Slots项目机台制作流程规范

Table of Contents

谁需要看这个文档？ 3

为什么这个规范很重要？ 3

制作的整体流程 3

流程的几个问题 4

策划 4

美术（UI，特效，动画） 5

美术（仅UI） 5

美术（仅特效和动画） 6

程序 6

音乐音效 6

几个协作的问题 6

svn的使用 7

最后一点 7

# 谁需要看这个文档？

参与制作Slots项目机台的所有人员

# 为什么这个规范很重要？

1 机台是Slots项目里最核心的玩法，每个版本里都会新增机台，这个流程会一直重复迭代下去

2 制作机台需要不同的职位频繁互相协作，需要规范来清晰定义每个职位的职责，保证不同职位之间以最高效的方式协作

3 之前的开发过程中出现了很多问题导致团队的开发效率降低的情况，并且有的问题经常在重复出现。遵循一定的规范可以规避绝大部分问题

# 制作的整体流程

按照时间顺序组织，标记为小数字的表示可以并行

1 策划调研，设计机台主题和玩法，出设计文档

（时间点：当前版本的上上周一就已经完成。如果是紧急插进来的机台，则至少是当前版本的上周一）

2 策划组织该机台的相关制作人员开会，介绍机台的设计，定下设计文档

（时间点：当前版本的上上周一开始做。如果是紧急插进来的机台，则至少是当前版本的上周一）

3

（1）策划配置机台Excel表，包括GameSetting和新机台表

（2）UI开始制作所有资源，包括机台和可能的小游戏，完成后经过美术和策划审核定下初稿，确保后面不会大改。（时间点：当前版本的上周五完成并切图）

（3）特效制作用于进机台的临时的特效文件，比如包括Symbol的特效和卷轴的各种特效

4 在机台配置表，UI，临时特效文件完成后，程序改少量代码，以及调试各种可能的问题，最后确保可以正常进机台spin

5

（1）策划继续机台的数值工作

（2）UI做机台背景，pay table，运行后调整symbol等

（3）特效（和可能的动画师）开始做机台内的特效

（4）程序写新玩法的逻辑

（5）音效师制作新的音乐音效（时间点：当前版本的第一周周五或二周周一开始）

6 如果有小游戏，制作的流程顺序基本为：策划，UI，程序，特效，音效

7 制作基本完成后，策划验收，提出改进的反馈，然后各职位分别去改。策划开会讨论新机器要调优的内容（时间点：当前版本的第一周周五下午4点，仅需该机台相关的制作人员参与）。下周周会讨论新机台制作的问题

8 持续调优新机器（时间点：当前版本的第二周周一至周三）

9 出包QA测试

# 流程的几个问题

1 UI初稿确定没有大的问题后，才由后面程序，特效，动画开始制作，避免出现UI改动后导致其他职位大面积返工的情况

2 pay table原则上由UI制作，如果UI来不及，可能让策划代劳

3 任何人看到机台中不是自己导致的问题，就反馈给策划和相关的制作人员。比如美术可以反馈程序和策划的问题，程序可以反馈美术和策划的问题

# 策划

1 机台配置Excel文档由策划主导和维护（包括Basic表），程序提供支持。所以在新建机台配置文档后，需要核对每一个字段，确保字段在该机台里是正确配置的。不需要的字段应该删掉，比如不是收集玩法的机台配置里不应该出现任何包含Collect关键字的配置

2 Basic表的字段很多很杂，并且一直在扩展。由程序负责维护一个Basic表字段的解释性文档。新玩法对应的新字段由程序负责添加，并告知策划，并加入解释性文档，但策划需负责老玩法的所有字段配置。Basic表目前有点乱，后续做新机台时，程序会拆分核心玩法的配置和表现层的配置，用空行隔开，方便维护，策划在维护时也遵循这个分离的规则

3 新建机台配置文档时可以从其他相似机台复制配置文档过来然后改名，但注意不要在Unity下做复制操作，否则自动导出逻辑会立即生成一堆最终不需要的asset导出文件。正确做法是在Unity外复制和改名，然后点下Unity，就会自动导出asset

4 改完Excel后点下Unity，会自动导出asset文件，上传时不要忘了传asset文件

5 目前Excel导出机制有个无法解决的bug，有时候excel导出的asset文件里有一堆空的配置行，然后导致程序出现各种莫名其妙的问题。所以在删除excel的若干行时，务必做如下操作：勾选删除区域，点右键，点Delete，再点右键，点Clear Contents，然后自动导出的asset文件就不会有空行了。如果仍不确定，可以用文本编辑器查看下asset文件是否有空行。确保上传的asset没有空行，可以正常运行游戏

