**新生态UGC AI游戏生成系统设计与实现**

**摘要**

游戏本质上是一种艺术与科技融合的产物，技术的应用与发展在发展在很大程度上决定着游戏的制作和效率，本文将详细概述新生态UGC AI游戏生成系统设计与实现,将视频制作简单划分为三个阶段，实现comfyui与unity接口，unity主界面制作流程，ai场景及美工布局，重点分析AIGC对游戏产业与动画产业的革命性影响，快速模拟2d游戏的生成，虽在实践应用中有诸多局限，但在整体创作过程中，整体游戏框架的搭建及流程实现是本次创作实践核心目标。

关键词：AIGC；comfyui；unity；游戏；

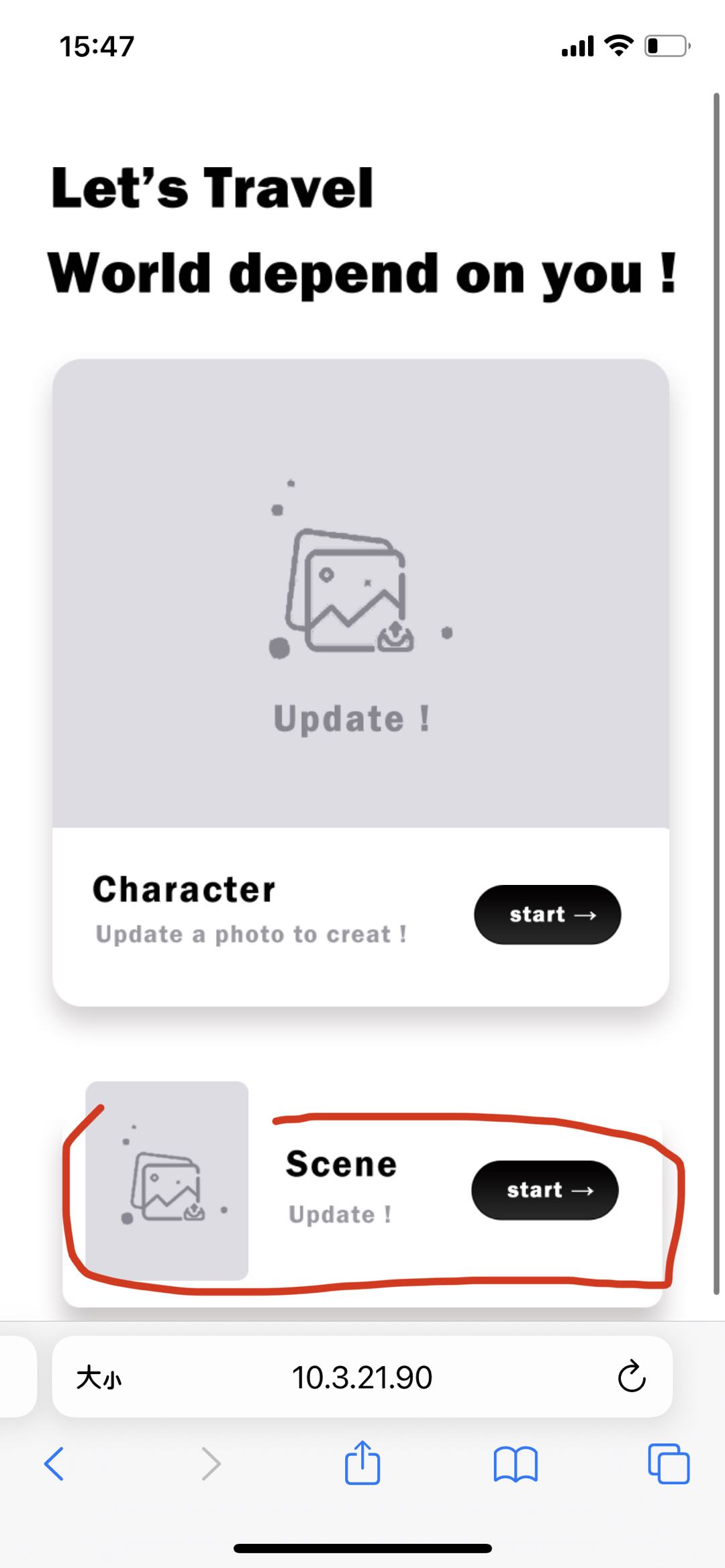
**1.引言**

AIGC领域已经推出了一系列强大的平台和产品，但我们始终致力于优化产品体验，以满足用户的多元化需求。在我们现有的产品体验中，用户反映工具使用门槛较高并且缺乏面向美术宣发的AI游戏生成工具。这些反馈让我们看到了市场的需求和改进的空间。

