U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Relatório da Unidade Curricular

Introdução

à

Programação Criativa

António Fernando V. C. C. Coelho

Professor Auxiliar do DEI/FEUP

Junho de 2018

"Educar é semear com sabedoria e colher com paciência."

Augusto Cury

Relatório sobre uma unidade curricular no âmbito das provas de agregação no ramo de conhecimento em Engenharia Informática, e de acordo com o Decreto-Lei no 239/2007, de 19 de junho, que aprova o regime jurídico do titulo académico de agregado.

Índice

1 INTRODUÇÃO 1

2 MOTIVAÇÃO 2

3 CONCEITOS BASE E TRABALHO RELACIONADO 3

3.1 PENSAMENTO COMPUTACIONAL 3

3.2 LUDIFICAÇÃO 4

3.3 TRABALHO RELACIONADO 4

4 CONTEXTO 6

4.1 A DIMENSÃO DA TECNOLOGIA NA ÁREA CIENTÍFICA DOS MEDIA DIGITAIS 6

4.2 PANORAMA DA INTRODUÇÃO À PROGRAMAÇÃO NA ÁREA DOS MEDIA DIGITAIS 6

4.2.1 O Mestrado em Multimédia da Universidade do Porto 6

4.2.2 Panorama Nacional 7

4.2.3 Panorama Internacional 8

4.3 LINGUAGENS DE PROGRAMAÇÃO ADOTADAS 9

5 OBJETIVOS 10

5.1 COMPETÊNCIAS 10

5.2 OBJETIVOS DE APRENDIZAGEM 10

6 CONTEÚDOS PROGRAMÁTICOS 13

6.1 PROGRAMA 13

6.2 DEMONSTRAÇÃO DA COERÊNCIA DOS CONTEÚDOS PROGRAMÁTICOS COM OS OBJETIVOS DA UNIDADE CURRICULAR 14

6.3 SELEÇÃO DA LINGUAGEM DE PROGRAMAÇÃO 14

6.4 BIBLIOGRAFIA 15

6.4.1 Bibliografia principal 15

6.4.2 Bibliografia complementar 16

7 METODOLOGIA DE ENSINO 17

7.1 ABORDAGEM METODOLÓGICA 17

7.2 AVALIAÇÃO 17

7.3 PROJETO DE LUDIFICAÇÃO DA APRENDIZAGEM 18

7.3.1 Definição dos objetivos 18

7.3.2 Delinear os comportamentos alvo 18

7.3.3 Descrever os Jogadores 19

7.3.4 Elaborar os ciclos de atividade 19

7.3.5 Não esquecer a diversão (motivação e envolvimento); 21

7.3.6 Implementar as ferramentas apropriadas 21

7.4 DEMONSTRAÇÃO DA COERÊNCIA DAS METODOLOGIAS DE ENSINO COM OS OBJETIVOS DE APRENDIZAGEM DA UNIDADE CURRICULAR 24

8 CONCLUSÕES 25

REFERÊNCIAS 26

ANEXO A – TABELA DE ATIVIDADES DE DESENVOLVIMENTO DE JOGOS 27

ANEXO B – FICHA A3ES 28

ANEXO C – CURSOS NA ÁREA DOS MEDIA DIGITAIS 35

Glossário

A3ES	Agência de Avaliação e Acreditação do Ensino Superior
ACM	Association for Computing Machinery
API	Interface de programação de aplicações (*Application Programming Interface*)
B-Learning	Blended-learning
ECTS	Sistema Europeu de Transferência e Acumulação de Créditos (*European Credit Transfer and Accumulation System*)
FEUP	Faculdade de Engenharia da Universidade do Porto
HTML	HyperText Markup Language
IDE	Ambiente de Desenvolvimento Integrado (*Integrated development environment*)
IGDA	International Game Developers Association
LMS	Sistema de gestão da aprendizagem (*Learning Management System*)
MM	Mestrado em Multimédia (da Universidade do Porto)
PBL	Aprendizagem baseada em problemas (*Problem-based Learning*)
PLE	Ambiente de ensino da programação (*Programming Learning Environment*)
SW	Software
UC	Unidade curricular
UP	Universidade do Porto
UX	Experiência do utilizador (*User Experience*)

1 Introdução

O ensino na área científica dos Media Digitais[1] tem vindo a evoluir para responder aos desafios que se deparam à nossa sociedade, particularmente o de criar tecnologias que efetivamente proporcionem valor acrescentado ao ser humano, tanto na vertente profissional como pessoal. O perfil do profissional da área dos media digitais é assim híbrido e diversificado, conjugando competências nas áreas do Design, da Arte, da Tecnologia e da Comunicação.

A dimensão da Tecnologia na formação em Media Digitais tem como objetivo principal desenvolver competências para o projeto e desenvolvimento de ferramentas multimédia. Alia-se assim a necessidade de desenvolver no estudante o pensamento computacional e de o dotar com a capacidade efetiva de desenvolver software em projetos de média complexidade.

Neste documento apresenta-se a descrição de uma unidade curricular (UC) de introdução à programação no âmbito do Mestrado em Multimédia (MM) da Universidade do Porto (UP), a ser lecionada a partir do ano letivo 2018/19. Para além do foco das necessidades específicas desta unidade curricular, este documento faz uma reflexão sobre como se pode abordar a aprendizagem da programação nos cursos da área dos Media Digitais / Multimédia, com base na experiência pessoal do docente em mais de 20 anos de carreira e na mais recente investigação científica na área da ludificação (*gamification*) e da aprendizagem baseada em jogos.

Seguindo a teoria de aprendizagem do Construcionismo, apresenta-se o conceito de "Introdução à Programação Criativa", que alia um foco no desenvolvimento do pensamento computacional, através de um conjunto de dinâmicas pedagógicas ativas. Através de uma aprendizagem baseada em problemas (PBL – *Problem-based learning*) que apelam à criatividade (jogos, por exemplo) desenvolvem-se as competências básicas para o desenvolvimento de software de baixa/média complexidade. A unidade curricular é desenhada numa perspetiva de ludificação (*gamification*), criando ciclos iterativos curtos que promovam atividade regular de aprendizagem, conjugando uma componente supervisionada pelo docente e outra a desenvolver pelo estudante em autonomia.

Seguindo a metáfora de Alberto Cury, o docente toma o papel de "agricultor" que semeia desafios nos terrenos férteis das mentes dos estudantes criativos, e que vai regando e nutrindo o seu crescimento ao longo do semestre, fomentando as competências já adquiridas previamente, com o objetivo de fortalecer o crescimento destas novas competências.

A estrutura deste documento espelha esse objetivo. Na secção 2 aborda-se a motivação para se introduzirem novas práticas pedagógicas no ensino da programação, particularmente na área dos Media Digitais. Na secção 3 apresentam-se os conceitos base do Pensamento Computacional e da Ludificação, seguindo-se o trabalho de investigação científica desenvolvido que suporta esta abordagem. Na secção 4 descreve-se o contexto do ensino da Programação na área dos Media Digitais e na secção 5 definem-se os objetivos da unidade curricular, tanto ao nível das competências como dos objetivos de aprendizagem. Na secção 6 são definidos os conteúdos programáticos, a seleção da linguagem de programação a adotar e a bibliografia. A secção 7 foca-se na metodologia de ensino, incluindo a avaliação, descreve-se com detalhe o projeto de ludificação da aprendizagem. Finalmente na secção 8 apresentam-se as conclusões.

[1] Considera-se neste documento que área dos Media Digitais integra, em sentido lato, todos os media não tradicionais que utilizam suporte digital. Embora muitas vezes utilizado de como sinónimo, assume-se que o Multimédia é uma subárea dos Media Digitais que se focam na integração de mais do que um tipo de suporte digital para a informação.

2 Motivação

A unidade curricular de Introdução à Programação Criativa é proposta no sentido de dar os fundamentos do desenvolvimento de software, mas integrando também uma introdução aos sistemas interativos, particularmente, o design e desenvolvimento de jogos digitais. Pretende-se assim alavancar a aprendizagem da programação, um processo tipicamente analítico, com o design e desenvolvimento de jogos digitais, dando largas à capacidade criativa dos estudantes.

O ensino introdutório da programação é um processo que requer a aquisição de competências do foro cognitivo, que se enquadram no que se define como Pensamento Computacional (Wing, 2006). O processo de aprendizagem da programação possui uma complexidade elevada (Jenkins, 2002), que causa dificuldades específicas que têm vindo a ser constatadas ao longo de vários anos (Piteira, 2013). Esta conjugação visa assim superar este problema bem identificado da falta de motivação, e mesmo até ansiedade, para a aprendizagem da programação por um número significativo de estudantes destes ciclos de estudo.

Adicionalmente, não se apresenta apenas uma instância de uma unidade curricular de Introdução à Programação, mas sim um modelo e um conjunto de ferramentas para a criação de uma unidade curricular de programação no âmbito de um ciclo de estudos específico da área dos Media Digitais. Decidiu-se denominar esta unidade curricular de "Introdução à Programação Criativa" de forma a melhor orientar o foco para estudantes com perfis em áreas relacionadas com o desenvolvimento criativo.

Tendo em vista o público alvo desta unidade curricular, a metodologia de ensino reveste-se de um caráter inovador, fruto do trabalho de investigação desenvolvido pelo autor e conjugando três conceitos:

- B-Learning - *Blended-learning*
- PBL - *Problem-based Learning*
- Ludificação (*gamification*)

Seguindo o espírito de Bolonha, a utilização de *b-learning* é fundamental para sustentar a autonomia do estudante neste processo, com materiais de autoaprendizagem. O processo de aprendizagem é centrado no estudante dos Media Digitais, de caráter heterogéneo e motivado para o desenvolvimento de projetos criativos. A opção pela integração do design e desenvolvimento de Jogos Digitais possibilita uma abordagem pedagógica de PBL, potenciando as dinâmicas individuais distintas e as dinâmicas de grupo, que se pretendem promover desde cedo.

Finalmente, todo este processo de aprendizagem é desenhado com base numa estratégia de ludificação de forma a assegurar um processo contínuo de trabalho em autonomia, com elevados níveis de motivação e envolvimento, e sempre sob supervisão do docente. Para tal, é essencial o suporte das atividades através de uma plataforma de gestão da aprendizagem (LMS – *Learning Management System*).

3 Conceitos Base e Trabalho Relacionado

A metodologia de ensino baseia-se nos conceitos do Pensamento Computacional, como competência fundamental para o sucesso na unidade curricular, e na ludificação para a motivação e emancipação do processo de aprendizagem.

A preparação desta unidade curricular foi suportada por trabalho de investigação do autor, na supervisão de teses de doutoramento, dissertações de mestrado e projetos de investigação.

3.1 Pensamento Computacional

O Pensamento Computacional foi definido por Jeannette Wing em 2006 (Wing, 2006), embora o termo já tivesse sido publicado anteriormente. Neste artigo seminal considera-se que a forma como se resolve um problema computacional pode ser utilizada para resolver muitos dos problemas complexos do quotidiano. Deixa de ser uma competência específica da informática, para se tornar uma competência essencial na formação de um indivíduo.

