lista de mensagens. Do outro lado o RectangleAgent receberá a mensagem e ao interpretá-la corre a

função MoveLeft(), que retornará se foi bem sucedida ou não. Caso tenha sido, envia uma Answer com o tipo “YES”, caso contrário, envia “NO”. O CircleAgent recebe a mensagem e decide como prosseguir.

Faseamento do projeto

Relativamente ao faseamento do projeto, o grupo pretende dividir o desenvolvimento em 4 fases.

Fase 1 - Criação do protocolo de comunicação

Numa primeira fase, pretende estabelecer-se um protocolo de comunicação entre os dois agentes, delineando quais os comandos que irão ser necessários para que os agentes possam efetivamente comunicar um com o outro.

Fase 2 - Implementação do algoritmo A* adaptado a cada agente

Para que os agentes apanhem um diamante é necessário que calculem qual o caminho mais curto para chegar a esse diamante. No entanto, consoante a disposição de obstáculos e de diamantes no nível, haverá diamantes que não conseguem ser apanhados pelo CircleAgent, pelo RectangleAgent ou por ambos.

Desta forma, pondera-se implementar o algoritmo A* para descobrir o caminho mais curto para cada diamante, tendo em conta as restrições de movimento de cada agente e restringindo o algoritmo a estas limitações. Assim, o algoritmo A* do CircleAgent terá restrições diferentes do algoritmo A* do RectangleAgent.

Fase 3 - Implementação de um algoritmo de negociação

Depois de estar determinado o caminho mais curto de cada agente a cada diamante, será útil que os agentes comuniquem entre si de forma a delinear a estratégia que lhes permitirá concluir o nível no menor tempo possível. Por exemplo, se o círculo estiver mais próximo de um determinado diamante do que o retângulo, deverá ser o círculo a apanhá-lo.

Fase 4 - Implementação dos movimentos físicos

Ao estar completamente definida a estratégia para apanhar todos os diamantes, resta apenas executá-la. Como o cenário de jogo é calculado através de um motor de física, será necessário implementar os movimentos de cada um dos agentes de forma a que consigam seguir o caminho definido nos passos anteriores.

Para além dos movimentos individuais já definidos, serão também implementados movimentos conjuntos, para serem usados em cenário de cooperação.

Um dos exemplos mais simples de movimentos cooperativos no Geometry Friends é a possibilidade de o círculo saltar para cima do retângulo, de forma a poder saltar para plataformas com maior altura.

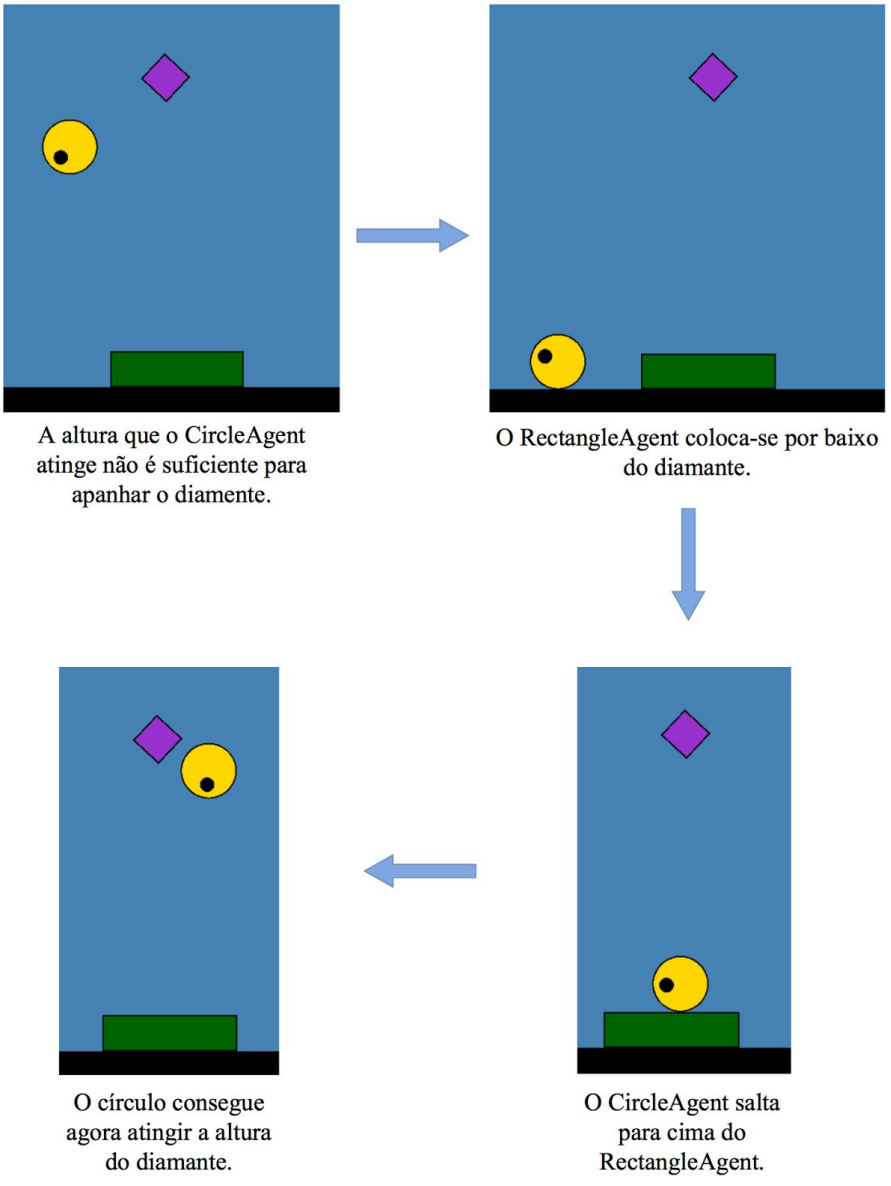

Figura 2 - Movimento ilustrativo da possível cooperação entre agentes

Recursos

Bibliografia

INESC. “Geometry Friends | Cooperation Puzzle Game.” Geometry Friends Cooperation Puzzle Game, INESC, http://gaips.inesc-id.pt/geometryfriends/.

D'Inverno, Mark, et al. Foundations and Applications of Multi-Agent Systems. Springer, 2002.

Vlassis, Nikos. A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence. Morgan & Claypool, 2007.

Oliveira, Eugénio. Agentes e Inteligência Artificial Distribuída, http://paginas.fe.up.pt/~eol/AIAD/aiad1718.html.

Software

Sistema operativo: Windows
IDE: Visual Studio
Linguagem: C#