StoryBook Component Description

# Introduction

StoryBook is a writing platform for stories that can be presented in various formats. A story writer wants to develop a story in the manner which best suits the writer, the story itself, and possibly the presentation format. One of the design goals of this project will be to free the writer from:

* Having to develop the story in a purely linear fashion,
* Having an order of the development steps imposed upon them,
* Allowing only one presentation of the story on a single medium,
* Having only a single author by enabling cooperative authoring by multiple authors,
* Having to be an expert in a particular medium to develop a story for that medium.

StoryBook aims to support the following media for the presentation of the story:

* Print, like a regular book,
* Interactive text where the user can interact with the story to determine the direction of the story,
* Graphic novels, which can also have an interactive component,
* Augmented graphic novels which can have a combination of 2D graphics and 3D scenes and allow for user interaction,
* Interactive movies which show video clips to represent each scene and can allow user interaction,
* Video games which allow user interaction to determine the next scene to be presented to the player,
* Mixed media presentations which combine some or all of the above.

# StoryBook Components

* Story -- The story describes everything that we are writing. This will be something like a play, movie or game. The story will contain all of the elements listed below.
  + Presentation – this determines whether the game should be a series of 2D scenes, like a graphic novel, a series of purely 3D scenes which would create an interactive movie, or a mixed presentation where 2D scenes are displayed like a graphic novel and 3D scenes are displayed like a frame in a graphic novel, but expand into 3D scenes when they are clicked on.
  + Layout – This determines the layout of the 2D frames. They can be arranged several to a page like you would see in a graphic novel. They can be displayed once at a time so that each of the frames occupies the entire viewport or they can be scrolled either horizontally or vertically across the viewport.
  + GUI – this describes a graphical user interface that will overlay the display of the story in the viewport. The GUI will be able to perform actions which will affect the state of the game. One of the windows can be defined to contain the display of the scenes. If no window is specified to contain the scenes, they will be displayed on the root window of the viewport.
  + Global Variables – this is a set of global variables that can be accessed by the GUI or any of the code in the scenes which are displayed. Variables will reside in namespaces and will use the dot notation to qualify variable names with the names of the namespace in which they reside. By convention, the names of all namespaces should be capitalized and the names of variables should start with a lower case letter and can be in mixed case. Constant variable names can be entirely in upper case. The name of the global name space is “Global” while the name space for each scene is simply the name of the scene. The scene which is currently being displayed can also be referenced by the name space “CurrentScene”. Variables have types int, float, bool, string and scene. Scene is a reference to the current scene. All variables have the value NIL before being assigned. Any variable which does not have a value will also be NIL and actions can set the value of a variable to be NIL. There are also pre-defined global variables:
    - currentSceneRef – a variable of type scene referencing the current scene. If there is no current scene, this has the value NIL.
    - playTime – the time in fractional seconds since the reader started reading the story.
  + Roles – Roles describe the various characters which will be in the game. Roles can be played by various real characters and a real character can be substituted for a role. You can develop a role before you have thought out the details of the character which will play the role. You can develop your story using roles and later substitute real characters for the roles. In fact, you can assign characters to roles while the game is being played. For example, in a murder mystery, the name of the murderer might be determined by how the game is played. As a result, one of the characters can be assigned to the role of the murderer once the identity of the murderer has been determined.
    - Scenes – a list of scenes where this role is played
    - Characters – a list of characters playing this role
  + Characters – These are the real characters in the game. They can be humans, aliens, robots or similar beings. Characters have a mixture of fixed and variable attributes. The attributes of the characters are shown below:
    - Description – a textual description of the character
    - Image – zero or more 2-D images of the character which can be used in various scenes
    - Model – zero or more 3D models of the character which can be used in various scenes
    - Relationships – zero or more relationships this character has with other characters
    - Scenes – a list of scenes in which the character appears
    - Roles – a list of roles played by this character
    - Variable Attributes – these are a list of attributes which can be defined by the writer. It might be a list of desires, strengths, etc. Once the attribute is defined for one character in a scene, it will appear in the list of attributes for all characters in the scene.
  + Relationships – a relationship is used to describe the relationship between roles or characters. It could represent a familial relationship like daughter or an emotional relationship like love or a contractual relationship like employee. No relationships are pre-defined and the writer can define any relationships required for the story.
    - List of relationships – this contains a list of the relationships defined and can show the roles or characters to which the relationships apply.
  + Summary – This is a textual summary of the story. It is written in a linear fashion and portions of the story can be linked to one or more scenes where that part of the story plays out. In addition, roles and characters named in the story are automatically linked to the definition of those roles and characters.
  + Timeline – this is a high-level view of the most important events in the story. Each of these events is linked to the point in the summary that provides the textual description of this part of the story. It is also linked to the scene (if applicable) that will visualize this point on the timeline. This is a graph that will allow the parts of the story to be laid out one after another, in a sequence. This sequence might not correspond to the linear flow of the flow of the story, allowing for flashbacks in the presentation, cut scenes as the technique of showing the ending before the rest of the story. The goal is to allow the author to write the summary in pure linear time and allow time to be re-arranged in the timeline. Branches in the timeline can also be created to show the alternate story lines which can be created based on actions taken by the reader. It will be possible to create new nodes in the story line and join them to sections of the summary and scenes. You can join a node to the summary and scene when it is created, or you can do it at a later time. This will allow you to use the timeline as an outlining tool to get the steps in the story down and then fill in the details of the summary and scene at a later point. There could be multiple timelines, each of which allows for a different presentation of the same story. When the story starts, there will be some way to select which version of the timeline will be followed. The timeline also allows scenes to be grouped and associated with a layout. For the automatic layouts, all you need to do is to indicate which scenes should use a particular layout. For arbitrary layouts the author has built manually, you can associate each scene with a particular window in a layout.
