

Ensaio de Reescrita de Comportamentos em Videojogos com Base no Ajuste e Computação de Modelos de Petri Net

Sérgio D. do Santos Licínio G. Roque

Universidade de Coimbra, Dep. Eng. Informática, Portugal

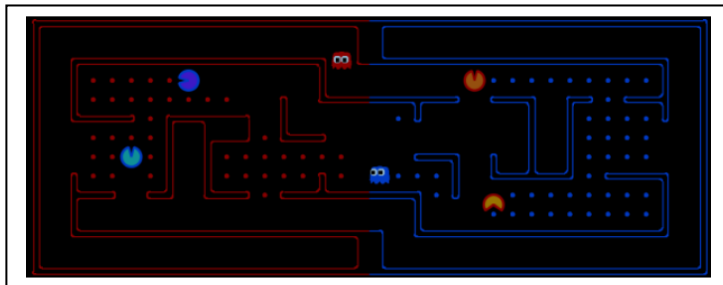


Figura 1 - Pacman: Capture the Flag – protótipo com agentes baseados em modelo Petri Net adaptavel

Abstract

Os autores apresentam um estudo de ensaio de modelação e reescrita de comportamentos de agentes em videojogos com base em modelos de Petri Nets. Conclui-se pela utilidade do método experimentado como forma de introduzir variedade na experiência, controlando dinamicamente o nível de dificuldade de jogo e obtendo a satisfação por parte dos jogadores.

Keywords: Geração Automática de Conteúdo, Redefinição Automática de Comportamentos, Ajuste Dinâmico de Dificuldade

Authors' contact:

sdsantos@student.dei.uc.pt
lir@dei.uc.pt

1. Introdução

Um dos aspectos com que os game designers mais se preocupam é a longevidade do seu jogo. No entanto, é relativamente difícil motivar jogadores a jogar repetidamente, assim como exige um custo muito elevado a produção massiva de conteúdos distintos. Um mecanismo que permita a renovação automática do que realmente torna um jogo atractivo é uma procura recorrente na indústria. Neste trabalho foi estudado, implementado e avaliado um método para a auto-reescrita de comportamentos em jogos, baseado na forma como são jogados. Apresenta-se a seguir o desenvolvimento de um protótipo para teste de conceito onde se descreveram os comportamentos dos agentes com base em modelos numa linguagem formal utilizando Petri Nets. Estes modelos foram automaticamente adaptados entre sessões de jogo, isto é, foram reescritos, com base em critérios e métodos definidos previamente. Os resultados obtidos são satisfatórios e encorajadores já que demonstram

potencial de introdução de variedade e, portanto, longevidade num jogo.

2. Trabalho Relacionado

Existem diversos jogos com elementos que se modificam a si próprios. Já o popular jogo *Diablo* da Blizzard Entertainment rearranjava aleatoriamente os níveis, adversários e itens de cada vez que o jogador o terminava, destacado-se por motivar os jogadores a voltar a jogá-lo frequentemente. Contudo, este tipo de aplicações perdem por excluir o jogador humano do processo de recriação do jogo. No outro extremo encontram-se os jogos baseados em *crowdsourcing*, onde o processo de recriação é deixado a cargo dos jogadores, que podem criar novos níveis, personagens, etc.. Um exemplo recente é o jogo *LittleBigPlanet*, da Media Molecule. No entanto, este tipo de abordagens limita em muito o controlo dos resultados pelo *game designer*. Pelo meio, e perto do objectivo deste trabalho, surgem experiências como o jogo *Façade* [Mateas & Stern 2003]. Neste caso os autores procuraram aumentar a imersão do jogador através de uma forma de interacção natural, que condiciona o jogo e especialmente a história que irão experienciar.

Um conceito de jogo é composto por muitos elementos diferentes organizados numa estrutura muito específica. Um sistema tão complexo terá que ser abordados de diferentes perspectivas e a diferentes níveis de abstracção. Myers [2009] procurou identificar os componentes mínimos de um jogo, ficando com quatro elementos: regras, objectivos, oposição e representação. Alertou ainda que se o conceito for demasiado objectivo, uma especificação completa, o jogo transforma-se numa simulação, e se pelo contrário, se deixar demasiado em aberto, ficamos com uma simples brincadeira. Hunicke et al. [2004] apresentaram na Game Developers Conference a *framework* MDA (Mechanics, Dynamics e Aesthetics).

