Politécnico Castelo Branco
Escola Superior de Tecnologia

ESCOLA SUPERIOR DE TECNOLOGIA

Enunciado do Mini-Projeto de Computação Gráfica

Tecnologias da Informação e Multimédia

Alien Invasion, versão de 2/2025 (27)

1. Enquadramento e Objetivos

O mini-projeto de Computação Gráfica convida os alunos a desenvolver, com base nos conhecimentos adquiridos ao longo do semestre, um jogo de Shot'Em Up do tipo "Space Invaders". O desenvolvimento poderá ser efetuado em Xcode (macOS) ou Visual Studio 2019 CE (Windows), com a utilização das APIs OpenGL e GLUT (macOS) ou FreeGLUT (Windows). Podem ser utilizadas APIs específicas para som (por exemplo a irrKlang) e para texturas (por exemplo a SOIL). Recomenda-se sempre tentar efetuar as funcionalidades básicas primeiro e só depois abraçar os desafios de texturas e som.

2. Descrição

O jogo deverá ser o mais intuitivo possível, guiando o utilizador na experiência de jogo. O jogo deve ter como base de cenário o ambiente espacial interplanetário, mas ao contrário do popular Space Invaders que os inimigos descem e a nave do jogador apenas se desloca da esquerda para a direita, no Alien Invasion a deslocação é mais liberal. As mecânicas de jogo são da responsabilidade dos alunos, que decidem o que acontece em caso de colisão de uma bala dos inimigos com o elemento controlado pelo jogador, assim como quando um inimigo colide diretamente.

As naves inimigas não aparecem todas no início do nível, à medida que este se vai desenvolvendo elas vão aparecendo. O sentido de deslocação é sempre o mesmo dentro de um dado nível. A nave do jogador deixa de ter limitação na sua deslocação, podendo inclusivamente deslocar-se para trás das naves inimigas e atingi-las dessa forma. Para isso deve ser possível rodar a nave do jogador em intervalos mínimos de noventa graus. Se num dado nível uma ou mais nave(s) inimiga(s) desaparece(m) do ecrã pode não implicar perda de vida por parte do jogador. O jogador passa ao nível seguinte quando todas as naves inimigas tiverem sido abatidas ou não tiver vidas ou energia disponível.

A Figura 1ilustra o conceito que se pretende do jogo a implementar. A figura é apenas uma orientação para definição da área de jogo, pois por exemplo a indicação do número de vidas e a pontuação são informações a adicionar que não estão ilustradas. Além disso, a criatividade de cada game designer pode introduzir alterações ao cenário apresentado.

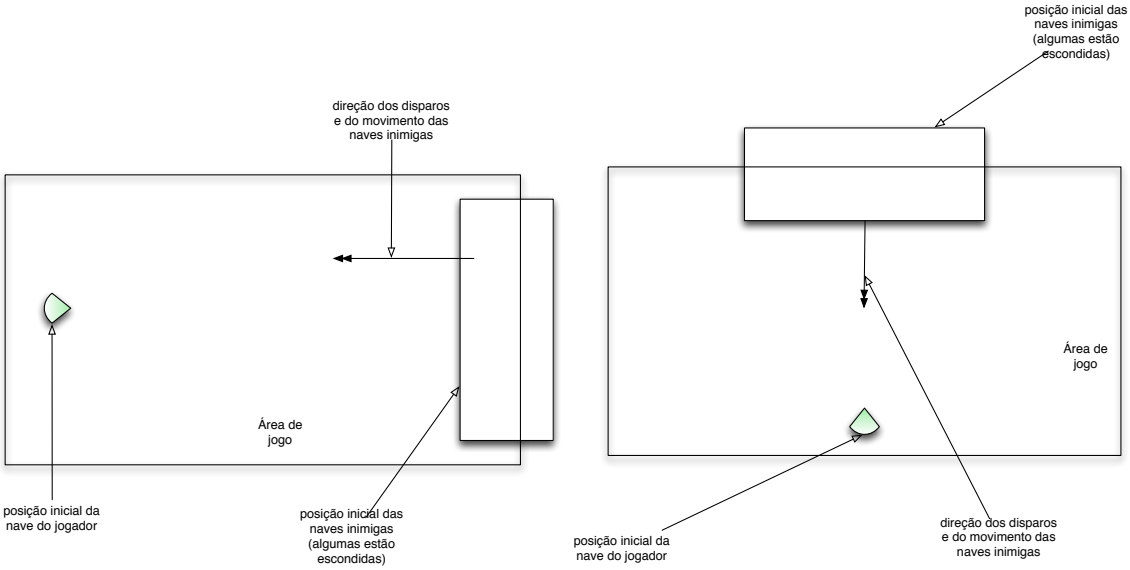

Figura 1 – Modelos simplificados do jogo a desenvolver.