为了解决这些问题，我们已经设计和实现了一个全新的面向UGC（User Generated Content）的AI游戏生成工具。这个工具旨在让用户以更低的门槛和更直观的方式生成符合个人风格和偏好的游戏。这不仅能增加用户参与度，使用户在游戏中体验到乐趣，更加让用户体验并参与游戏创作中，同时也可以促进用户与平台之间的交流和互动。我们相信，这款工具能让用户乐于分享自己的成果，从而帮助我们的产品在社交媒体上获得更广泛的传播。

具体来说，这款工具包含了以下特点和功能：

1. **用户友好的操作界面**：我们精心设计了直观而易用的操作界面，即使对AI和游戏生成技术没有基础知识的用户也可以快速上手。



1. **自定义帧动画动作生成**：用户可以根据自己的想法，进行详细的定制设置，如角色形象、动作、场景、配乐等，实现真正意义上的个性化创作。



1. **一键分享功能**：用户可以一键将自己创作的游戏分享到各大社交媒体，与好友们分享创作的乐趣。

我们期待这款新的AI游戏生成工具能给用户带来全新的创作体验，并帮助AIGC的产品走进更多用户的生活。

**2.概念定义**

用户通过拍摄或上传个人照片，产生与个人形象关联的帧动画角色动作和场景，真正做到了用户内容生产（UGC）和游戏化的体验。用户不仅可以在虚拟世界中创建自己的立绘形象，还可以以此作为角色在游戏中打怪前进。主要拥有以下功能特性：

* **角色定制**：用户可以对他们的角色进行详细的定制，如更换服装、添加配饰等，使角色更具个性化。
* **游戏多样性**：我们可以设计多种游戏模式和任务，为用户提供丰富的游戏体验。
* **实时反馈**：系统可以提供实时的反馈，让用户了解角色的状态和进度，增加游戏的紧张感和趣味性。