A chave do pensamento computacional é a "abstração", ou seja, a forma como se apreende o essencial do problema e se vai decompondo em subproblemas até se encontrarem padrões de soluções que possam ser facilmente automatizáveis através de um algoritmo.

Embora várias abordagens tenham considerado diferentes processos cognitivos, salientam-se os quatro principais:

- Abstração – Focar na informação mais relevante ignorando os detalhes de implementação.

Considera-se a abstração procedimental, ou seja, a forma como os procedimentos (ou funções) encapsulam os detalhes de implementação do sistema computacional, quer através de primitivas da própria linguagem, bibliotecas de funções ou mesmo frameworks para áreas de aplicação específicas. É também feita uma abordagem à definição de funções por parte do programador e, finalmente, uma breve abordagem à programação orientada por objetos.

Outro conceito relevante é o de abstração de dados, abordando-se a forma como se estruturam os dados para a resolução dos problemas. Para além de uma abordagem às estruturas básicas disponibilizadas pelas linguagens de programação (vetores, listas e cadeias de caracteres) são também abordadas outras estruturas comuns como filas, pilhas e tabelas.

- Decomposição – Abordar um problema através da sua decomposição em subproblemas mais simples e de fácil resolução.

A capacidade de abordar um problema e ser capaz de o decompor em subproblemas mais simples é também essencial. É seguida uma abordagem estruturada, através da definição de diferentes módulos (funções ou procedimentos) e de forma hierárquica (*top-down*) para o estudante criar a sua solução.

- Desenho de algoritmos – desenvolver uma solução detalhada para um problema, passo a passo.

O desenho de algoritmos é a competência mais primária do programador, que se pretende desenvolver. Ser capaz de olhar para um (sub)problema simples e desenvolver uma sequência detalhada de instruções que definam sem ambiguidade as operações a realizar para o solucionar. São introduzidas as estruturas de decisão, nas diversas formas (condicional, alternativa e múltipla) bem como a recursividade e a iteração, na automatização dos processos. Neste último caso abordando os ciclos (condicional e contado).

- Reconhecimento de padrões – Identificar semelhanças entre problemas ajuda a encontrar soluções padronizadas.

Na decomposição de um problema em subproblemas é possível identificar características tipificadas, que podem ser facilmente resolvidas com algoritmos padronizados. É o caso, por exemplo, da pesquisa e ordenação, mas também da decomposição e mapeamento de estruturas de dados.

3.2 Ludificação

A ludificação (*gamification*), também frequentemente traduzida como gamificação, define-se como a utilização de técnicas e elementos de design de jogos em contextos que não são de jogo (Deterding et al., 2011). Trata-se de um processo de *UX design* (design da experiência).

O processo de ludificação pressupõe uma abordagem baseada na "jornada do jogador", iniciando-se numa primeira fase de "*Onboarding*" onde o utilizador identifica os objetivos que deve atingir, é confrontado com as "regras de jogo" e motivado a avançar. Numa segunda fase, de "*Scaffolding*", constrói-se um percurso de aprendizagem que promove a aquisição de competências de forma progressiva, até finalmente se atingir uma fase final de maior desafio onde o utilizador demonstra ter adquirido as competências e possuir domínio do conhecimento fundamental, o "*Endgame*".

O framework *Werbach's 6 steps to Gamification* (Werbach, 2012), é composto por um processo de design com seis passos:

1 Definir os objetivos de negócio;
2 Delinear os comportamentos alvo;
3 Descrever os jogadores;
4 Elaborar os ciclos de atividade;
5 Não esquecer a diversão (motivação e envolvimento);
6 Implementar as ferramentas apropriadas.

3.3 Trabalho Relacionado

Este trabalho pedagógico é alavancado num conjunto de projetos de investigação, desenvolvidos pelo autor, nos quais se inclui a participação no projeto europeus H2020 BEACONING (Bourazeri, 2017), a supervisão de uma tese de doutoramento (Baptista, 2017) e de três dissertações de mestrado (Monteiro, 2017; Faria, 2017; Reis, 2017).

O projeto BEACONING[2] centra-se no desenvolvimento de jogos pervasivos (que têm em consideração o contexto envolvente do jogador) para o ensino formal, não formal ou informal numa metáfora de "aprendizagem em qualquer lugar e em qualquer altura". Neste âmbito, foi desenvolvida uma dissertação (Faria, 2017) adaptando o framework do projeto ao ensino da programação. O estudante joga um jogo baseado na localização, segundo uma narrativa de "caça ao tesouro" onde, através do seu *smartphone*, vai sendo desafiado para resolver problemas de programação.

A utilização de jogos na aprendizagem e em particular a certificação da aquisição de competências, foi alvo de uma tese de doutoramento (Baptista, 2017) que derivou no desenvolvimento do método de certificação triádico, que acabou por servir de base também ao modelo de avaliação (e de *learning analytics*) do projeto BEACONING. Alavancou uma outra dissertação de mestrado (Reis, 2017) onde se explora a aprendizagem da programação, com base em PBL, utilizando o desenvolvimento de jogos digitais como motor do processo de

[2] http://www.beaconing.eu/

aprendizagem, mas assegurando que o conjunto de tutoriais a desenvolver possibilita a aquisição das competências pretendidas de acordo com o modelo de certificação triádica.

Na dissertação (Monteiro, 2017), abordou-se a ludificação do ensino da programação. Foram identificados os elementos de jogo e atividades passíveis de ludificação, bem como as possibilidades de integração em LMS (neste caso o Moodle). Foram também desenvolvidos dois estudos, um com estudantes e outro com docentes, em que se demonstrou a motivação de ambos os grupos para a ludificação da aprendizagem e a valorização positiva que davam à utilização de pontuação como mecanismo de avaliação contínua.

4 Contexto

A unidade curricular de "Introdução à Programação Criativa" enquadra-se no ensino dos Media Digitais, embora possa ser alargada a outras áreas de formação. Nesse sentido apresenta-se um estudo sobre o contexto do perfil do profissional desta área, em particular do Multimédia, e dos cursos oferecidos atualmente na Universidade do Porto e em outras universidades, tanto em Portugal como a nível internacional.

4.1 A dimensão da Tecnologia na área científica dos Media Digitais

A dimensão da Tecnologia na formação em Media Digitais tem como objetivo principal desenvolver competências para o projeto e desenvolvimento de ferramentas multimédia. Logo, é fundamental uma forte componente prática do curso, para a aquisição destas competências, aliada a uma componente teórica de suporte que envolve áreas científicas como a Computação Gráfica, a Computação Sonora e Musical, a Interação Humano Computador e os Jogos Digitais. Mas é também essencial formação de base nas áreas das Ciências da Computação e da Engenharia Informática.

A Tecnologia dos Media Digitais é analisada num contexto abrangente em termos de áreas de aplicação (como a Educação, a Arte, a Cultura, a Comunicação, a Saúde, o Entretenimento, etc), focando-se na capacidade de criar sistemas interativos que auxiliem o ser humano a desenvolver processos complexos, maximizando não só a eficácia da tecnologia na resolução desses problemas, mas também a interação do ser humano com essa tecnologia.

A formação ao nível do Ensino Superior (universitária ou politécnica) desenrola-se em três ciclos, com objetivos distintos. Se numa formação de 1º ciclo a Tecnologia é lecionada, nas diferentes componentes, como suporte nuclear das competências a adquirir, num 2º ciclo estas competências são muitas vezes complementares a uma formação prévia em área afim, à qual se pretende adicionar competências de desenvolvimento de ferramentas multimédia. Já num 3º ciclo, é fomentada a inovação tecnológica pelo desenvolvimento de novos métodos, processos ou ferramentas.

4.2 Panorama da Introdução à Programação na área dos Media Digitais

Antes de definir a unidade curricular importa fazer um estudo das melhores práticas nas universidades que lecionam cursos de Introdução à Programação na área dos Media Digitais. Inicia-se pela apresentação do curso onde a unidade curricular vai ser lecionada.

4.2.1 O Mestrado em Multimédia da Universidade do Porto

O Mestrado em Multimédia[3] é um 2º ciclo de estudos oferecido conjuntamente por cinco unidades orgânicas da Universidade do Porto (FEUP, FBAUP, FCUP, FEP e FLUP). Esta orgânica reflete o caráter heterogéneo do profissional que se pretende formar, com estudantes provenientes de diversas áreas como o design, o ensino, as ciências naturais, as ciências da comunicação ou as artes. Não existe uma formação de 1º ciclo em Multimédia na UP, pelo que o curso se adapta a esta diversidade oferecendo quatro especializações:

- Cultura e Artes
- Educação
- Música Interativa e Design de Som
- Tecnologias Interativas e Jogos Digitais

[3] http://fe.up.pt/mm

O plano de estudos do Mestrado em Multimédia é organizado em 4 semestres letivos, correspondendo a um total de 120 ECTS. O primeiro ano letivo é composto por 10 unidades curriculares semestrais de 6 ECTS, divididas por ambos dos semestres. O segundo ano é composto por uma dissertação anual de 48 ECTS e duas unidades curriculares optativas de 6 ECTS no 3º semestre.

A unidade curricular de Introdução à Programação é oferecida como optativa no 1º e 3º semestres. O objetivo desta UC é fornecer competências de programação aos estudantes provenientes de outros ciclos de estudos que não disponibilizem unidades curriculares de programação no plano de estudos. O MM fornece ainda outras UC optativas na área das tecnologias que se podem articular com esta UC: Sistemas Digitais Interativos, Tecnologias Web e Jogos Digitais.

4.2.2 Panorama Nacional

Fazendo uma análise a nível nacional, através dos cursos registados no portal da A3ES[4] na área dos Media Digitais, verifica-se que pesquisando as palavras "media" e "digital" a oferta de unidades curriculares de introdução à programação se centra nos cursos de Multimédia. Os restantes são mais focados nas Ciências da Comunicação, no Audiovisual ou em outras áreas menos tecnológicas e não possuem unidades curriculares de programação. Alargou-se a pesquisa aos cursos de Computação Gráfica e de Jogos Digitais.

Analisando a oferta ao nível do 1º ciclo (Anexo C.1), verifica-se que todos os cursos têm pelo menos uma unidade curricular de programação, sendo que os cursos com maior pendente tecnológica, como os de Jogos Digitais e de TIC possuem até quatro unidades curriculares. Em termos de esforço (ECTS) existe também uma grande diversidade, desde os 3 aos 12 ECTS, sendo o valor mais comum os 6 ECTS (Figura 1). A maior parte destas UC surgem desde o primeiro ano, com apenas duas exceções, os cursos de Arte Multimédia (UL) e de Design e Animação Multimedia (IPP).

