**1. Write a complete game script**

Working from the concept document you created for [Assignment 06](http://users.wpi.edu/%7Ebmoriarty/imgd1001/a06.html), sketch out a *complete* script for your game.

**The purpose of the script is to provide the engineer(s) on your team with the information they need to actually implement the game using Ren'Py.**

The format of your script will vary, depending on the design of your game. But it will probably need to include at least some of the following elements:

* **Character list.** List every significant character that will appear in the game. Anyone who speaks, performs an action, or whose presence is necessary for the story is significant.

|  |  |  |  |
| --- | --- | --- | --- |
| Character Name | Location | Placeholder File Name | Other placeholders associated |
|  |  |  |  |
|  |  |  |  |

* **Location list.** List every location that will appear in the game. In most visual novels, each location is signified by a unique background image.

20 minute job

* **Practical screen list.** List every screen that is *not* a location, but is necessary to complete the game. All games will include at least a title screen and a credits screen. You may also need screens for navigation, visual menus, cut-scenes, close-ups of interesting objects, and other purposes.

1 hour

* **Flowcharts.** Create one or more flowcharts that indicate how players will progress through your game from start to finish. These charts can be made at various levels of detail. For example, you might have a master chart that shows the overall scene transition structure, with additional charts showing the flow of individual scenes. Every major decision point, story location and practical screen should appear on a chart somewhere.

Master flowchart: 20 minutes

Conversation flowchart: 30 minutes

* **Scene scripts.** Write a complete script for every scene in your game. Typically, a "scene" is everything that happens in a single location until the location changes. Each scene script should include:
  + Scene and location name.
  + List of all characters appearing in the scene.
  + Narration/dialog. Write down every line spoken by the narrator (if any) and by each character. Lines should be clearly labeled with the name of the character speaking. Your writing can be first-pass and rough at this point, but try to capture *all* of the narration and dialog essential to the game.
  + Changes/animation/sound cues. If the appearance of something changes during the course of your narrative (somebody enters or leaves a scene, a character moves, smiles or frowns, a door opens or closes, etc), note the change and when it should take place. Also explain when sound effects/music should be played or changed.
  + Menu/Dialog choices. If the player is presented with choice(s) during a scene, show what choices are available and how the scene will proceed after each choice is made.

Lobby sequence

Conversation sequence for each character

Elevator sequence

Hallway sequence

Room sequence

* **Cut scene descriptions.** If your game has any non-interactive cut scenes, describe them completely, listing all backgrounds, characters, dialog, animations, sound effects and/or music required.

Pawn shop sequence

Jail sequence

Transition between apartment and hotel

**2. Create an asset list**

When your script is complete, go through it and create an Excel spreadsheet of *every* audiovisual asset your engineer(s) will need to build your prototype:

* Practical screens (title, credits, navigation, etc)
* Background screens
* Character art, including all variants (clothing changes, facial expressions, asleep/awake/dead, etc).
* Multiframe animations (twinkling stars, doors that open/close, etc).
* User interface elements (buttons, hover states, etc).
* Sound effects and music.

If you don't know how to use Excel yet, it's time to learn. Excel is part of Microsoft Office, installed on every PC at WPI, and also available for download on your personal machine.

Each entry in your spreadsheet should include the following information:

1. **Filename.** Decide the actual filename that will be assigned to the asset. It's a good idea to use a naming convention that suggests the purpose of the asset. For instance, backgrounds could start with the prefix "bg\_", characters with the prefix "ch\_", etc. Include the file extension ("bg\_park.png").
2. **Description.** Just a few words is enough, but make it unique and unambiguous. Example: "Park background, morning."
3. **Specs.** Show the expected pixel dimensions of images/sprites, or the approximate duration of sounds and music.
4. **Source.** The source of the asset, including full credits if the asset is borrowed.
5. **Status.** The production status of the asset. Usually one of the following: Placeholder, In Progress, Done.

Here is a [sample asset list](http://users.wpi.edu/%7Ebmoriarty/imgd1001/docs/Sample.xls) to get you started. Right-click the link and select Save Link As to grab it.

**2. Create placeholder assets**

Create or borrow a working placeholder for *every* asset in your spreadsheet.

**Make sure your placeholders are useful and usable.** Every piece of placeholder art should be unique, and easily identifiable just by looking at it or listening to it. The specifications of the assets should meet the requirements listed in the spreadsheet, so that they can be imported and wired into your game by your engineers.

When you're done, you should have an actual file (with matching filename) associated with *every* item on your asset list.

**Alternate submission strategy**

In lieu of a game script and folder of placeholder assets, I will accept a *complete, fully working* Ren'Py prototype of your game, incorporating *useful* placeholders for all art, music and sound effects.

* The prototype should be compiled as a Windows executable. By default, Ren'Py will also create Macintosh and Unix versions. Refer to the Ren'Py documentation to learn how to disable this.
* Be sure to thoroughly test the prototype before submitting it. If a programming error prevents me from seeing part of your game, your grade may suffer.
* Move the folder containing your prototype to a different PC and make sure it works there, too. If it doesn't work when moved, it probably won't work when you send it to me.

Your team is still expected to provide an Excel spreadsheet that lists all of your assets, as described above.

No extra credit is awarded for submitting a prototype instead of a script. However, by completing your prototype early, your team earns more time for polishing and perfecting your final game, which is likely to improve your final project grade.