High Level Architecture of Game Assembly Application

The Game Assembly application has a two layer architecture with the first layer containing the components and the second layer containing the entities. This architecture splits the logic of building the game into one layer and provides a second layer of entities which contain all information needed to build the game.

The first layer in the two layer architecture contains three main components: the game layer loading component, the game building component, and the export component. The game layer loading component retrieves the specified entities from the repository (currently xml files) and converts them into Java objects. Additionally, it wires the game layers together resolving any dependencies required for assembly. A game layers object is created in this process and can be passed to the game building component. The game building component consists of three sub-components that perform the actual construction of the game. The first of which is the structure which calls upon the other two sub-components, the theme component and the locale component, to build the individual acts which will comprise the game. The theme sub-component builds the story line which consists of the intro and outro acts of the game. The locale builds all interior acts which are made up of the lessons and challenges as well as intro and outro story sequences. After receiving all acts from the other sub-components, the structure wires the acts together into a game. Once the game is built it can be passed to the final component, the export component, which exports the game into the desired format (currently an xml file).

The second layer in the two layer architecture simply contains the entities which encompass all data needed to build the game. These entities are comprised of challenge, characters, lesson, locale, subject, and theme. The challenge entity represents the challenge presented at the end of the lesson and is currently limited to a multiple choice quiz of varying length. The characters entity contains information about what names and assets should be used for each of the four supported character types. The lesson entity encompasses a single lesson which directs the student to learn some objective. The locale entity holds all information describing the location of the experience such as the background, character positions, background and foreground object positions and locations of where text can be displayed. This allows other entities to simply reference a named location, such as a character’s speech bubble, and provide the accompanying text. The subject entity describes the overall subject that is being taught and any introductory text. The theme is the story aspect of the game and contains not only the information used to build the intro and outro acts, but also story snippets which surround each learning act to progress the story throughout the learning experience.

Components:

Structure

The structure is the main component which is called upon to build the game. The structure knows the basic form of the game without knowing intimate details about the acts. The basic form of the game is separated into three parts: the intro story act (the introduction), the inner acts, and the outro story act (the conclusion). The structure calls upon the theme sub-component to build both the intro and outro acts. The number of inner acts can vary depending on how many lesson entities are provided as input. These inner acts are built by the locale sub-component. The structure depends on both the theme and locale sub-components and provides a game object as output. It has a single entry point called createGame which calls upon the sub-components to build the acts and assembles them into a game which is returned.

Theme

The theme sub-component is responsible for building the story acts which includes the intro and outro acts. It builds all screens for each of these acts by combining the information contained in itself, the subject entity, and the characters entity. It has no other dependencies or inputs. It provides two methods for creating these acts: getIntro which creates the intro act and returns it as a list of screens which are internally wired together, and getOutro which creates the outro act and also returns it as a list of screens which are internally wired together.

Locale

The locale sub-component is responsible for building each of the learning acts. These are all interior acts contained between the two story acts created by the theme. It builds the screens for each act from the information stored in itself, the characters entity, the theme entity, and the lesson and challenge entities. It has no other dependencies. It has a single entry point which is getAct and accepts a learning act id as input. This learning act id represents which of the inner acts should be built (ie which lesson and challenge entities to use). As output it produces a list of screens which are internally wired together.