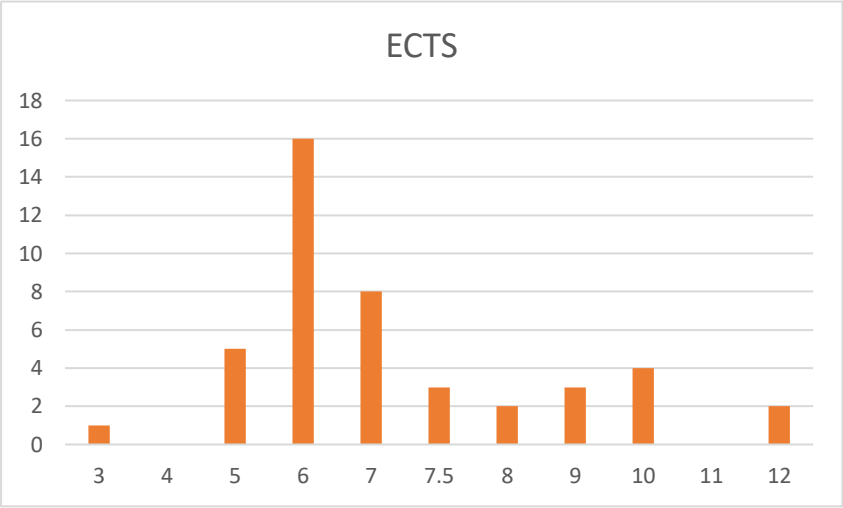

Figura 1 - Número de UC com determinado esforço (ECTS)

Em relação ao 2º ciclo (Anexo C.2), verifica-se uma situação bastante distinta, dado que 60% dos cursos não têm qualquer unidade curricular de programação, e os restantes 40% têm apenas uma unidade curricular de programação, metade dos quais apenas como opção[5]. Em termos de esforço (ECTS) existe uma grande uniformidade nos 6 ECTS, com apenas uma exceção com

[4] http://www.a3es.pt

[5] Nesta análise não foi considerado o curso " Tecnologia e Arte Digital " pois encontra-se em reestruturação.

7,5 ECTS. O facto de muitos cursos não possuírem UC de programação prende-se, na maior parte dos casos, por serem uma continuidade de 1º ciclos nessa mesma escola ou cursos que têm como pré-requisitos competências de programação, como é o caso dos Mestrados em Computação Gráfica ou em Jogos Digitais. A oferta de unidades curriculares de opção é uma forma de potenciar uma formação mais completa quando os candidatos têm uma formação base muito diversificada, como é o caso do Mestrado em Multimédia da UP.

Em termos da primeira linguagem de programação há uma maior abrangência da utilização de Processing, seguindo-se Javascript e depois Python, C e Java[6].

4.2.3 Panorama Internacional

Numa perspetiva internacional, a oferta de unidades curriculares de programação na área dos Media Digitais tem evoluído ao longo dos últimos anos, em diversos níveis. Tanto a nível metodológico, como a nível das ferramentas e linguagens de programação. Existem assim abordagens muito diversificadas, que se tentou analisar através de um conjunto de cursos internacionais[7], listados nos anexos C.3 e C.4.

Ao nível do mestrado (anexo C.4) é pouco comum encontrar unidades curriculares de introdução à programação, em particular nos EUA, dado estes tendem a ser muito focados na especialização em cursos de nível avançado. No caso do Mestrado em Multimédia da UP, esta unidade curricular surge como optativa dado não haver uma licenciatura da área na UP que forneça competências de programação, e associa-se a uma outra de Tecnologias Web. Um mestrado semelhante é o MSc in Interactive Digital Media do Trinity College, em Dublin. A oferta é de duas unidades curriculares (5 ECTS cada), a primeira de introdução à programação (em Python) e a segunda de programação Web (em Javascript).

Sendo o foco analisar as UC de introdução à programação na área dos Media Digitais, fez-se uma análise mais aprofundada ao nível dos 1º ciclos (anexo C.3), onde a oferta é mais diversificada. Nos cursos analisados na europa, verifica-se uma abordagem centrada nos curricula tradicionais da área de Ciências da Computação / Informática, que seguem os curricula ACM, com esforço associado entre os 5 ECTS e os 7,5 ECTS. Em cursos de desenvolvimento de Jogos Digitais (caso da Universitat de Girona) há por vezes um maior foco na programação, utilizando neste caso dois semestres com um total de 15 ECTS (9 + 6) e utilizando uma linguagem de mais baixo nível como o C++. Nos EUA encontram-se dois tipos de abordagens à Introdução da Programação: uma mais tradicional e uma outra muito focada na utilização da programação como uma ferramenta de mediação digital para a arte. Exemplos deste último caso são os cursos de *Foundations of Creative Coding* do Game and Mobile Media Applications (GAMMA) program da University of Texas at Austin e *Introduction to Computing for Creative Practice* do curso *IDeATe - Integrative Design, Arts, and Technology* de Carnegie Mellon University. A metodologia e os tópicos são mais focados nos aspetos da criação de aplicações multimédia para imagem e som, e utiliza-se sobretudo Processing ou Javascript com a biblioteca p5.js (derivada de Processing). Nos restantes cursos faz-se uma abordagem centrada no Pensamento Computacional e no desenvolvimento de programas com linguagens dinâmicas e de alto nível como Python.

---

[6] A informação disponível não permitiu uma cobertura global de todos os cursos, pelo que esta afirmação é fundamentada apenas parcialmente, sendo posteriormente aprofundada na secção 4.3.

[7] Foi feita uma amostra de conveniência no caso europeu e selecionaram-se as principais universidades ao abrigo de protocolos de cooperação na área dos Media Digitais.

4.3 Linguagens de Programação adotadas

Ao nível da seleção da linguagem de programação, há várias abordagens que passam pela utilização de paradigmas imperativo (ou procedimental), orientação por objetos ou funcional. O paradigma imperativo potencia uma abordagem simples, baseado na abstração procedimental, potenciando a decomposição estruturada do problema em blocos que definem o fluxo da informação. Já as linguagens orientadas por objetos possuem um elevado nível de abstração e encapsulamento, mas requerem abordagens mais complexas ao nível da modelação conceptual do problema. As linguagens funcionais possibilitam um nível de abstração procedimental elevado, conjugado com uma simplicidade que pode ser útil para trabalhar o pensamento algorítmico. Em cursos de Informática ou de Ciências da Computação é possível encadear um conjunto de unidades curriculares que aborde os distintos paradigmas, dando uma formação mais sólida. Num estudo em universidades europeias, verificou-se que a maior parte das UC introdutórias são lecionadas em C (155), seguindo-se C++ (51), Java (28), Pascal (26) e Python (19) (Aleksić & Ivanović, 2016). Consultando o "Tiobe Programming Community Index"[8], um índice de popularidade das linguagens de programação na indústria do software, verifica-se esta tendência, com a predominância de Java, C e C++, seguidos de Python em 5º lugar e Javascript em 8º, estas duas últimas linguagens com forte crescimento. Este crescimento é também observado ao nível do ensino, em particular da utilização de Python (Murphy & Davenport, 2017; Guo, 2014).

[8] https://www.tiobe.com/tiobe-index/

5 Objetivos

A unidade curricular "Introdução à Programação Criativa" tem como objetivos gerais dotar os cursos da área dos Media Digitais com um dos pilares da dimensão tecnológica – o pensamento computacional – e as competências para o desenvolvimento de aplicações multimédia interativas. Estando orientada para profissionais com uma vertente criativa, pretende-se que a aprendizagem da programação surja articulada com as bases do design e programação de jogos digitais, permitindo aos estudantes, por um lado, terem uma perceção da complexidade inerente a um sistema interativo e o suporte tecnológico de um motor de jogo e, por outro, a utilização de bibliotecas e *frameworks* no desenvolvimento de SW. Esta unidade curricular pode assim articular-se com outras unidades curriculares em tópicos como os Sistemas Interativos e o Desenvolvimento de Jogos Digitais, a Programação Web, a Computação Gráfica, a Computação Física, a Computação Sonora e Musical e a Interação Humano computador, mesmo a um nível de aprendizagem mais prático ou laboratorial.

Esta secção apresenta as competências que se pretende desenvolver nos estudantes e os objetivos de aprendizagem que se pretendem atingir.

5.1 Competências

Em termos gerais, pretende-se que o estudante adquira as competências englobadas no pensamento computacional, acrescido da capacidade efetiva de desenvolver programas de baixa / média complexidade[9] utilizando uma linguagem de programação.

No final desta unidade curricular o estudante deverá adquirir as seguintes competências:

- Conhecimentos:
  - Desenho de algoritmos;
  - Criação de programas numa linguagem de programação;
  - Desenvolvimento de aplicações interativas;
  - Fundamentos do design de jogos digitais.
- Atitudes e comportamentos:
  - Estruturação dos problemas informáticos de acordo com os pilares do pensamento computacional;
  - Utilização de frameworks multimédia no desenvolvimento de aplicações interativas;
- Aptidões e habilidades:
  - Utilização de IDE - Ambiente de Desenvolvimento Integrado (*Integrated Development Environment*).

5.2 Objetivos de aprendizagem

Os objetivos de aprendizagem foram definidos com base nas recomendações do ACM, tanto o de *Computer Science* como o de *Computer Engineering*[10]. Também foram incluídos objetivos de aprendizagem da área dos jogos digitais, de acordo com o *IGDA Curriculum Framework*[11], particularmente no tópico *Conceptual Game Design*, que inclui os conceitos e componentes fundamentais de um jogo, as mecânicas e as dinâmicas. Aborda-se também o tópico do *Game*

[9] Por média complexidade entende-se a capacidade de desenvolver aplicações interativas simples que funcionem de forma autónoma, ou sejam baseadas em frameworks que potenciem outras arquiteturas de software.

[10] https://www.acm.org/education/curricula-recommendations

[11] http://www.igda.org/group/game-education

Programming, com enfoque particular na arquitetura de um motor de jogo e dos módulos que o compõem.

Com base nas competências definidas na secção anterior e nos documentos de base, classificados de acordo com a taxonomia de Bloom, o estudante deverá ser capaz de atingir os seguintes objetivos de aprendizagem:

- (1) Conhecimento

  Programação:
  - Reconhecer os elementos constituintes de um programa de computador;
  - Identificar o léxico e a sintaxe de uma linguagem de programação;
  - Descrever os diferentes tipos de dados;
  - Descrever as regras para a definição de variáveis e da atribuição de valores.

  Jogos Digitais:
  - Reconhecer os elementos constituintes de um jogo digital e descrever a sua funcionalidade básica;
  - Descrever a arquitetura de um motor de jogo e listar as funcionalidades de cada módulo;
- (2) Compreensão

  Programação:
  - Selecionar um algoritmo adequado a um determinado problema de programação;
  - Selecionar a estrutura de dados mais adequada a um determinado problema de programação;
  - Explicar o algoritmo subjacente a um programa;
  - Comparar a eficiência de diferentes algoritmos.

  Jogos Digitais:
  - Explicar as mecânicas e as dinâmicas de um determinado jogo digital;
- (3) Aplicação

  Programação:
  - Aplicar as regras de sintaxe e o léxico de uma determinada linguagem de programação para o desenvolvimento de um programa informático;
  - Modificar o código fonte de um jogo digital simples, de forma a experimentar variações simples dos elementos de jogo;
  - Verificar a funcionalidade de um programa através da definição de um bloco de testes.

  Jogos Digitais:
  - Alterar um jogo digital pela variação das mecânicas do jogo base.
- (4) Análise

  Programação:
  - Examinar um código fonte numa linguagem de programação específica e corrigir os erros semânticos (bugs) existentes;
  - Otimizar um código fonte no sentido de o tornar mais eficiente.

  Jogos Digitais:
  - Desenvolver novas dinâmicas para um jogo digital.

