

Linguagens e Ambientes de Programação (2019/2020) [Eng.Inf. - DI/FCT/UNL]

Enunciado do 3º Projeto Prático (JavaScript)

Artur Miguel Dias

Datas

- 22/maio (23:30) - Divulgação da versão provisória do enunciado
 - 25/maio (18:00) - Divulgação da versão final do enunciado
 - 30/maio (20:00) - Data limite de entrega sem penalização
 - 03/junho (22:00) - Data e hora limite de entrega com atraso. Um valor de penalização por cada dia de atraso.
-

Changelog

- 22/maio - Possíveis correções ou melhorias deste enunciado serão assinaladas aqui.
-

Objetivos

Por favor, escreva o programa que se descreve a seguir num estilo orientado pelos objetos impecável. Concretamente, organize o seu programa como uma comunidade descentralizada de objeto autónomos que interajam por meio de mensagens. Além disso, fatorize bem as suas classes e escreva código extensível: pense que, de futuro, podemos desejar acrescentar novos tipos de personagens e que isso deverá poder ser feito adicionando apenas código novo, sem praticamente alterar quase nada do código já escrito.

Em termos de organização, o principal desafio deste trabalho é descobrir como escrever um programa envolvendo múltipla entidades que interagem de formas variadas, sem com isso comprometer a extensibilidade do código. (A extensibilidade, foi o tema da aula teórica 22.)

O objetivo deste projeto é desenvolver em HTML e JavaScript uma página Web interativa que implemente este jogo. Deverá criar um formulário HTML com um retângulo gráfico (canvas), mais alguns elementos tais como botões e caixas de texto. É dentro do canvas que decorre a ação do jogo. Os elementos envolventes servem para atuar sobre o jogo (e.g. um botão "Restart" para recomençar o jogo), e para mostrar informação sobre o decorrer do jogo (e.g. mostrar número de unidades de tempo que já passaram).

Como ponto de partida do seu trabalho, use estes dois ficheiros: [LodeRunner.html](#) e [LodeRunner.js](#). Mais tarde, nas regras de submissão, será explicada a forma de submeter estes dois ficheiros no Mooshak. Utilize ainda o ficheiro [LodeRunner0.js](#), mas este não é para submeter e não é suporto ser alterado.

O código de partida que lhe oferecemos já fornece tudo o que é necessário para lidar com gráficos, teclado e eventos de animação. O programa já está metade feito. O que falta é programar o comportamento dos atores dentro das respetivas classes, respeitando rigorosamente este enunciado. **Atenção que o código do ficheiro "LodeRunner.js" não é sagrado.** Você poderá precisar de apagar partes do código e trocar por

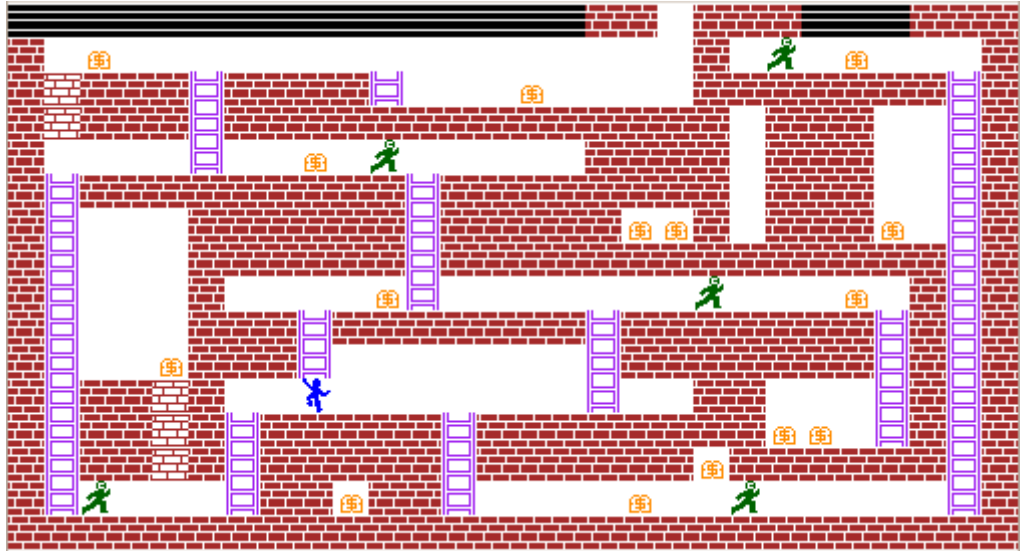
código novo. De certeza que vai precisar de inventar bastantes novos métodos nas classes. Inclusivamente poderá criar classes novas.