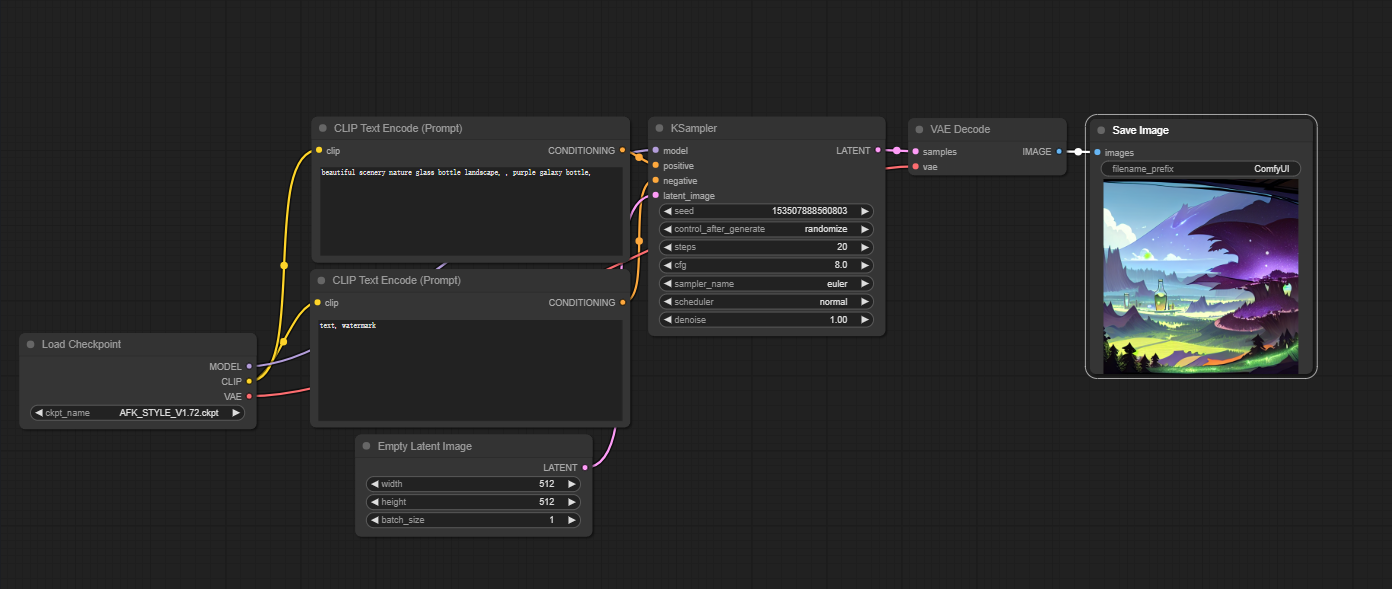
**3.技术路线和系统设计**

**3.1 技术栈**

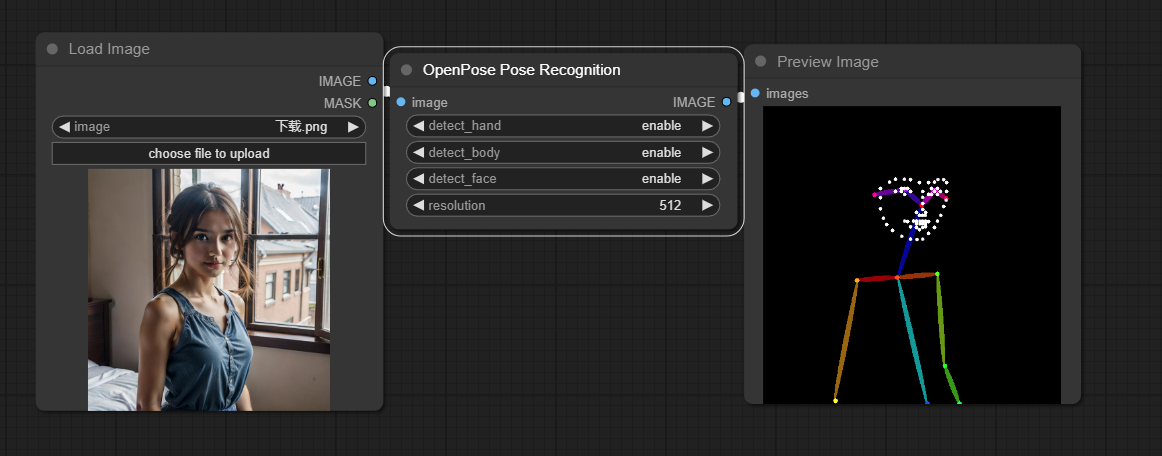
AI部分基于 Comfyui + Animdiff +Openpose + PidiNet实现图生动画资产流程，平台系统采用JS + Python Flask来编写前后端，Unity+C#实现AI序列帧获取与视频生成。

**亮点：**首次从0到1实现前端到Comfyui到unity的全自动化AI视频生成系统，用户只需一键拍照上传就能得到目标风格化视频，大大降低用户使用门槛。

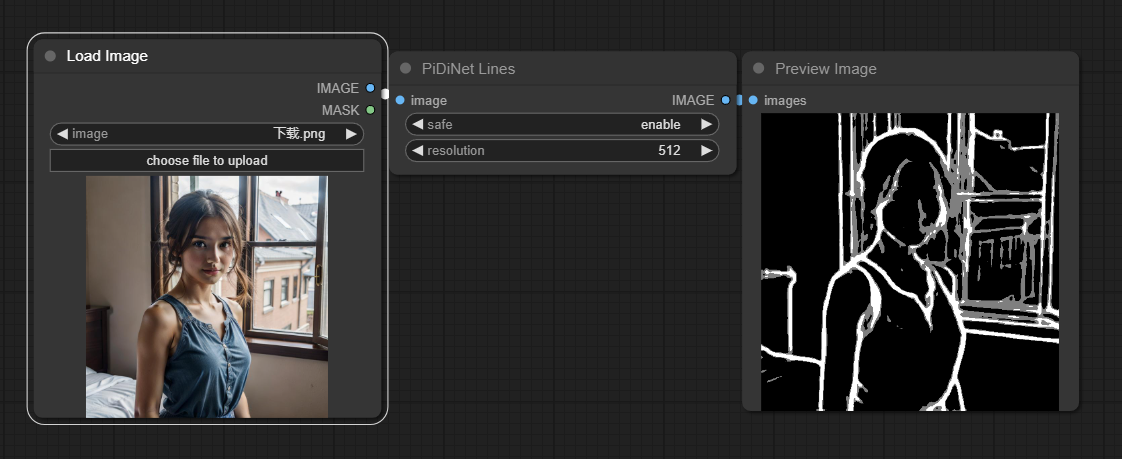
1. **Comfyui**：这是一个基于stable diffusion的节点式工作框架，对用户友好，可以方便地创建出直观和响应式的AI图像与动画。



1. **Animdiff**：这是一个动画差异化工具，可以用来生成动画序列帧，实现角色的各种动作。
2. **OpenPose**：这是一个实时多人关键点检测库，适用于身体、脸部和手部估计，可以帮助将实体特征转换为动画角色。基于Comfyui内部一个Controlnet插件。