- (5) Síntese

  Programação:
  - Desenhar um algoritmo para solucionar um determinado problema informático;
  - Desenvolver um programa numa linguagem de programação para solucionar um determinado problema;
  - Criar uma aplicação interativa;
  - Criar abstrações específicas para resolver um determinado problema de programação.
- (6) Avaliação

  Programação:
  - Selecionar os algoritmos de pesquisa e ordenação mais eficientes para um determinado problema;
  - Avaliar a eficiência de um algoritmo.

6 Conteúdos programáticos

Os conteúdos programáticos seguem os objetivos de aprendizagem, convergindo para os tópicos principais do ensino da programação, adaptando componentes dos sistemas interativos e dos jogos digitais de acordo com os curricula ACM e IGDA.

6.1 Programa

O programa estrutura-se em 6 tópicos, 4 na área da Programação e 2 na área dos Jogos Digitais:

1.	**Fundamentos da Programação**
1.1.	Léxico e sintaxe de uma linguagem de programação;
1.2.	Variáveis e tipos de dados primitivos;
1.3.	Expressões;
1.4.	Input/Output básico, incluindo ficheiros;
1.5.	Estruturas de controlo condicionais;
1.6.	Funções e passagem de parâmetros;
1.7.	Recursividade e iteração: ciclos condicionais e contados;
1.8.	Teste e depuração de programas.
2.	**Estruturas de Dados**
2.1.	Vetores;
2.2.	Cadeias de caracteres;
2.3.	Listas, pilhas, filas e tabelas;
2.4.	Abstrações de dados e sua implementação;
2.5.	Seleção da estrutura de dados mais adequada.
3.	**Projeto de Software**
3.1.	Conceitos básicos (abstração, decomposição, encapsulamento, reutilização e teste);
3.2.	Programação estruturada;
3.3.	Programação orientada por objetos;
3.4.	Programação baseada em abstrações: bibliotecas e frameworks.
4.	**Gráficos e Visualização**
4.1.	Representação de imagem: modelo vetorial e raster;
4.2.	Cor e modelos de cor;
4.3.	Desenvolvimento de aplicações gráficas;
4.4.	Interfaces gráficas.
5.	**Conceito e design de Jogos Digitais**
5.1.	Conceito de jogo e de elementos de jogo;
5.2.	A narrativa de jogo;
5.3.	Mecânicas de jogo;
5.4.	Design dos mundos de jogo (níveis).

6. Programação de Jogos Digitais
6.1. Arquitetura de um motor de jogo;
6.2. Programação gráfica;
6.3. Programação de áudio;
6.4. Lógica de jogo;
6.5. Input e gestão de eventos;
6.6. Física e deteção de colisões;
6.7. Inteligência artificial aplicada aos jogos.

6.2 Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular

Os conteúdos programáticos desta UC baseiam-se nas recomendações da ACM em termos do ensino da programação e da IGDA em termos de Jogos Digitais. A forma como esses tópicos se articulam e integram no percurso de aprendizagem formam os pilares da Introdução à Programação Criativa.

Os fundamentos da programação são os pilares para o desenvolvimento do pensamento computacional e dos primeiros programas, que dão lugar à descoberta dos conceitos base dos jogos digitais, logo nos primeiros desafios. São os jogos digitais que, através da exploração das suas mecânicas e dinâmicas, levam depois o estudante ao domínio dos tipos e estruturas de dados, bem como das estruturas de controlo. À medida que o estudante aprofunda as suas competências de programação é possível abordar tópicos mais complexos como o projeto de software e o paradigma da orientação por objetos. Os projetos de jogos potenciam ainda os tópicos de programação de aplicações multimédia, como os gráficos e a visualização, as interfaces e ainda a exploração da arquitetura e dos diversos módulos de um motor de jogo.

6.3 Seleção da linguagem de Programação

De acordo com a secção 4.3 verifica-se uma tendência a nível internacional para a utilização da linguagem Python na Introdução à Programação. No caso particular do ensino na área dos Media Digitais existe também a opção pelo Javascript e pelo Processing, um ambiente de programação desenvolvido para a área criativa, baseado na linguagem Java.

Tanto Python como Javascript são linguagens dinâmicas e que tanto potenciam a programação no paradigma imperativo como orientado por objetos. Possuem um conjunto de bibliotecas e API (*Applications Programing Interface*) bastante alargado, particularmente na componente de sistemas interativos e multimédia e têm uma adoção forte ao nível do mercado de trabalho. Processing tem um conjunto de funcionalidades vasto para a área do multimédia e um ambiente de programação simplificado, baseado numa metáfora de sistemas interativos.

Python é utilizada com grande frequência na indústria, tanto para programação dinâmica, como na automação de tarefas em software, e até mesmo na computação física, como é o caso do Raspberry Pi[12] ou micro:bit[13]. Javascript é a linguagem da Web, atualmente potenciada ainda mais com o surgimento da versão 5 do HTML. Processing tem vindo a ser utilizada com frequência para a programação de aplicações multimédia simples, mas é um ambiente mais

[12] https://www.raspberrypi.org/
[13] http://microbit.org/

limitado em termos adoção industrial. Verifica-se uma tendência de substituir o Processing por Javascript com p5.js[14] ou por Python com Processing.py[15].

Tendo em consideração a UC do Mestrado em Multimédia, com 1 semestre de duração e 6 ECTS, optou-se pela utilização de Python. É uma linguagem dinâmica e que se integra bem com uma aprendizagem visual e com um IDE profissional, sendo uma linguagem emergente no mercado de trabalho. Javascript tem também um grande potencial, particularmente para a programação Web, no entanto essa característica torna mais difícil a depuração de código e mesmo o próprio ambiente de desenvolvimento Web torna-se mais complexo de configurar para um principiante. Processing tem uma curva de aprendizagem mais curta, através da simplificação do ambiente de aprendizagem, mas oculta algumas das questões relevantes no desenvolvimento de software. Embora estas duas linguagens tenham grande potencial, seria preferível adotá-las como segunda linguagem num curso com mais de uma unidade curricular de programação na área dos Media Digitais.

Optou-se assim pela seleção de Python com Pygame, mantendo em aberto a possibilidade de no futuro se optar pela adoção de Javascript com p5.js.

6.4 Bibliografia

Os estudantes abordam os conteúdos programáticos através de tutoriais de aprendizagem desenvolvidos pelo autor, fornecidos no LMS da UC, e que incorporam, adicionalmente, pequenas notas e referências aos tópicos de aprendizagem abordados.

No entanto, é essencial que antes da aula os estudantes estudem os capítulos relevantes da bibliografia principal, e durante a aula possam consultar os Web sites de referência da linguagem.

Também é definida uma bibliografia complementar, para explorar fora da aula, em autonomia, e para os tópicos relacionados com os Jogos Digitais.

6.4.1 Bibliografia principal

A bibliografia principal é focada na aprendizagem dos conteúdos programáticos na linguagem de programação selecionada (Python). Apresentam-se duas propostas, uma em português e outra em inglês:

- Ernesto Costa (2015). Programação em python - fundamentos e resolução de problemas, FCA.
  ISBN: 978-972-722-816-4
- Bill Lubanovic (2015). Introducing Python: Modern Computing in Simple Packages, O'Reilly.
  ISBN: 978-1-449-35936-2

E também nos Web sites de referência para a linguagem Python e para a biblioteca Pygame:

- Python.org: https://www.python.org/
- Pygame.org: https://www.pygame.org/docs/

[14] https://p5js.org/
[15] http://py.processing.org/

6.4.2 Bibliografia complementar

Os seguintes livros, também disponíveis online, apresentam vários tutoriais para a construção de jogos em Python e com Pygame, que os estudantes podem desenvolver fora da aula:

- Al Sweigart (2016), Invent Your Own Computer Games with Python, 4th edition, No Starch Press.
  ISBN: 978-1593277956
  Available online at: https://inventwithpython.com/invent4thed/
- Al Sweigart (2012), Making Games with Python and Pygame, CreateSpace Independent Publishing.
  ISBN: 978-1469901732
  Available online at: http://inventwithpython.com/pygame/

Os tópicos relacionados com Jogos podem ser explorados com os seguintes livros, um em português e outro em inglês:

- Carlos Martinho (2014), Pedro Santos e Rui Prada; Design e Desenvolvimento de Jogos, FCA - Editora de Informática, Lda.. ISBN: 978-972-722-762-4
- Jesse Schell (2008), The art of game design, CRC Press. ISBN: 978-0-12-369496-6

7 Metodologia de Ensino

Na preparação desta unidade curricular houve uma especial atenção à criação de mecanismos que assegurem a motivação e o trabalho continuado dos estudantes, essenciais para o sucesso na aprendizagem da programação. Para tal criou-se um plano de atividades ludificado que possibilite ao estudante adquirir competências no âmbito do pensamento computacional e a capacidade de desenvolver projetos de software para sistemas interativos de média complexidade.

A unidade curricular do Mestrado em Multimédia funciona num regime de uma aula teórico-prática semanal com três horas de duração, ao longo das 14 semanas de um semestre letivo. Pelo que é fomentado o trabalho fora da aula, em autonomia, ao longo de todo o semestre.

A ludificação do processo de aprendizagem admite também uma avaliação com percursos diferenciados para cada estudante, possibilitando aos estudantes progredirem com a velocidade que mais se adequa às suas capacidades e necessidades.

7.1 Abordagem metodológica

A metodologia de ensino é baseada numa abordagem construcionista, com PBL, definindo-se percursos de aprendizagem com base em problemas baseados em jogos digitais. A dificuldade inerente a construir uma base problemática coerente com o processo de aprendizagem levou à adoção do método de certificação triádica (Baptista, 2017), derivando na matriz apresentada no Anexo A – Tabela de atividades de desenvolvimento de jogos.

O método de certificação triádica serve de guia à elaboração dos tutoriais, assegurando a abordagem completa aos conteúdos programáticos e sua disponibilização na sequência mais adequada. Estes tutoriais são desenvolvidos pelo estudante na aula, sob supervisão do docente e, no final da aula, são lançados desafios de programação com base nos jogos desenvolvidos, para serem resolvidos pelo estudante em autonomia. A leitura da bibliografia principal é realizada antes da aula, de forma a que possa ser revista e discutida pelo docente, na aula, antes da realização do tutorial.

É também utilizada ludificação para motivar os estudantes a adotar um ritmo de aprendizagem regular para conseguirem adquirir as competências necessárias em cada fase do semestre.

A avaliação da UC é heterogénea, conjugando uma componente distribuída, com desafios semanais e provas práticas em computador, com um exame final, onde o estudante demonstra as competências adquiridas de forma mais abrangente.

7.2 Avaliação

A avaliação da UC é distribuída com exame final.

A componente distribuída tem um peso de 70% e o exame final 30%.

A avaliação distribuída é composta por duas componentes:

- Provas práticas em computador (duração de 60 a 90 min);

São definidas 4 provas práticas em computador que permitem avaliar a capacidade de, em autonomia, o estudante ser capaz de colocar em funcionamento algoritmos para resolver determinados problemas de baixa/média complexidade. A correção é automática e realizada por computador.