Quanto ao ficheiro "LodeRunner.html", o formulário HTML que lhe oferecemos é apenas um esboço que você deverá melhorar para criar uma interface com alguma utilidade para o jogo.

Lode Runner

Introdução

O inventor do jogo [Lode Runner](#) foi [Doug Smith](#), em 1983 estudante da Universidade de Washington. No verão desse ano, ele tentou vender o jogo a algumas companhias de software e recebeu algumas ofertas interessantes. Acabou por escolher a empresa [Brøderbund](#) que lhe pagou uma quantia inicial de 10.000 dólares mais 23% sobre todos os lucros futuros.



O jogo Lode Runner é simples de jogar e combina ação em tempo real com elementos de estratégia. Estes fatores juntos tornaram-no particularmente popular.

Neste projeto de programação, pedimos-lhe que programe em JavaScript uma variante do jogo Lode Runner, muito parecida com a versão original para Macintosh. Usaremos os gráficos originais, mais os cenários oficiais correspondentes aos 16 primeiros níveis. No nosso programa, as três diferenças mais importantes relativamente ao jogo original são as seguintes: (1) a movimentação dos personagens é mais rudimentar, sendo feita ao nível do quadrado e não do pixel; (2) as regras de disparo da arma pelo herói são diferentes; (3) os buracos criados pela arma são simples casas vazias que nunca se regeneram.

Funcionamento geral do programa

O maléfico Império de Bungeling confiscou uma enorme quantidade do ouro dos seus legítimos proprietários. A missão do herói é infiltrar-se no Edifício do Tesouro e tentar reaver a fortuna roubada. Para isso, ele terá de percorrer diversos ecrãs de plataformas, escadas e cordas, tentando sempre evitar qualquer contacto com os guardas robóticos que protegem o ouro; um tal contacto causaria ao herói a morte.

O herói tem duas técnicas à disposição para evitar os robôs. A primeira consiste em simplesmente fugir, correndo através de plataformas e usando escadas e cordas; também se pode deixar cair numa plataforma ou numa corda mais abaixo. A segunda técnica envolve uma arma que, apesar de não ter eficácia quando usada diretamente contra os robôs, permite ao herói abrir buracos no chão, através dos quais os robôs caem para as plataformas mais abaixo.

Para dificultar a tarefa do nosso herói, por vezes parte do ouro aparece enterrado, e nesse caso o herói precisa de usar a sua arma para abrir buracos e chegar ao ouro. Também para complicar a vida do herói, quando um robô passa por uma bolsa de ouro, normalmente o robô agarra na bolsa e transporta-a até outro local.

Logo que o herói recolhe todo ouro, surge na parte superior do ecrã uma escada nova. Essa escada permite ao herói aceder ao nível seguinte do jogo.

O mundo

O mundo onde decorre o jogo é um retângulo de 28 x 16 casas. Os limites do retângulo são intransponíveis.

Existem 16 cenários distintos, já definidos dentro do ficheiro "LodeRunner0.js". Quando o programa arranca, o cenário correspondente ao nível 1 é automaticamente carregado e o jogo começa.

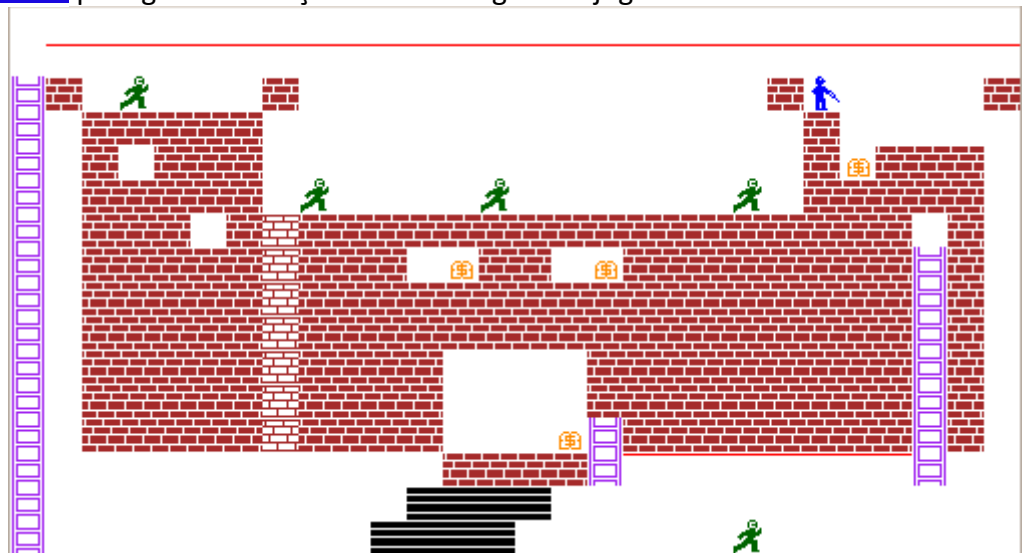
Repare que os atores móveis podem circular por cima de alguns dos atores passivos (por exemplo, o Herói pode estar a subir uma Escada). Por isso, internamente, o mundo precisa definido usando duas camadas (duas matrizes), para não se perder informação.

