Hero6 Progress Database Outline

This database represents the current development of Hero6, an adventure game developed using Adventure Game Studio. The web interface related to the database offers a review capability of development while allowing for the modification, removal, or addition of data.

The typical adventure game has a third-party perspective where the player controls a character or characters in the pursuit of a goal via an interaction of characters and the environment. The basic requirements for an adventure game are characters which populate the game, rooms that the characters may occupy or navigate between, sprites (i.e., artistic renderings) of characters for looping into animations, and dialog (i.e., feedback) that the player receives from character interaction. The web interface represents these four basic areas of development; however, the database handling the data is more complex.

Characters

Characters have the following subjective attributes:

- Initial room location (optional) and the X, Y position in the room (required, defaults set)
    - Characters without an initial starting room are not loaded by the game engine; however, the constraint regarding the starting X, Y position is a protection against an event where a script loads the character in a room without specifying its X, Y position.
- A character name (used in game) and a reference name (used in scripting) (both required, defaults set)
- Five basic views (optional, set to null when associate view is deleted)
- The text color used when the character speaks (required, default set)
    - Colors consist of red, green, and blue (rgb) attributes (required, table is already populated with all possible 65,536 color types)
    - The rgb attributes for a color can be modified, but there are always 65,536 colors available

Views

A view refers to a collection of sprites looped into an animation. Like a film, views are a series of frames with each frame containing a sprite. The sprite is the artwork; the frame is its container. To preserve resources, frames are organized into loops, and loops are repeated to generate animation effects like walking, running, or talking.

For the purposes of database organization, the 'view' table is simply a reference to a script name (required). The collections of frames that compose a view are stored in 'viewLoopsFrame' by a triple attribute key (view id, loop id, and frame number). Each frame has:

- A parent view (required, key, set to cascade on delete)

- A parent loop (required, key, set to cascade on delete)
- A sprite image (required, default set, no action on delete since sprites should only be added or updated)
- A variable for displaying the sprite flipped on its vertical axis (required, default set)

A sprite contains file location attributes for the image on the web server:

- path (optional, a null value results in current web directory, default set)
- file name (required, default set)
- extension (required, default set).

A loop is a method for storing a series of frames, the first eight loops are typically reserved for directional movement; however, the loop direction attribute (required) is expandable for more than eight directions and defaults to 'No special purpose' for loops higher than seven (0 through 7 account for the eight principal directions). Loops, like sprites, can only increase in size or be modified, but never deleted.

Rooms

A room is a rendered environment encompassing the entire display. For the purposes of database organization, the 'room' table contains a reference to the room's description (optional). The elements of a room are its background image and a directional relationship with other rooms.

A room may have multiple background images displayed in sequence to simulate an animated environment. Each row on the 'roomBackground' table has a background id (i.e., 'bid') that allow for multiple room backgrounds (required, key). It also contains file location attributes for the image on the web server:

- path (optional, a null value results in current web directory, default set)
- file name (required, default set)
- extension (required, default set).

Each room is an origination point for travelling to another room location (i.e., destination) by exiting in a specific direction. This path from one room to another occurs via an exit direction. There are eight principal exit directions (direction is required); however, the 'exitDirection' table is expandable allowing for additional directions. Exit directions, like sprites and loops, can be added or modified, but never deleted. The 'roomExitDirection' table combines the exit direction for each room (both are required and create a double key) and its corresponding destination room (required).

Dialogs

The player receives text feedback from characters during an interaction. This database stores a reference to each dialog along with its attributes:

- script name (required, default set)
- character id of the speaker (optional, but recommended)

Long-form ER Diagram

The entity-relationship diagram included with this document details the interaction of these tables. Long-form relationship interpretations are:

'characters' to 'dialog':

- A character can read one or more dialogs
- A dialog can be read by zero or one character

'characters' to 'color':

- A character speaks (via text) in one and only one color
- A color is used (via text speech) by zero or more characters

'characters' to 'view':

- A character can have zero or one blink|idle|normal|speech|think view
- A view can by the |idle|normal|speech|think view for zero or more characters

'characters to 'room':

- A character starts in zero or one room
- A room is the starting location for zero or more characters

'viewLoopsFrame' to 'view':

- A view contains only one instance of its loops and frames
- Each set of loops and frames is the only instance of a view

'viewLoopsFrame' to 'loops':

