

**Objectivo:** Explorar o modelo relacional-estendido no suporte à manipulação de informação descrita através de estruturas complexas. Aprofundar os aspectos de modelação e de utilização de extensões espaciais. Utilização do sistema de gestão de base de dados PostgreSQL e sua extensão PostGIS. Utilização de uma ferramenta (e.g., Quantum GIS ou uDig) para representação em camadas (“layers”) de informação espacial. Conhecer a relação entre os dados complexos e os mecanismos para a sua visualização. Explorar formas de recuperar informação (e.g., via SQL) que contenha dados alfanuméricos e de estrutura complexa.

## Problema:

Consideremos, como inspiração, o conceito de “realidade aumentada” (*augmented reality*) onde entidades do mundo real coexistem com entidades virtuais. Esta coexistência projeta-se num **novο espaço virtual complementado pela realidade**. Vamos focar-nos num **“espaço virtual” (2D) caracterizado pela existência de vários tipos de terreno e de vários objetos móveis que se deslocam nesses terrenos**.

**Terrenos, rios e objetos móveis.** Cada tipo de **terreno**, para além de um **nome (único)** tem também o **efeito de alterar a velocidade (linear) dos objetos que sobre ele se deslocam**. Assim que um objeto entra num terreno a sua velocidade é afectada, num valor percentual (e.g.,  $\pm 0\%$  a  $100\%$ ), e o efeito permanece enquanto o objeto estiver nesse terreno. Notar que a alteração da velocidade é o “aspecto visível” pois a alteração na velocidade resulta da variação sobre a aceleração. **Um terreno, de determinado tipo, pode interseccionar terrenos de outros tipos**. Por exemplo, um terreno do **tipo floresta** pode **conter um terreno do tipo pântano**; note-se que o mesmo **pântano pode cruzar várias florestas**. Assume-se que o **terreno que inclui todos os restantes não precisa de estar explicitamente representado** e não afecta a mobilidade dos objetos. Note-se que há uma hierarquia de inclusão entre terrenos (e.g., a floresta inclui pântanos e possivelmente não o inverso) pelo que **um objeto é afectado pelo tipo de terreno que estiver no nível de maior profundidade da hierarquia de terrenos que o contém**. Para além de terrenos existem também rios que têm **precisamente o mesmo efeito sobre os objetos que sobre ele se deslocam**. No entanto, nem todos os objetos são afectados do mesmo modo pelos terrenos (ou rios). **Existem tipos diferentes de objetos móveis pelo que a redução (percentual) da velocidade depende do tipo de terreno (e.g., uma floresta não reduz tanto como um pântano)**.

**Objetos móveis.** Cada **objeto móvel é um ponto geográfico** que representa, na perspectiva do movimento, todo o objeto. O objeto **tem também uma orientação** (ângulo, medido em graus ou radianos), **uma velocidade e aceleração linear** (com componentes x e y), e uma **velocidade e aceleração angular**. **Admitindo que “g\_posicao” representa a posição geográfica do objeto, “orientacao” a sua orientação, “velocidade.linear” um vector com as componentes (x e y) da velocidade, “velocidade.angular” a velocidade angular do objeto e “tempo” um instante de tempo**, podemos descrever o processo de actualização destas variáveis que descrevem o movimento (cinemática) do seguinte modo:

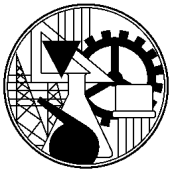
- $g\_posicao := g\_posicao + velocidade.linear * tempo$
- $orientacao := orientacao + velocidade.angular * tempo$
- $velocidade.linear := velocidade.linear + aceleracao.linear * tempo$
- $velocidade.angular := velocidade.angular + aceleracao.angular * tempo$
- a aceleração (linear e angular) pode ser constante ou ser calculada para: a) simular uma perseguição, ou b) simular uma interação, do tipo “jogo”, com o utilizador (onde o utilizador controla o movimento).

Considera-se que **cada objeto móvel representa o centróide de um componente geométrica que lhe “dá corpo”**. Ou seja, associado a cada objeto móvel **existe uma “figura geométrica” que define o aspecto visual do objeto móvel**. **Todos os movimentos devem ser mantidos em histórico para permitir visualizar (e.g., com QuantumGIS) as trajetórias dos vários objetos móveis ao longo do tempo**.