  + Layouts – Layouts define how scenes are arranged in the viewport. This could be one scene that fills the entire screen, a series of scenes that scroll across the screen, or multiple scenes displayed at once. When multiple scenes are displayed at once, you can elect to have them laid out like a graphic novel with controls to display the next page. In the novel layout, you will be able to specify the size of the frames and they will be laid out automatically. You can also specify an arbitrary layout with windows to hold different scenes on the screen. You will be able to connect actual scenes to the locations in the timeline. Layout can be named and saved so they can be easily referenced from the timeline. Layouts can also have a GUI associated with them.
  + Scenes – this is a list of scenes in the story. Scenes can be either 2D scenes, 3D scenes or a mixture of the two. Scenes can simply flow onto the next scene or can contain decision points whereby the reader can interact with the scene to determine the direction the story takes from that point forward. It is possible to have scenes which contain just a dialog which forces the reader to make a decision which will affect the next scene which is displayed. Scenes have various attributes associated with them.
    - List of roles – a list of the roles involved in the scene
    - Characters – a list of characters in the scene
    - Graphic – In the case of 2D scenes, this is a 2D image or solid colour definition which can be used as a background for the scene. You will also be able to specify the size of the scene as part of the graphic. Setting the size of the scene might cause the graphic to be resized or will provide the size for a solid colour frame. 3D frames can have the name of a 3D scene in the host graphics system which will be displayed. If a 3D frame is used, then a 2D graphic to represent that 3D scene in the 2D world can also be used. A graphic can also be a reference to a video clip for the scene. It will have a size the video clip can be presented.
    - Objects – a list of objects in the scene that the reader can interact with
      * Name – the name of the object which can be used to link the object tto an object in the scene that has the same name.
      * Event listeners – a series of event listeners associated with the object that will specify an action to be taken when the event occurs.
      * Position – an optional position which specifies the location of the object in the scene when the object is first created in the scene.
      * Description – a textual description of the object in the scene
      * Story link – one or more links to the relevant part(s) of the summary that describe the object or its use in the scene.
      * Decision action – an action that will allow the reader to make some type of decision that will affect the outcome of the game. This could be the answer to a question, a selection from a menu of choices, or some other simple interaction.
    - Dialogs – a dialog is a series of sentences that form the interaction between two or more characters. Each sentence is associated with a role, character or object that will say the sentence. A sentence can be of several different forms:
      * Simple Sentence – this is just a statement. It might or might not elicit a response from other members of the conversation.
      * Simple question – a simple question requires an answer and can be answered by any other member of the conversation.
      * Random question – asks one of a series of questions which are selected at random while the story is being read.
      * Choice question – a series of questions where the reader is able to select the question to be asked.
      * Variable question – a question selected from a list by using a variable to determine which question number should be used. The variable can be affected by an action performed by the user in the scene.
      * Simple answer – a straight-forward answer to a question.
      * Random answer – an answer randomly selected from a list of answers when the story is being read.
      * Choice answer – an answer selected by the reader from a list of choices.
      * Variable answer – an answer selected from a list by using a variable to determine which answer number should be used. The variable can be affected by an action performed by the user in the scene.
      * Each sentence will have the following attributes
        + The role, character or object which says it or a random selection or a choice based on the setting of a variable.
        + The text of the sentence (optional)
        + The audio file for the sentence (optional)
        + An action to be performed before, during, or after saying this (optional)
        + The name of a callout in which the sentence should be displayed (optional)
        + The next sentence to be said (optional)
        + One or more links to the summary where this conversation occurs.
    - Callouts – these are the areas where dialog is displayed. In a 2D scene, the callouts have been drawn by the artist into the scene. The software will then draw the text over the position of the callout drawn by the artist. The text will be rendered as a rectangular area. In a 3D scene, you can have callouts at either fixed positions or relative to the position of a character or object in the scene. Since the callouts are not pre-drawn by the artist in a 3D scene, you can select the shape of the callout to be used. Each callout is associated with one or more speakers and displays the sentences said by them. In most cases, there is one callout per speaker. When the callout is shared among several speakers, the sentences of all speakers are displayed in the same callout. Callouts have the following properties:
      * Name – a unique name for the callout
      * Speaker list – a list of roles, speakers or object that utter sentences to be displayed in the callout
      * Position – the position of the callout
      * Relative-to – an object the position is relative to. If this is not specified, the position is absolute.
      * Background color – the colour of the background
      * Text colour – the colour of the text
      * Text font – the font name and size
      * Scrolling – whether the text scrolls towards the top or whether new text overwrites the previous text.
    - Variables – These are variables that are defined in the scene. Actions performed within the scene can set the value of the various variables. Variables can be of type string, int, bool or float. Variables can be initialized at the start of the scene and will retain their values until the scene is finally destroyed. Variables can be made constant so that the actions within the scene cannot modify the value of the variable after it has been initialized.
      * Pre-defined Variables – these are variables which are pre-defined when a frame is created. Therefore, the implementation can depend on these variables existing in every frame.
        + nextFrame – This is a string which contains the name of the next frame in the story. It can be initialized to a value when the frame is designed and this value will be assigned when the frame is instantiated. If none of the actions in the frame modify the value, it can be used to represent a static relationship to the next frame. Actions can modify the value of this variable in response to actions taken by the reader and this will result in a branching storyline.
        + sceneName – the name of the scene as a constant string.
        + scenePresentation – a string telling you whether this is a 2D scene or a 3D scene.
        + sceneGraphicHeight – the height of a 2D scene or the 2D representation of a 3D scene. This is zero if not applicable to this scene.
        + sceneGraphicWidth – the height of a 2D scene or the 2D representation of a 3D scene. This is zero if not applicable to this scene.
    - Menus – menus provide a way for the user to interact with the game. They can be similar to a dialogue but do not represent a dialogue between two characters in the game, but rather a dialogue between the story and the reader. They are limited to asking a question and presenting a list of choices from which the reader can make a selection. There needs to be support for cascading menus that allow for more complexity in the choices available to the reader.