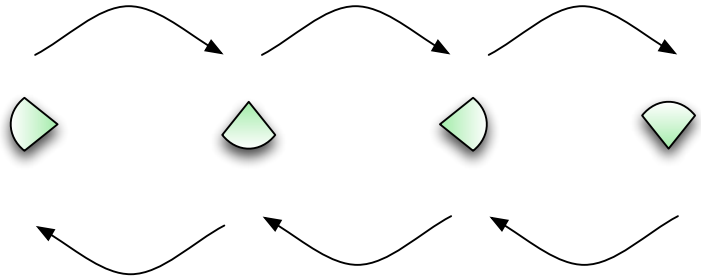

Figura 2 - Rotação da nave do jogador.

A criatividade na definição do GDD e implementação é encorajada, no entanto devem ser implementadas as seguintes funcionalidades básicas:

- Suporte a início de jogo, pausa, retomar e fim de jogo
- Suporte a ajuda
- Sistema de vidas do jogador, com continuidade/fim de jogo
- Nível único com suporte a ganha e perde
- Suporte a disparos por parte da nave do jogador
- Pontuação por inimigo destruído
- Gestão de projéteis (destruição adequada nas estruturas de dados)
- Movimentação autónoma das naves inimigas
- Suporte a colisões

Os estudantes poderão encarar os seguintes desafios:

- Disparos da parte dos inimigos
- Suporte a *powerups* diversos
- Mudança de aspeto do jogo (temas gráficos de cores)

- Gravação de estado de jogo e posterior resumo
- Diversos tipos de inimigos e pontuação, por exemplo boss
- Sistema de níveis, com diferentes estruturas de naves inimigas e diferentes sentidos de deslocação
- Configuração dos parâmetros de jogo (número de vidas, nível inicial de dificuldade, entre outros)
- Carregamento de configuração de níveis a partir de ficheiro de configuração
- Diversos tipos de tiros nave jogador
- Som
- Texturas

Chama-se a especial atenção para o facto de este mini-projeto ter um peso de 80% na avaliação, devendo, portanto, também ocupar 80% das horas fora de contato previstas de acordo com os ECTS da Unidade Curricular. De acordo com o desenvolvimento efetuado durante as aulas, o código entregue deve funcionar quer em sistemas Windows, quer em macOS com xcode. Chama-se especial atenção para este fato, pois o docente apenas tem acesso a sistema macOS.

3. Possível abordagem

O planeamento, modelação e GDD é fundamental para o sucesso. Não se começa a fazer código, a codificação da solução advém do GDD e não o contrário. Por isso existe um momento de avaliação intermédio que avalia isso mesmo. Em termos de programação domine bem a programação orientada a objetos e orientada ao evento que é abordada em duas aulas práticas iniciais. É fundamental não deixar dúvida alguma por esclarecer, se necessário utilize o horário de atendimento marcando com o docente uma sessão zoom.

As três aulas laboratoriais que se seguem da prática laboratorial vão abordar as transformações geométricas. As transformações geométricas são essenciais para definir os objetos em jogo, ainda sem se preocupar com a movimentação autónoma. A animação autónoma é abordada na ficha 3 e nessa altura poderá trabalhar os inimigos e as balas.

Nunca esquecer que as aulas práticas ajudam na forma de raciocinar, mas é a aula TP de "Introdução ao Desenvolvimento de Jogos" que vai permitir entender o que é um GDD! Discuta com o docente a sua abordagem e caso queria partilhe com ele numa sessão presencial ou via zoom o seu GDD atual para obter feedback.

O relatório intercalar deverá conter a abordagem formal ao jogo através de um GDD. Não vai conter código, a explicação da implementação é reservada para o relatório final.

