半面授校级硕士
电子游戏编程

tech 科学技术大学

tech 科学技术大学

半面授校级硕士
电子游戏编程

模式:**混合式(在线+临床实践)**
时间:**12个月**
学位: **TECH 科技大学**
网络访问: **www.techtitute.com/cn/video-games/hybrid-professional-master-degree/hybrid-professional-master-degree-video-game-programming**

目录

01

介绍

电子游戏创作领域已得到进一步巩固，并吸引了越来越多的年轻人。在电子游戏行业，支撑一款优秀游戏的支柱之一就是程序设计。创建基本指令并决定其整体功能的处理过程对于游戏故事和开发的顺利进行至关重要。这个半面授学习资格证书为游戏专业人士提供了一个专业化的学习机会，使他们能够在 游戏行业的职业生涯中更上一层楼。这个学位完全100%在线学习，适合你的需求，包括在视频游戏创作和开发工作室实习。所有这些都将使你在最优秀专家的指导下，在职业生涯中不断进步。

通过这个半面授校级硕士，成为电子游戏行业最优秀的程序员之一"

电子游戏产业潜力巨大。日益增长的需求和 游戏玩家 本身的要求，导致他们对游戏作品精益求精。高素质的专业编程团队为每项创作提供了高质量的支持。

电子游戏编程半面授校级硕士顺应了当前市场的需求，市场对高度参与创作的专业人才的需求日益增长。创意起着重要作用，但如果没有扎实的知识，就不可能制作出高质量的视频游戏。

因此，这个专业为学生提供了详尽的编程和软件工程基础知识，深入探讨了数据结构和算法，并教授了面向对象编程和引擎规范。这个课程还涉及实时编程，为电子游戏专业人员提供完整的半面授校级硕士。

为了实现在电子游戏编程专业领域取得进步的目标，学生们将拥有该领域的专业教师团队，他们将随时指导和辅导学生。此外，视频摘要、案例研究和额外阅读等互动内容将补充 TECH 在这个 100% 在线学位课程中提供的大量教学大纲，并提供 公司实习机会。

因此，学生可以在一家领先的电子游戏编程工作室进行为期 3 周的现场强化学习，从而完成这一学术课程。一个理想的专业环境，在这里你可以现场测试最先进的工作方法、程序和技术，以创作出高质量的作品。这是一个独特的机会，只有TECH这个世界上最大的数字大学能够提供。

这个**电子游戏编程半面授校级硕士**包含市场上最完整和最新的课程。主要特点是：

- 开发 100 多个电子游戏编程案例，由在视频游戏行业拥有丰富经验的编程专家和大学教授主讲
- 其图形化、示意图和突出的实用性内容，以其为构思，为那些对专业实践至关重要的医学学科提供科学和保健信息
- 由电子游戏编程和开发方面的专家介绍案例研究的发展
- 这个课程的内容图文并茂、示意性强、实用性强为那些视专业实践至关重要的学科提供了科学和实用的信息
- 可以进行自我评价过程的实践练习，以提高学习效果
- 其特别强调创新方法
- 理论课、向专家提问、关于有争议问题的讨论区和这个反思性论文
- 可从任何连接互联网的固定或便携设备上访问内容
- 这将由理论讲座、向专家提问、关于争议性问题的讨论论坛和个人反思工作来补充
- 可以从任何有互联网连接的固定或便携式设备上获取内容
- 此外，你还可以在全国最好的医院之一进行临床实习

> 掌握回溯技术，通过这个半面授校级硕士课程打造最佳冒险游戏”

这个 100% 在线的课程让你有机会在工作室实习，与顶级程序员一起测试自己"

这个半面授校级硕士将使你成为电子游戏编程领域的标杆。立即报名吧。

通过这个半面授校级硕士课程，你可以有效地开发应用于电子游戏引擎的应用程序。

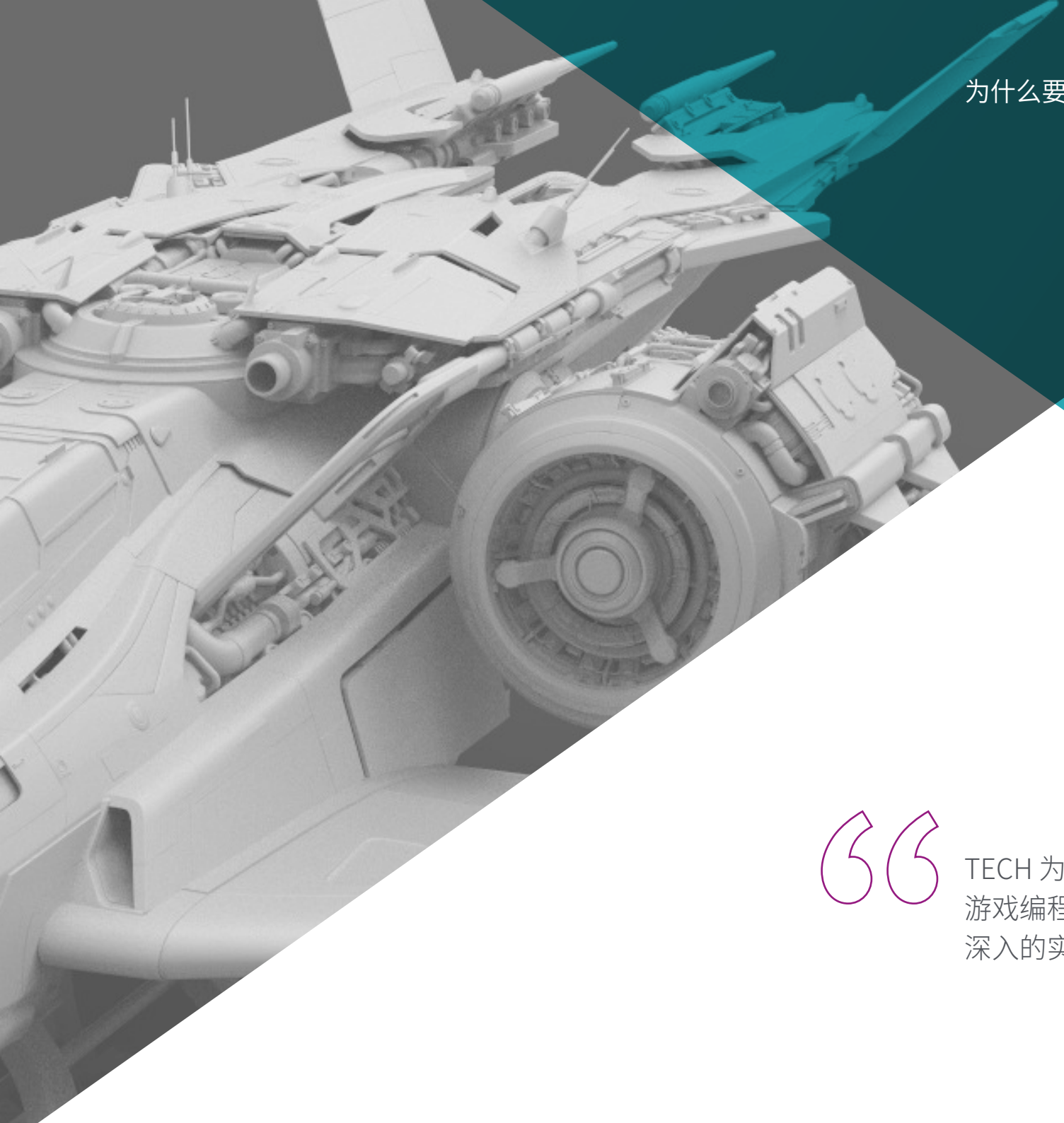
这个校级硕士具有专业化性质，采用半面授课程学习模式，旨在更新在大型创意工作室工作的视频游戏专业人员的知识，他们需要高水平的资格认证。内容以最新的科学证据为基础，以教学方式为导向，将理论知识与 游戏 行业的实践相结合，理论与实践相结合的元素将促进知识的更新，并有助于电子游戏编程方面的决策制定。

由于采用了最新教育技术开发的多媒体内容，它们将使电子游戏专业人员能够进行情景式学习，也就是说，模拟环境将提供身临其境的学习编程，在真实环境中进行培训。这个课程的设计重点是基于问题的学习，通过这种方式，你必须尝试解决整个课程中出现的不同专业实践情况。为此，你将获得由知名专家制作的新型交互式视频系统的帮助。

02

为什么要选择这个半面授校级硕士?

