【声明场景 走廊】

* 背景：hallway

【声明场景：房间】

* 背景：room

【声明场景：归路】

* 背景：pathway

【声明角色：苏语涵】

* 立绘：
  + 正常：
    - 平静：语涵正常平静
    - 微笑：语涵正常微笑
    - 尴尬：语涵正常尴尬
    - 无奈：语涵正常无奈
    - 严肃：语涵正常严肃
  + 解释：
    - 平静：语涵解释平静
    - 微笑：语涵解释微笑
    - 尴尬：语涵解释尴尬
* 立绘位置设置：立绘默认方案（屏底高度=0.5，顶部高度=0.75）
  + 中间：横向位置=0.5
* 头像：
  + 平静：占位
* 发言
  + 名字颜色：#00FFFF
  + 内容颜色：#00FFFF

【声明角色：苏语泉】# 注：苏语泉的所有立绘都还没开始做，所以所有表情等都会被忽略

* 发言：
  + 名字颜色：#007FFF
  + 内容颜色：#007FFF

【声明别名：我，目标=苏语泉】

【声明别名：语涵，目标=苏语涵】

【章节：主章】

【设置背景音乐：”Gymnopedie No 1”】

【切换场景：走廊】

介绍会等会就开始了，现在去和老姐汇合吧。

应该是旁边这间吧。

空空荡荡的小教室，就是这里。

【切换场景：房间】

【角色入场：苏语涵】

语涵（微笑）：你来了啊。

我（微笑）：嗯，录像器材我也搬来了。

如果介绍会没有人来，我们的B计划是把介绍录制下来传到网上。为此，像三脚架这样的装备还是带来比较好。

语涵（平静）：嗯。希望预感不要应验。

我（平静）：介绍会的消息，到现在还是零转发？

语涵（尴尬）：至少有人点赞了。。

我（叹气）：。。。

语涵（微笑）：往好的方面想，没人来的话我们也省下了去租大教室的钱，这可是一周的饭钱。

我：道理我都懂，不过。。

语涵（平静）：好了，到此为止。

语涵（微笑）：我再记一下文稿，你出去看看，拉拉客吧。

我（平静）：好。

。。。

我：姐，预定的开始时间已经过了十分钟了，用B计划吧。

语涵（无奈）：好的。。

【长发言：苏语涵（微笑）】

你们好，我是语涵计划执行组的负责人，苏语涵，这里是语涵计划的介绍会。

今天我们将介绍我们组仍在研发中的试验性的视觉小说辅助开发程序：语涵编译器。虽然该程序离正式完成仍然有待时日，但我们希望能够吸引任何对此感兴趣的潜在合作伙伴的注意，帮助我们及早发现问题、改正问题，甚至帮助我们更好或更早地完成该项目。

【#注：这里的时长应在游戏内设置不一样的字体颜色来提醒玩家，等游戏全部完成后再更新这里的时间】

【切换角色状态：解释,微笑】

在介绍会正式开始前，我再简单介绍一下流程。首先，我将详细介绍今天的主角，语涵编译器。这是一个辅助视觉小说游戏开发的工具，它的介绍将首先包含其功能、使用场景，还有适用语涵编译器时的视觉小说制作流程。之后，我将简单描述关于语涵编译器未来的开发计划。