4. Datas, grupos e organização da entrega

As datas mais importantes são as de limite de entrega do relatório intercalar e a entrega final do trabalho. Não dê menos importância ao relatório intercalar, ele pode ser a diferença entre passar e não passar. Planeie com antecedência para não ser apanhado sem tempo. A componente prática laboratorial apenas exige a realização deste trabalho, pelo que tem todas as condições para ter sucesso.

O processo de entrega compreende sempre o envio via Moodle (não serão aceites entregas por outra via, nomeadamente entregas por email serão completamente ignoradas) de um único ficheiro .ZIP com os conteúdos necessários. Por questão de organização o ficheiro zip deverá ter como nome os números dos executantes. Por exemplo os alunos 62011000 e 62011001 entregam o ficheiro zip 62011000_62011001.zip. Não utilize carateres especiais ou acentuados para o nome de qualquer ficheiro, mesmo dos que estiverem dentro do zip. Também se aconselha a não utilização de carateres especiais dentro do código, mesmo que seja nos comentários. Para datas de submissão consultar a plataforma moodle.

5. Critérios de avaliação – época de frequência

5.1. Relatório intercalar (10%)

Este relatório deverá conter a parte de modelação do jogo, bem como indicação das classes a implementar e suas funcionalidades básicas. Os alunos podem seguir um modelo do tipo GDD – Game Design Document. Avalia-se a capacidade de abordar o problema, a qualidade de escrita, a organização do relatório. Não é necessário criar qualquer apresentação nesta fase. A submissão é efetuada através de um único ficheiro PDF.

5.2. Entrega final (70%)

A entrega final tem três componentes que são avaliadas de forma distinta – relatório (15%), apresentação (10%) e execução (35%). Os critérios de avaliação para cada componente são os que se apresentam nas tabelas seguintes:

Relatório (15%)		
Escrita	Estrutura	Conteúdo
3	2	10

- Escrita (2%)

  A qualidade de escrita é importante e deve ser avaliada. O relatório não deve ter erros ortográficos, caso os alunos optem por não respeitar o acordo ortográfico

devem referi-lo em local apropriado. A construção das frases e parágrafos deve ser cuidada e fornecer informação útil. A linguagem deve ser cuidada, precisa e técnica. A formatação e tipo de letra devem ser adequados, sem elementos gráficos desnecessários.

- Estrutura (1%)

  A estrutura de um relatório vê-se no índice, mas não só. Os capítulos devem estar bem encadeados, no sentido em que a informação que é utilizada no seguinte deve estar explicada no anterior e nunca vice-versa. A formatação do documento deve ser adequada, com paginação. A capa deve ser sóbria e fornecer todos os elementos necessários para identificar o trabalho e os seus executantes (nomes e números) de forma inequívoca.

- Conteúdo (10%)

  Sem dúvida a parte mais importante do relatório. A descrição da execução deve ser concisa e fornecer, sem necessidade de consultar código, visão sobre o conteúdo do trabalho. As figuras devem ser legíveis, estar bem identificadas (legendas) e explicadas. O sistema de referências bibliográficas deve ser referenciado no texto (por exemplo numericamente). O conteúdo deve dar uma visão da engenharia da solução, nomeadamente descrição exaustiva dos algoritmos, classes e objetos criados. Deve também ser fornecida informação sobre a forma de jogar o jogo, ou seja, o jogo visto sob o ponto de vista do jogador e não sob o ponto de vista técnico.

Em relação à apresentação, esta é obrigatória. Ou seja, trabalho que não é apresentado oralmente não é avaliado.

Apresentação (10%)		
Acetatos/Discurso	Demonstração	Questões
2	5	3

- Acetatos/Discurso (2%)

  Os acetatos ou sistema equivalente devem ser fáceis de seguir, de referir em caso de perguntas e sobretudo fáceis de ler, por isso atenção ao tema e esquema de cores que utilizar. Os acetatos ou equivalente não devem estar cheios de texto para os alunos lerem, mas devem apenas ter tópicos/figuras que os alunos explicam/expandem na apresentação. Claro está que as figuras devem ser legíveis. No caso de grupos de 2 elementos também é avaliada a percentagem de tempo que cada um intervém, bem como a sincronização entre os elementos do grupo. O discurso deve ser fácil de seguir e coerente, sem grandes pausas.

- Demonstração (5%)

  O jogo deve sempre ser apresentado ao vivo, por isso antes de se deslocar para a apresentação tenha em execução todo o software necessário. Em grupos de 2 elementos é necessário que ambos descrevam uma parte do jogo. Devem ser abordados todos os aspetos do jogo, com ênfase nos mais importantes, sem repetições.

