**Story Blocker System Overview and Examples**

William Bares, College of Charleston
Donald Schwartz, Marist College

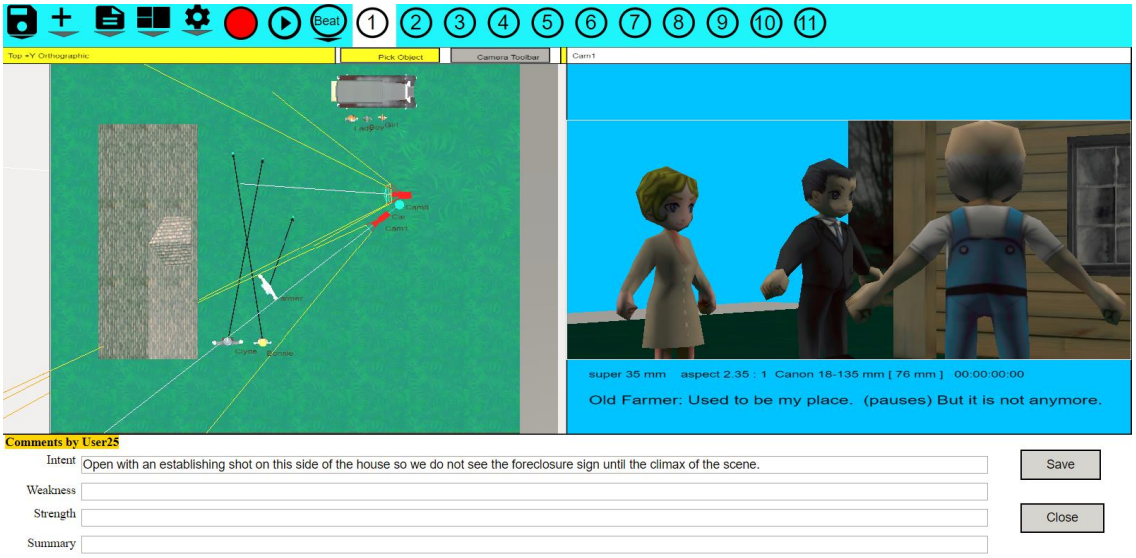
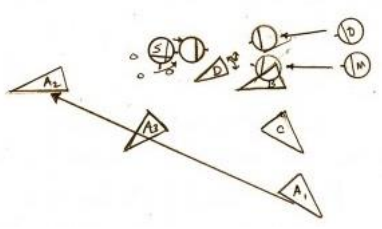



Fig. 1. *Story Blocker* Web-based, collaborative filmmaking system

## DESIGN GOALS

The motivation behind the design of *Story Blocker* was to create an accessible Web-based learning system that can offer more frequent, quality hands-on learning activities to overcome the problems often encountered when teaching the art of film, including the need for expensive equipment and movie sets, along with the time needed for all students to do hands-on activities with equipment. The system has the following technical design goals:

- Web-based application that runs on tablets, notebook computers, or desktop computers.
- Simulates look-and-feel of traditional filmmaking tools including storyboards and overhead diagrams. Overhead diagrams illustrate the relative positions and movements using a notation of labeled circles and arrows (Figure 2).
- Low-polygon 3-D character and object models that can be rendered in real-time on mobile devices.
- Server-side storage of user accounts and movie projects to enable collaboration, sharing, and communication.
- Easy-to-use interface that requires minimal learning time.




- Fig. 2. Sample top-down blocking diagram showing camera and character moves.

## IMPLEMENTATION

The *Story Blocker* system is a client-server architecture with a lightweight client managing the GUI and scene rendering and sending and receiving messages via a socket connection with the server. The server processes client messages, accesses the database, and transmits result messages to clients. The client encodes outgoing messages as JavaScript Object Notation (JSON) text, which is sent to the server over a Web socket connection.

*Server-side system.* The server is implemented as multi-threaded C++ code that maintains a queue of active client socket connections and a queue of received messages. On receipt of a message, the server converts the incoming JSON message into an equivalent C++ structure. It inspects the command type attribute and dispatches the appropriate command handler routine, which accesses the database, computes camera solutions, and packages the result object. Lastly, the server serializes the result object into JSON text and sends the message to the client-side Web application which initiated the incoming message.