编程需要坚实的理论知识,但毫无疑问,日常实践是锤炼在游戏行业的专业技能的关键。因此,TECH 创建了这一学位,将算法设计、智能系统和实时编程等领域最先进的在线教学大纲与一流编程工作室的实践实习完美地结合在一起。通过这种方式,学生将更全面地了解电子游戏编程、所使用的程序和最新技术。所有这一切,始终由该领域最优秀的专家提供指导。



TECH 为你提供了在电子游戏编程方面进行有效而深入的实践实习的机会"

1. 升级到最新的可用技术

近年来,技术为电子游戏编程领域带来了革命性的变化,有利于制作出质量更高、更逼真的游戏。因此,为了拉近学生与这项技术的距离,TECH 创建了这个半面授校级硕士,专业人员将在这里学习电子游戏引擎、当前的技术挑战和创建本体的软件。这将使你掌握最新的技术。

2. 汲取最优秀专家的专业知识

在这一学术旅程中,学生将始终得到电子游戏编程领域最优秀专家的指导。因此,在理论学习阶段,毕业生将有一个在该行业具有丰富经验的优秀教学团队陪伴,而在现场实习期间,他/她将与真正的专家一起,这些专家是工作室团队的成员,将在工作室进行实践阶段的学习。

3. 进入一流的临床环境

TECH 对所有实习的工作室和公司都进行了严格的筛选。这样,学生就能在一流的专业环境中学习电子游戏编程。通过这种方式,你将能够亲眼目睹专业程序员的日常工作,以及为获得高质量职称所使用的技术和方法。

4. 将最好的理论与最先进的实践相结合

通过这一学位,TECH 适应了程序员的日常工作,因此采用了理论与实践相结合的方法,摆脱了长时间的学习,将重点放在关键概念上。因此,这个学术机构提供了一种创新的学习模式,使学生能够掌握必要的知识,在高质量视频游戏的编程方面处于领先地位。

5. 拓展知识的前沿领域

TECH提供了不仅在国内而且在国际中心进行这种实践项目的可能性。通过这种方式,专家将能够扩大他或她的领域,并赶上最好的专业人员,他们在一流的医院和不同的大陆执业。只有 TECH 这所全球最大的数字大学才能提供这样一个独特的机会。

> 你将在自己选择的中心
> 进行完全的实际沉浸"

03

目标

这个半面授校级硕士课程的设计将使学生获得必要的技能，以便在需要合格人才的竞争激烈的课程中取得进步。因此，这个教学的主要目标是确保学生利用本教学提供的所有工具，成为优秀的视频游戏开发人员。为此，本资格证书提供了最新的多媒体内容、补充读物和学习系统 Relearning、以重复内容为基础，有助于巩固概念。

想要推出一款多人游戏吗？学会将你的想法转化为实际的多人游戏编程语言"

总体目标

- 确保学生熟悉电子游戏所使用的不同编程语言和方法。为此，将深入研究学位的产生过程及其在不同阶段的整合。此外，电子游戏专业人员还将学习电子游戏设计的基础知识和主要理论知识。此外，在这个课程结束时，你将能够理解编程的作用，并开发网络游戏和多人视频游戏

> 这个程序将让你在职业上取得进展。通过这个半面授校级硕士课程，你将能够掌握任何数据结构和算法”

具体目标

模块 1. 编程的基础知识

- 了解计算机的基这个结构、软件和通用编程语言
- 分析计算机程序的基这个要素，如不同类型的数据、运算符、表达式、语句、I/O和控制语句
- 解释算法，这是软件开发的必要基础

模块 2. 数据结构和算法

- 学习算法设计的主要策略，以及算法计算的不同方法和措施
- 区分算法的功能、策略和主要已知问题中的使用实例
- 理解回溯法及其主要用途

模块 3. 面向对象的编程

- 了解面向对象问题的不同设计模式
- 理解软件开发中文档和测试的重要性
- 管理线程和同步的使用，以及解决并发编程中的常见问题

模块 4. 视频游戏机和设备

- 知道主要输入和输出外围设备的基这个操作
- 理解不同平台的主要设计含义
- 研究设备和系统的结构、组织、运作和相互联系
- 理解移动设备和视频游戏平台的操作系统和开发工具包的作用

模块 5. 软件工程

- 区分软件工程的基础知识，以及软件过程和其开发的不同模式，包括敏捷技术
- 认识需求工程，它们的开发、阐述、协商和验证，以了解与软件质量和项目管理有关的主要标准

模块 6. 视频游戏引擎

- 发现电子游戏引擎的功能和架构
- 了解现有游戏引擎的基这个特征
- 正确和有效地应用于视频游戏引擎的程序应用
- 选择最合适的范式和编程语言为游戏引擎应用编程

模块 7. 智能系统

- 建立与代理理论和代理架构有关的概念及其推理过程
- 吸收信息和知识概念背后的理论和实践，以及表现知识的不同方式
- 理解语义推理器、基于知识的系统和专家系统的工作原理

模块 8. 实时编程

- 分析实时编程语言区别于传统编程语言的主要特征
- 理解计算机系统的基这个概念
- 掌握应用实时编程的主要基础和技术的能力

模块 9. 网页游戏设计和开发

- 能够设计游戏和交互式网络应用程序，并编写相应的文档
- 评估游戏和交互式网络应用程序的主要功能，以便进行专业和正确的交流

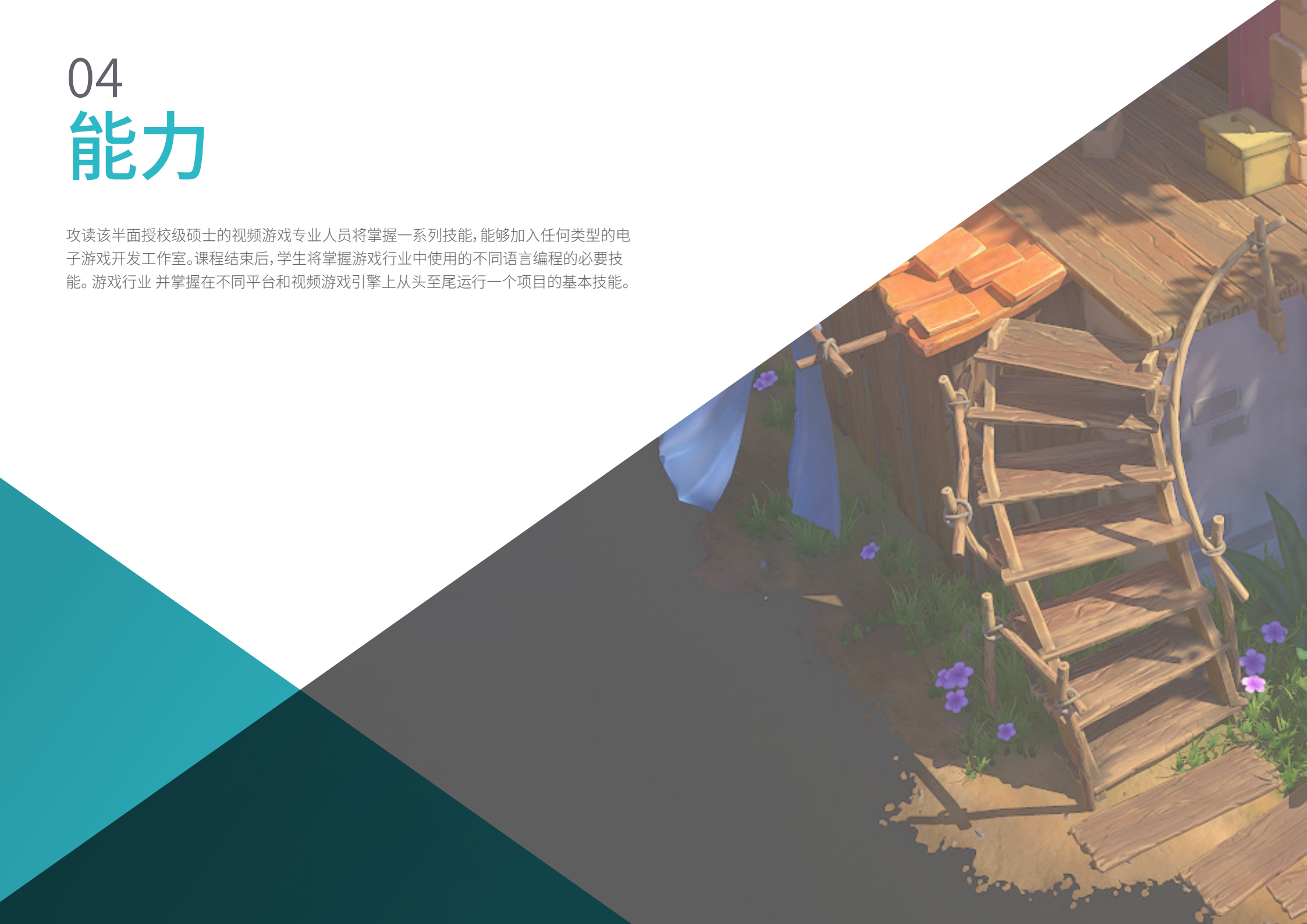
模块 10. 多人游戏网络和系统

- 描述传输控制协议/互联网协议（TCP/IP）架构和无线网络的基这个操作
- 分析应用于视频游戏的安全问题
- 掌握开发多人在线游戏的能力

04

能力

攻读该半面授校级硕士的视频游戏专业人员将掌握一系列技能，能够加入任何类型的电子游戏开发工作室。课程结束后，学生将掌握游戏行业中使用的不同语言编程的必要技能。游戏行业 并掌握在不同平台和视频游戏引擎上从头至尾运行一个项目的基本技能。