A mecânica compreende as regras e conceitos que especificam formalmente o jogo/sistema. A dinâmica especifica o comportamento durante o jogo da interação sistema-jogadores. Finalmente a estética enuncia as respostas emocionais evocadas pela dinâmica do jogo. Estes três conceitos encontram-se interligados e ordenados desde o domínio do *game designer* até ao domínio do jogador. Pretende-se que com esta decomposição formal seja possível analisar e compreender qualquer jogo. Mateas e Wardrip-Fruin [2009] registaram o termo *operational logic* que define “an authoring (representational) strategy for specifying the behaviors a system must exhibit”. Esses comportamentos são decompostos no seu processo abstracto, domínio, estratégia de representação e interpretação da audiência. De notar que estas *operational logics* trabalham a um nível mais baixo que a *framework* MDA. Referem-se tanto ao significado como à concreta implementação de um elemento de jogo reconhecível. São dados como exemplo o suporte de navegação no espaço de jogo ou alocação de recursos.

Várias abordagens já foram analisadas na literatura, para ajudar a tarefa de formalizar o conceito de um jogo. Partindo de uma notação semelhante às máquinas de estado, as **Petri Nets** são um formalismo alternativo e mais económico que recorre frequentemente a uma notação gráfica para permitir representar fenómenos mais complexos com menos elementos. As Petri Nets são definidas graficamente por locais (círculos), *tokens* (marcadores simples ou diferenciados), transições (quadrados ou rectângulos) e arcos (setas) conectando os locais origem e destino das transições. Os locais representam tipicamente condições ou recursos, enquanto as transições representam as acções que dependem e/ou produzem os *tokens*, usados para representar a sinalização ou recursos necessários [Murata 1989]. Os sistemas cibernéticos auto-regulados são normalmente baseados na interacção, considerando tanto as entradas e saídas como os mecanismos internos, logo próximos do conceito de jogo adaptativo que procuramos, e são particularmente úteis para a descrição de ocorrências de feedback positivo ou negativo, comuns em jogos [Salem & Zimmerman 2004]. No nosso estudo optámos pela representação do modelo de regulação com base em Petri Nets pela sua expressividade e extensibilidade. Um estudo prévio da sua aplicação em *game design* já tinha sido apresentado por Araújo e Roque [2009], para diferentes elementos de jogo. Serão de seguida descritas quatro aplicações possíveis por nós estudadas.

3. Âmbito da Modelação

As Petri Nets poderão ser usadas para especificar os comportamentos de agentes autónomos, considerando o estado do jogo e as acções do jogador, e podem ser redefinidas e ordenadas facilmente alterando os arcos que definem os locais origem e os locais de produção de tokens para cada transição modelada. A reescrita

dos comportamentos pode ser feita, em resposta ou com base na análise nos comportamentos do jogador. Decidiu-se começar por ensaiar uma aplicação à volta da reescrita da inteligência artificial de um jogo, por ser aparentemente mais simples de testar e por prometer um impacto significativo na percepção que o jogador teria da variedade de comportamento do jogador. Adicionalmente, o conceito de jogo a implementar também deveria ser simples, para que o desenvolvimento fosse focado nos factores que queremos estudar - a reescrita dinâmica de comportamentos - de uma forma que fosse perceptível ao nível da experiência de jogo.

O construção de um jogo completo e complexo foge ao âmbito deste trabalho. Pretende-se antes desenvolver um caso de estudo claro pela aplicação do mecanismo de actualização de comportamentos num aspecto específico de um jogo conhecido, para assim se demonstrar a potencialidade do conceito e método de modelação apresentados. Optou-se por usar um jogo facilmente reconhecível, o Pacman, e adicionar-lhe alguns elementos de competição para permitir uma jogabilidade diferente e mais variada. O conceito original do jogo foi apresentado num torneio universitário de inteligência artificial, sob o nome de “Pacman: Capture the flag”, onde se pretendia que estudantes universitários programassem agentes para competir entre si. Existem duas equipas de pacmans no jogo, cada uma com metade do mapa como zona defensiva e a restante como zona atacante. Cada equipa é composta por múltiplos pacmans. Na zona defensiva, um pacman age como um fantasma, podendo capturar pacmans adversários, enviando-os de volta para a sua zona defensiva. Na zona atacante, um pacman pode recolher itens do terreno adversário, e assim ganhar pontos. O jogo termina após um limite de tempo ou quando uma equipa capturar toda a comida do adversário. Na Figura 1 é possível observar um jogo a decorrer, entre as equipas azul e vermelha.