*Database.* The database backend for *Story Blocker* consists of a two-layer design, which includes an abstract interface and an implementation developed using *Microsoft SQL Server 2015*. In addition, a simulated database component developed on the C++ standard collection library is included. The database tables that are most closely related to the examples described in this paper are described below.

Information about the users of the system includes login details, contact information, and social score details about the user's movies.

Details about the movies created or copied by the users include descriptive details (name, description, version), the movie's script and shot list, and access policies. The movie script includes a list of beats, each of which contains a description of the character(s) involved in the beat, their action(s) during the beat, their emotion(s) and dialog for the beat, any director notes for the beat, and a URL description of optional audio dialog. Data about each character includes his/her name, archetype (e.g., heroine), and description. Comments include feedback given by other users, directors and teachers.

A description of each Scene of a movie includes the scene name and description, a list of all the 3-D objects within the scene, and the timeline for the scene. The timeline breaks the scene into beats, specifying the start time and duration for each beat of the scene, and changes in position or orientation of scene objects. Access policies define who can read, copy and modify a movie.

*Client-side system.* The client-side graphical user interface (GUI) is implemented using JavaScript and HTML with theThree.JS graphics library used for interactive 3-D rendering, Howler.js to support audio, and Grid.JS to support editable tables used for writing characters, scripts, and shot lists, in addition to browsing lists of available movie projects. The Three.JS library is used to manage the position, size, and orientation of all virtual 3-D objects and render the animated scenes.

## EXTENDED EXAMPLE

Students and instructors begin using *Story Blocker* by visiting the home page where they are prompted to either login or register as a new user by choosing a unique name and password. We illustrate the functionality of the *Story Blocker* user interface by recreating a scene from *Bonnie and Clyde* (1967), in which the title characters meet a farmer and his family whose house had been foreclosed by a bank. Figure 3 presents a completed edit that closely follows the edits of the actual movie. The numbers below each frame correspond to beat numbers, each annotated by an abbreviated version of the dialogue or action descriptions.

For example, to begin an instructor creates a new movie project for *Bonnie and Clyde* by entering the movie title and a brief text description. After creating a movie, the user visits the movie content workflow menu which presents options to define characters, write the script, write a shot list, create blocking layout of animated characters, props, and cameras, screen a movie and write shot breakdowns, or screen a movie and write critique comments.

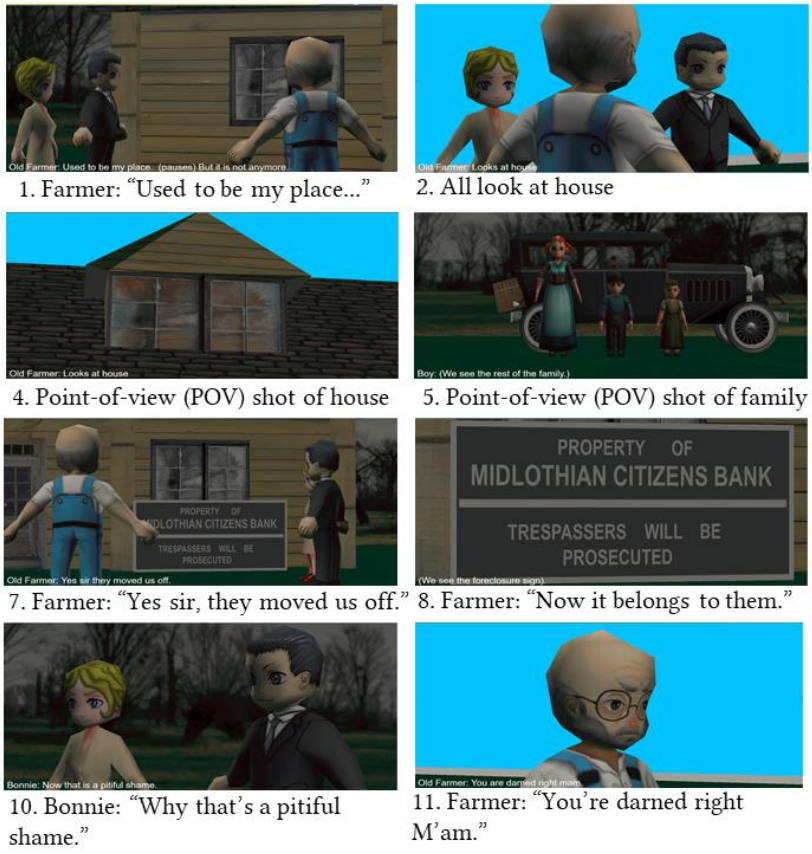



Fig. 3. Summary of camera shot edits for the Bonnie and Clyde scene.