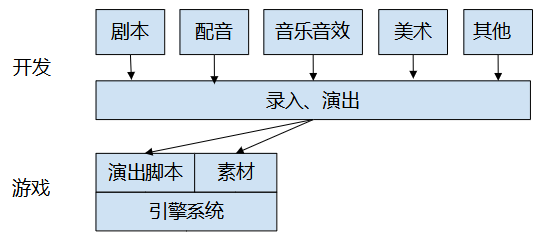
【切换角色状态：微笑】

最后将会有一个问答环节，我们将一一解答各位与会者的问题。

那我们现在开始。

【切换角色状态：解释】

为了更好地理解语涵编译器的使用场景和设计理念，我们先来回顾一下视觉小说的大致制作过程。

图1：视觉小说的大致制作过程与游戏框架

这里是一张简略的关于制作过程的示意图，我们先从这张图的下方开始。首先，视觉小说制作的最终产品是游戏本身，这是玩家实际获取的部分。由于视觉小说的玩法以及交互设计的相对固定，或者说成熟，每部视觉小说都有一部分很少需要改动、可在不同游戏间通用的程序框架，我们称之为游戏**引擎**。比如RenPy，面包引擎等。除了这部分通用的程序框架外，我们把每个视觉小说独有的部分归为两类：一类是**演出脚本**，包含游戏内的大多数文本内容和对美术、音乐等资源的引用；另一类是**素材**，包含如背景音乐、音效、语音、视频、特效等内容。如果把游戏引擎视为一个解释器的话，那么演出脚本可以对应程序部分，而素材就是数据部分。

【切换角色状态：平静】

虽然游戏只需要开发者提供演出脚本和素材，但开发者一般不会直接基于他们的格式来进行开发。这是因为，首先，编写演出脚本需要开发者对游戏的所有细节都有确定的计划，比如在场的每名角色应该处在什么位置、有怎样的表情，每个人会说什么。这在游戏开发初期，连情节都没有完全确定的情形下是无法确定的。另外，演出脚本和素材都需要符合引擎系统的要求。比如每个引擎对演出脚本都有自己的格式要求，或者说语法，这使得对演出脚本的修改实际上需要编程的能力和思维开销；而素材可能需要转换为特定的格式，图片需要有特定的大小，这些来自引擎的要求使得对它们的准备和修改都需要更高的代价。所以一般情况下，开发者需要提前准备所有内容，争取在完成演出脚本和素材之后尽可能少地修改它们。

这张图的上半段，标注着“开发”的这一片描述的是视觉小说开发者的一般流程。一般来说，视觉小说有以下几个部分需要分别准备。首先是剧本，剧本包含了视觉小说故事的所有情节和人物设定等信息，是“视觉小说”名字中的“小说”部分。其次，视觉小说的美术部分包含了人物的立绘、背景图、剧情原画等，对应“视觉小说”名字中的“视觉”部分。另外，音频资源也是不可或缺的。背景音乐可以渲染情绪、烘托氛围，人物配音可以补全人物性格、传递情绪和状态，特殊音效可以吸引玩家注意、丰富演出，等等。最后，部分视觉小说可能还包含其他类型的资源，包括特殊脚本、插件，或是小游戏那样丰富玩法的部分。由于各个视觉小说制作的情况都不同，这些资源的准备顺序没有必然的关系，比如大多数游戏需要先有剧本再准备其他资源，而二次创作可能会在已有人物立绘、音乐等资源的情况下补足剧本。不过不论这些素材的准备顺序如何，将他们转化为引擎可以用的演出脚本和素材都需要录入和演出这两个步骤，我们在这里把它们简化为一个环节。综上所述，理想情况下视觉小说制作的一般流程是，先筹备各种素材，再进行录入和演出，最后使用引擎生成完整的游戏，一气呵成。

【切换角色状态：正常，严肃】【收起图片：大致制作过程与游戏框架】#这里应该能够匹配到图1

但是，理想是美好的，现实是残酷的，这样的理想流程很难在现实世界重现，各种因素都有可能导致额外的波折。比如，经验不足的制作组有可能在剧本不够完善的情况下，开始准备人物立绘，结果在演出阶段，剧本需要的立绘、差分需要再补，原来准备的立绘却用不上。又比如，在游戏基本成型，开始测试时发现各种问题需要修改剧本和演出，等等。每一点这样的浪费、返工，造成的后果都将由制作者和玩家承担。最糟糕的是，在准备剧本的阶段对其他资源需求预估的不足，有限的制作预算无法满足雄心勃勃的剧本，项目成为一个吞噬资源的无底洞，最后半路夭折。