- Avaliação Contínua

A metodologia de ensino baseia-se num projeto de ludificação que, através de um sistema de pontuação, avalia a atitude do estudante ao longo de todo o percurso de aprendizagem. Os pontos obtidos pelo estudante são convertidos na avaliação contínua. Neste caso particular, um máximo de 1000 pontos corresponde a 100% na avaliação contínua.

A fórmula de cálculo da classificação final será:

PPC * 70% + AC * 30%

PPC = média aritmética das classificações nas provas práticas em computador

AC = avaliação contínua

7.3 Projeto de ludificação da aprendizagem

O projeto de ludificação do processo de aprendizagem teve em conta o framework Werbachk's 6 steps for Gamification (Werbachk, 2012). Descreve-se de seguida, de acordo com os seis passos que compõem o método.

7.3.1 Definição dos objetivos

O principal objetivo a atingir é desenvolver um ambiente de aprendizagem motivador e envolvente que incentive os estudantes a estudarem programação de forma regular, mantendo objetivos de curto prazo (semanais). Desta forma é possível alavancar a aquisição de competências do pensamento computacional e a capacidade de desenvolver em autonomia programas de baixa/média complexidade, fundamentais para o progresso ao longo do curso.

Como objetivo adicional, pretende-se incorporar as competências criativas dos estudantes de Multimédia para fazer uma abordagem introdutória ao desenvolvimento jogos digitais, servindo também de componente motivadora para a aprendizagem da programação.

7.3.2 Delinear os comportamentos alvo

O segundo passo foca-se na definição dos comportamentos que os estudantes devem adotar (ou corrigir) de forma a atingir os objetivos definidos no primeiro passo.

Descrevem-se de seguida os principais comportamentos a fomentar:

- Ler os módulos do livro de estudo antes da aula teórica-prática

De forma a que a componente mais teórica das aulas teorico-práticas (TP) possa ter uma participação mais ativa e promover maior dinamismo e discussão é importante que o estudante leia previamente os módulos de estudo designados na semana anterior.

- Resolver exercícios de programação de forma regular

É essencial que os estudantes pratiquem de forma regular (semanalmente) exercícios de programação em autonomia e que se submetam a uma auto-avaliação formativa de forma a detetarem por si próprios as suas principais lacunas para as poderem superar posteriormente com o docente, na aula.

- Promover a individualização do processo de aprendizagem

Os estudantes têm percursos de aprendizagem muito diversificados e possuem conhecimentos prévios heterogéneos relativamente à programação. O ritmo dos processos de aprendizagem torna-se assim assimétrico entre os estudantes, pelo que é importante promover a diferenciação do método de aprendizagem consoante as suas competências, evitando demasiada pressão nos

estudantes com maior dificuldade e evitando a desmotivação nos estudantes com maiores competências.

- Assiduidade e pontualidade

É essencial que os estudantes frequentem as aulas e sejam assíduos, maximizando tanto o ritmo das aulas presenciais como a capacidade de supervisão dos docentes.

7.3.3 Descrever os Jogadores

Sendo estudantes de um 2º ciclo, têm um perfil muito heterogéneo, tanto ao nível da formação de base, como a nível sociodemográfico, desde recém-licenciados com pouco mais de 20 anos até profissionais da área e com grande experiência. Acrescente-se que é normal a frequência desta unidade curricular, que é optativa, por estudantes de outros cursos e estudantes de mobilidade.

7.3.4 Elaborar os ciclos de atividade

De acordo com os comportamentos alvo identificados definiram-se dois ciclos de atividades: um ciclo de atividades ao longo do semestre, que pretende mapear a "jornada do jogador" e um outro ciclo semanal que mapeia um sistema de pontuação que fomenta atividades regulares por parte dos estudantes.

O ciclo semestral desenvolveu-se em 14 semanas de aulas, sendo 2 semanas utilizadas para o "Onboarding", 10 para o "Scaffolding" e as 2 semanas finais para o "Endgame".

A Figura 2 apresenta um gráfico ilustrativo do ciclo de atividades semestral.

Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Fase	Onboarding		Scafolding										Endgame	
Avaliação						PP1				PP2			PP3	PP4

Figura 2 - Ciclo de atividades semestral

Na primeira semana de aulas, é feita a apresentação da unidade curricular onde se descrevem os objetivos de aprendizagem, os primeiros conceitos e o funcionamento da UC, nomeadamente o processo de ludificação: as "regras do jogo". Nas duas primeiras aulas, são referidos os conceitos fundamentais da Programação bem como as bases da linguagem de programação a adotar ao longo do semestre como ferramenta essencial para o desenvolvimento das competências do pensamento computacional.

Nesta fase inicial não se aborda a linguagem diretamente, mas sim através de um PLE (*Programming Learning Environment*), como o RUR-PLE[16] no caso de Python, ou um jogo para aprendizagem do pensamento computacional. Neste caso pode-se utilizar o Code Combat[17], um jogo de aventura, em ambiente web, para aprendizagem das linguagens Javascript ou Python, entre outras. Desta forma vai-se introduzindo os conceitos fundamentais do pensamento computacional e o léxico e a sintaxe da linguagem de programação através das mecânicas de jogo.

[16] http://rur-ple.sourceforge.net/

[17] http://codecombat.com/

A partir da 3ª semana de aulas, e ao longo de 10 semanas, o estudante entra na fase de "Scaffolding" onde a aprendizagem vai ser promovida em ciclos semanais, desenvolvendo desafios de dificuldade crescente. O estudante vai progredindo por três níveis designados de "Aprendiz", "Pro" e "Perito":

- Aprendiz (0 pontos)

  O nível inicial.

- Pro (500 pontos)

  Permite eliminar a pontuação da pior prova prática. Desta forma é possível que o estudante recupere de alguma dificuldade que tenha ocorrido na realização de uma das provas.

- Perito (1000 pontos)

  Também permite eliminar a pior prova prática... Mas ainda pode substituir uma outra prova prática em computador por um projeto.

Finalmente, na fase de "Endgame" os estudantes são sujeitos a uma avaliação final, o grande desafio, composta por duas provas práticas que cobrem todos os tópicos lecionados, ou o projeto no caso dos estudantes que tenham atingido o nível de "Perito".

A passagem entre níveis faz-se através de um sistema de pontuação onde o estudante vai conquistando pontos por cada desafio superado. Os níveis e a pontuação permitem ao estudante ter uma medida da sua progressão ao longo do processo de aprendizagem e fornecem recompensas muito relevantes, tais como eliminar a pior prova prática ou substituir uma outra prova prática por um projeto. Estas recompensas são importantes para permitir ao estudante recuperar de dificuldades pontuais no percurso de aprendizagem e individualizar o seu percurso caso prefira desenvolver um projeto. O sistema de pontuação assegura que esta possibilidade só é atribuída aos estudantes que têm uma média das duas primeiras provas práticas superior a 75% (15 valores em 20) e que cumpriram a grande maioria dos desafios (nível de "Perito").

Sendo o objetivo do estudante a obtenção dos 1000 pontos, são atribuídos diversos desafios ao longo de um ciclo semanal com 10 iterações onde o estudante pode conquistar 1110 pontos. Estes cerca de 10% adicionais dão alguma tolerância para evitar que o estudante possa perder uma oportunidade por um problema pessoal ou evento imprevisível.

O ciclo semanal é apresentado na figura seguinte:

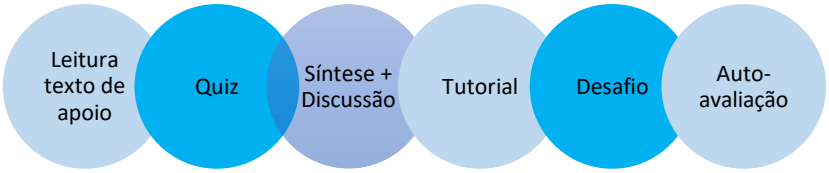

Figura 3 - Ciclo de atividades semanal

A primeira atividade do ciclo semanal é a leitura do texto de apoio e, para validar que o estudante leu o módulo, é utilizado um jogo *multiplayer* tipo *quiz* no início da aula (Kahoot[18]). Este jogo permite ao docente resumir os tópicos mais relevantes da leitura do texto de apoio e validar se foram compreendidos pelos estudantes, bem como fomentar a assiduidade dos estudantes no início da aula pois são contabilizados pontos pela participação e por acertar num

[18] https://create.kahoot.it/

mínimo de respostas certas. A aula prossegue com uma breve síntese e discussão dos conteúdos programáticos dessa semana através de um conjunto de exemplos, desenvolvidos de forma interativa com os estudantes, estimulando a sua participação. Na segunda parte da aula é apresentado um exercício (tutorial) para ser desenvolvido em autonomia pelo estudante, com a supervisão do docente. A participação nas aulas teórico-práticas é contabilizada na pontuação e no final da aula são apresentados desafios, com base nos jogos, para desenvolver em casa. Finalmente, no final da semana os estudantes têm que se submeter a auto-avaliação formativa, com suporte num sistema de correção automática.

7.3.5 Não esquecer a diversão (motivação e envolvimento);

O tópico dos jogos digitais fornece uma componente de diversão, permitindo que os exercícios desenvolvidos em aula sejam depois jogados pelo estudante. Adicionalmente, é também possível variar as mecânicas e dinâmicas do jogo, no sentido de o estudante criar a sua própria versão, fomentando as suas capacidades criativas. De certa forma, procura-se obter o que os psicólogos denominam "IKEA efect" (Norton et al. 2012) que promove uma perceção de maior valor pelos bens que foram produzidos pelo próprio.

Embora no futuro este componente vá ser mais trabalhado, nesta fase a abordagem foca-se na motivação intrínseca dos estudantes, garantindo que o estudante se apercebe da sua progressão e que lhe é fornecido *feedback* imediato. No caso do projeto dá-se a liberdade para os estudantes desenvolverem um projeto do seu gosto.

7.3.6 Implementar as ferramentas apropriadas

Um projeto de ludificação requer a utilização de um conjunto de ferramentas para a sua operacionalização. Neste caso desenvolveu-se um sistema de pontuação, um sistema de correção automático e um conjunto de tutoriais baseados no desenvolvimento de jogos.

- <u>Sistema de pontuação</u>

O processo de ludificação alicerçou-se num sistema de pontuação que permite ao estudante monitorizar o seu progresso e delinear o seu método de trabalho de acordo com o nível que pretende atingir ("Aprendiz", "Pro" e "Perito").

Apresenta-se de seguida a tabela de pontos e duração das atividades:

Desafio	pontos	Nº desafios	total	duração
Leitura (quiz)	15	10	150	120 min
Aula	10	12	120	180 min
TPC	30	10	300	120 min
Autoavaliação	20	7	140	60 min
PPC1, PPC2	200	2	400	60 min

Tabela 1 - Pontuação atribuída ao estudante nas atividades ludificadas

As atividades do ciclo semanal somam 710 pontos o que possibilita atingir o nível de "Pro" (500 pontos), permitindo eliminar a pior classificação das provas práticas, mas não a de "Perito". Este nível requer que o estudante atinja os 1000 pontos e obtenha um mínimo de 75% (300 pontos) em ambas as provas práticas em computador iniciais (PPC1 e PPC2). Ou seja, é feita uma filtragem pelos estudantes que demonstrem capacidade para poderem desenvolver o projeto opcional em autonomia. As atividades fora das aulas têm uma duração média semanal de 5 horas durante estas 10 semanas.