## Actions

Actions are performed in response to events which happen. This could be saying something in a dialogue, making a selection from a menu, pressing a key on the keyboard or clicking on an object with the mouse.

Actions can be combined to form scripts. Scripts can be named and saved so that the same script can be used with several actions. The combination of actions into scripts can be done via a visual scripting interface that allows scripts to be created by dragging and dropping actions to form a sequence of actions.

The basic set of actions will include:

* MoveObject -- moves an object to a new location
  + Parameters
    - Object – the object to move
    - Location – the location to move to
    - Relative – if this is selected, then location is an offset from the current position of the object
    - Destination Object – this can be used to specify that the object should move to the location of this object. If this is specified, then the location will be a distance to stop before reaching the destination object.
    - Speed – the speed the object should travel with an option to accelerate at the beginning and decelerate at the end.

# Minimal Viable Product

The minimal viable product (MVP) is defined as the smallest implementation which can be built and demonstrates a core set of functionality of the system. The MVP for StoryBook is:

* Client-Side
  + Login screen
  + Screen to select / create / delete stories from a list of existing stories
  + Screen to enter characters and their descriptions
  + Screen to enter the summary of the story
  + Screen to define scenes where much of the functionality is missing, You can indicate the characters, parts of the summery involved, and links to graphics for the scene
  + Timeline screen that allows the sections of the summary to be arranged and to be connected to scenes
* Server-Side
  + Functionality to generate the required web pages
  + Ability to store the data in memory
  + Ability to persist the data to a database