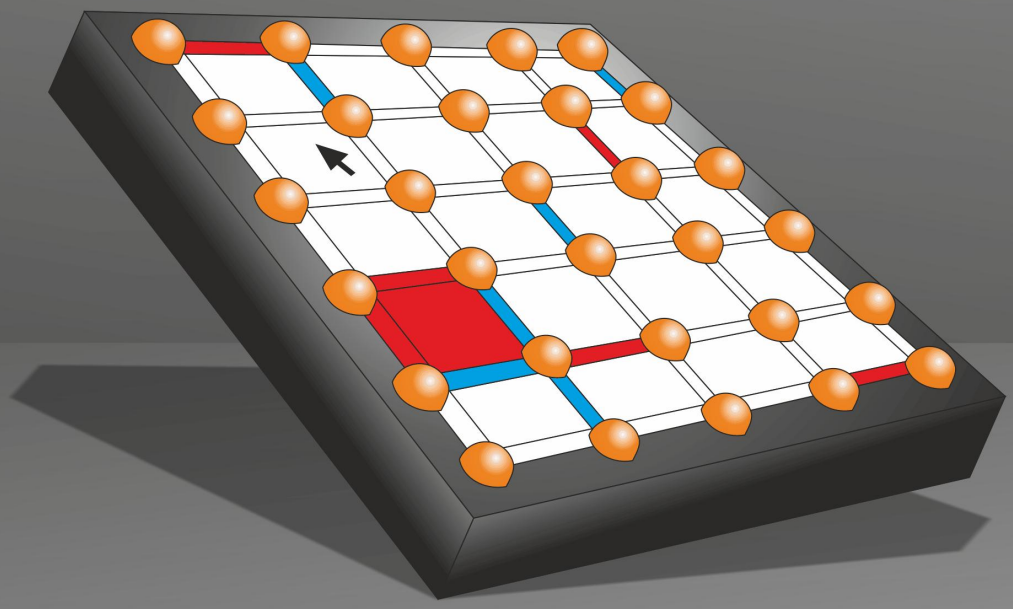
UNIVERSIDADE DE ÉVORA

dOtes & bOxes

Programação I



Daniel Ramos 29423
Simão Ramos 29035

2011/2012

Introdução

O jogo de papel “Dots and Boxes”, jogado entre dois oponentes, sejam estes humanos ou humano contra computador, é um jogo de simples estratégia que tem como objetivo a conquista do máximo de quadrados presentes no tabuleiro. O tabuleiro é uma grelha de quadrados delimitada em comprimento e largura pelo número de quadrados que se queira, podendo este ser um simples quadrado ou um retângulo com quaisquer medidas. Normalmente os quadrados não são visíveis até estes serem conquistados no decorrer do jogo, e a demarcar o espaço de jogo estão vários pontos que representam os vértices dos quadrados. O objetivo é ligando os vários pontos entre si, lado a lado, vez a vez, ate todas as ligações estarem completas, e quando todos os vértices de um quadrado são ligados entre si esse é preenchido com a cor ou símbolo identificativo do jogador que fez a ultima ligação, ganhando este direito a uma jogada extra. O vencedor é o jogador que tiver preenchido mais quadrados no tabuleiro.

O objetivo deste trabalho é implementar este jogo em Python em várias vertentes:

- modo de texto, em que o jogador interage com um tabuleiro construído num sistema de coordenadas do 1º quadrante de um referencial ortonormado (ordenadas e abcissas são positivas, e o ponto (0,0) encontra-se no canto inferior esquerdo).

O programa deverá desenhar uma grelha de pontos consoante a decisão do jogador; fazer as ligações entre as coordenadas pretendidas reconhecendo jogadas anteriores e impedindo a repetição das mesmas; reconhecer quando todas as quatro ligações em volta do quadrado estão preenchidas e atribuir esse quadrado ao jogador que fez a ultima ligação dando-lhe a possibilidade de repetir uma nova jogada; reconhecer quando o utilizador insere algo diferente das coordenadas de pontos requeridas pelo jogo; reconhecer o jogador vencedor e os empates…

-modo gráfico, usando a suíte de programas em Python, *Swampy,* em que o programa apresenta ao jogador um tabuleiro de jogo construído com as predefinições pedidas pelo mesmo anteriormente.

O programa tem de ser capaz de apresentar um tabuleiro construído com retângulos e outras formas geométricas; terá de ser o máximo apelativo em termos visuais apelando à interação do jogador com o jogo; reconhecer o rato e as interações entre este e o tabuleiro; da mesma forma que o modo de texto tem de reconhecer jogadas anteriores e impedir a repetição, preencher os quadrados que têm todos os vértices ligados entre si, e preencher o mesmo com a cor do jogador…

-modo humano vs. Computador

Implementado em ambas as vertentes, modo de texto e gráfico, o programa terá de ser capaz de efetuar jogadas permitidas pelas regras, enquanto imprime algum grau de dificuldade ao jogo.