É importante realçar que, para além dos pontos permitirem a progressão ao longo dos níveis, obtendo as respetivas recompensas, também se mapeiam na classificação contínua. Assim, os 1000 pontos correspondem à obtenção de uma classificação de cerca de 20% da nota final, possibilitando que o estudante valorize o esforço despendido nestas atividades, atribuindo-lhes um valor tangível.

A pontuação é atualizada numa periodicidade semanal, sendo visualizada no LMS.

Fundamental para esta estratégia é a utilização de uma ferramenta de correção automática de exercícios de programação. Desenvolvida inteiramente de raiz, funciona integrada com o servidor de e-mail da FEUP. Desta forma os estudantes podem realizar a sua auto-avaliação, através do envio de um e-mail para o endereço da unidade curricular, recebendo a classificação (em percentagem) e a descrição dos erros obtidos num prazo máximo de 5 minutos após a submissão. Esta ferramenta é também utilizada para a correção das provas práticas em computador possibilitando que os estudantes saibam a sua classificação logo após a realização da prova.

- <u>Atividades baseadas em problemas centrados no desenvolvimento de jogos digitais</u>

O método de ensino foi desenvolvido com base na abordagem PBL e foca-se num conjunto de problemas centrados no desenvolvimento de jogos digitais. Assim, cada jogo digital promoverá o desenvolvimento de determinadas competências. Foi realizada uma seleção de jogos que promovesse uma progressão suave, para o qual se utilizou o método de certificação triádica (Baptista, 2017). Este método permite assegurar que o conjunto de jogos que foram selecionados permitem trabalhar os conteúdos programáticos até ao nível da taxonomia de Bloom especificado nos objetivos de aprendizagem.

Cada problema é composto por 4 passos que permitem a aprendizagem dos conceitos base. Começando por jogar o jogo, o estudante apercebe-se do que se pretende obter no final. Seguidamente, um tutorial conduz à construção desse jogo, fazendo com que o estudante se vá apercebendo de como as diversas estruturas do programa se mapeiam nas mecânicas e dinâmicas do jogo. Após esta fase, surgem como exercícios variações das mecânicas do jogo, promovendo o domínio das principais estruturas dos conteúdos programáticos selecionados para a atividade. Finalmente, apelando à criatividade do estudante, é proposto como desafio a construção de um novo jogo com mecânicas semelhantes. Este ciclo varia ao longo da fase de *Scaffolding*, sendo que os primeiros tutoriais são apenas orientados às duas primeiras atividades, sendo acrescentada a terceira a partir das primeiras semanas e a quarta atividade, a construção de um novo jogo, apenas nas finais.

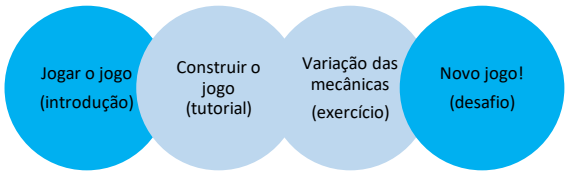

Figura 4 - Ciclo de atividades atividades baseadas em problemas

Descrevem-se de seguida os tópicos e jogos associados utilizados na aula na fase de *scaffolding* (10 semanas):

1	Introdução ao Python: Ambiente de desenvolvimento (IDE) e execução em linha de comando; léxico e sintaxe; variáveis e tipos de dados primitivos; expressões. Um primeiro jogo em Python. Estruturas de decisão. • Jogo da moeda (cara ou coroa) • Jogo "Pedra, papel, tesoura"
2	Recursividade e iteração: ciclos, o ciclo de jogo. Abstração procedimental: funções. Definição dos testes e deteção de erros semânticos (debugging). • Jogo de adivinhar um número de 1 a 100 • Jogo de cartas "21"
3	Estruturas de dados: Vetores. • Mastermind • Roleta
4	Estruturas de dados: Cadeias de caracteres. • Jogo da forca • Sopa de letras
5, 6	Estruturas de dados: listas, filas, pilhas e tabelas. Introdução ao Pygame. • Jogo da serpente – listas • Corrida de automóveis – filas • Torres de Hanói – pilhas • Jogo de aventura de texto – tabelas
7	Anatomia de um motor de Jogo. Gráficos e visualização. • Pong
8, 9	Abstração de dados. Classes de objetos. • Pacman • Jogo de Plataformas
10	Algoritmos de pesquisa e ordenação. Eficiência dos algoritmos.

A matriz de desenvolvimento do método de certificação triádica (anexo A) é composta pela tabela de "conteúdos programáticos", seguindo-se uma coluna "alvo" com o nível da taxonomia de Bloom desejado:

1 - Conhecimento
2 - Compreensão
3 - Aplicação
4 - Análise
5 - Síntese
6 - Avaliação

As colunas seguintes apresentam o número de tutoriais que abordam o tópico ("N"), os níveis médio ("Média") e máximo ("Max") da taxonomia de Bloom atingidos. Neste caso o nível máximo deve ser igual ou da coluna "Alvo". Finalmente, cada uma das colunas seguintes

descreve como cada tutorial de jogo deve abordar cada um dos tópicos definidos e a semana em que deve ser desenvolvido.

7.4 Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

A metodologia de ensino é baseada num conjunto de dinâmicas pedagógicas ativas que promovem a aquisição das competências definidas nos objetivos de aprendizagem, seguindo uma perspetiva PBL. Essas competências são adquiridas através de um conjunto de tutoriais de criação de jogos digitais, que são desenvolvidos segundo o modelo de certificação triádica, assegurando a cobertura da totalidade desses objetivos de aprendizagem.

A ludificação do processo de aprendizagem possibilita também uma avaliação com percursos diferenciados para cada estudante, possibilitando a sua progressão com a velocidade que mais se adequa às suas capacidades e necessidades.

8 Conclusões

As competências associadas ao pensamento computacional são essenciais para o profissional dos Media Digitais. Particularmente no Multimédia, o desenvolvimento de competências para o desenvolvimento de aplicações interativas é fundamental. A unidade curricular de Introdução à Programação Criativa procura desenvolver estas competências, de forma criativa, assumindo a metáfora do jogo digital para a compreensão dos principais conceitos de programação.

Procurou-se inovar a componente pedagógica, baseando a aprendizagem em problemas de construção de jogos digitais que concretizem os diversos objetivos de aprendizagem definidos pelas principais associações da área (ACM e IGDA). Esta abordagem é consolidada com os resultados de projetos de investigação científica na área da aprendizagem baseada em jogos, particularmente utilizando o método de certificação triádica. Esta unidade curricular, tendo sido desenvolvida para o Mestrado em Multimédia da UP, pode ser assim adaptada para outros cursos.

Adicionalmente, projetou-se um conjunto de atividades ludificadas que promovam uma progressão regular na aprendizagem, essencial para a uma primeira abordagem à programação. A partir de um conjunto de objetivos bem definidos, foram definidos os comportamentos alvo e o perfil dos estudantes, sendo definidos os ciclos de envolvimento e as ferramentas necessárias, sem esquecer que a aprendizagem deve ser divertida...

Pretendeu-se com o desenvolvimento desta unidade curricular de "Introdução à Programação Criativa" desenvolver uma "introdução criativa à programação", que motive e cative os estudantes da área dos Media Digitais, consolidada pela experiência pedagógica e em resultados científicos desenvolvidos pelo autor.

Referências

Aleksić, V., & Ivanović, M. (2016). Informatics in Education, vol. 15, np. 2, pp. 163-182.

Baptista, R. (2017). Jogos Sérios para Treino e Certificação de Competências (tese de doutoramento), Programa Doutoral em Media Digitais, UP.

Bourazeri, A., Arnab, S., Heidmann, O., Coelho, A., & Morini, L. (2017). Taxonomy of a gamified lesson path for STEM education: The Beaconing approach. In Proceedings of the 11th European Conference on Games Based Learning ECGBL 2017, Graz, Austria, pp. 29-37.

Deterding, S., Dixon, D., Khaled, R. & Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". In Proceedings of the 15th International Aca-demic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11). ACM, New York, NY, USA, pp. 9-15.

Faria, J. (2017). Jogos Pervasivos para o Ensino (dissertação de mestrado). Mestrado Integrado em Engenharia Informática e Computação, FEUP.

Guo, P. (2014). Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. Communications of the ACM, July, 2014.

Jenkins, T. (2002). On the Difficulty of Learning to Program. In 3rd Annual LTSN-ICS Conference. Loughborough, UK, pp. 53–58.

Monteiro, J. (2017) Ludificação do ensino da programação: um caso de estudo (dissertação de mestrado), Mestrado em Multimédia, UP.

Murphy, E., Crick, T. & Davenport, J.H. (2017). An Analysis of Introductory Programming Courses at UK Universities. The Art, Science, and Engineering of Programming, vol. 1, no. 2, article 18.

Norton, M., Mochon, D., & Ariely, D. (2012). The IKEA effect: When labor leads to love. In Journal of Consumer Psychology, vol. 22, no. 3, July 2012, pp.453-460.

Piteira, M., & Costa. C. (2013). Learning computer programming: study of difficulties in learning programming. In Proceedings of the 2013 International Conference on In-formation Systems and Design of Communication (ISDOC '13). ACM, New York, NY, USA, pp. 75-80.

Reis, L. (2017). Definição de um plano de atividades e de um conjunto de conteúdos pedagógicos para o ensino da programação através do desenvolvimento de jogos digitais (dissertação de mestrado). Mestrado Integrado em Engenharia Informática e Computação, FEUP.

Werbach, K., & Hunter, D. (2012). For the win: How game thinking can revolutionize your busi-ness. Wharton Digital Press.

Wing, J. M. (2006). Computational thinking. In Communications of the ACM, vol. 49, no. 3, pp. 33–35.