Assista alguns minutos deste [video](https://www.youtube.com/watch?v=PWwyhymcDxI) para ganhar intuição sobre as regras do jogo.

Os atores

Vamos conhecer os atores do jogo, nomeadamente o comportamento e objetivos de cada um.

Há duas grandes classes de atores: (1) os atores ativos, que se podem movimentar através do mundo: Herói e Robôs; (2) os atores passivos, que não se podem movimentar: Tijolo, Pedra, Escada, Corda, Ouro, Chaminé, Vazio.



<https://www.youtube.com/watch?v=PWwyhymcDxI>

- Herói

Em cada nível, o **objetivo** do Herói é apanhar o mais rapidamente possível todo o Ouro, ao mesmo tempo tentando evitar ser morto por algum Robô.

O Herói é controlado pelo teclado. Movimenta-se na horizontal e na vertical; nunca nas diagonais. Normalmente, o Herói só se movimenta quando há realmente teclas de movimento premidas e quando o contexto envolvente não impede o movimento. Mas, numa situação de queda, o Herói movimenta-se por si mesmo, de forma independente do teclado.

O Herói consegue correr através de plataformas, subir/descer Escadas e percorrer Cordas. Também se pode deixar cair numa plataforma ou numa Corda mais abaixo, atravessando espaços livres e Chaminés. O salto pode ser feito a partir duma plataforma, duma Escada ou duma Corda. Se o Herói cair num buraco, atravessa-o e vai continuando a cair até atingir uma plataforma, Escada ou Corda mais abaixo. Mas, se ele cair num buraco duma plataforma rente ao chão, já não se conseguirá libertar e fica derrotado.

O Herói também é derrotado se algum Robô o conseguir alcançar e tocar-lhe.

Sempre que o Herói passa por um bolsa de Ouro, agarra-a e conserva-a durante todo o nível.

O Herói dispõe duma arma que lhe permite abrir buracos no chão das plataformas, mas apenas em zonas de Tijolos. O número de tiros à disposição do Herói é ilimitado. Em condições ideais, um tiro tem dois efeitos: (1) um Tijolo é substituído por um espaço livre; (2) em virtude do recuo da arma, o Herói recua uma casa e fica parado com a arma na mão, podendo continuar a dar mais tiros e a recuar mais, se o desejar. Mas nem sempre os dois efeitos se verificam ao mesmo tempo:

- Para um tiro abrir efetivamente um buraco, têm de verificar as duas seguintes condições:
 - O tiro tem de ser sobre um Tijolo da plataforma, numa posição adjacente ao Herói, para o lado em que ele está virado;
 - Por cima do Tijolo a destruir não pode estar Tijolo ou Pedra.
- Para o recuo do disparo se efetivar, têm de verificar as duas seguintes condições:
 - Nas costas do Herói pode estar qualquer coisa menos Tijolo ou Pedra, porque estes o herói não consegue atravessar.
 - Aos pés da casa nas costas do Herói tem de estar Tijolo, Pedra, Escada. Estes são os únicos atores que oferecem garantidamente sempre chão firme ao recuo do Herói e assim o recuo do Herói nunca tem o risco de queda.

O Herói vence um nível se as três seguintes condições se verificarem:

- Já apanhou todo o ouro disponível. (Note que, nesta situação aparece uma escada suplementar.)
- Está sobre uma escada.
- Encontra-se no topo do cenário, portanto numa posição com coordenada $y = 0$.

- Pedras

As Pedras não se movimentam e são elementos constituintes das plataformas.

As plataformas são vias de passagem horizontal do jogo. Podem ser percorridas pela parte de cima, tanto pelo Herói como pelos Robôs.

As Pedras são indestrutíveis.