除了容易产生以上问题之外，以上流程漫长的周期也容易导致不必要的问题和困难。对于没有商业化运营、纯粹靠热爱来驱动的制作组来说，人力财力的局限导致项目经常需要以年为单位来进行。这样漫长的时间对于经验不足、生活不稳定的制作组成员来说是巨大的煎熬。因为成品的游戏直到最后才会成型、可玩，所以在那之前，制作者的心血基本没有正向的反馈，并且随时可能需要返工。当制作组成员因为各种原因退出开发时，这可能会引起连锁的损失及质量缺陷。比如画师退出的话，为了保持绘画风格一致，所有由该画师负责的美术可能需要重新设计、绘制。如果剧本的原作者退出，接替的作者对角色以及故事的理解出现偏差或不一致，则有可能导致故事出现逻辑错误或前后不一致，给作品留下难以挽回的遗憾。如果制作组成员接连退出，整个项目很可能就悄然停止。

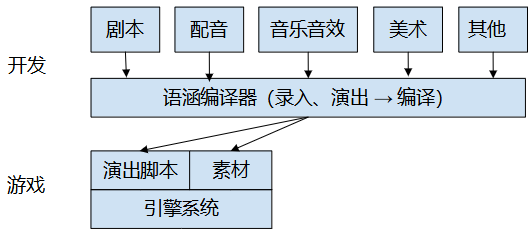
【#以上就是背景部分，接下来介绍语涵编译器】

【切换角色状态：平静】

以上是我们总结的，目前对视觉小说游戏制作的一般流程和可能的问题。我们现在将展示的语涵编译器的主要目标是努力缩短从创作到游戏生成的周期，以减少上述提到的问题。

【切换角色状态：微笑】

设想一下，如果在剧本成型的第一天就能生成一个可运行的游戏原型，所有新的素材刚准备完成就能在游戏中实机验证，相信制作组就能够更有成就感、更有动力地继续开发创作，同时也有更多的时间及早发现问题并进行修复。更进一步，如果制作组采取更合适的流程，在剧本完成时能够更准确地了解所需的美术音乐资源，那么由错误预估资源需求导致的浪费和返工就能大量减少。

图2：（有语涵编译器下的）制作流程与框架

简而言之，语涵编译器是一个将上图中的录入、演出步骤自动化的程序。在制作组成员用文档进行剧本创作、用表格来记录素材信息时，如果文档与表格符合一定的格式，语涵编译器将能够读取这些文件，直接生成所选引擎的演出剧本和素材。生成的脚本、素材等都可以以游戏引擎工程的形式输出，方便制作者直接生成可运行的游戏。如果您有一定的编程基础的话，换句话说，如果文档、表格形式的剧本和素材信息等内容满足语涵编译器的**语法**，语涵编译器就可以将它们**编译**为所支持的游戏引擎的脚本、素材，或是工程。

【切换角色状态：平静】

在之后的介绍中，“编译”一词指代使用语涵编译器从脚本等资源生成游戏工程的过程。目前语涵编译器只支持RenPy引擎，我们将在未来添加对更多引擎的支持。

【#等实际准备完结的时候再更新支持引擎的列表】

【收起图片：“（有语涵编译器下的）制作流程与框架”】

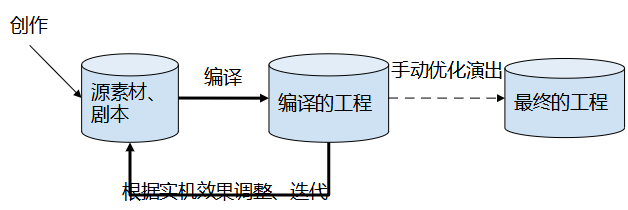
有人可能会觉得，既然剧本文档也需要符合语涵编译器的格式，那么这似乎与直接使用演出脚本进行剧本创作没什么区别。