6 Reel表里Wild个数变化后，记得更新下Payout，NearHit表里的Reel1Wild，Reel2Wild, Reel3Wild里的数组使之个数保持一致

7 确保Payout，NearHit表里所有出现过的symbol名字必须得在Symbol表里有配置，否则运行时会有错误信息提示找不到symbol

8 理解Payout表的算法设计：任何一个symbol组合都是从上到下匹配的，优先匹配上面的行。所以在设计Payout时要确保不会有一个symbol组合能同时匹配两行，而你需要的是下面这一行的中奖，这个是违背算法逻辑的

9 新建表或者改表后可能出现运行时错误，遇到错误后请先检查下配置文档是否有问题（包括空行的问题），然后看下错误log，很多错误log是程序在检测到配置错误时特意打印出来给策划看的提示信息，请不要忽略。如果仍无法解决问题，就发截图到群里@程序

10 需求文档请尽量详细，对美术和程序的制作都会有帮助。如果文档有任何改动，请及时传达给相关制作人员

11 需求都用jira管理，没有遗漏。这样美术和程序可以直接通过看jira来知道自己的任务和进度

12 jira上的需求和bug应尽量描述详细，能重现的bug说清楚重现方式，有错误信息的就把log截图。其他任何可疑的有关的线索都可以写上去

# 美术（UI，特效，动画）

1 和策划确定好机台是下载或非下载，这个决定了资源的路径

2 资源命名尽量规范，最好用英文，不能的话就用拼音。注意大小写，一般一个单词的首字母用大写

3 注意不要有重复的资源命名，比如ABC.controller和ABC.anim就是重复的资源命名，在项目里会导致问题

4 每个机台资源都记得设置AssetBundle名，以该机台名字命名，首字母小写，比如m10。设置方法是：选中该资源，界面右下方有个asset bundle选项，点下拉表选择，如果是新机台可能需要添加名字。漏掉该设置会导致运行时加载不到资源，程序会运行不起来，所以添加任何机台资源请务必记得设置AssetBundle名

5 每张图片都应设置正确的压缩方式，具体参考文档《Slots项目图片压缩事项》。常见的设置比如：是否带透明通道，保持4的倍数，尽量用九宫格。建议美术Unity下安装下iOS和安卓的开发包，这样可以修改压缩方式

6 尽量以节省资源的方式制作，这个是在一开始制作时就要考虑的，会影响到具体的制作方式。有时候甚至需要牺牲一点效果来换取资源的很大节省，这个对产品很有好处。注意美术的职责不仅包括做出好的美术效果，也包括在可接受的美术效果内尽量缩小美术资源

7 学习理解Unity的常见组件，比如Transform和RectTransform，只有正确配置了才有正确的自适配效果。其他常用组件包括但不限于Image，Text，Canvas等

8 改完美术资源后，务必进机台运行看一下，一是看实际运行的效果，二是看能否正常进机台，有没有改错了配置导致游戏不能正常运行。之前出现过很多次美术改错了资源导致机台不能正常运行的情况，会比较耽误程序的时间。其实只需要改动后运行游戏检查下，就能及时发现问题

# 美术（仅UI）

1 注意机台背景prefab，需要和excel里的配置对应起来，比如一些图片以镜像方式复用的，就需要改下excel配置

2 机台卷轴遮罩：目前有3种模式可以用，分别对应资源是3遮罩，单遮罩，镜像遮罩，并且可以选用slice、simple两种资源形式，注意需要填对专门的属性

3 一些机台上有特别的UI，需要在Excel里配置才能正常运行。比如万圣节的装饰Decorate，和一些位置参数，注意在Excel里改配置和上传导出的asset文件

4 如果UI由于某些原因在预计时间内没有做完，导致程序无法按计划搭UI，可以考虑先切给程序一些临时的图，差不多尺寸但图的内容无所谓，程序搭UI。然后UI做完正式的图后直接上传svn

# 美术（仅特效和动画）

1 尽量复用各种图片，粒子资源，有时候图片可以尽可能小尺寸然后在界面里拉大。总之尽量以最节省资源的方式制作

2 注意临时的特效prefab后来改为正式的以后，确保没有保留任何不用的GameObject。特效资源由特效管理，如果程序生成了临时的特效prefab，最后决定不用的话，就由特效来删掉