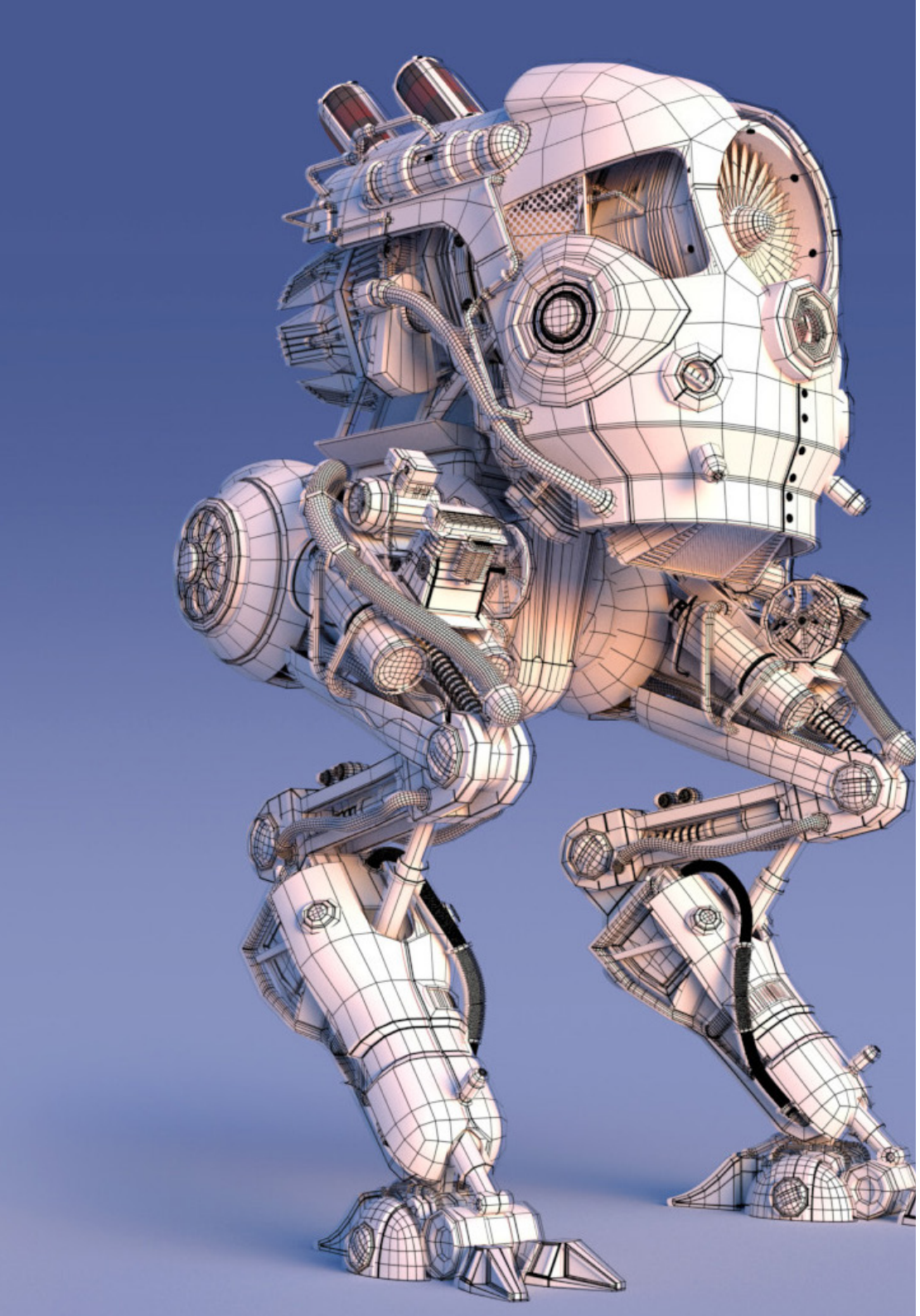


掌握为 Xbox One、PS4 和 Wii U-Switch 电子游戏编程的必要技能"

总体能力

- 设计一个视频游戏的所有阶段，从最初的想法到最后的推出
- 成为专业视频游戏程序员
- 深入研究开发的所有部分，从最初的架构，到玩家角色的编程以及游戏过程中涉及的所有元素
- 获得项目的整体愿景，能够为视频游戏设计中出现的不同问题和挑战提供解决方案

> 在编程之前，要了解玩家的体验，分析电子游戏的玩法。通过这个半面授校级硕士，你将学会如何取得成功"



具体能力

- 了解成为一名专业电子游戏开发者所需的软件
- 了解玩家的体验，知道如何分析视频游戏的可玩性
- 了解电子游戏编程过程的理论和实践程序
- 掌握对电子游戏世界最有用的编程语言
- 将所学的编程整合到不同类型的游戏机和平台上
- 为网络和多人电子游戏编程
- 吸收电子游戏引擎的概念，以便能够正确编程
- 将软件工程的知识应用于电子游戏编程

05

教学规划

这个电子游戏编程半面授校级硕士课程由 10 个专业模块组成，为学生提供完整而广泛的电子游戏开发内容。这个学位课程的教学人员将能够深入了解编程的基本原理，同时掌握最新的、以实践为导向的知识。这样，学生们就能接受高质量的教学，并能随时下载和查阅多媒体格式的创新内容。100% 在线理论培训让你可以根据自己的需要自由分配教学任务。

旨在为你提供最先进和最新的电子游戏编程内容的课程"

模块 1. 编程的基础知识

1.1. 程序设计入门
- 1.1.1. 计算机的基本结构
- 1.1.2. 软件
- 1.1.3. 编程语言
- 1.1.4. 计算机应用程序的生命周期

1.2. 算法设计
- 1.2.1. 问题的解决
- 1.2.2. 描述性技术
- 1.2.3. 算法的元素和结构

1.3. 程序的要素
- 1.3.1. C++语言的起源和特点
- 1.3.2. 开发环境
- 1.3.3. 程序概念
- 1.3.4. 基本数据类型
- 1.3.5. 运算符
- 1.3.6. 表达方式
- 1.3.7. 句子
- 1.3.8. 输入和输出数据

1.4. 控制语句
- 1.4.1. 句子
- 1.4.2. 分叉
- 1.4.3. 循环

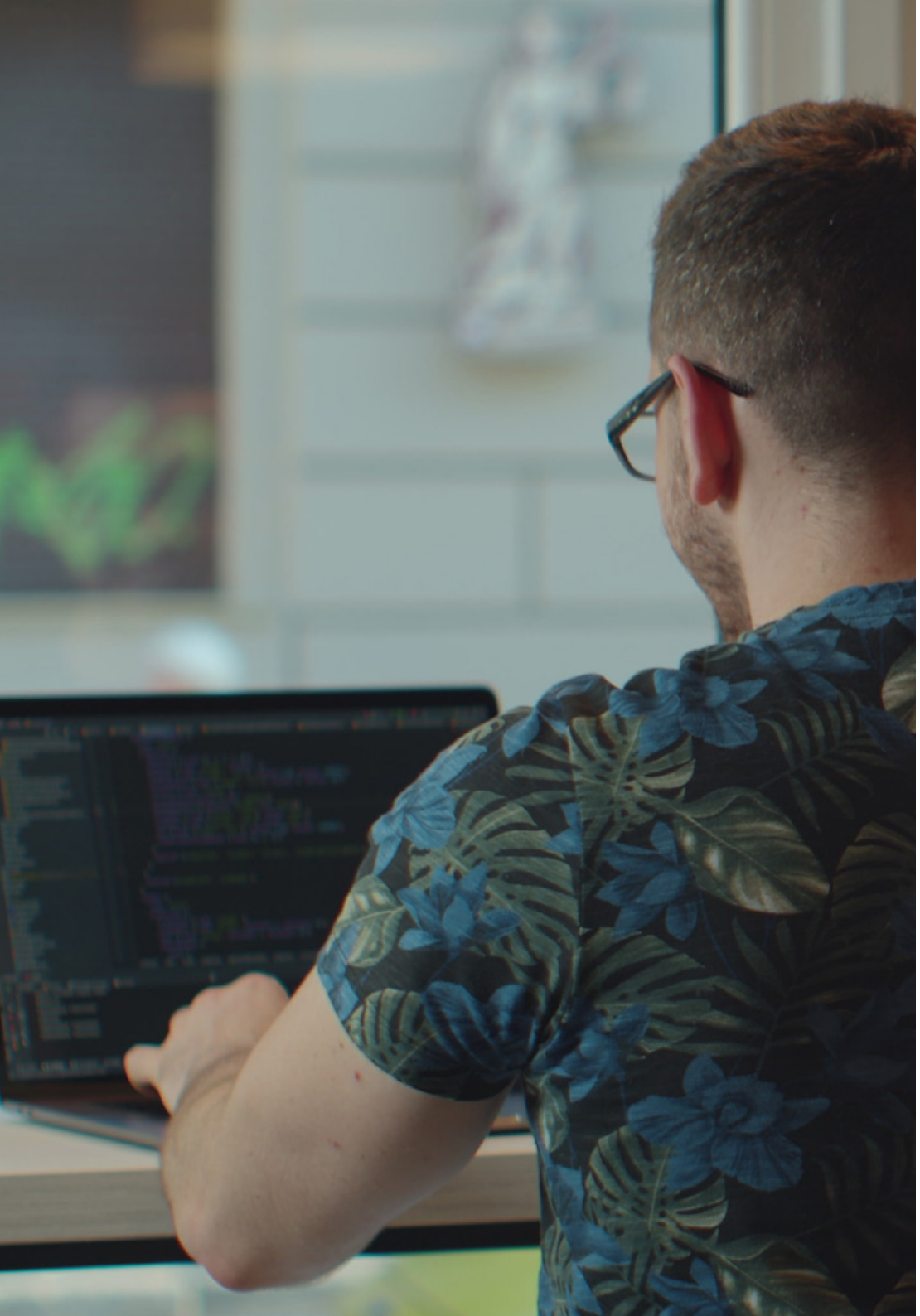
1.5. 抽象和模块化：函数
- 1.5.1. 模块化设计
- 1.5.2. 功能和效用的概念
- 1.5.3. 函数的定义
- 1.5.4. 函数调用的执行流程
- 1.5.5. 函数原型
- 1.5.6. 结果返回
- 1.5.7. 调用函数：参数
- 1.5.8. 通过引用和值传递参数
- 1.5.9. 标识符范围

1.6. 静态数据结构
- 1.6.1. Arrays
- 1.6.2. 阵列。多面体
- 1.6.3. 搜索和排序
- 1.6.4. 链字符串的 I/O 函数
- 1.6.5. 结构。連結
- 1.6.6. 新数据类型

1.7. 动态数据结构：指针
- 1.7.1. 概念指针定义
- 1.7.2. 运算符和指针操作
- 1.7.3. 指针Arrays
- 1.7.4. 指针 和数组
- 1.7.5. 指向字符串的指针
- 1.7.6. 结构体指针
- 1.7.7. 多重间接
- 1.7.8. 指向函数的指针
- 1.7.9. 将函数、结构和数组作为函数参数传递