【切换角色状态：微笑】

我们接下来就通过对这个误区的解答来体现语涵编译器的独特之处。游戏引擎的演出脚本本质上是最终的程序，引擎能且只能按照脚本中的指令执行，所以演出脚本需要补完游戏中的所有细节，包括转场效果、素材标识名这种不会在剧本创作阶段就定下的内容。而语涵编译器读入的剧本文档不需要这么多细节，更像是一个需求文档，除了剧本中明确指定的内容外，语涵编译器将使用内置的算法来填充所有没有给出的细节，拥有较高的自由度。与此同时，内置的算法也可以随语涵编译器的更新而改进。我们在继续开发语涵编译器时也会努力使得对文档与表格的格式尽可能宽松、减少限制，使其贴近常规剧本的形式，并且努力提高自动演出的质量。

【切换角色状态：平静】

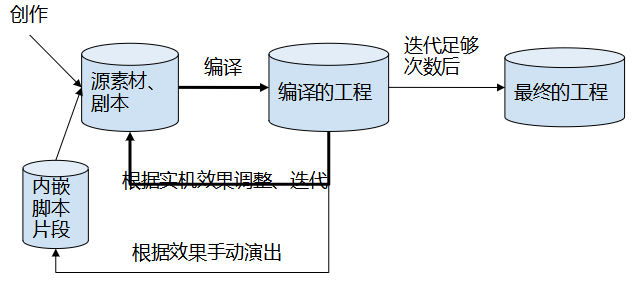
为了更好地使用语涵编译器，我们推荐对视觉小说的制作过程进行一部分变更，来最大化使用语涵编译器的收益。

图3：基础版管线

当语涵编译器不够成熟时，我们预计自动演出的实际效果与制作者想要的理想效果仍有较大差距。在这种情况下，制作者仍然可以使用语涵编译器提供的快速迭代来更好更精准地准备各类素材。如图所示，在这种情况下，首次创作后完成的内容，即使和最终游戏内容相比只有一小部分，也可以编译为游戏工程进行实机测试，并以此指导后续的修改与继续开发。等到所有素材和剧本故事完成，制作者可以以编译出的游戏工程为基础，手动修改不够好的部分。手动优化演出完成后，所有的流程与不使用语涵编译器时的流程一致。

【切换角色状态：微笑】

当语涵编译器足够成熟，自动演出的大部分内容能够满足成品质量要求时，可能需要手动修改的部分会变得很少，此时我们推荐如图的流程：

图4：进阶版管线

【收起图片：基础版管线】

如果需要手动演出的部分不多，制作者可以选择在剧本、素材中内嵌对应游戏引擎的脚本，语涵编译器将完整地把这部分脚本复制到编译出的工程中。在这样的流程中，语涵编译器全程参与视觉小说游戏的开发，制作者可以根据实机效果反复迭代至成品质量。如果您有编程基础的话，可以这样理解，语涵编译器将剧本视作像C++这样高级编程语言，内嵌游戏引擎脚本的行为可以对应于在C++源代码中内嵌X86等架构的汇编指令。

【收起图片：进阶版管线】

以上是语涵编译器现在的基本功能和用法，除此之外，语涵编译器的设计决定了它可以不止做这些。由于在流程上它处在原始剧本和游戏引擎工程之间，我们将持续努力，将它拓展为一个视觉小说制作的辅助**体系**。换句话说，我们将在现有功能的基础上添加更多被游戏制作者需要的、与具体引擎选择无关的功能。首先是**需求整理**，制作者可以在编写中的剧本里标注一些资源需求，这里需要一个怎样的背景，某个角色需要怎样的立绘动作、表情，等等。语涵编译器在读取剧本时便可以首先在游戏中自动生成用于占位的素材，保证游戏可运行；其次可以生成关于所有未被满足的资源需求的报告，制作者可以根据这个报告去准备其他素材，减少素材方面的浪费与返工。制作者也可以生成关于游戏内已有资源的报告（如已经准备了哪些立绘、表情差分等），方便之后的创作与演出。其次是各类**分析、检查**，例如语涵编译器未来可以帮忙检查中文语法，检查是否有剧本编写错误会导致剧情卡死。还有就是**优化**功能，比如立绘分层导出以缩小游戏体积，不同声音的音量平衡、裁剪首尾空白，或者可以与人工智能结合，自动提升图片分辨率，等等。