1. **PidiNet**：这是一个用于图像处理的深度学习网络，可以进行图像边缘检测，帮助我们从用户上传的照片生成个性化的立绘形象。



**3.2技术解析**

**Stable Diffusion**：是一种[扩散模型](https://zh.wikipedia.org/wiki/%E6%89%A9%E6%95%A3%E6%A8%A1%E5%9E%8B)（diffusion model）的变体，叫做“潜在扩散模型”（latent diffusion model; LDM）。扩散模型是在2015年推出的，其目的是消除对训练图像的连续应用[高斯噪声](https://zh.wikipedia.org/wiki/%E9%AB%98%E6%96%AF%E5%99%AA%E5%A3%B0)，可以将其视为一系列去噪[自编码器](https://zh.wikipedia.org/wiki/%E8%87%AA%E7%BC%96%E7%A0%81%E5%99%A8)。Stable Diffusion由3个部分组成：[变分自编码器](https://zh.wikipedia.org/wiki/%E5%8F%98%E5%88%86%E8%87%AA%E7%BC%96%E7%A0%81%E5%99%A8)（VAE）、[U-Net](https://zh.wikipedia.org/wiki/U-Net)和一个文本编码器。与其学习去噪图像数据（在“像素空间”中），而是训练VAE将图像转换为低维[潜在空间](https://zh.wikipedia.org/wiki/%E6%BD%9C%E7%A9%BA%E9%97%B4_(%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0))。添加和去除高斯噪声的过程被应用于这个潜在表示，然后将最终的去噪输出解码到像素空间中。在前向扩散过程中，高斯噪声被迭代地应用于压缩的潜在表征。每个去噪步骤都由一个包含[ResNet](https://zh.wikipedia.org/w/index.php?title=%E6%AE%98%E5%B7%AE%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1&action=edit&redlink=1)骨干的U-Net架构完成，通过从前向扩散往反方向去噪而获得潜在表征。最后，VAE解码器通过将表征转换回像素空间来生成输出图像。研究人员指出，降低训练和生成的计算要求是LDM的一个优势。

去噪步骤可以以文本串、图像或一些其他数据为条件。调节数据的编码通过[交叉注意机制](https://zh.wikipedia.org/wiki/%E6%B3%A8%E6%84%8F%E5%8A%9B%E6%9C%BA%E5%88%B6)（cross-attention mechanism）暴露给去噪U-Net的架构。为了对文本进行调节，一个预训练的固定CLIP ViT-L/14文本编码器被用来将提示词转化为嵌入空间。

**Lora**: LoRA模型全称是：Low-Rank Adaptation of Large Language Models，可以理解为Stable-Diffusion中的一个插件，仅需要少量的数据就可以进行训练的一种模型。在生成图片时，LoRA模型会与大模型结合使用，从而实现对输出图片结果的调整。假如一张图片是由背景，服饰，脸型，躯干，姿态等等组成，我们可以训练LORA模型对这些进行微调，在大模型的基础上微调成自己想要的图片。

**rembg**:是一个用来从图片中移除背景的工具。它使用一种名叫U-2-Net的深度学习模型来达成目标。U-2-Net是一个基于U-Net的深度学习架构，特别用于解决图像分割问题。  
一般来说，rembg的运作流程是这样的：

1. 首先，它会接收一个包含了待处理图片（可以是.jpg或.png格式）的输入。
2. 然后，它会使用U-2-Net模型来预测图片中的每个像素是前景（即目标对象）还是背景。
3. 最后，rembg会根据U-2-Net模型的预测结果，将背景像素全部移除（即设置为透明），只保留前景像素。

在输出结果时，为了保证对图片的最小影响，rembg会尽量保持原图的质量，即它会将处理后的图片以无损.png格式输出。

**Flask**:是一个用Python编程语言编写的微型Web开发框架。其特性包括：

1. 轻量级：Flask是一个微型的框架，基本上只提供了路由和模板服务的功能。其他大部分功能，比如表单验证、数据库抽象层、登录系统等等，都需要自己实现或者使用第三方插件，这让Flask变得非常灵活和轻量。