- Tijolos

Os Tijolos não se movimentam e, tal como as Pedras, também são elementos constituintes das plataformas.

Um Tijolo pode ser destruído por um disparo da arma do herói, ficando um espaço livre no seu local.

- Chaminés

As Chaminés não se movimentam e comportam-se para todos os efeitos como espaços livres. Herói e Robôs podem cair e atravessá-las em queda.

As Chaminés são indestrutíveis.

- Escadas

As Escadas não se movimentam e constituem vias de passagem vertical do jogo. Podem ser subidas e descidas pelo Herói e pelos Robôs.

As Escadas são indestrutíveis.

- Cordas

As Cordas não se movimentam e constituem vias de passagem horizontal do jogo. Podem ser atravessadas pelo Herói e pelos Robôs.

As Cordas são indestrutíveis.

- Ouro

As bolsas de Ouro não se movimentam autonomamente, mas podem ser agarradas pelo Herói e pelos Robôs.

Quando o Herói agarra uma bolsa de Ouro, ele nunca mais a larga. Quando um Robô apanha uma bolsa de Ouro, ele só a guarda temporariamente.

As bolsas de Ouro são indestrutíveis.

- Robôs

Os Robôs dispõem de movimento autónomo e andam sempre atrás do Herói. Movimentam-se na horizontal e na vertical; nunca nas diagonais.

Os Robôs têm as mesmas capacidades de movimentação do Herói. Por exemplo, conseguem correr através de plataformas, subir/descer Escadas e percorrer Cordas.

Os Robôs são pouco inteligentes e só se preocupam em diminuir a distância em linha reta ao Herói, mesmo que isso seja inútil pelo facto de haver barreiras pelo caminho.

Quando um Robô passa por uma bolsa de Ouro, o Robô agarra-a e conserva-a durante um período de tempo fixo. Depois o Robô largará a bolsa na primeira oportunidade que surgir. O Robô só pode largar a bolsa sobre um Tijolo ou uma Pedra. O Robô também larga imediatamente a bolsa, no caso de cair num buraco aberto pelo herói.

Um Robô só consegue transportar uma bolsa de cada vez. Se já estiver transportando uma bolsa, ele ignora as outras bolsas.

Quando um Robô cai num buraco aberto pelo Herói, o Robô fica bloqueado no buraco durante um tempo fixo (não continua a cair). Quando passa o tempo de bloqueio, o Robô liberta-se, saindo pelo lado de cima. O buraco fecha-se, nascendo lá um Tijolo.

Teclas

O Herói é controlado através teclado, por meio das seguintes teclas:

- 'O', 'J', '←' - para a esquerda.
- 'P', 'L', '→' - para a direita.
- 'Q', 'I', '↑' - para cima.
- 'A', 'K', '↓' - para baixo.
- tecla ' ' - disparar no lado em que está voltado.

O jogo Lode Runner original usa o conjunto de teclas IJKL.

A parte de gestão das teclas já está programada, exceto o caso da tecla ' ' (barra de espaço).

Partes não especificadas do programa

Você deve respeitar as regras do jogo e tudo o que está escrito neste enunciado.

Contudo, há aspetos do programa que este enunciado não especifica.

Por exemplo, o enunciado não diz quais são os controlos HTML junto ao canvas. Algo de esteticamente elegante e com 4 elementos úteis, relevantes para o programa, já permitirá obter a nota máxima nesta

parte, que é de apenas 1 valor.

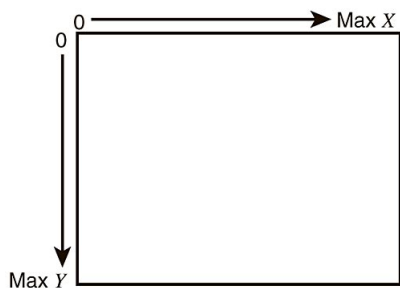
O enunciado não diz o que acontece quando termina um nível, com vitória ou derrota do Herói. O enunciado também não diz como se contabiliza a pontuação.

Estes e outros aspetos não especificados, você tem a liberdade de os tratar da forma que lhe parecer melhor para obter um bom programa, agradável de usar.

O sistema de coordenadas

O jogo decorre num mundo a 2 dimensões, onde cada casa é identificada por um par de coordenadas.