【切换角色状态：平静】

除了方便视觉小说制作者外，我们还将努力使语涵编译器项目具有更深的意义。首先，在语涵编译器内部设计稳定、游戏引擎支持足够丰富后，语涵编译器可以让制作者对游戏引擎的选择有更大的自由度。制作者可以自由选择最适合的引擎，甚至同时尝试不同的引擎。会编程的制作者也可以将视觉小说内容整合进其他游戏类型，不必拘泥于已有的视觉小说引擎。用编程的语言来说，使用编译器可以提高游戏的可移植性。其次，如果有人需要自研引擎，可以通过修改语涵编译器、添加对自研引擎的支持来快速将已存在的游戏移植到新引擎中进行测试。如果新引擎不需要用户手动演出、主要依靠编译器，则引擎的设计可以进一步简化，把更多任务转移到语涵编译器中完成。这样可以缩短引擎的开发时间，或者改善引擎的重点指标，比如可以做得更小或者更省资源，等等。除了编译器与引擎的交互外，我们相信语涵编译器的存在也能更好地服务于制作组内或是与合作方之间的合作。比如对于提升制作组内的合作，由于语涵编译器实现了从素材到游戏实机的自动化，绕开了人工的录入步骤，因而在合理的项目管理下，所有成员都能即时看到自己与其他人的改动，都能以实机为标准进行讨论。当语涵编译器的体系足够成熟后，除素材制作所需的专业软件外，所有视觉小说制作所需的软件都可以移至云服务器，制作组成员可以更好地用多平台、多系统进行开发。

。。。

接下来是提问时间，语泉你把我们准备的问题都问一遍吧。

【切换角色状态：微笑】#默认发言者还是语涵，所以这个应该对语涵起效

【选项：名称=提问，结束动作=循环】

* 为什么不另做一个引擎或是拓展一个已有引擎，而是采用编译器的形式？
  + 首先，我们所设想的功能以编译器的形式呈现是最合适的，现有引擎都能用上，用户不需要舍弃旧知识。
  + 其次，我们设想的编译器是一个可以绕过的阶段。
  + 如果出于各种原因，用户中途决定不使用我们的程序，用户一定可以把带语涵编译器标注的剧本当成普通的剧本继续开发，不会浪费努力。
* 为什么在现在已经有可运行的演示的情况下仍然将该程序归类为实验性质？还有哪些工作没有完成？
  + 首先，语涵编译器目前所支持的命令还无法使用部分常见的视觉小说引擎的功能，比如使用变量、条件判断、设置图案位置、转场等。
  + 其次，现阶段我们仍然没有足够的用户文档和足够易用的接口，辅助调试的功能都还没有做。使用好该编译器仍然需要较强的编程技能，用户很难在没有我们帮助的情况下运行。
  + 还有就是，编译器内部一些设计决定仍然需要评估，部分核心代码可能仍然需要迭代、大改。
  + 我们需要在更多人用之前把这些已知的问题修好。
* 没有其他问题了
  + 【跳出循环】

【默认发言模式】

语涵（微笑）：完成！这样就好了。

我：好的，等会回去之后就把视频传到网上吧。

语涵：没问题。

【切换场景：归路】

收拾完东西离开教学楼时，天色已经完全暗淡下来。

我（叹气）：唉，我现在有点担心，万一我们辛辛苦苦把这个编译器做完了，但最后没有一个人用，那怎么办。

语涵（尴尬）：我们的论文不都是发表了之后就什么也没发生么，应该说有人用才是中大奖了。

我：。。。太真实了。

语涵（微笑）：好了好了，别想这么多了。今天到此为止，明天继续加油！

我（微笑）：加油！