O código-fonte da aplicação original era *open source* pelo que o foco do trabalho foi o desenvolvimento de um novo agente, controlado por uma inteligência artificial definida formalmente pela Petri Net reescrita. Um dos agentes seria controlado por um jogador. O comportamento dos restantes agentes, da mesma equipa e adversários, seria controlado pela Petri Net. Cada agente possui os seguintes elementos: sensores, acções, memória e regras, que serão mapeados na Petri Net. Os sensores serão representados por locais com geradores associados. A cada iteração, quando um sensor é activado, introduz um token no local respectivo da rede. As acções são representadas por transições. A memória é representada através de locais, um para cada transição, porque irão memorizar os tokens transmitidos pelas transições que disparam. As regras são os arcos que ligam os locais dos sensores às transições das acções, e que ligam os locais de memória de volta às transições. Os sensores implementados para cada agente, identificados e

modelados como locais da Petri Net, foram: a) *Na zona de defesa?* b) *Na zona de ataque?* c) *Comida à vista?* d) *Colega à vista?* e) *Inimigo à vista?* f) *A ganhar?* g) *A perder?* As acções implementadas para cada agente, identificadas e modeladas como transições, foram: a) *Perseguir inimigo;* b) *Perseguir comida;* c) *Procurar comida;* d) *Fugir de fantasma;* e) *Proteger fronteira;* f) *Atacar;* g) *Defender.* Na Figura 2 apresenta-se um modelo parcial da Petri Net exemplificativa de um agente. Os locais S1, S2, S3 e S4 são locais de memória das acções do agente.

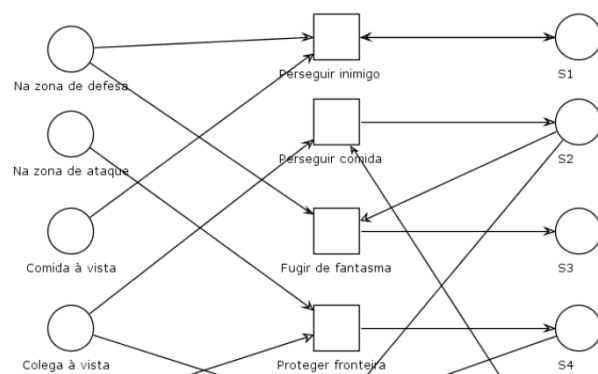


Figura 2 - Aspecto da Petri Net de um Agente

As ligações aos locais à esquerda podem ser interpretados como representando condições para o disparo das acções representadas pelas transições ao centro. Os locais à direita actuam como memória de actuação do agente ajudando a regular a selecção de acções a cada passo com base na história de acções anteriores, de que resultaram *tokens* nessas posições.

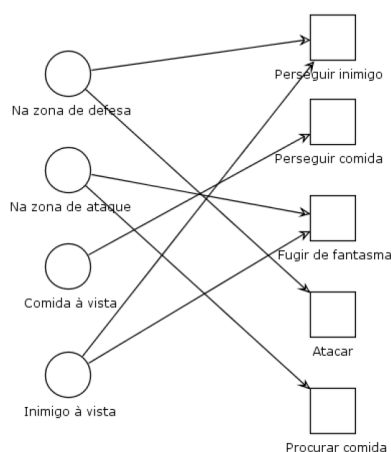


Figura 3 - Regras iniciais na Petri Net que define o comportamento inicial de um agente do jogo (*bot*)