1.8. 文件
- 1.8.1. 基这个概念
- 1.8.2. 文件操作
- 1.8.3. 文件类型
- 1.8.4. 文件的组织
- 1.8.5. C++ 文件简介
- 1.8.6. 文件管理

1.9. 递归
- 1.9.1. 递归的定义
- 1.9.2. 递归类型
- 1.9.3. 优点和缺点
- 1.9.4. 考虑因素
- 1.9.5. 递归-迭代转换
- 1.9.6. 递归栈



1.10. 测试和文档
- 1.10.1. 程序测试
- 1.10.2. 白盒测试
- 1.10.3. 黑盒测试
- 1.10.4. 测试工具
- 1.10.5. 程序文档

模块 2. 数据结构和算法

2.1. 算法设计策略简介
- 2.1.1. 递归
- 2.1.2. 分而治之
- 2.1.3. 其他策略

2.2. 算法的效率与分析
- 2.2.1. 效率测量
- 2.2.2. 测量输入的大小
- 2.2.3. 测量执行时间
- 2.2.4. 最坏情况、最好情况和平均情况
- 2.2.5. 渐近符号
- 2.2.6. 非递归算法的数学分析准则
- 2.2.7. 递归算法的数学分析
- 2.2.8. 算法的实证分析

2.3. 排序算法
- 2.3.1. 协调概念
- 2.3.2. 冒泡排序
- 2.3.3. 选择排序
- 2.3.4. 插入排序
- 2.3.5. 合并排序 (Merge_Sort)
- 2.3.6. 快速排序 (Quicksort)

2.4. 带树的算法
- 2.4.1. 树的概念
- 2.4.2. 二叉树
- 2.4.3. 树游览
- 2.4.4. 表示表达
- 2.4.5. 有序二叉树
- 2.4.6. 平衡二叉树

2.5. heaps算法
- 2.5.1. Heaps
- 2.5.2. 堆排序算法
- 2.5.3. 优先队列

2.6. 图形算法
- 2.6.1. 代表
- 2.6.2. 行程宽度
- 2.6.3. 行程深度
- 2.6.4. 拓扑排序

2.7. Greedy算法
- 2.7.1. Greedy策略
- 2.7.2. Greedy策略元素
- 2.7.3. 货币兑换
- 2.7.4. 旅人的问题
- 2.7.5. 背包问题

2.8. 搜索最小路径
- 2.8.1. 最短路径的问题
- 2.8.2. 负弧和循环
- 2.8.3. Dijkstra的算法

2.9. 图上的Greedy算法
- 2.9.1. 最小生成树
- 2.9.2. Prim 算法
- 2.9.3. Kruskal 算法
- 2.9.4. 复杂性分析

2.10. Backtracking
- 2.10.1. Backtracking
- 2.10.2. 替代技术

模块 3. 面向对象的编程

3.1. 面向对象编程简介
- 3.1.1. 面向对象编程简介
- 3.1.2. 分类的设计
- 3.1.3. 用于问题建模的 UML 简介

3.2. 分类之间的关系
- 3.2.1. 抽象和继承
- 3.2.2. 高级继承概念
- 3.2.3. 多态性
- 3.2.4. 组合和聚合

3.3. 面向对象问题的设计模式简介
- 3.3.1. 什么是设计模式?
- 3.3.2. 工厂模式
- 3.3.3. 单例模式
- 3.3.4. 观察者模式
- 3.3.5. 复合模式

3.4. 例外情况
- 3.4.1. 什么是例外?
- 3.4.2. 异常捕获和处理
- 3.4.3. 引发例外
- 3.4.4. 创建例外

3.5. 用户界面
- 3.5.1. Qt简介
- 3.5.2. 定位
- 3.5.3. 什么是事件?
- 3.5.4. 事件:定义和捕捉
- 3.5.5. 用户界面的发展

3.6. 并发编程简介
- 3.6.1. 并发编程简介
- 3.6.2. 进程和线程的概念
- 3.6.3. 进程或线程之间的交互
- 3.6.4. C++ 的线程
- 3.6.5. 并发编程的优缺点

3.7. 线程管理和同步
- 3.7.1. 线程的生命周期
- 3.7.2. 线程类型
- 3.7.3. 线程规划
- 3.7.4. 线程组
- 3.7.5. 守护线程
- 3.7.6. 同步
- 3.7.7. 锁定机制
- 3.7.8. 沟通机制
- 3.7.9. 监视器

3.8. 并发编程中的常见问题
- 3.8.1. 生产者消费者的问题
- 3.8.2. 读者和作者的问题
- 3.8.3. 哲学家的晚餐问题

3.9. 软件文档和测试
- 3.9.1. 为什么记录软件很重要?
- 3.9.2. 设计文件
- 3.9.3. 使用文档工具

3.10. 软件测试
- 3.10.1. 软件测试简介
- 3.10.2. 证据的类型
- 3.10.3. 单元测试
- 3.10.4. 集成测试
- 3.10.5. 验证测试
- 3.10.6. 系统测试

模块 4. 视频游戏机和设备

4.1. 电子游戏编程的历史
- 4.1.1. 雅达利时期(1977-1985年)
- 4.1.2. NES和SNES时期(1985-1995)
- 4.1.3. PlayStation / PlayStation 2时期(1995-2005)
- 4.1.4. Xbox 360、PS3和Wii时期(2005-2013)
- 4.1.5. Xbox One、PS4和Wii U - Switch时期(2013年至今)
- 4.1.6. 未来

4.2. 视频游戏中游戏性的历史
- 4.2.1. 简介
- 4.2.2. 社会背景
- 4.2.3. 结构图
- 4.2.4. 未来

4.3. 适应现代社会的发展
- 4.3.1. 基于运动的游戏
- 4.3.2. 虚拟现实技术
- 4.3.3. 扩增实境
- 4.3.4. 混合现实

4.4. 统一性ScriptingI和实例
- 4.4.1. 什么是script这个?
- 4.4.2. 我们的第一个script这个
- 4.4.3. 添加一个script这个
- 4.4.4. 打开一个script这个
- 4.4.5. 单行为
- 4.4.6. 调试

4.5. 统一性Scripting这个 II 和示例
- 4.5.1. 键盘和鼠标输入
- 4.5.2. Raycast
- 4.5.3. 实例化
- 4.5.4. 可变因素
- 4.5.5. 公共变量和序列化变量

4.6. 统一性Scripting III和实例
- 4.6.1. 获取组件
- 4.6.2. 修改组件
- 4.6.3. 测试
- 4.6.4. 多个对象
- 4.6.5. 对撞机和触发器
- 4.6.6. 四元数

4.7. 外围设备
- 4.7.1. 演变和分类
- 4.7.2. 外围设备和接口
- 4.7.3. 当前外设
- 4.7.4. 下一个未来

4.8. 电子游戏:未来的前景
- 4.8.1. 基于云的游戏
- 4.8.2. 缺少控制
- 4.8.3. 身临其境的现实
- 4.8.4. 其他选择

4.9. 建筑学
- 4.9.1. 电子游戏的特殊需求
- 4.9.2. 架构的演变
- 4.9.3. 当前架构
- 4.9.4. 架构之间的差异

4.10. 开发套件及其演变
- 4.10.1. 简介
- 4.10.2. 第三代开发套件
- 4.10.3. 第四代开发套件
- 4.10.4. 第五代开发套件
- 4.10.5. 第六代开发套件

模块 5. 软件工程

5.1. 软件工程与建模导论
- 5.1.1. 软件的性质
- 5.1.2. Webapps的独特性
- 5.1.3. 软件工程
- 5.1.4. 软件流程
- 5.1.5. 软件工程实践
- 5.1.6. 软件神话
- 5.1.7. 这一切是如何开始的?
- 5.1.8. 面向对象的概念
- 5.1.9. UML 简介

5.2. 软件流程
- 5.2.1. 整体流程模型
- 5.2.2. 规范的流程模型
- 5.2.3. 专业流程模型
- 5.2.4. 统一流程
- 5.2.5. 个人和团队流程模型
- 5.2.6. 什么是敏捷?
- 5.2.7. 什么是敏捷流程?
- 5.2.8. Scrum
- 5.2.9. 敏捷流程工具包

5.3. 指导软件工程实践的原则
- 5.3.1. 指导流程的原则
- 5.3.2. 指导实践的原则
- 5.3.3. 沟通原则
- 5.3.4. 规划原则
- 5.3.5. 建模原则
- 5.3.6. 施工原则
- 5.3.7. 部署原则

5.4. 了解需求
- 5.4.1. 需求工程
- 5.4.2. 建立基础
- 5.4.3. 需求调查
- 5.4.4. 用例开发
- 5.4.5. 需求模型的细化
- 5.4.6. 需求的协商
- 5.4.7. 验证要求

5.5. 需求建模:场景、信息和分析类
- 5.5.1. 需求分析
- 5.5.2. 基于场景的建模
- 5.5.3. 提供用例的 UML 模型
- 5.5.4. 数据建模概念
- 5.5.5. 基于类的建模
- 5.5.6. 类图

5.6. 需求建模:流程、行为和模式
- 5.6.1. 需求建模策略
- 5.6.2. 面向流的建模
- 5.6.3. 状态图
- 5.6.4. 建立行为模型
- 5.6.5. 序列图
- 5.6.6. 通讯图
- 5.6.7. 需求建模模式

5.7. 设计理念
- 5.7.1. 软件工程背景的设计
- 5.7.2. 设计过程
- 5.7.3. 设计理念
- 5.7.4. 面向对象的设计理念
- 5.7.5. 设计模型