Sendo estes os principais objetivos em mente durante a programação do jogo “Dots and Boxes” predispusemo-nos a resolvê-los na sua completude, perfeitamente conscientes que existirão bastantes mais problemas subjacentes à programação do jogo com os quais nos depararemos e outros que nem conhecimento teremos eu que por sorte ou consistência do código escrito serão resolvidos, ou não…

Desenvolvimento

O programa foi delineado por partes que correspondiam às diretrizes principais do trabalho (modo de texto, modo gráfico, humano vs. Computador

Modo de Texto:

Este modo está construído dentro de um loop principal responsável por várias questões.

Está responsável por verificar se há quadrados completos no tabuleiro do jogo, e para o caso de ter acabado de se preencher um permite ao último jogador, jogar de novo. Dentro deste loop existe outro que verifica também as ligações que foram efetuadas anteriormente, e constrói o tabuleiro de jogo durante o mesmo. O tabuleiro de jogo por falta de opções gráficas é construído recorrendo ao uso de strings. Aquando de alguma alteração do tabuleiro de jogo, o programa imprime um tabuleiro novo com a alteração anterior.

Pergunta também pelas coordenadas que os utilizadores terão de pôr para as jogadas, verificando se o input é permitido e devolvendo uma mensagem de alerta e a permissão de voltar a colocar a jogada.

Este loop principal verifica também qual o jogador a jogar em seguida, quando está todo o tabuleiro jogado, verificando para os casos de empate, e decidindo quem ganha.

Modo gráfico:

Para o modo gráfico do jogo “Dots and Boxes” foi utilizado a suíte de Programas em Python, Swampy, e entre estes, o módulo World, do qual é importado o Canvas (um componente de uma interface gráfica do usuário (GUI)). Usando este widget no qual é possível desenhar todo o tipo de formas geométricas, usaram-se retângulos para construir as linhas e colunas que separam os vários vértices dos quadrados, representados por círculos.

O Canvas no entanto define o ponto (0,0) como o centro da janela. Sendo este o primeiro problema necessário a resolução, uma vez que todas as coordenadas de desenho para poderem ser usadas mais eficientemente teriam de ser positivas. Este problema foi resolvido usando uma função shift que transformava o ponto (0,0) num ponto que se quisesse na janela do canvas, colocando-o então no canto inferior esquerdo da janela, mas dando-lhe uma margem entre o tabuleiro e a janela por uma questão puramente estética. Muito do código foi rearranjado após a introdução da função shift, uma vez que até à descoberta desta foi desenhado o tabuleiro com as coordenadas predefinidas pelo Canvas, tornando-se incrivelmente difícil por vezes formar algoritmos para os cálculos de algumas coordenadas, e outras completamente infrutíferas e falhadas.

Em seguida era necessário rearranjar a janela do canvas conforme as dimensões do tabuleiro dando também a respetiva margem por questões de estética.

Na construção do tabuleiro usaram-se dois loops (for) que desenham o tabuleiro linha a linha, desenhando ao longo do eixo x, e em seguida ao longo do eixo y, tomando como pontos de referência os cantos inferiores esquerdos de cada agrupamento de objetos. Foram portanto usadas classes para construir os objetos presentes no tabuleiro com os cantos anteriormente mencionadas como referência. A escolha de classes permitiu resolver vários

problemas como as colisões do rato entre os objetos, originando a mudança de cor ou não do mesmo, dependendo da viabilidade da jogada e do jogador.

Em seguida, introduziram-se as funções necessárias para interagir o rato com a janela de jogo, sendo necessário também uma inversão de coordenadas para adaptar à anterior inversão na janela do Canvas, e assim haver congruência entre as coordenadas do rato e da janela.

As colisões em conjunto com as verificações das jogadas ou não permitidas foram o passo seguinte. Em primeiro pensou-se na área dos objetos, não a usando diretamente. Para verificar se a colisão em questão alterava algo ou não no jogo, implementaram-se várias restrições que delimitavam as coordenadas onde poderia ser algo alterado e em consequência identificar o que foi alterado e alterando-o de facto. Um redesenho anterior do tabuleiro tornou-se bastante útil neste passo, uma vez que nesse os retângulos que simbolizavam as linhas e colunas, intersectavam-se nos vértices, e no desenho final estes topam vértice a vértice. As colisões ficaram confinadas aos retângulos das linhas e colunas.

Nas verificações das jogadas usou-se um dicionário para armazenar as tuples com as coordenadas correspondentes ao objeto alterado anteriormente. A verificação ocorre aquando de nova colisão o programa “varre” o dicionário e se neste estiver presente o objeto que foi tocado, a jogada não se dá.

Para alterar a cor dos quadrados quando os lados adjacentes estão todos preenchidos usou-se o mesmo dicionário onde estão armazenadas as tuples das coordenadas das jogadas anteriores, e o programa verifica esse dicionário em busca, e contando com a jogada anterior dos lados adjacentes aos quadrados, e os que estiverem preenchidos altera a cor do quadrado consoante o jogador.

O programa no fim do jogo conta o número de quadrados com determinada cor (corresponde a um jogador) e aquele que tiver maior número ganha o jogo.