- Questões (3%)

  Os alunos deverão responder às questões formuladas de forma sucinta e concisa. As questões podem ser formuladas dentro da apresentação em si, nomeadamente aquando da demonstração.

Execução (45%)			
Código	Componente Gráfica	Jogabilidade	Funcionalidades
5%	10%	10%	20%

- Código (5%)

  O código deve ter uma estrutura adequada e ser facilmente legível (nomenclatura, indentação). Os comentários devem cingir-se ao sistema Doxygen ao nível das classes, métodos/funções e atributos. A estrutura de dados deve ser adequada ao problema e o mais eficiente possível (minimizar uso de memória, minimizar o número de ciclos).

- Componente gráfica (10%)

  A parte gráfica deve responder de forma eficaz e apelativa aos objetivos do jogo. A forma como as APIs OpenGL e freeGLUT são utilizadas também é aqui avaliado. Os objetos gráficos devem ser flexíveis e as transformações aplicadas de forma adequada e eficiente. Aqui também se avalia a gestão de janelas e viewports.

- Jogabilidade (10%)

  O jogo como aplicação interativa deve ser fácil de compreender e jogar. Os controlos devem facilitar o manuseamento do jogo. O objetivo do jogo deve ser bem explícito e fácil de compreender. A parte gráfica também tem um papel importante na jogabilidade.

- Funcionalidades (20%)

  Quais as funcionalidades implementadas? Movimentação dos elementos, controlo de projéteis, início/pausa/resumo do jogo. Funcionalidades avançadas. Diversos níveis de dificuldade e orientação. Suporte a vidas, entre outros.

6. Critérios de avaliação – outras épocas

Tenha em atenção que nas restantes épocas perde o resultado do relatório intercalar. A entrega fora da época de frequência tem três componentes que são avaliadas de forma distinta – relatório (20%), apresentação (10%) e execução (50%). Os critérios de avaliação para cada componente são os que se apresentam nas tabelas seguintes:

Relatório (20%)		
Escrita	Estrutura	Conteúdo
4%	1%	15%

Aplica-se a mesma obrigação em relação à apresentação nas restantes épocas.

Apresentação (10%)		
Acetatos/Discurso	Demonstração	Questões
2	5	3

Execução (50%)			
Código	Componente Gráfica	Jogabilidade	Funcionalidades
5%	12.5%	12.5%	20

7. Bibliografia

- Richard S. Wright, Nicholas Haemel, Graham Sellers, Benjamin Lipchak, "OpenGL SuperBible: Compreensive Tutorial and Reference", Addison-Wesley Professional, 5th edition, ISBN-13: 978-0321712615, 2010
- OpenGL – The Industry's Standard for High Performance Graphics, http://aluno.opengl.org, acedido em setembro de 2014
- OpenGL wikibook – http://en.wikibooks.org/wiki/OpenGL_Programming, acedido em setembro de 2014
- The FreeGLUT project, http://freeglut.sourceforge.net, acedido em setembro de 2014
- John F. Hughes, Andries van Dam, Morgan McGuire, David Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley, "Computer Graphics: Principles and Practice", Addison Wesley Professional, 3rd edition, ISBN-13: 978-0321399526, 2013
- James D. Foley, Andries van Dam, Steve K. Feiner, John F. Hughes, "Computer Graphics: Principles and Practice in C", Addison-Wesley Professional, 2nd edition, 1995

- Dave Shreiner, The Kronos OpenGL ARB Working Group, “OpenGL Programming Guide: The Official Guide to Learning OpenGL, versions 3.0 and 3.1)”, Addison-Wesley Professional, 7th edition, ISBN-13: 978-0321552624, 2009
- Shalini Govil-Pai, “Principles of Computer Graphics: Theory and Practice Using OpenGL and Maya®”, Springer, ISBN-13: 978-0387955049, 2005
- David Bud Leiser, http://aluno.budleiser.com, acedido em setembro de 2014
- Alguns GDD de jogos conhecidos – http://aluno.sloperama.com/advice/specs.html, acedido em setembro de 2014
- Exemplo de GDD por David Bud Leiser, https://aluno.youtube.com/watch?v=MDPRyc53WiI, acedido em setembro de 2014