5.8. 架构设计
- 5.8.1. 软件架构
- 5.8.2. 架构流派
- 5.8.3. 架构风格
- 5.8.4. 架构设计
- 5.8.5. 替代架构设计的演变
- 5.8.6. 使用数据流的架构映射

5.9. 组件级和基于模式的设计
- 5.9.1. 什么是组件?
- 5.9.2. 基于类的组件设计
- 5.9.3. 在组件级别执行设计
- 5.9.4. 传统组件设计
- 5.9.5. 基于组件的开发
- 5.9.6. 设计模式
- 5.9.7. 基于模式的软件设计
- 5.9.8. 架构模式
- 5.9.9. 组件级设计模式
- 5.9.10. 用户界面设计模式

5.10. 软件质量和项目管理
- 5.10.1. 质量
- 5.10.2. 软件质量
- 5.10.3. 软件质量困境
- 5.10.4. 实现软件质量
- 5.10.5. 软件质量保证
- 5.10.6. 行政范围
- 5.10.7. 工作人员
- 5.10.8. 产品
- 5.10.9. 这个过程
- 5.10.10. 项目
- 5.10.11. 原则和实践

模块 6. 视频游戏引擎

6.1. 视频游戏和信息通信技术
- 6.1.1. 简介
- 6.1.2. 机会
- 6.1.3. 挑战
- 6.1.4. 结论

6.2. 视频游戏引擎的历史
- 6.2.1. 简介
- 6.2.2. 雅达利时代
- 6.2.3. 80年代时代
- 6.2.4. 早期发动机90年代时代
- 6.2.5. 当前引擎

6.3. 视频游戏引擎
- 6.3.1. 发动机的类型
- 6.3.2. 视频游戏引擎的部件
- 6.3.3. 当前引擎
- 6.3.4. 为我们的项目选择一个引擎

6.4. 游戏制作引擎
- 6.4.1. 简介
- 6.4.2. 情景设计
- 6.4.3. 插件和动画
- 6.4.4. 碰撞
- 6.4.5. GML中的脚本

6.5. 虚幻引擎4。简介
- 6.5.1. 什么是虚幻引擎4?它的理念是什么?
- 6.5.2. 材料
- 6.5.3. 介面
- 6.5.4. 动画片
- 6.5.5. 粒子系统
- 6.5.6. 人工智能
- 6.5.7. FPS

6.6. 虚幻引擎4。可视化脚这个
- 6.6.1. 蓝图 和 可视化脚本理念
- 6.6.2. 调试
- 6.6.3. 变量的类型
- 6.6.4. 基这个流量控制

6.7. Unity 5引擎
- 6.7.1. 用C#和Visual Studio编程
- 6.7.2. 创建 Prefabs
- 6.7.3. 使用Gizmos来控制电子游戏
- 6.7.4. 自适应的引擎二维和三维

6.8. 戈多引擎
- 6.8.1. 戈多的设计理念
- 6.8.2. 面向对象的设计和组合
- 6.8.3. 全部包含在一个包中
- 6.8.4. 免费和社区驱动的软件

6.9. RPG Maker引擎
- 6.9.1. RPG Maker哲学
- 6.9.2. 以此作为参考
- 6.9.3. 创造一个有个性的游戏
- 6.9.4. 成功的商业游戏

6.10. Source 2引擎
- 6.10.1. Source 2 哲学
- 6.10.2. Source和Source 2：发展
- 6.10.3. 社区使用：视听内容和视频游戏
- 6.10.4. Source 2引擎的未来
- 6.10.5. 修改和成功的游戏

模块 7. 智能系统

7.1. 代理人理论
- 7.1.1. 概念的历史
- 7.1.2. 代理定义
- 7.1.3. 人工智能中的代理
- 7.1.4. 软件工程中的代理

7.2. 代理人架构
- 7.2.1. 代理的推理过程
- 7.2.2. 反应性代理
- 7.2.3. 演绎代理
- 7.2.4. 混合代理
- 7.2.5. 比较

7.3. 信息和知识
- 7.3.1. 数据、信息和知识之间的区别
- 7.3.2. 数据质量评估
- 7.3.3. 数据采集方法
- 7.3.4. 信息获取方式
- 7.3.5. 知识获取方式

7.4. 知识表述
- 7.4.1. 知识表示的重要性
- 7.4.2. 通过其角色定义知识表示
- 7.4.3. 知识表示的特征

7.5. 这个体论
- 7.5.1. 元数据介绍
- 7.5.2. 这个体论的哲学概念
- 7.5.3. 这个体论的计算概念
- 7.5.4. 领域这个体和更高层次的这个体
- 7.5.5. 如何建立一个这个体论?

7.6. 这个体语言和这个体构建软件
- 7.6.1. RDF、Turtle 和 N3 三元组
- 7.6.2. RDF模式
- 7.6.3. OWL
- 7.6.4. SPARQL
- 7.6.5. 简介用于创建这个体的不同工具
- 7.6.6. Protégé安装和使用

7.7. 语义网
- 7.7.1. 语义网的现状和未来
- 7.7.2. 语义网应用

7.8. 其他知识表示模式
- 7.8.1. 词汇
- 7.8.2. 全球视野
- 7.8.3. 分类法
- 7.8.4. 叙词表
- 7.8.5. 大众分类法
- 7.8.6. 比较
- 7.8.7. 思维导图

7.9. 知识表征的评估和整合
- 7.9.1. 零阶逻辑
- 7.9.2. 一阶逻辑
- 7.9.3. 描述性逻辑
- 7.9.4. 不同类型逻辑之间的关系
- 7.9.5. Prolog：基于一阶逻辑的编程

7.10. 语义推理器、基于知识的系统和专家系统
- 7.10.1. 推理概念
- 7.10.2. 推理机的应用
- 7.10.3. 基于知识的系统
- 7.10.4. MYCIN，专家系统的历史
- 7.10.5. 专家系统的元素和架构
- 7.10.6. 专家系统的创建

模块 8. 实时编程

8.1. 并发编程基础
- 8.1.1. 基本概念
- 8.1.2. 并发
- 8.1.3. 并发优势
- 8.1.4. 并发和硬件

8.2. Java 中支持并发的基这个结构
- 8.2.1. Java 中的并发
- 8.2.2. 线程创建
- 8.2.3. 方法
- 8.2.4. 同步

8.3. Threads,、生命周期、优先级、中断、状态、执行程序
- 8.3.1. 线程
- 8.3.2. 生命周期
- 8.3.3. 优先事项
- 8.3.4. 中断
- 8.3.5. 状况
- 8.3.6. 执行者

8.4. 互斥
- 8.4.1. 什么是互斥?
- 8.4.2. Dekker算法
- 8.4.3. 彼得森算法
- 8.4.4. Java中的互斥

8.5. 状态依赖
- 8.5.1. 依赖注入
- 8.5.2. 模式在 Java 中的实现
- 8.5.3. 注入依赖的方法
- 8.5.4. 例子

8.6. 设计模式
- 8.6.1. 简介
- 8.6.2. 创建模式
- 8.6.3. 结构模式
- 8.6.4. 行为模式

8.7. Java库的使用
- 8.7.1. Java 中的库是什么?
- 8.7.2. Mockito-all, mockito-core
- 8.7.3. Guava
- 8.7.4. Commons-io
- 8.7.5. Commons-lang, commons-lang3

8.8. shaders编程
- 8.8.1. 3D 和光栅管线
- 8.8.2. 顶点着色
- 8.8.3. 像素着色：照明 I
- 8.8.4. 像素着色：照明 II
- 8.8.5. 后期效果

8.9. 实时编程
- 8.9.1. 简介
- 8.9.2. 中断处理
- 8.9.3. 进程间的同步和通信
- 8.9.4. 实时规划系统

8.10. 实时规划
- 8.10.1. 概念
- 8.10.2. 实时系统的参考模型
- 8.10.3. 规划政策
- 8.10.4. 周期性计划者
- 8.10.5. 具有静态属性的调度器
- 8.10.6. 具有动态属性的调度器

模块 9. 网页游戏设计和开发

9.1. 网络的起源和标准
- 9.1.1. 互联网起源
- 9.1.2. 万维 World Wide Web
- 9.1.3. 网络标准的出现
- 9.1.4. 网络标准的兴起

9.2. HTTP 和客户端-服务器结构
- 9.2.1. 客户端-服务器角色
- 9.2.2. 客户端-服务器通信
- 9.2.3. 最近的历史
- 9.2.4. 集中计算

9.3. 网络编程：简介
- 9.3.1. 基这个概念
- 9.3.2. Web 服务器准备
- 9.3.3. HTML5 基础知识
- 9.3.4. HTML 表单

9.4. HTML 简介和示例
- 9.4.1. HTML5 历史
- 9.4.2. HTML5 元素
- 9.4.3. APIS
- 9.4.4. CCS3

9.5. 文件对象模板
- 9.5.1. 什么是文档对象模型?
- 9.5.2. 使用DOCTYPE
- 9.5.3. 验证 HTML 的重要性
- 9.5.4. 存取元素
- 9.5.5. 创建元素和文这个
- 9.5.6. 使用 innerHTML
- 9.5.7. 删除文这个元素或节点
- 9.5.8. 读写元素的属性
- 9.5.9. 操纵元素样式
- 9.5.10. 一次附加多个文件

9.6. CSS 简介和示例
- 9.6.1. CSS3 语法
- 9.6.2. 样式表
- 9.6.3. 标签
- 9.6.4. 选择器
- 9.6.5. 使用 CSS 进行网页设计

9.7. JavaScript 简介和示例
- 9.7.1. 什么是JavaScript?
- 9.7.2. 语言简史
- 9.7.3. JavaScript 版这个
- 9.7.4. 显示对话框
- 9.7.5. JavaScript 语法
- 9.7.6. 理解 Scripts
- 9.7.7. 空间
- 9.7.8. 注释
- 9.7.9. 功能
- 9.7.10. 页内和外部 JavaScript