2. 灵活性：Flask的核心代码非常简洁，用户可以按照自己的需求随意添加第三方的插件或者工具，如数据库ORM、表单验证等。

3. 模板系统：Flask内置了Jinja2模板引擎，可以方便地生成动态HTML页面。

4. 支持RESTful：Flask支持RESTful风格的URL，使得URL的设计更加简洁和一致。

Flask的架构：

1. WSGI(Web Server Gateway Interface)：Flask实现了WSGI应用接口，可以与多种Web服务器配合使用。

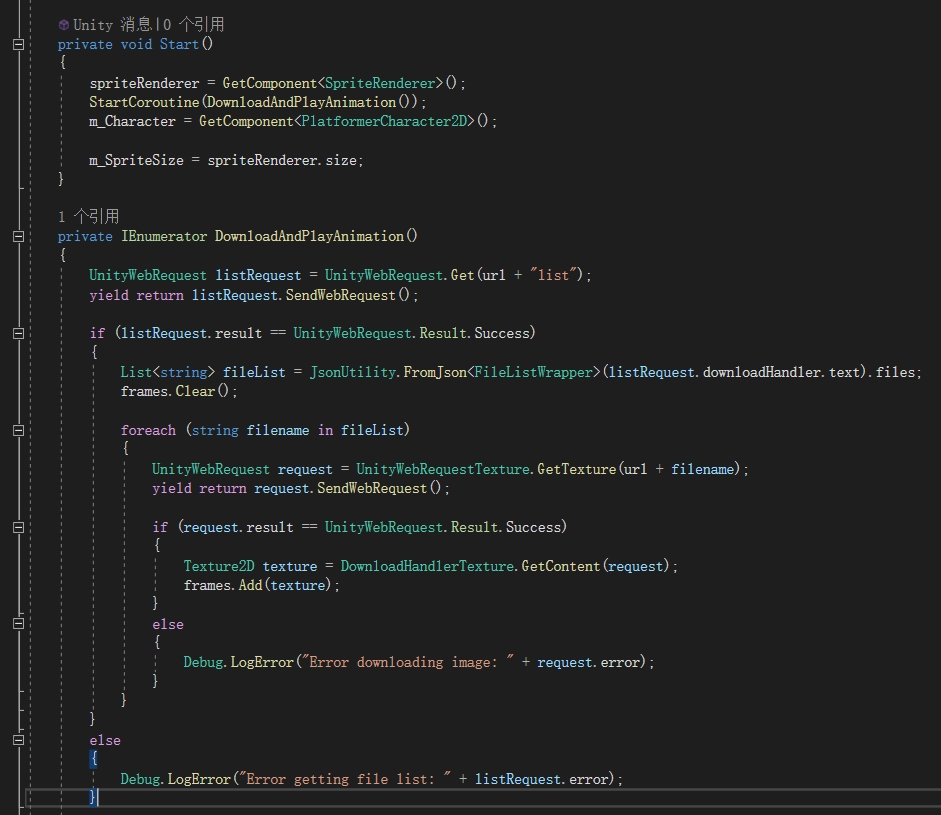
2. 路由系统：Flask的路由系统由 Werkzeug routing类实现，通过装饰器标注来进行URL和视图函数的绑定。

3. 模板引擎：使用Jinja2模板引擎，可以生成动态的HTML页面。

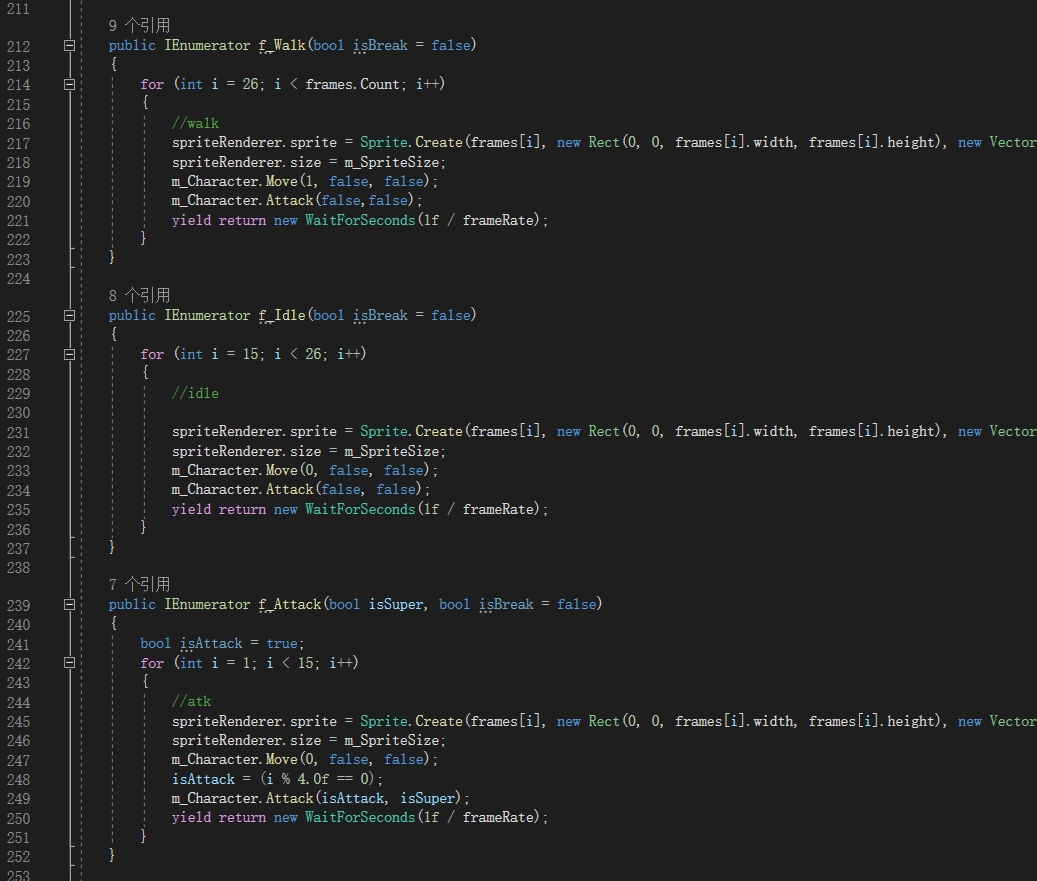
4. 请求和会话：Werkzeug库提供了Request和Response对象来处理HTTP请求和响应，而Flask则提供了session对象来处理会话（即跨请求的数据存储）。