Anexo A – Tabela de atividades de desenvolvimento de jogos

Conteúdos programáticos	Alvo	N	Média	Max	Moeda (cara ou coroa)	"Pedra, papel, tesoura"	Adivinha	"21"	Mastermind	Roleta	Forca	Sopa de letras	Serpente	Corrida de automóveis	Torres de Hanói	Aventura de texto	Pong	Pacman	Plataformas
					1	1	2	2	3	3	4	4	5	5	6	6	7	8	9
1. Fundamentos da Programação																			
1.1. Léxico e sintaxe de uma linguagem de programação	**3**	**15**	**2,67**	**3**	1	2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.2. Variáveis e tipos de dados primitivos	**3**	**15**	**2,67**	**3**	1	2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.3. Expressões	**3**	**14**	**2,79**	**3**		2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.4. I/O básico, incluindo ficheiros	**3**	**15**	**2,47**	**3**	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3
1.5. Estruturas de controlo condicionais	**4**	**15**	**3,13**	**4**	1	2	2	2	3	3	3	3	4	4	4	4	4	4	4
1.6. Funções e passagem de parâmetros	**4**	**13**	**3,23**	**4**			1	1	3	3	3	3	4	4	4	4	4	4	4
1.7. Recursividade e iteração: ciclos condicionais e contados	**4**	**13**	**3,23**	**4**			1	1	3	3	3	3	4	4	4	4	4	4	4
1.8. Teste e depuração (debug) de programas	**4**	**12**	**3,42**	**4**				1	3	3	3	3	4	4	4	4	4	4	4
2. Estruturas de Dados																			
2.1. Vetores	**3**	**14**	**2,36**	**3**	1	1	1	1	2	3		3	3	3	3	3	3	3	3
2.2. Cadeias de caracteres	**3**	**15**	**1,73**	**3**	1	1	1	1	1	1	2	3	2	2	2	3	2	2	2
2.3. Listas, Pilhas, filas e tabelas	**3**	**7**	**2,57**	**3**									1	2	3	3	3	3	3
2.4. Abstrações de dados e sua implementação	**3**	**7**	**2,14**	**3**									1	2	2	2	2	3	3
2.5. Seleção da estrutura de dados mais adequada	**2**	**7**	**1,43**	**2**									1	1	1	1	2	2	2
3. Projeto de Software																			
3.1. Conceitos básicos;	**3**	**13**	**2,46**	**3**			1	2	2	2	2	2	3	3	3	3	3	3	3
3.2. Programação estruturada;	**3**	**15**	**2,80**	**5**	1	1	2	2	3	3	3	3	3	3	3	3	3	4	5
3.3. Programação orientada por objetos;	**3**	**3**	**3,33**	**5**													2	3	5
3.4. Programação baseada em abstrações: bibliotecas e frameworks	**3**	**6**	**2,50**	**3**									1	2	3		3	3	3
4. Gráficos e Visualização																			
4.1. Representação de imagem: modelo vetorial e raster;	**3**	**6**	**2,33**	**3**									1	2	2		3	3	3
4.2. Cor e modelos de cor	**3**	**6**	**2,33**	**3**									1	2	2		3	3	3
4.3. Desenvolvimento de aplicações gráficas	**5**	**6**	**2,83**	**5**									1	2	2		3	4	5
4.4. Interfaces gráficas	**3**	**6**	**1,83**	**3**									1	1	1		2	3	3
5. Fundamentos e design de Jogos Digitais																			
5.1. Conceito de jogo e fatores humanos;	**3**	**15**	**2,80**	**3**	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
5.2. Narrativas interativas e não lineares;	**3**	**2**	**3,00**	**3**												3			3
5.3. Mecânicas de jogo;	**4**	**15**	**2,93**	**4**	1	2	2	2	3	3	3	3	3	3	3	4	4	4	4
5.4. Design dos mundos de jogo (níveis);	**5**	**6**	**2,67**	**5**									1	2		2	2	4	5
6. Programação de Jogos Digitais																			
6.1. Arquitetura de um motor de jogo	**2**	**6**	**1,50**	**2**									1	1	1		2	2	2
6.2. Programação gráfica	**3**	**6**	**2,33**	**3**									1	2	2		3	3	3
6.3. Programação de áudio	**2**	**4**	**1,75**	**2**										1			2	2	2
6.4. Lógica de jogo	**4**	**14**	**2,36**	**4**	1	1	1	1	2	2	2	2	3	3	3		4	4	4
6.5. Input e gestão de eventos	**2**	**14**	**1,50**	**2**	1	1	1	1	2	1	1	1	2	2	2		2	2	2
6.6. Física e deteção de colisões	**2**	**5**	**1,80**	**2**									1	2			2	2	2
6.7. Inteligência artificial	**2**	**3**	**1,67**	**2**													1	2	2

Anexo B – Ficha A3ES

6.2.1.1. Unidade curricular:

PT	Introdução à Programação Criativa
ENG	Introduction to Creative Programming

6.2.1.2. Docente responsável e respetiva carga letiva na unidade curricular (preencher o nome completo):

PT	António Fernando Vasconcelos cunha Castro Coelho (42 TP)
ENG	António Fernando Vasconcelos cunha Castro Coelho (42 TP)

6.2.1.3. Outros docentes e respetivas cargas letivas na unidade curricular:

PT	
ENG	

6.2.1.4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes) / Learning outcomes of the curricular unit:

Nota: A informação deverá incidir, não sobre os objetivos da própria UC, mas sim sobre os objetivos de aprendizagem (isto é, os conhecimentos, aptidões e competências que se espera que, no final da UC, os estudantes tenham adquirido).

PT

Em termos gerais, pretende-se que o estudante adquira as competências englobadas no pensamento computacional, acrescido da capacidade efetiva de desenvolver programas de baixa / média complexidade utilizando uma linguagem de programação.

No final desta unidade curricular o estudante deverá adquirir as seguintes competências:

Conhecimentos:

- Desenho de algoritmos;
- Criação de programas numa linguagem de programação;
- Desenvolvimento de aplicações interativas;
- Fundamentos do design de jogos digitais.

Atitudes e comportamentos:

- Estruturação dos problemas informáticos de acordo com os pilares do pensamento computacional;
- Utilização de frameworks multimédia no desenvolvimento de aplicações interativas;

Aptidões e habilidades:

- Utilização de IDE (Ambientes de Desenvolvimento Integrado).

ENG

In general terms, it is intended that the student acquire the competences encompassed in computational thinking, plus the effective ability to develop low / medium complexity programs using a programming language.

At the end of this course the student should acquire the following competences:

Knowledge:

- Algorithm design;
- Development of programs in a programming language;
- Development of interactive applications;
- Fundamentals of digital game design.

Attitudes and behaviors:

- Structuring of computer problems according to the pillars of computational thinking;
- Use of multimedia frameworks in the development of interactive applications;

Skills and abilities:

- Effective use of IDE (Integrated Development Environments).

6.2.1.5. Conteúdos programáticos / Syllabus

PT

1. Fundamentos da Programação

Léxico e sintaxe e de uma linguagem de programação; variáveis e tipos de dados primitivos; expressões; input/output; estruturas de controlo condicionais; funções e passagem de parâmetros; recursividade e iteração: ciclos condicionais e contados; teste e depuração de programas.

2. Estruturas de Dados

Vetores; cadeias de caracteres; listas, pilhas, filas e tabelas; abstrações de dados.

3. Projeto de Software

Programação estruturada; programação orientada por objetos; bibliotecas e frameworks.

4. Gráficos e Visualização

Representação de imagem; cor e modelos de cor; aplicações gráficas; interfaces gráficas.

5. Conceito e design de Jogos Digitais

Conceito e elementos de jogo; narrativa; mecânicas; design dos mundos de jogo (níveis).

6. Programação de Jogos Digitais

Arquitetura de um motor de jogo; programação gráfica; programação de áudio; lógica de jogo; input e gestão de eventos; física e deteção de colisões; inteligência artificial.

ENG

1. Programming Fundamentals

Vocabulary and syntax of a programming language; variables and primitive data types; expressions; input / output; conditional control structures; functions and parameters; recursion and iteration: conditional and the for loop; testing and debugging of programs.

2. Data Structures

Vectors; strings; stacks, lists and tables; abstractions.

3. Software Project

Structured programming; object-oriented programming; libraries and frameworks.

4. Graphics and Visualization

Image representation; color and color models; graphical applications; graphical interfaces.

5. Concept and Design of Digital Games

Concept and elements of digital games; narrative; mechanics; level design.

6. Digital Games Programming

Architecture of a game engine; graphic programming; audio programming; game logic; input and event management; physics and collision detection; artificial intelligence.

6.2.1.6. Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular / Demonstration of the syllabus coherence with the curricular unit's objectives:

PT

Os conteúdos programáticos baseiam-se nas recomendações do ACM em termos de ensino da programação e da IGDA em termos de Jogos Digitais. A forma como esses tópicos se articulam e integram no percurso de aprendizagem formam os pilares da Introdução à Programação Criativa.

Os fundamentos da programação são os pilares para o desenvolvimento do pensamento computacional e dos primeiros programas, que dão lugar à descoberta dos conceitos base dos jogos digitais, logo nos primeiros desafios. São os jogos digitais que, através da exploração das suas mecânicas e dinâmicas, levam depois o estudante ao domínio dos tipos e estruturas de dados, bem como das estruturas de controlo. À medida que o estudante aprofunda as suas competências de programação é possível abordar tópicos mais complexos como o projeto de software e o paradigma da orientação por objetos. Os projetos de jogos potenciam ainda os tópicos de programação de aplicações multimédia, como os gráficos e a visualização, as interfaces e ainda a exploração da arquitetura e dos diversos módulos de um motor de jogo.

ENG

The syllabus is based on the ACM Curricula for programming and also on the IGDA Curriculum Framework for Digital Games. The way these topics are articulated and integrated in the learning path compose the pillars of the Introduction to Creative Programming.

The fundamentals of programming are the pillars for the development of computational thinking and the first programs, which give rise to the discovery of the basic concepts of digital games, as early as the first challenges. It is the digital games that, through the exploration of their mechanics and dynamics, later lead the student to master the types and structures of data, as well as the control structures. As the student deepens his programming skills it is possible to address more complex topics such as software design and the object orientated paradigm. Game design further enhances the programming topics of multimedia applications such as graphics and visualization, interfaces, and the exploration of the architecture and the distinct modules of a game engine.

6.2.1.7. Metodologias de ensino (avaliação incluída) /Teaching methodologies (including evaluation):

PT

A metodologia de ensino baseia-se num modelo pedagógico construcionista, que alia um foco no desenvolvimento do pensamento computacional, através de um conjunto de dinâmicas pedagógicas ativas, desenvolvendo através de problemas que apelam à criatividade (jogos digitais) as competências básicas para o desenvolvimento de software de baixa/média complexidade. A unidade curricular é desenhada numa perspetiva de ludificação (*gamification*), criando ciclos iterativos curtos que promovem uma atividade regular de aprendizagem, conjugando uma componente supervisionada pelo docente e outra a desenvolver pelo estudante em autonomia.

A avaliação da UC é distribuída com exame final.

A componente distribuída tem um peso de 70% e o exame final 30%.

A avaliação distribuída é composta por duas componentes:

- Provas práticas em computador (duração de 60 a 90 min);

São definidas 4 provas práticas em computador que permitem avaliar a capacidade de, em autonomia, o estudante ser capaz de colocar em funcionamento algoritmos para resolver determinados problemas de baixa/média complexidade. A correção é automática por computador.

- Avaliação Contínua

Os pontos obtidos pelo estudante nas atividades ludificadas são convertidos na avaliação contínua. Neste caso particular, um máximo de 1000 pontos corresponde a 100% na avaliação contínua.

A fórmula de cálculo da classificação final será:

$$PPC * 70\% + AC * 30\%$$

PPC = média das provas práticas em computador

AC = avaliação contínua

ENG

The teaching methodology is based on a constructionist pedagogical model, which combines a focus on the development of computational thinking, through a set of active pedagogical dynamics, developing the basic skills for the development of low / medium complexity software through problems that appeal to creativity (digital games). The course is designed in a gamification perspective, creating short iterative cycles that promote regular learning activity, combining activities supervised by the teacher and other activities developed autonomously by the student.

The assessment of the course is distributed with a final exam.

The distributed component has a weight of 70% and the final exam 30%.

The distributed evaluation consists of two components:

- Practical exams (duration of 60 to 90 min);

4 practical exams are preformed using a computer that assess the ability of the student to develop algorithms to solve certain low / medium complexity problems. The assessment is automatic and performed by computer.