3 Symbol动画在游戏里都是播放整数秒的时间，比如1秒，2秒，所以Symbol动画的周期得是整数秒时间，典型的是1秒

# 程序

1 负责给其他职位提供技术支持

2 遇到重复性的工作时，考虑写工具把工作自动化。如果其他职位有减少重复劳动的需求，可以告知程序一起讨论是否有自动化的必要

3 将新建机台的部分工作自动化。程序提供一个脚本，可以自动生成几个空的临时占位prefab：MapMachine，Pay Table，Background，（以及可能的特效prefab），用于正常进机台。这个工具策划在配完表后就可以使用。最终效果是，新建一个机台时，当策划和特效的资源准备好后，策划运行下工具，程序再加一点代码，新机台的基本逻辑就可以正常运行

4 在做新机台的逻辑时，改的逻辑有可能影响到老机台的运行。这个可能的影响只有写该逻辑的程序能意识到，所以程序在改逻辑时要保持敏感，一旦意识到可能影响老机台，就告知策划和QA，把老机台加入该版本的测试用例

5 搭UI时如果有些Object可能在某些情况下不需要用到，就做成另一个prefab然后动态加载，这样可以节省内存资源

6 程序搭UI应该还有些best practice可以遵守，这个以后再补充

# 音乐音效

暂无

# 几个协作的问题

1 谁来搭UI

程序负责搭UI框架，同时尽量调到和效果图接近的效果。但在基本逻辑完成后，美术效果靠UI负责细调

2 谁维护机台配置表

策划主导，负责维护。新机台加了新玩法的配置字段由程序加到配置表里，并告知策划和写入解释性文档。新机台里如果有老玩法的配置，由策划负责维护，在新建数值表的时候一并把所有字段核对一遍，改掉错误的，删掉不用的。如果有不清楚的就查文档或者咨询程序。注意错误的配置可能会导致机台运行不了。

配置表里有少量字段也需要美术去维护。在多方可能改表的情况下，谁先上传svn谁就不会遇到冲突（谁先传谁牛逼）

3 谁负责打机台资源包

打机台资源包只针对下载机台，目前打机台资源包的功能已经做成工具并集成到jenkins上，使用非常方便

新机台的机台资源包由程序负责打包，因为程序在新制作时会更清楚什么时候需要打包

老机台的机台资源包如果有改动，由策划负责重打，因为策划更清楚美术的改动需求，以及是否有改动。出问题时由程序支持和修复

版本结束前，策划和美术沟通下哪些老机台的资源有改动过，策划确保所有改动的机台都有重打资源包

# svn的使用

1 请花点时间熟悉svn的基本概念和操作

2 Assets目录下的每个文件都有对应的meta文件。新建或者改动任何文件后，不要漏传meta文件，不要漏传meta文件，不要漏传meta文件！

3 任何人做的临时改动，请记得在上传svn前改回去，不要上传自己的临时改动

4 原则上，上传svn前应该浏览下所有要上传的文件有哪些，是否有任何不是自己的改动。如果改动的文件是可读的，就看下具体改动了什么，是否有任何非预期的改动。确认都是自己预期的改动再上传。如果上传前不清楚某个改动是否应该上传，可以随时咨询程序

5 一般更新svn时，svn会自动合并线上的改动和你本地的改动，大部分时候改动是可以正常合并的。但如果线上和本地正好改了同一个地方，svn就会报冲突，常见的图标是感叹号。一旦出现冲突，就必须解决冲突后再上传。注意未解决的冲突上传上去会导致别人更新后资源完全无法使用，耽误他人的时间。解决冲突的最保险方式是，先revert，然后把自己之前的改动重做一遍

6 为了尽量降低冲突发生的概率，养成一个好习惯：更频繁地做更新和上传

7 保持svn本地文件的干净，也就是不要保留无意义的改动和冲突。有时候对某些文件无意的改动和不解决的冲突就会导致游戏运行出错。遇到这种问题，先尝试revert这些不干净的文件，然后运行看看是否解决

# 最后一点

我们项目的开发是围绕Unity来做的，很多工具和流程都是在Unity上。所以请每位同学多花点时间学习下Unity，至少熟悉Unity里和自己工作相关的常用功能。这样我们才能更高效更顺畅地制作游戏。