9.8. JavaScript 中的函数
- 9.8.1. 函数说明
- 9.8.2. 函数表达式
- 9.8.3. 呼叫函数
- 9.8.4. 递归
- 9.8.5. 嵌套函数和闭包
- 9.8.6. 变量的保存
- 9.8.7. 多重嵌套函数
- 9.8.8. 命名冲突
- 9.8.9. 关闭
- 9.8.10. 功能参数

9.9. PlayCanvas开发网页游戏
- 9.9.1. 什么是 PlayCanvas?
- 9.9.2. 项目设置
- 9.9.3. 创建对象
- 9.9.4. 加入物理学
- 9.9.5. 添加模型
- 9.9.6. 更改重力和场景设置
- 9.9.7. 脚本
- 9.9.8. 摄像机控制

9.10. 用于开发网页游戏的Phaser
- 9.10.1. 什么是Phaser?
- 9.10.2. 加载资源
- 9.10.3. 建设世界
- 9.10.4. 平台
- 9.10.5. 玩家
- 9.10.6. 添加实物
- 9.10.7. 使用键盘
- 9.10.8. 拿起Pickups
- 9.10.9. 积分和得分
- 9.10.10. 反弹炸弹

模块 10. 多人游戏网络和系统

10.1. 多人视频游戏的历史和演变
- 10.1.1. 1970 年代：早期多人游戏
- 10.1.2. 90年代：Duke Nukem, Doom, Quake
- 10.1.3. 多人视频游戏的兴起
- 10.1.4. 本地和在线多人游戏
- 10.1.5. 派对游戏

10.2. 多人商业模式
- 10.2.1. 新兴商业模式的起源和运作
- 10.2.2. 在线销售服务
- 10.2.3. 免费玩
- 10.2.4. 小额支付
- 10.2.5. 广告
- 10.2.6. 按月付款订阅
- 10.2.7. 按场付费
- 10.2.8. 先试后买

10.3. 这个地游戏和网络游戏
- 10.3.1. 这个地游戏:开始
- 10.3.2. 派对游戏:任天堂与家庭团聚
- 10.3.3. 网络游戏:开端
- 10.3.4. 网络游戏的演变

10.4. OSI 模型:第一层
- 10.4.1. OSI模型:简介
- 10.4.2. 物理层
- 10.4.3. 数据链路层
- 10.4.4. 网络层

10.5. OSI 模型:第二层
- 10.5.1. 传输层
- 10.5.2. 会话层
- 10.5.3. 表示层
- 10.5.4. 应用层

10.6. 计算机网络和互联网
- 10.6.1. 什么是计算机网络?
- 10.6.2. 软件
- 10.6.3. 硬件
- 10.6.4. 服务器
- 10.6.5. 网络存储
- 10.6.6. 网络协议

10.7. 移动和无线网络
- 10.7.1. 移动网络
- 10.7.2. 无线网络
- 10.7.3. 移动网络运营
- 10.7.4. 数字技术

10.8. 安全
- 10.8.1. 人身安全
- 10.8.2. 电子游戏中的黑客 和 作弊器
- 10.8.3. 防作弊安全
- 10.8.4. 反作弊安全系统分析

10.9. 多人游戏系统:服务器
- 10.9.1. 服务器托管
- 10.9.2. MMO视频游戏
- 10.9.3. 专用游戏服务器
- 10.9.4. 局域网

10.10. 多人视频游戏设计和编程
- 10.10.1. 虚幻多人游戏设计基础
- 10.10.2. Unity 多人游戏设计基础
- 10.10.3. 如何让多人游戏变得有趣?
- 10.10.4. 超越命令:多人控制的创新

> 由于采用了以重复关键概念为基础的Relearning方法,你将能够减少长时间的学习"

06

实习

电子游戏编程专业人员在完成在线理论培训后，将在创意工作室和电子游戏开发商实习。通过这种方式，学生将能够直接接触到该行业的开发团队，并能够展示他们所学到的一切。在这一阶段的学习中，学生将有一名辅导员陪伴，直到课程结束。

在这个教学的实践阶段，与优秀的开发者一同坐下来，共同合作"

电子游戏编程课程的实习期为 3 周。周一至周五，学生将在创意工作室和视频游戏开发室上课，每天由一名专家进行连续 8 小时的实践教学。

在这一实践阶段，学生可以运用在混合式硕士学位课程中学到的所有编程知识。与公司的其他专业人员一起，你将能够与团队的其他成员协调，并学习主要的编程工具和软件。通过这种方式，你可以获得接近电子游戏行业现实的实习机会。

在该工作室学习期间，专业人员不仅可以使用常用的编程语言，还能设计多人视频游戏。越来越多的电子游戏令 玩家着迷。

实践部分将由学生积极参与，执行每个竞争领域的活动和程序（学会学习和学会实践），在教师和其他培训同学的陪伴和指导下，促进团队合作和多学科整合，作为游戏编程实践的横向能力（学会成为和学会与人相处）。

下文所述程序将构成培训实践部分的基础，其实施将取决于中心自身的可用性和工 作量，拟议的活动如下：

> 在一个能够为你展示作为游戏程序员潜力的工作室接受培训"

模块	实践活动	数量
创建数据结构和算法	采用 回溯技术，用于创建数据和算法	1
	参与分析提高效率的算法	
	执行输入尺寸和运行时间测量任务	
	用树、堆、图和贪婪创建排序算法 堆、图和 贪婪排序算法	
面向对象的编程	在创建对象时使用 工厂模式、单件模式 、观察者模式 和 复合 模式 来创建对象	1
	在创建对象时创建、捕捉和管理异常	
	执行并行编程	
	采用封锁和通信机制	
	创建软件文档和测试	
实时编程	创建和同步 线程	1
	程序 着色器	
	用 Java 实现该模式并使用 Java 库	
	创建后期项目	
	中断处理、同步和进程间通信	
网页游戏设计和开发	使用 HTML 表单进行网络编程	1
	使用 DOCTYPE 和 innerHTML 开发网页游戏	
	使用 PlayCanvas 开发网页游戏	
	建立网页游戏设计和开发项目	

责任保险

这个机构的主要关注点是保证受训者和公司实践培训过程中所需要的其他合作者的安全。为实现这一目标而采取的措施包括应对整个教学过程中可能发生的任何事件。

为此，这个教育实体承诺购买民事责任保险，以应对在工作经验中心逗留期间可能出现的任何意外情况。

这份针对受训者的责任保险应是全面的，并应在实践培训期开始前投保。这样，专业人员就不必担心会出现意外情况，而且在中心的实践课程结束前都有保障。

> 为此，大学将购买民事责任保险，以应对在工作实习中心逗留期间可能出现的任何意外情况”

实践培训的一般条件

该计划的实习协议的一般条件将如下:

1.辅导:在半面授校级硕士期间,学生将被分配到两名辅导员,他们将全程陪伴学生,解决可能出现的任何疑惑和问题。一方面,将有一位属于工作安置中心的专业导师,他将随时指导和支持学生。另一方面,也会有一名学术导师,其任务是在整个过程中协调和帮助学生,解决他们的疑惑,并为他们可能需要的东西提供便利。通过这种方式,专业人员将一直陪同,并能够咨询任何可能出现的疑问,包括实践和学术方面的疑问。

2. 时间:实习计划将有连续三周的实践培训时间,分布在每周五天,每天8小时。出勤的日子和时间表将由中心负责,并适当提前通知专业人员,提前足够的时间以方便其组织。

3.不出席:如果在半面授校级硕士程开始的当天没有出现,学生将失去同样的权利,没有报销或更改日期的可能性。在没有正当/医疗理由的情况下缺席超过两天,将导致学生辞去实习,因此,自动终止实习。在实习过程中可能出现的任何问题都必须及时和紧急地报告给学术导师。

4.证书:通过半面授校级硕士的学生将收到一份证书,认可他们在有关中心的逗留。

5.雇佣关系:半面授校级硕士不构成任何形式的雇佣关系。

6.以前的学习经历:一些中心可能要求提供以前的学习证明,以便参加半面授校级硕士。在这些情况下,有必要向TECH实习部出示该证明,以确认所选中心的分配。

7.不包括:半面授校级硕士不包括本条件中未描述的任何内容。因此,它不包括住宿、前往实习城市的交通、签证或任何其他未描述的服务。

然而,学生可以向他们的学术导师咨询这方面的任何疑问或建议。他/她将提供所有必要的信息以方便办理手续。

07

我在哪里可以进行临床实习?