- Continuous assessment

The points obtained by the student in the gamified activities are converted into the continuous assessment. In this particular case, a maximum of 1000 points corresponds to 100% in the continuous assessment.

The formula for the final classification is:

$$PE * 70\% + CA * 30\%$$

PE = average of the practical exams

CA = continuous assessment

6.2.1.8. Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular / Demonstration of the coherence between the teaching methodologies and learning outcomes:

PT

A metodologia de ensino é baseada num conjunto de dinâmicas pedagógicas ativas que promovem a aquisição das competências definidas nos objetivos de aprendizagem, seguindo uma perspetiva PBL. Essas competências são adquiridas através de um conjunto de tutoriais de criação de jogos digitais, que são desenvolvidos segundo o modelo de certificação triádica, assegurando a cobertura da totalidade desses objetivos de aprendizagem.

A ludificação do processo de aprendizagem possibilita também uma avaliação com percursos diferenciados para cada estudante, possibilitando a sua progressão com a velocidade que mais se adequa às suas capacidades e necessidades.

ENG

The teaching methodology is based on a set of active pedagogical dynamics that promote the acquisition of the competences defined in the learning outcomes, following a PBL perspective. These competencies are acquired through a set of tutorials for creating digital games, which are developed according to the triadic certification model, ensuring the coverage of all of these learning objectives.

The gamification of the learning process also allows an assessment with distinct learning paths for each student, enabling them to progress with the speed that best suits their abilities and needs.

6.2.1.9. Bibliografia de consulta/existência obrigatória:

PT

Bibliografia principal:

- Ernesto Costa (2015). Programação em python - fundamentos e resolução de problemas, FCA. ISBN: 978-972-722-816-4
- Bill Lubanovic (2015). Introducing Python: Modern Computing in Simple Packages, O'Reilly. ISBN: 978-1-449-35936-2
- Python.org: https://www.python.org/
- Pygame.org: https://www.pygame.org/docs/

Bibliografia complementar:

- Al Sweigart (2016), Invent Your Own Computer Games with Python, 4th edition, No Starch Press. ISBN: 978-1593277956
  Available online at: https://inventwithpython.com/invent4thed/
- Al Sweigart (2012), Making Games with Python and Pygame, CreateSpace Independent Publishing. ISBN: 978-1469901732
  Available online at: http://inventwithpython.com/pygame/
- Carlos Martinho (2014), Pedro Santos e Rui Prada; Design e Desenvolvimento de Jogos, FCA - Editora de Informática, Lda.. ISBN: 978-972-722-762-4
- Jesse Schell (2008), The art of game design, CRC Press. ISBN: 978-0-12-369496-6

ENG

Main bibliography:

- Ernesto Costa (2015). Programação em python - fundamentos e resolução de problemas, FCA. ISBN: 978-972-722-816-4
- Bill Lubanovic (2015). Introducing Python: Modern Computing in Simple Packages, O'Reilly. ISBN: 978-1-449-35936-2
- Python.org: https://www.python.org/
- Pygame.org: https://www.pygame.org/docs/

Complementary bibliography:

- Al Sweigart (2016), Invent Your Own Computer Games with Python, 4th edition, No Starch Press. ISBN: 978-1593277956
  Available online at: https://inventwithpython.com/invent4thed/
- Al Sweigart (2012), Making Games with Python and Pygame, CreateSpace Independent Publishing. ISBN: 978-1469901732
  Available online at: http://inventwithpython.com/pygame/
- Carlos Martinho (2014), Pedro Santos e Rui Prada; Design e Desenvolvimento de Jogos, FCA - Editora de Informática, Lda.. ISBN: 978-972-722-762-4
- Jesse Schell (2008), The art of game design, CRC Press. ISBN: 978-0-12-369496-6

Anexo C – Cursos na área dos Media Digitais

A tabela C.1 apresenta os cursos de 1º ciclo em Portugal:

Instituição	Curso	Unidades curriculares de programação	1ª L.P.
IPCA	Engenharia em Desenvolvimento de Jogos Digitais	Paradigmas de Programação I (1ºa 1ºs 9 ECTS) Paradigmas de Programação II (1ºa 2ºs 9 ECTS) Introdução à Programação 3D (2ºa 1ºs 9 ECTS) Programação 3D (2ºa 2ºs 6 ECTS)	
IPCB	Tecnologias da Informação e Multimédia	Programação I (1ºa 1ºs 6 ECTS) Programação II (1ºa 2ºs 6 ECTS) Programação III (2ºa 1ºs 5 ECTS) Linguagens Web (2ºa 2ºs 5 ECTS)	
IPL	Jogos Digitais e Multimédia	Introdução à Programação (1ºa 1ºs 7 ECTS) Programação de Jogos (1ºa 2ºs 6 ECTS) Desenvolvimento Web (2ºa 1ºs 6ECTS)	Python
IPP	Design e Animação Multimedia	Programação (3ºa 2ºs 6 ECTS Op.)	Javascript
IPP	Engenharia da Computação Gráfica e Multimédia	Introdução à Programação (1ºa 1ºs 7 ECTS) Algoritmia e Programação (1ºa 2ºs 6 ECTS)	Java
IPP	Licenciatura em Multimédia	Programação I (1ºa 2ºs 7 ECTS) Programação II (2ºa 1ºs 6 ECTS) Programação III (2ºa 1ºs 6 ECTS)	C#
IPP	Tecnologias e Sistemas de Informação para a Web	Algoritmia e Estruturas de Dados (1ºa 1ºs 7 ECTS) Programação Orientada a Objetos (1ºa 2ºs 7 ECTS) Programação Web I (2ºa 1ºs 7 ECTS) Programação Web II (2ºa 2ºs 7 ECTS)	Javascript
IPT	Informática e Tecnologias Multimédia	Programação e Algoritmia (1ºa 1ºs 7 ECTS) Linguagens de Programação (1ºa 2ºs 6 ECTS) Programação Orientada por Objetos (2ºa 1ºs 6 ECTS) Técnicas Avançadas de Programação (2ºa 2ºs 6 ECTS)	C
IPV	Tecnologias e Design de Multimédia	Introdução à Programação (1ºa 1ºs 7,5 ECTS) Estruturas de Dados (1ºa 2ºs 8 ECTS) Técnicas Avançadas de Programação (2ºa 1ºs 8 ECTS)	
ISMAI	Artes e Multimédia	Programação Criativa (1ºa 1ºs 5 ECTS)	Processing
ISMAI	Tecnologias de Comunicação Multimédia	Técnicas de Programação I (1ºa 2ºs 5ECTS) Técnicas de Programação II (2ºa 2ºs 5ECTS Op)	Processing

Lusófona	Videojogos	Fundamentos de Programação (1ºa 1ºs 6 ECTS) Introdução à Computação (1ºa 1ºs 6 ECTS) Linguagens de Programação I (1ºa 2ºs 6 ECTS) Linguagens de Programação II (2ºa 1ºs 6 ECTS)	Javascript
UA	Novas Tecnologias da Comunicação	Laboratórios Multimédia I, II, III e IV[19]	Javascript
UC	Design e Multimédia	Introdução à Programação e Resolução de Problemas (1ºa 1ºs 12 ECTS) Programação Multimédia (1ºa 2ºs 12 ECTS)	Processing
UL	Arte Multimédia	Computação Multimédia (3ºa 1ºs 6ECTS - ramo "Ambientes Interactivos ")	Processing
UL	Design de Comunicação[20]	Programação Gráfica	Processing
UMa	Design de Media Interativos	Paradigmas da Programação (1ºa 1ºs 7,5 ECTS) Programação Orientada por Objetos (2ºa 2ºs 7,5 ECTS)	
UTAD	Comunicação e Multimédia	Fundamentos de Algoritmia e Programação (1ºa 2ºs 3 ECTS)	C , Java

A tabela C.2 apresenta os cursos de 2º ciclo em Portugal:

Instituição	**Curso**	**Unidades Curriculares de programação**	**1ª L.P.**
IPCA	Engenharia em Desenvolvimento de Jogos Digitais	-	
IPP	Sistemas e Media Interativos	Programação Criativa (1ºa 1ºs 6 ECTS)	
ISMAI	Tecnologias de Informação, Web e Multimédia	-	
Lusófona	Sistemas de Comunicação Multimédia	-	
UA	Comunicação Multimédia	-	
UAb+UP+UC	Computação Gráfica	-	
UBI	Design Multimédia	-	
UC	Design e Multimédia	Programação (1ºa 1ºs 6 ECTS Op)	Processing
UL	Arte Multimédia	-	
UL	Design de Comunicação e Novos Media	Linguagens da Programação Aplicadas ao Design (1ºa 2ºs 6 ECTS Op)	Processing
UM	Tecnologia e Arte Digital[20]	Programação para a Arte Digital I (1ºa 1ºs 5 ECTS) Programação para a Arte Digital II (1ºa 2ºs 5 ECTS)	
UMa	Design de Media Interativos	Tópicos Avançados em Informática (1ºa 2ºs 7,5 ECTS)	C
UNL	Novos Media e Práticas Web	-	
UP	Multimédia	Introdução à Programação (1ºa 1ºs S 6 ECTS op)	Python
UTAD	Multimédia	Paradigmas Emergentes para a Web (1ºa 1ºs 6 ECTS)	

[19] Neste 1º ciclo as programações são enquadradas nas unidades curriculares de Laboratório Multimédia
[20] Curso em restruturação

A tabela C.3 apresenta os cursos internacionais de 1º ciclo / BSc

University of Kent (UK)	Multimedia Technology and Design	Introduction to Programming (7,5)	C
University of Sussex (UK)	Games & Multimedia Environments	Introduction to Programming (7,5)	Java
Goldsmiths, University of London (UK)	Digital Arts Computing	Introduction to Programming (7,5)	
Coventry University (UK)	Multimedia Computing	Introduction to Computing	Python
Universitat Pompeu Fabra (ES)	Design and Production of Videogames	Fundamentals of Programming (5)	
Universitat de Girona (ES)	Disseny i Desenvolupament de Videojocs	Metodologia i tecnologia de la programació I e II (9+6)	C++
Carnegie Mellon University (US)	IDeATe - Integrative Design, Arts, and Technology	Introduction to Computing for Creative Practice	Javascript + p5.js
UT Austin (US)	The GAMMA Program	Foundations of Creative Coding	Processing
UT Austin (US)	The GAMMA Program	Computer Fluency	Python
UT Austin (US)	The GAMMA Program	Intro to Problem Solving and Programming	Python
UT Austin (US)	The GAMMA Program	Elements of Computers and Programming	Python
Northwestern University (US)	Media Arts and Game Design	Computer Code as Expressive Medium	Processing

A tabela C.4 apresenta os cursos internacionais de 2º ciclo / MSc

Instituição	**Curso**	**Unidades Curriculares de programação**	**1ª L.P.**
Trinity College Dublin (IE)	Interactive Digital Media	Programming for Digital Media (5+5)	Python + Javascript
University of Limerick (IE)	Interactive Media	Digital Media Software and Systems, Visual Coding	
University of Portsmouth (UK)	Digital Media	Mobile device application design and programming (15)	
Leeds Beckett University (UK)	Creative Technologies	-	
University of Applied Sciences Upper Austria (AT)	Interactive Media	-	