## Objectivos base:

1. Desenhar o modelo conceptual usando a notação do entidade-associação estendido com pictogramas espaciais (EA-EPE) que descreve o problema acima enunciado.
2. Implementar o EA-EPE com o modelo relacional-estendido e a extensão espacial (e.g., usando PostGIS).
3. Implementar, diretamente na base de dados, o comportamento dos objetos móveis e das correspondentes figuras geométricas.
4. Construir uma base de dados com (pelo menos) 2 tipos de terreno e (pelo menos) 2 terrenos de cada tipo e com (pelo menos) 2 objetos móveis com as respectivas figuras geométricas. Registrar, na base de dados, o trajeto dos 2 objetos móveis e das respectivas figuras geométricas. Mostrar o efeito, sobre a velocidade (via alteração da aceleração), da passagem, dos objetos, em terrenos e rios.
5. Garantir que os objetos móveis têm uma velocidade máxima e uma aceleração máxima que nunca são excedidos. Esses valores máximos estão registados na base de dados e qualquer trajetória garante sempre estas restrições de valores máximos (para velocidade e aceleração).
6. Considerar que um objeto móvel pode perseguir outro mantendo registo dessa informação (sobre "quem persegue quem") na base de dados (notar que o modelo EA-EPE tem que representar este requisito). Construir trajetória de perseguição na qual se pode admitir, para o perseguidor, a seguinte formulação:
  - $aceleracao := normalizar( alvo.g\_posicao - perseguidor.g\_posicao ) * boost$   
onde "alvo" representa o objeto a ser perseguido, "perseguidor" o objeto que persegue e "boost" o incremento que se pretende imprimir à aceleração (que pode ser positivo ou negativo). Note que o objeto que é perseguido (i.e., o alvo) não "foge" do perseguidor (é "como se não soubesse que é perseguido").



## Objectivos mais ambiciosos:

7. Considerar um cenário de perseguição e fuga. Registrar na base de dados “quem persegue e quem foge” e estenda o modelo de simulação (na base de dados) de modo construir trajetórias com este cenário. Notar que a formulação do objeto que foge (i.e., o alvo) pode escrever-se como:
  - $aceleracao := - \text{normalizar}(alvo.g\_posicao - perseguidor.g\_posicao) * boost$   
ou seja o “alvo” (objeto que foge) afasta-se na medida (simétrica) da aproximação do perseguidor; o “boost” pode ajustar-se para dar vantagem ao perseguidor ou ao alvo. Note que agora o objeto que é perseguido (i.e., o alvo) já “foge” do perseguidor (é “como se soubesse que é perseguido”).
8. Permitir que diversos objetos móveis persigam simultaneamente um outro objeto (mas cada perseguidor apenas tem um alvo). O alvo pode, por exemplo, fugir em cada instante de quem lhe estiver mais próximo.
9. Alinhar a orientação de cada figura geométrica com a direcção do movimento. *Sugestão:* em cada instante considerar a nova orientação, do objeto, dada por `atan2( velocidade.linear.x, - velocidade.linear.y )`.
10. Construir uma interface que permita alterar em cada instante de tempo os valores da aceleração dos vários objetos móveis. Numa versão mais elaborada essa interface pode posicionar o cursor do rato sobre o botão de “Refresh” do QuantumGIS e enviar evento de “click” em cada instante de tempo e assim usar o QuantumGIS para ir fazendo, de modo automático, a apresentação da evolução da cena de perseguição. *Sugestão:* para controlar posição do cursor e enviar eventos para outras aplicações use `“java.awt.Robot”`.

## Algumas sugestões:

- Pode considerar que o espaço onde os objetos existem tem um formato “tipo toroidal”; isto simplifica o tratamento do movimento dos objetos pois o espaço deixa de estar limitado. Neste caso a próxima posição de um objeto deve “rodar” em cada uma das dimensões (e.g.,  $nextX := (X + deslocamentoEmX) \% lengthX$ , onde  $a \% b$  é o resto da divisão inteira de  $a$  por  $b$ ; ou função `mod(a,b)` no PostgreSQL).
- Pode considerar que uma perseguição termina assim que os objetos envolvidos distem, um do outro, um determinado valor predefinido e do qual se mantém registo na base de dados.
- Pode considerar que o objeto em fuga tem um movimento que segue uma rota predefinida; a rota pode ser representada por pontos marcados sobre os terrenos onde os objetos se movimentam.

**Implementação:** Construir “scripts” e executáveis de resposta às alíneas anteriores. O modelo final (constituído pelos “scripts” e restante código) deve permitir construir todo o modelo e diferentes trajetórias do modo automático. Integrar a base de dados com um visualizador de informação geográfica (e.g., Quantum GIS).

**Regras:** Entregar versão em papel do relatório de projeto, no máximo com 10 folhas; a primeira folha deve ter: título, nome da disciplina e curso, identificação do grupo, número e nome de cada elemento do grupo. Entregar (submeter no Moodle): em formato electrónico (CSI\_SIGM\_XX.zip, XX é número do grupo) o relatório e todo o sistema implementado com “.exe”, “.java”, “.py” ou “script” (.bat) para construir o sistema.

**Prazo de entrega:** 09.dez.2024 (para realizar discussões antes do início da época de exames, i.e., até 06.jan)