Tendo-se adoptado *Design Research* como metodologia de investigação, após algumas sessões de avaliação, verificou-se ser necessário realizar algumas extensões ao modelo do artefacto inicial. Em cada iteração é necessário escolher uma transição activa a definir a acção a ser desempenhada pelo agente no momento seguinte. Verificou-se que as regras de activação modeladas na versão original eram muito rígidas para este caso, pois exigiam que todos os locais ligados a uma transição tivessem pelo menos um *token*

para esta poder ser activada, situação mais rara do que se supunha. Por essas razões, alterou-se o modelo para considerar que na falta de uma transição activa fosse activada a que possuísse o maior número de locais com pelo menos um token. Em caso de empate é escolhida aleatoriamente uma transição entre as activas. Experimentou-se, na primeira sessão de jogo, começar com os arcos definidos aleatoriamente. Contudo, como o comportamento era demasiado variável, definiu-se um conjunto de arcos base (ver Figura 3) como esqueleto de partida para a definição de regras extra. As regras base não são fixas e poderão ser alteradas em actualizações seguintes da rede.

4. Ajuste Dinâmico

No ajuste dinâmico da Petri Net que define em runtime o comportamento dos agentes não-humanos (*bots*) consideraram-se dois sub-problemas. Primeiro, quando actualizar a rede: a) No final de uma sessão do jogo; b) Em tempo-real, durante a sessão de jogo. A actualização em tempo-real da Petri Net é uma tarefa cuja implementação requer a definição de uma métrica para avaliação contínua de desempenho. Na ausência desta métrica e sendo as sessões do jogo escolhido relativamente curtas, optou-se por começar por estudar a actualização do comportamento dos agentes entre sessões. Segundo, é necessário definir os critérios que irão influenciar a actualização. Algumas possibilidades consideradas: a) Com base nas acções dos jogadores; b) Com base nos estados intermédios que o jogo foi tomando; c) Com base nos pares estado-transição mais usados; d) Com base nos resultados do jogo da equipa; e) Actualização aleatória das transições.

Os critérios do método de actualização testado (designado “inteligente”) combinam os critérios com base em pares estado-transição e o resultado do jogo. São contabilizados quantos tokens passam através de cada arco e os arcos por onde passaram mais tokens presumem-se os mais influentes no comportamento da última sessão. Se o agente foi bem sucedido esses arcos são mantidos e os restantes substituídos por novos arcos sorteados. Se o agente não foi bem sucedido esses serão os arcos substituídos. Desta forma, pretende-se que os agentes evidenciem variedade nos comportamentos, mas que o nível de dificuldade vá acompanhando os resultados do jogador, correspondendo em simultâneo aos critérios: variedade de comportamentos e ajuste da dificuldade.

5. Avaliação com Playtesting

Após a implementação do comportamento dos agentes, pretendia-se avaliar vários aspectos relacionados com a percepção da experiência pelos jogadores. As questões essenciais eram: a) *A evolução no comportamento é perceptível?* b) *O comportamento é mais defensivo ou ofensivo?* c) *A evolução mantém o “interesse”?*

Para a validação dos resultados, foram desenvolvidas duas versões do jogo: uma com a

atualização “inteligente” anteriormente descrita e outra com a *atualização aleatória* da Petri Net, como referencial. Realizaram-se 4 sessões de playtesting, com um painel de alunos do ensino superior familiarizados com o jogo PacMan original. Foram executados simultaneamente 10 jogos com *atualização “inteligente”* e 10 jogos com *atualização aleatória*. A velocidade do jogo foi previamente ajustada ao jogador, após a realização de algumas sessões de ambientação, mantendo-se então constante. Após cada jogo foram avaliados vários aspectos numa escala diferencial semântica de 1 a 5, enquanto a pontuação e tempo de jogo foram registados automaticamente. No final das sessões foram ainda avaliados alguns aspectos de carácter geral. O sumário dos resultados obtidos durante as sessões de *playtesting* apresenta-se nas tabelas 1, 2 e 3.

Critério	Intel.	Alea.
Emotivos (1) .. Aborrecidos (5)	2,93	3,40
Espírito de Equipa (1) .. Individualista (5)	2,83	3,20
Agressivos (1) .. Passivos (5)	2,95	2,97
Fácil (1) .. Difícil (5)	3,20	2,77
Gostou (1) .. Não gostou (5)	2,33	3,13
Vitória (1) .. Derrota(0)	0,53	0,70
Pontuação média	11,13	10,43
Tempo de jogo médio	86,48	68,55

Tabela 1 – Médias, *atualização “inteligente”* ou “aleatória”