- Loops and frames are arranged by one or more loops
- Each loop is an arrangement of zero or more sets of loops and frames

'viewLoopsFrame' to 'sprite':

- Each frame has a one and only one sprite
- A sprite is used in zero or more frames

'room' to 'roomBackground':

- Every room has one or more backgrounds
- Every background is used in one or more rooms

'room' to 'roomExitDirection':

- Each origin room has one or more exits
- Each exit path has only one origin
- Each room is a destination from one or more exits
- Each exit leads to only one room

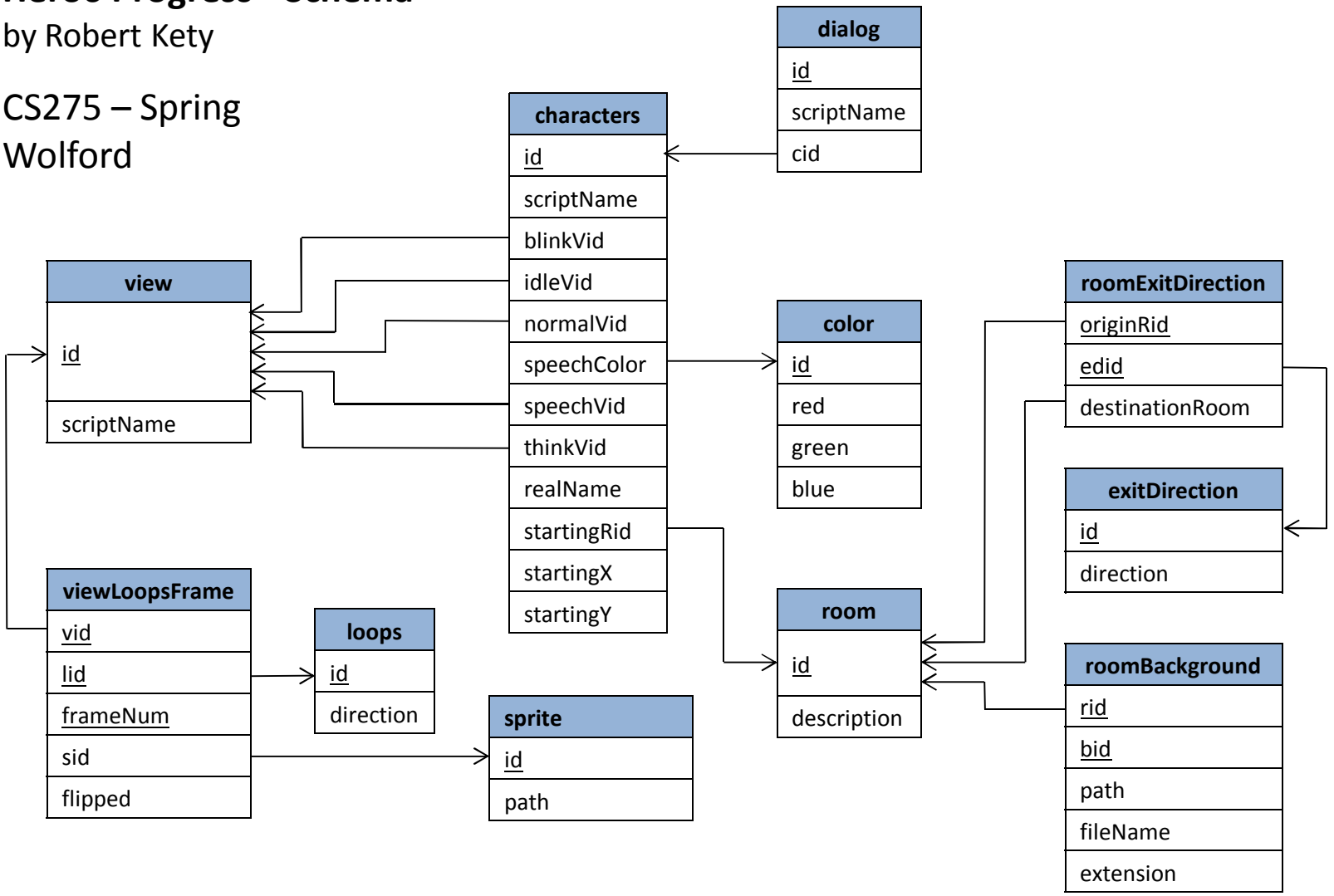
'roomExitDirection' to 'exitDirection':