5. 信号：Flask提供了一套信号系统，可以在特定的事件（如请求到来、请求结束等）发生时发送信号。用户可以定义信号处理函数，当相应的事件发生时自动执行这些函数。

**unity**:基于unity运行的游戏逻辑，从后端中获取全部的人物序列帧，根据不同的行为运行不同的序列帧动画，自动索敌并攻击，下图是从后端加载人物动画序列帧的部分代码：



下图是人物行为功能的部分代码：



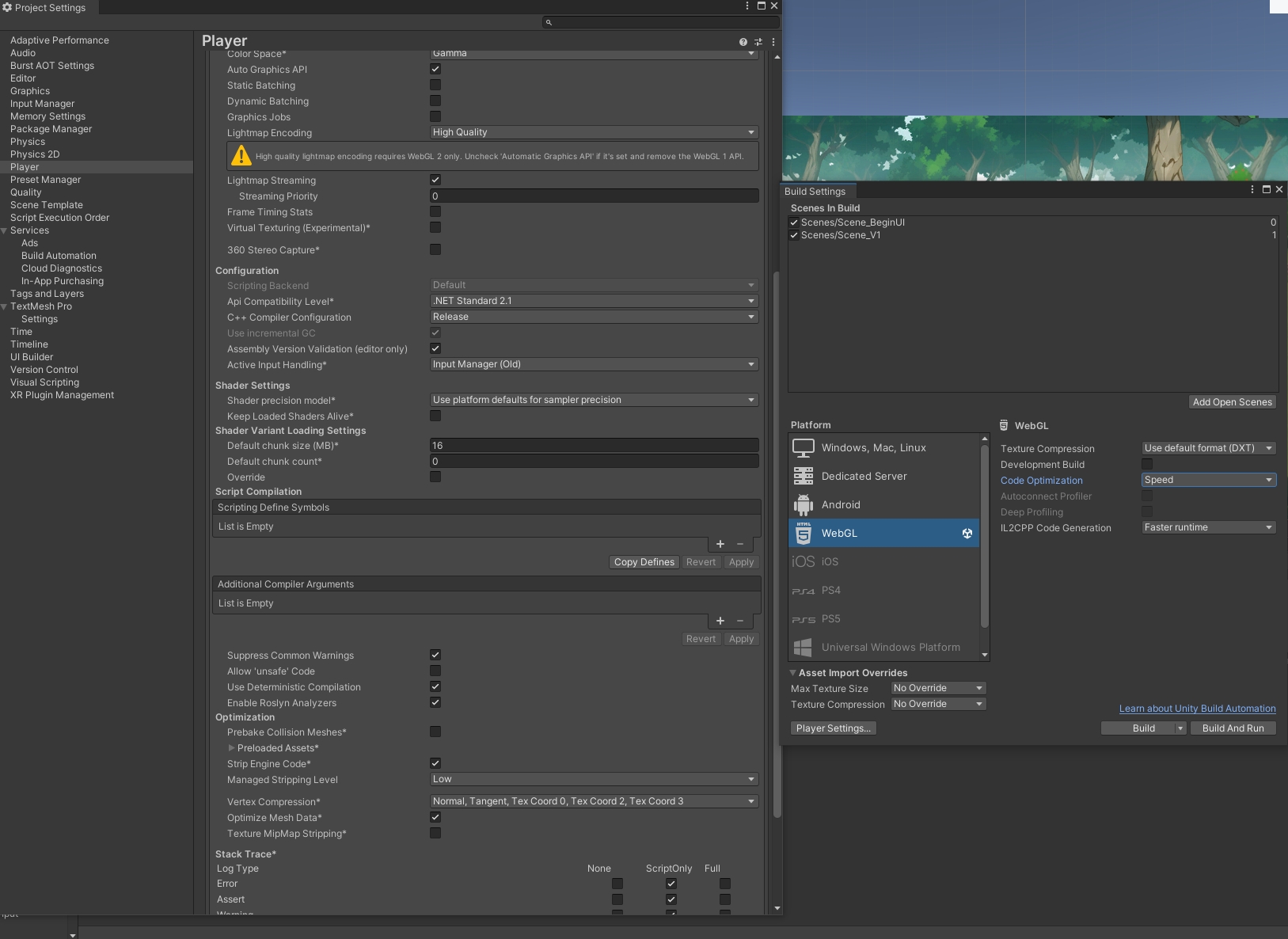
下图是unity制作流程主界面



游戏内部

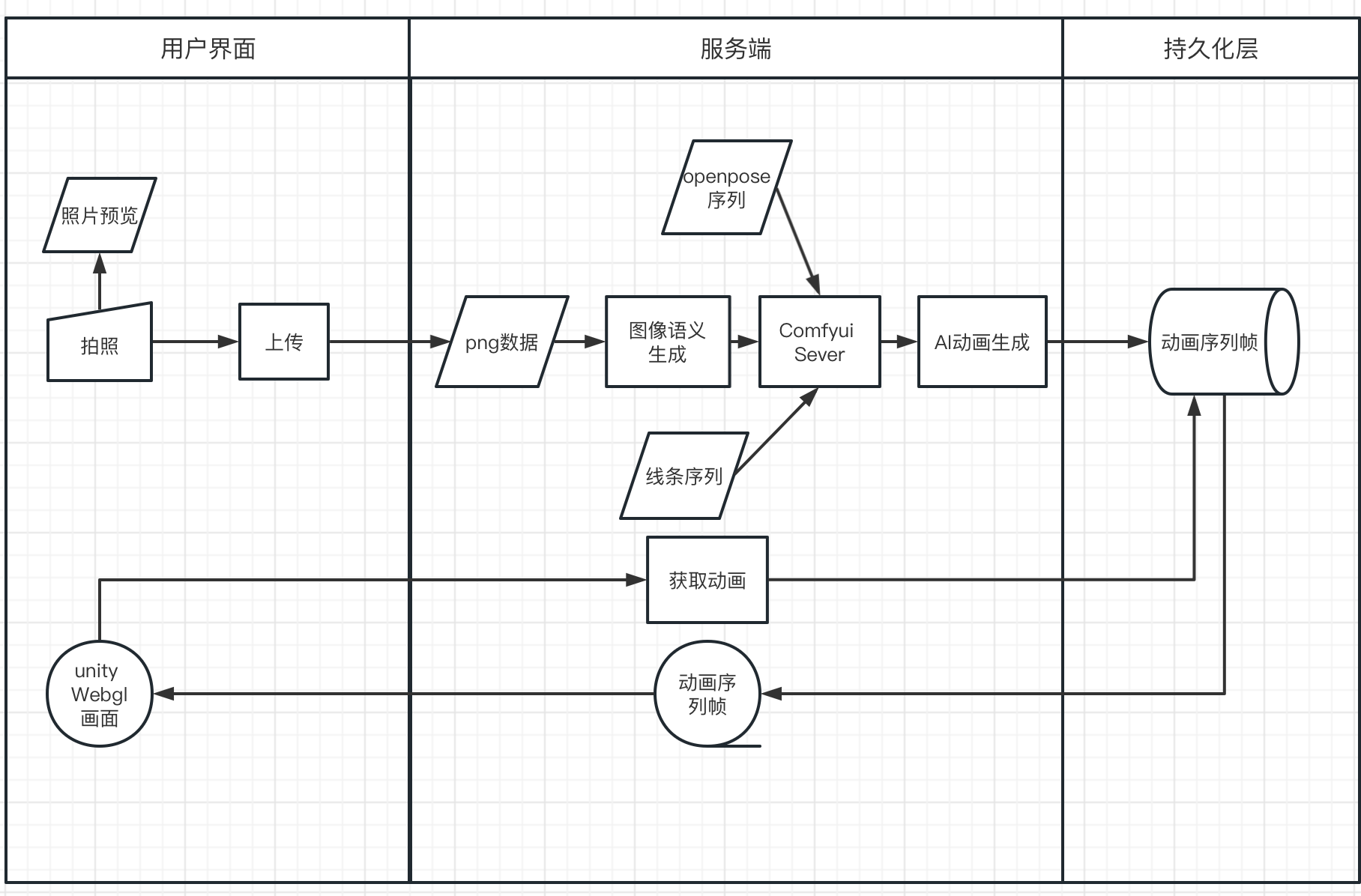


unity网页端打包时参数设置：