这个半面授校级硕士课程包括在参考工作室的实习，学生将在那里把学到的电子游戏编程知识付诸实践。通过这种方式，TECH 为学生提供了机会，使该专业更贴近希望获得符合当前行业需求的优质教育的视频游戏专业人士。

通过将你的所有知识应用于游戏行业的一家领先研究机构，实现你的目标"

学生可以在以下中心参加该半面授校级硕士学位的实践部分课程:

剧本设计

Startreming Games

国家
阿根廷

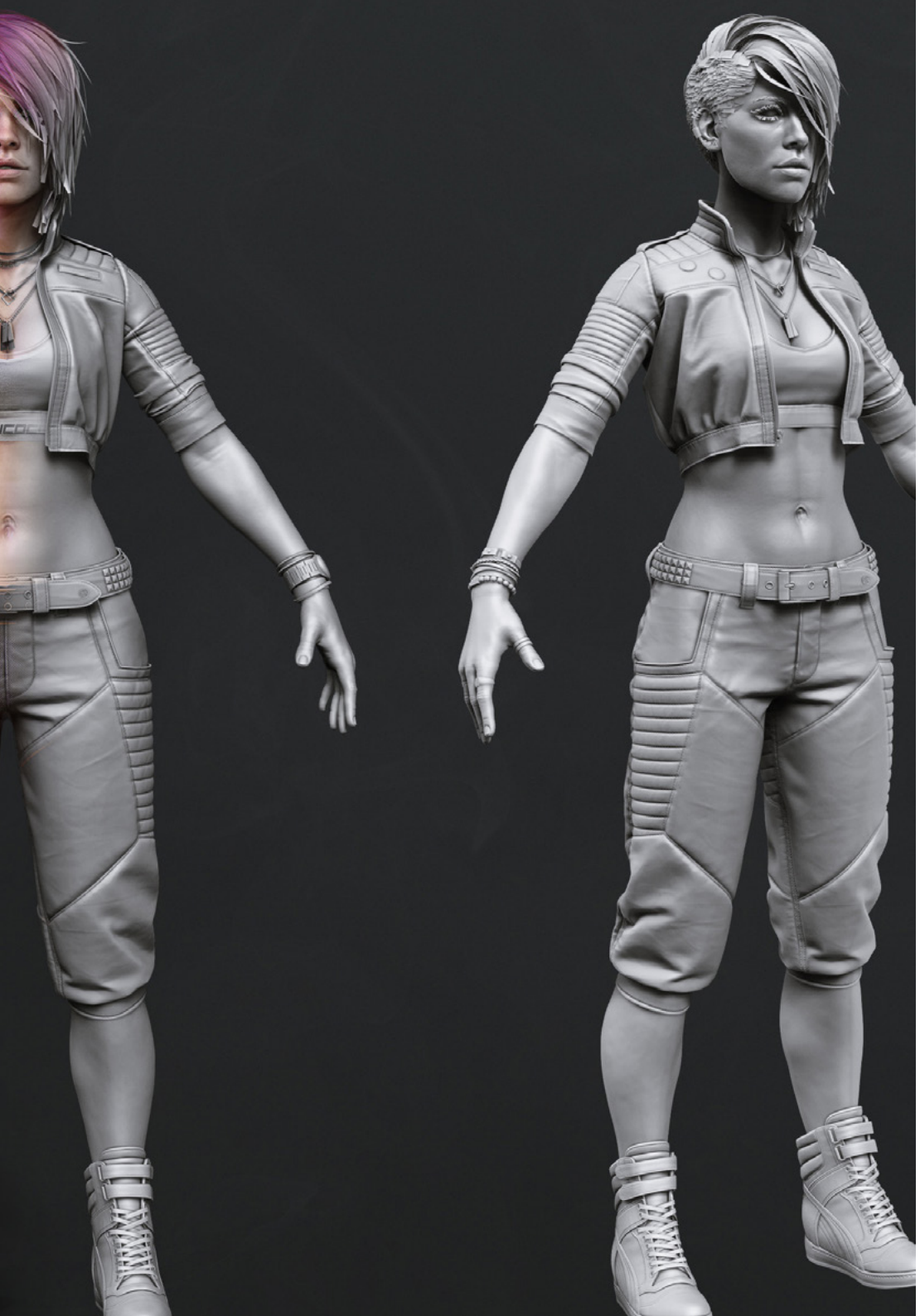
城市
门多萨

管理人员: Ruta 160 esquina Buenos Aires 88, San Rafael, Mendoza

独立的远程游戏开发工作室

相关实践项目:
-电子游戏编程
-3D硬表面建模

电子游戏编程



借此机会了解这个领域的最新发展，并将其应用到你的日常工作中"

08

方法

这个培训计划提供了一种不同的学习方式。我们的方法是通过循环的学习模式发展起来的: **Re-learning**。

这个教学系统被世界上一些最著名的医学院所采用，并被**新英格兰医学杂志**等权威出版物认为是最有效的教学系统之一。

发现 Re-learning，这个系统放弃了传统的线性学习，带你体验循环教学系统：这种学习方式已经证明了其巨大的有效性，尤其是在需要记忆的科目中”

案例研究，了解所有内容的背景

我们的方案提供了一种革命性的技能和知识发展方法。我们的目标是在一个不断变化，竞争激烈和高要求的环境中加强能力建设。

> 和TECH,你可以体验到一种正在动摇世界各地传统大学基础的学习方式”

你将进入一个以重复为基础的学习系统，在整个教学大纲中采用自然和渐进式教学。

学生将通过合作活动和真实案例，学习如何解决真实商业环境中的复杂情况。

一种创新并不同的学习方法

该技术课程是一个密集的教学计划，从零开始，提出了该领域在国内和国际上最苛刻的挑战和决定。由于这种方法，个人和职业成长得到了促进，向成功迈出了决定性的一步。案例法是构成这一内容的技术基础，确保遵循当前经济，社会和职业现实。

> 我们的课程使你准备好在不确定的环境中面对新的挑战，并取得事业上的成功”

在世界顶级商学院存在的时间里，案例法一直是最广泛使用的学习系统。1912年开发的案例法是为了让法律学生不仅在理论内容的基础上学习法律，案例法向他们展示真实的复杂情况，让他们就如何解决这些问题作出明智的决定和价值判断。1924年，它被确立为哈佛大学的一种标准教学方法。

在特定情况下，专业人士应该怎么做?这就是我们在案例法中面对的问题，这是一种以行动为导向的学习方法。在4年的时间里，你将面对多个真实案例。你必须整合你所有的知识，研究，论证和捍卫你的想法和决定。

Re-learning 方法

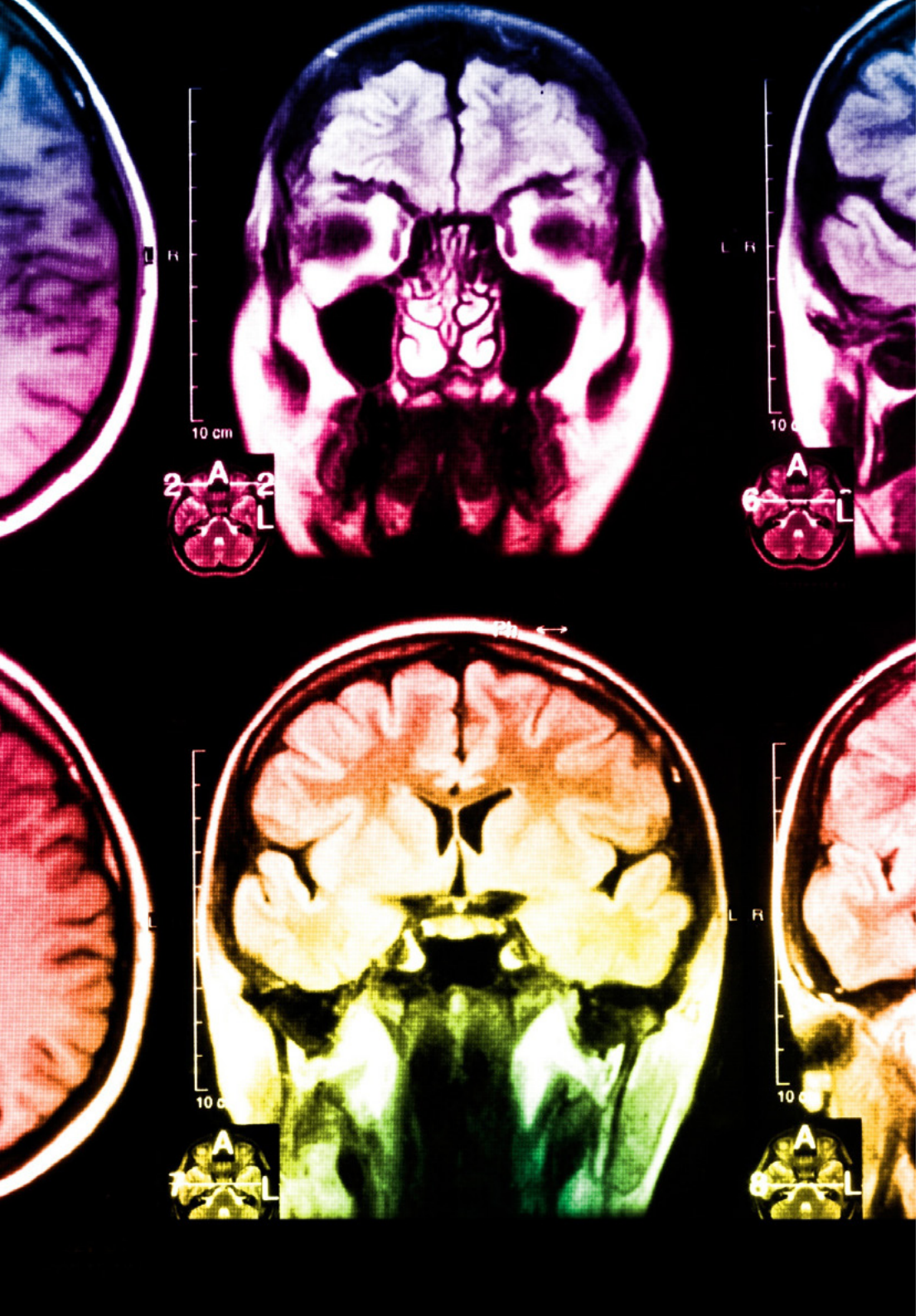
TECH有效地将案例研究方法与基于循环的100%在线学习系统相结合，在每节课中结合了8个不同的教学元素。