Nos sistemas gráficos, a **coordenada x** indica a distância em casas contada a partir da esquerda e a **coordenada y** indica a distância em casas contada a partir de cima. Portanto a origem do sistema de coordenadas está no topo esquerdo, as coordenadas x crescem da esquerda para a direita e as coordenadas y crescem de cima para baixo. O seguinte diagrama ajuda a tirar dúvidas:



Regras

- Em condições normais, este projeto seria realizado obrigatoriamente por grupos de dois alunos. Mas, tal deixou de ser obrigatório, considerando a situação especial em que nos encontramos. Também, considerando a situação especial em que nos encontramos, os alunos podem formar grupo com toda a liberdade, independentemente dos turnos em que estejam inscritos. **No caso deste terceiro projeto, não existe bónus por se fazer em grupo de dois.**
- Os grupos só podem ser de um aluno ou de dois alunos. Um projeto entregue por três ou mais alunos vale zero valores.
- Será ativado um concurso do Mooshak, que servirá para submeter os trabalhos. Os detalhes da forma de fazer a submissão serão divulgados nessa altura. Até lá preocupe-se apenas em escrever um bom programa.
- Depois do prazo limite ainda se aceitam trabalhos atrasados, mas com penalizações na nota. Mais detalhes nas primeiras linhas deste enunciado.
- Apesar de o projeto ser de grupo, cada aluno, a título individual, tem a responsabilidade de responder por todo o projeto. Assim, é indispensável que os dois membros de cada grupo programem efetivamente. O ideal seria que cada grupo dividisse o projeto em duas partes, e que o desenvolvimento de cada uma das duas partes fosse "comandada" por um aluno distinto, sempre em diálogo com o outro.
- Não há lugar a qualquer pré-inscrição dos grupos. Basta que nos trabalhos submetidos figurem nomes de alunos inscritos na cadeira.

- A nota máxima do projeto é 20 valores.
-

Avaliação

Os docentes responsáveis pela gestão e pela avaliação deste trabalho são Artur Miguel Dias e Luís Carvalho.

A maior parte da nota (cerca de 80%) será determinada de forma muito objetiva, testando o programa à mão e confirmando quais as funcionalidades implementadas.

Na parte mais subjetiva da apreciação da qualidade dos trabalhos serão tidos em conta aspetos, tais como:

- clareza e simplicidade das ideias programadas,
- qualidade do sistema de classes para modelar o jogo e obter extensibilidade,
- qualidade da página Web criada.

Não é obrigatório concluir todas as funcionalidades. Uma implementação parcial poderá ser classificada com um resultado positivo. Mas dê prioridade ao funcionamento do jogo de acordo com as regras, nem que seja só no primeiro nível.

Situação especial 1 - O programa não está escrito num estilo orientado pelos objetos, ou seja parte substancial do comportamento dos atores está programada em funções globais exteriores às classes. O programa será avaliado numa escala de 0 a 10 valores.

Situação especial 2 - O programa está escrito num estilo orientado pelos objetos, mas viola as regras da aula teórica 22 que ensinam a fazer programas extensíveis. O programa será avaliado numa escala de 0 a 15 valores.

Observações

- Os grupos são incentivados a discutir entre si aspetos gerais da realização do projeto (inclusivamente no fórum). No entanto os grupos devem manter uma certa distância entre si, não podendo partilhar ideias muito detalhadas ou ler código mútuo. A escrita de código exige esforço intelectual, mas só com esforço se consegue evoluir.
 - O objetivo deste projeto é levar os alunos a praticar. Um aluno que pratique de forma genuína ganha experiência e provavelmente terá mais facilidade em conseguir aprovação no exame final.
 - Cuidado com as fraudes. Por exemplo, se alguém dum grupo oferecer o projeto resolvido a um elemento de outro grupo, trata-se duma fraude envolvendo dois grupos (já tem acontecido). Também se um grupo deixa distraidamente a área aberta e se alguém de outro grupo "rouba" o projeto, então também se considera fraude dos dois grupos (já tem acontecido). Ainda um terceiro caso: se dois grupos se juntam para fazer o projeto conjuntamente e depois o entregam em duplicado, então também se considera fraude (já tem acontecido). Além destes três exemplos, existem muitas outras situações de fraude. Em suma, **cada grupo é responsável pelo seu projeto, tem de produzir código original**, e não o pode mostrar ou oferecer, direta ou indiretamente, o seu código a outro grupo. Note que é muito melhor ter zero num dos quatro projetos do que ser logo excluído da cadeira por motivo de fraude.
 - Poderá haver discussões dos projetos via Skype, mas será normal os docentes entenderem dispensar a maioria dos alunos da discussão.
-

Final

Bom trabalho! Esperamos que goste.