unity打包后网页端运行方法：python -m http.server 8001（端口号可随便写一个未被占用的）

**3.3 系统架构**



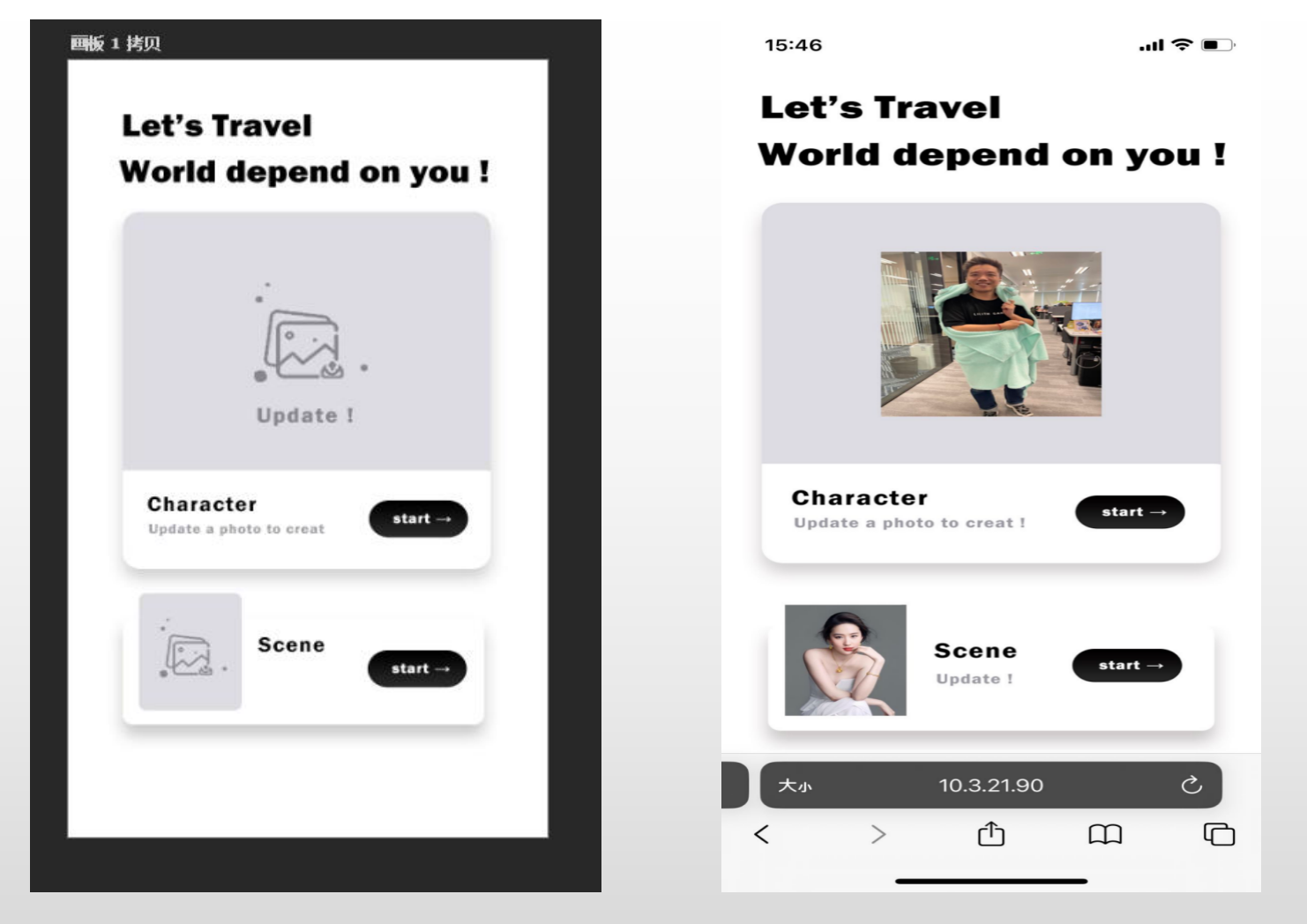
新生态UGC AI游戏生成系统设计与实现中的架构主要分为三层，用户层展示用户界面和相关用例处理，服务层用于数据处理，AI作品生成，并向持久化层发送消息以存储生成的作品，而用户层中的webgl界面可通过服务层去访问AI作品。

1. **用户层**：这是与用户交互的前端部分，负责展示用户界面和处理用户输入的请求。用户可以通过图形界面上传图片，并监视生成的动画过程。
2. **服务层**：服务层是系统的核心，负责处理用户请求，生成AI作品，并与持久化层通信以存储生成的作品。服务层接入AI技术（如Comfyui, Animdiff, Openpose, PidiNet）来生成用户请求的动画。
3. **持久化层**：这一层负责将生成的作品进行存储和管理。涉及到的技术包括数据库管理，文件系统存储等。

**4.最终预览**

**4.1用户界面**

用户界面中主要有选择图片和拍照，图片预览，图片上传，隐藏式动画播放窗口。



**4.2生成效果**

当用户点击上传，会生成个人形象立绘、各种不同动作的动画序列帧，触发隐藏式动画播放窗口播放AI生成的视频。

立绘生成样例展示：



动画序列帧生成样例展示：



最终动画播放效果前端展示：