我们用最好的100%在线教学方法加强案例研究：Re-learning。

> 2019年，我们取得了世界上所有西班牙语网上大学中最好的学习成果。

在TECH，你将用一种旨在培训未来管理人员的尖端方法进行学习。这种处于世界教育学前沿的方法被称为 Re-learning。

我校是唯一获准使用这一成功方法的西班牙语大学。2019年，我们成功地提高了学生的整体满意度（教学质量，材料质量，课程结构，目标……），与西班牙语最佳在线大学的指标相匹配。



在我们的方案中，学习不是一个线性的过程，而是以螺旋式的方式发生（学习，解除学习，忘记和重新学习）。因此，我们将这些元素中的每一个都结合起来。这种方法已经培养了超过65万名大学毕业生，在生物化学，遗传学，外科，国际法，管理技能，体育科学，哲学，法律，工程，新闻，历史，金融市场和工具等不同领域取得了前所未有的成功。所有这些都是在一个高要求的环境中进行的，大学学生的社会经济状况很好，平均年龄为43.5岁。

Re-learning 将使你的学习事半功倍，表现更出色，使你更多地参与到训练中，培养批判精神，捍卫论点和对比意见:直接等同于成功。

从神经科学领域的最新科学证据来看，我们不仅知道如何组织信息，想法，图像y记忆，而且知道我们学到东西的地方和背景，这是我们记住它并将其储存在海马体的根本原因，并能将其保留在长期记忆中。

通过这种方式，在所谓的神经认知背景依赖的电子学习中，我们课程的不同元素与学员发展其专业实践的背景相联系。

该方案提供了最好的教育材料，为专业人士做了充分准备：

学习材料

所有的教学内容都是由教授该课程的专家专门为该课程创作的，因此，教学的发展是具体的。

然后，这些内容被应用于视听格式，创造了TECH在线工作方法。所有这些，都是用最新的技术，提供最高质量的材料，供学生使用。

大师课程

有科学证据表明第三方专家观察的有用性。

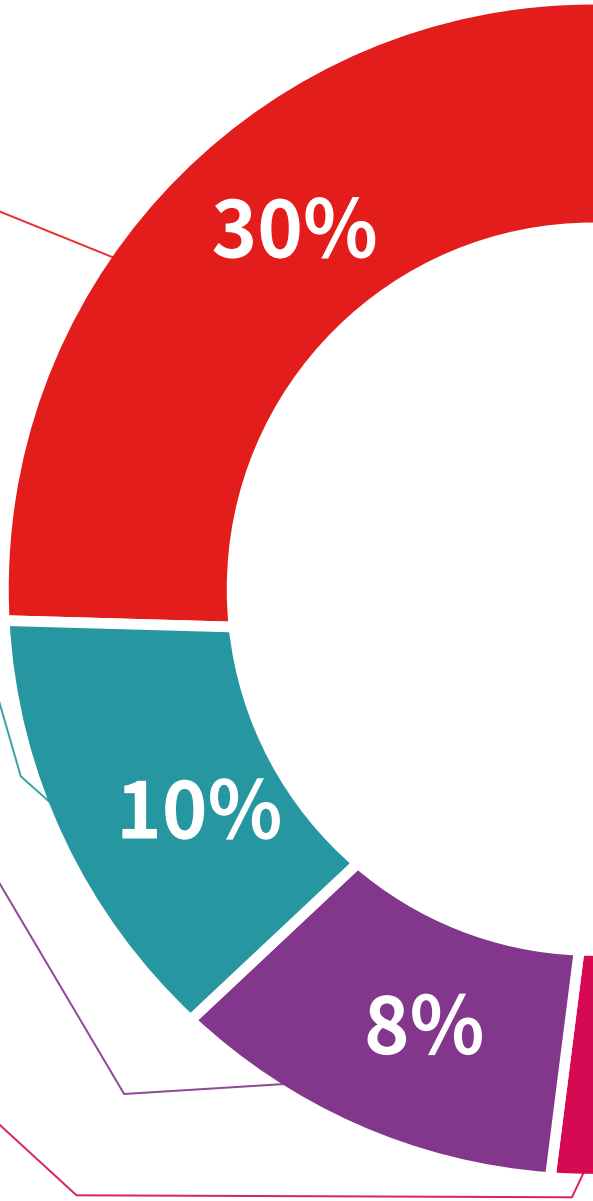
向专家学习可以加强知识和记忆，并为未来的困难决策建立信心。

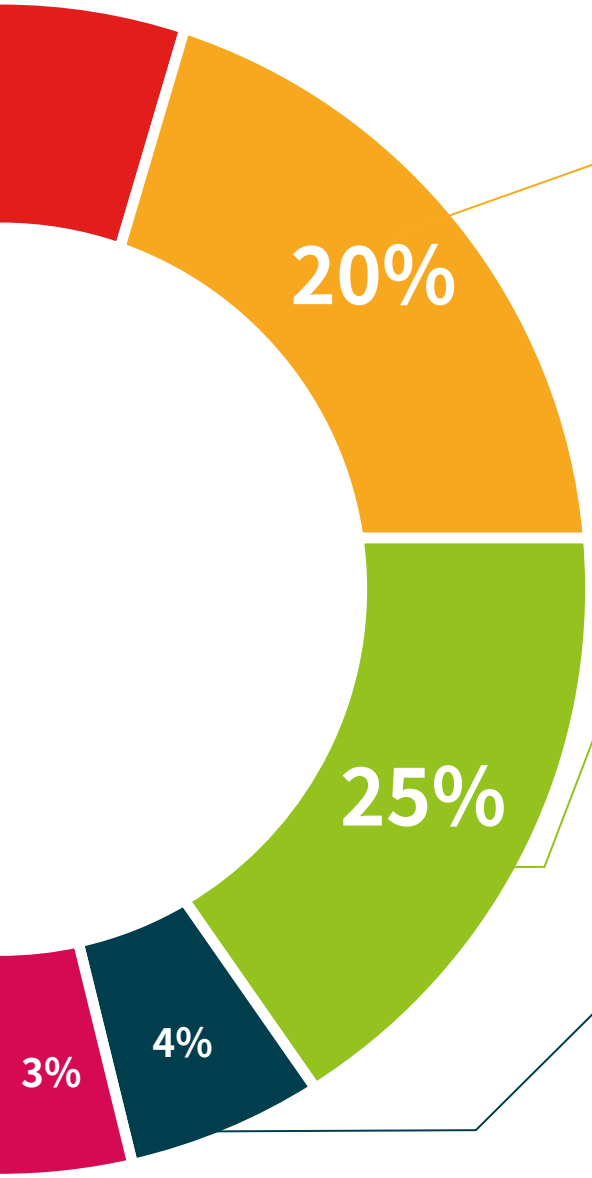
技能和能力的实践

你将开展活动以发展每个学科领域的具体能力和技能。在我们所处的全球化框架内，我们提供实践和氛围帮你取得成为专家所需的技能和能力。

延伸阅读

最近的文章，共识文件和国际准则等。在TECH的虚拟图书馆里，学生可以获得他们完成培训所需的一切。





案例研究

他们将完成专门为这个学位选择的最佳案例研究。由国际上最好的专家介绍，分析和辅导案例。

互动式总结

TECH团队以有吸引力和动态的方式将内容呈现在多媒体丸中，其中包括音频，视频，图像，图表和概念图，以强化知识。

这个用于展示多媒体内容的独特教育系统被微软授予“欧 洲成功案例”称号。

测试和循环测试

在整个课程中，通过评估和自我评估活动和练习，定期评估和重新评估学习者的知识:通过这种方式，学习者可以看到他/她是如何实现其目标的。

09

学位

电子游戏编程半面授校级硕士除了保证最严格和最新的培训外，还可以获得由TECH科技大学颁发的半面授校级硕士学位证书。

无需旅行或繁琐的程序，即可成功通过此课程并获得大学学位”

这个**电子游戏编程半面授校级硕士**包含了市场上最完整和最新的课程。

评估通过后，学生将通过邮寄收到**TECH科技大学**颁发的相应的**半面授校级硕士**学位。

TECH科技大学颁发的证书将表达在半面授校级硕士获得的资格，并将满足工作交流，竞争性考试和专业职业评估委员会的普遍要求。

学位：**电子游戏编程半面授校级硕士**

模式：**在线**

时长：**12个月**


tech 科学技术大学

授予

文凭

_____先生/女士，身份证号码_____

顺利通过并获得认证的课程是

半面授校级硕士

电子游戏编程

这是一个由本大学授予的学位，相当于1,620个小时，
开始日期是 dd/mm/aaaa,结束日期是dd/mm/aaaa。

截至2018年6月28日，TECH是一所被公共教育部认可的私立高等教育机构。

2020年六月17日

Tere Guevara Navarro女士
校长

Unique TECH Code: AFWORD23S techtitute.com/certificates


电子游戏编程半面授校级硕士

科目类型	小时
必修(OB)	1,500
选修 (OP)	0
外部实习 (PR)	120
硕士论文 (TFM)	0
	总计1,620

教学大纲概况

学期	学科	小时	性质
1º	编程的基础知识	150	必修
1º	数据结构和算法	150	必修
1º	面向对象的编程	150	必修
1º	视频游戏机和设备	150	必修
1º	软件工程	150	必修
1º	视频游戏引擎	150	必修
1º	智能系统	150	必修
1º	实时编程	150	必修
1º	网页游戏设计和开发	150	必修
1º	多人游戏网络和系统	150	必修

Tere Guevara Navarro女士
校长

tech 科学技术大学

*海牙加注。如果学生要求为他们的纸质资格证书提供海牙加注，TECH EDUCATION将采取必要的措施来获得，但需要额外的费用。

tech 科学技术大学

半面授校级硕士
电子游戏编程

模式：混合式（在线+临床实践）
时间：12个月
学位：TECH 科技大学

半面授校级硕士
电子游戏编程

tech 科学技术大学