Ritmo? Muito lento(1) ... Muito rápido(5)	3
Notou evolução? Nenhuma(1) ... Significativa(5)	3
Adversários? Defensivos(1) ... Ofensivos(5)	2,75
Dificuldade do jogo? Fácil(1) ... Difícil(5)	2,75

Tabela 2 – Valores médios para a apreciação geral

Correlação Gosto/Dificuldade	-0,5
Correlação Gosto/Tempo de jogo	-0,3

Tabela 3 – Correlação de dificuldade e tempo de jogo com a apreciação da experiência de jogo

Comparando os valores médios obtidos, vários indicadores apontam para a superioridade da versão com *atualização “inteligente”*. Apenas ocorreram vitórias do jogador em 53% dos jogos, ao contrários dos 70% na versão com *atualização aleatória*, destacando-se por ter jogos mais apreciados e mais equilibrados, julgando pela média “Gostou/Não gostou” e pelo tempo médio de jogo superior. Analisando alguns indicadores auxiliares, é possível verificar uma correlação entre o prazer retirado do jogo e a dificuldade. Neste caso, uma correlação inversa, deduzindo-se que os jogadores preferem jogos mais desafiantes e equilibrados. Em geral, os jogadores notaram evoluções significativas no comportamento dos agentes mas não distinguiram uma estratégia predominante, quer defensiva quer ofensiva, nem tendência para um jogo mais fácil ou difícil. Estes são resultados interessantes na medida em que um dos objectivos era manter a variedade de comportamentos, sem perda de competitividade. É ainda interessante notar a interpretação que os jogadores fazem de que na versão com *atualização “inteligente”* os agentes

parecem evidenciar maior “espírito de equipa” do que na versão aleatória.

6. Conclusões

Como conclusão deste trabalho é evidente que existe um potencial importante para uso das Petri Nets na modelação de jogos: a extensibilidade, expressividade e computabilidade dos modelos é essencial como ferramenta de design. Conclui-se que é possível introduzir variedade e dinamismo nos comportamentos com base na adaptação automática dos arcos de uma Petri Net, mantendo um nível adequado de competitividade. Os elementos de jogo descritos num modelo formal tornam-se facilmente reconfiguráveis, abrindo espaço de estudo de outros métodos ou mecanismos que introduzam variedade na experiência.

7. Trabalho Futuro

A *atualização* do jogo em tempo de jogo abre um leque grande de possibilidades, especialmente para jogos mais longos. A adaptação do comportamento do agente poderá resultar em jogos mais dinâmicos e complexos, para tal será necessário trabalhar a definição de *métricas de gameplay* que permitam a avaliação e o ajuste dinâmico de comportamentos. Para além da *atualização* ser feita com base nos comportamentos e resultados do próprio agente, poderia ainda o contexto ou informação dos agentes à sua volta, permitindo a emergência de comportamentos sociais complexos. Assim poderemos ter agentes que aprendam a jogar com base em fenótipos de estratégias experimentadas por outros agentes, incluindo humanos, introduzindo-se a comunicação, e através dela a coordenação, como um elemento essencial a uma experiência mais complexa e rica.

Referências

- MATEAS, M. AND STERN, A. Façade: An Experiment in Building a Fully-Realized Interactive Drama, Game Developers Conference, 2003.
- MYERS, D. In search of a minimalist game. In *Proceedings of DiGRA 2009 Conference*, West London, UK, 2009.
- HUNICKE, R., LEBLANC, M. AND ZUBEK, R. MDA: A formal approach to Game Design and Game Research. In *Proc. of Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*. 2004.
- MATEAS, M. AND WARDRIPE-FRUIIN, N. Defining operational logics. In *Proceedings of DiGRA 2009 Conference*, West London, UK, 2009.
- MURATA, T. Petri Nets: Properties, analysis and applications. *Proceedings of the IEEE*. 1989, Vol. 77.
- SALEN, K. AND ZIMMERMAN, E. Rules of Play: Game Design Fundamentals. s.l. : The MIT Press, 2004.
- ARAÚJO, M. AND ROQUE, L. Modeling games with petri nets. In *Proceedings of DiGRA 2009 Conference*, West London, UK, 2009.
- Contest: Pacman capture the flag. [Online] <http://inst.eecs.berkeley.edu/~cs188/fa07/projects/contest/contest.html>.