- Room exits are ordered by one or more directions
- Each direction is used to order zero or more room exits

Hero6 Progress - Schema

by Robert Kety

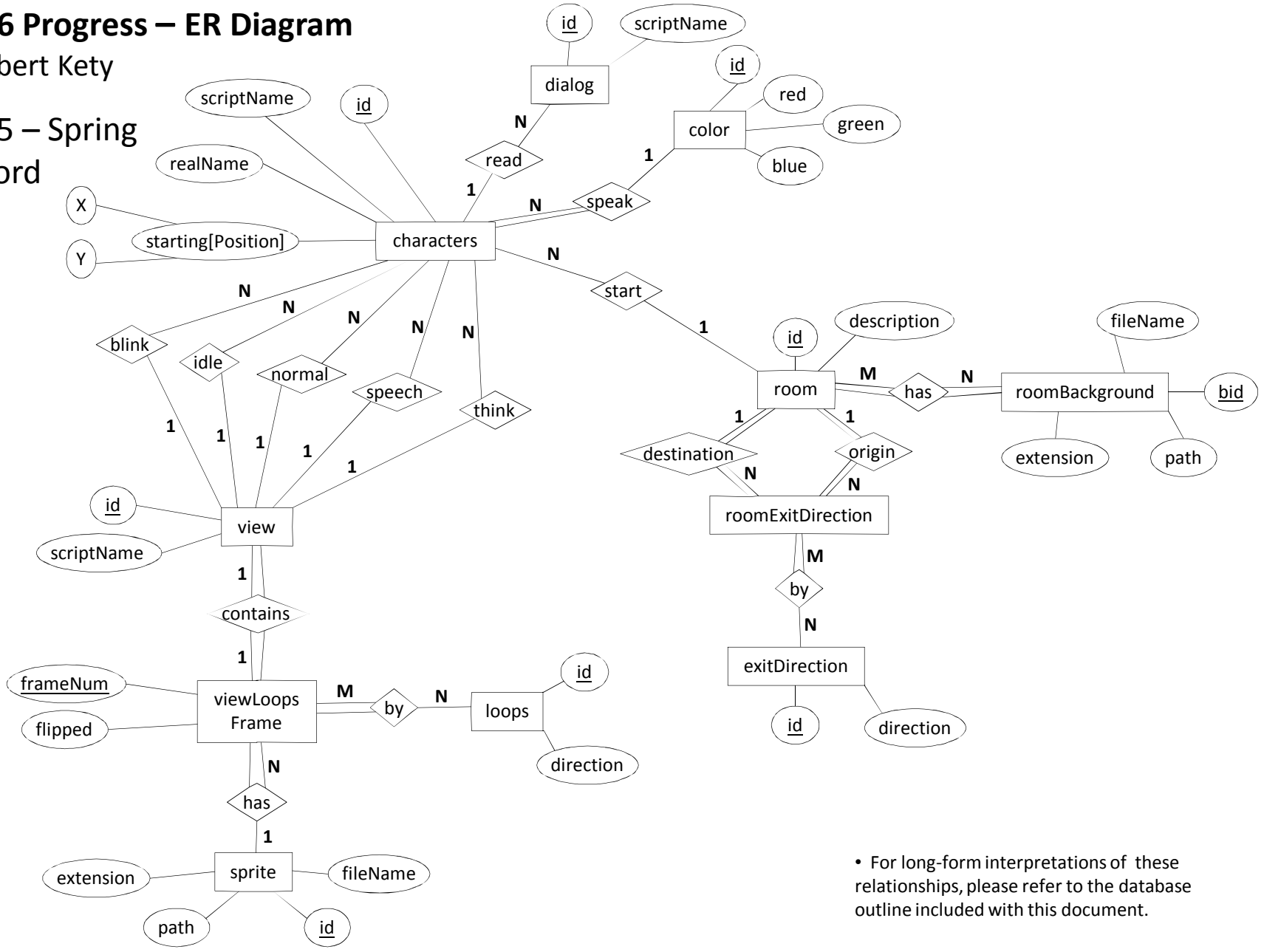
CS275 – Spring
Wolford



Hero6 Progress – ER Diagram

by Robert Kety

CS275 – Spring
Wolford



- For long-form interpretations of these relationships, please refer to the database outline included with this document.