The first step in defining the content of a movie is to define its cast of characters. Subsequent steps such as script writing present the user with pull-down menu choices populated with the names of these characters. For each character, the user enters a unique name, selects an archetype from a pull-down list of hero/heroine, sidekick, foil, villain, comic relief, or minor, and enters a brief description of that character's traits and motivations. For example, we might define the character Bonnie as a heroic archetype with a description of

"bored waitress who falls in love with Clyde to find adventure" (Figure 4). For brevity and clarity of text, only the first two characters are shown in Figure 4.

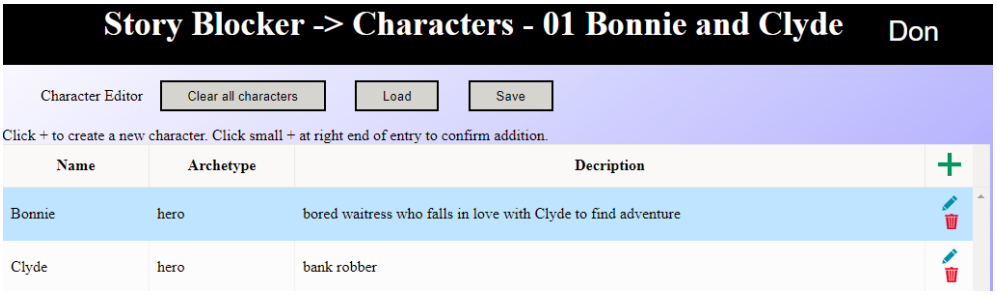



Fig. 4. Character entries for Bonnie and Clyde.

After defining a cast of characters, users may follow a sequential workflow in which they write the script that contains dialogue lines and character actions before entering the 3-D layout stage in which they block and animate characters and cameras. Alternately, a user may follow an iterative workflow in which they go directly to the 3-D layout stage where they layout characters and write script entries one beat at a time. In this example, we will follow a sequential workflow, which begins with script writing. In script writing, film students may take an instructor-provided story or create an original story, then divide the story into discrete logical units known as beats. One beat corresponds to one line of spoken dialogue or action. We simulate the first eleven beats of a scene in *Bonnie and Clyde* (1967), where a farmer tells Bonnie and Clyde how the bank foreclosed on his family's house (Figure 5). For legibility only the first three beats are shown.

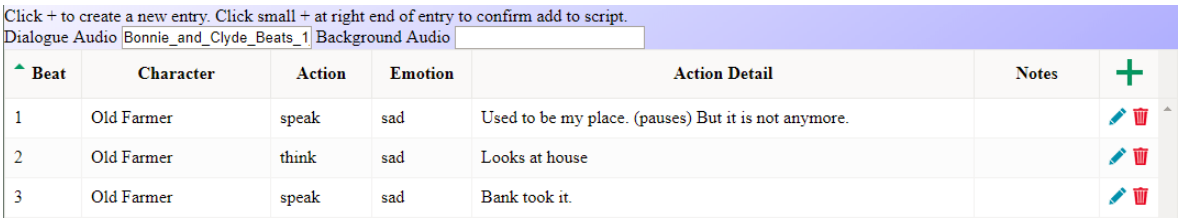



Fig. 5. Script writing and beat breakdown example.

In writing the script for each beat, the user selects the character from a pull-down list of characters defined using the Character Editor, then chooses the type of action as one of speak, move, facial expression, or think. Next, the user chooses the character's emotional cue as one of angry, sad, bored, happy, anxious, calm, fearful, timid, confident, loving, or jealous. After that, the user provides the textual content of that script entry by typing the dialogue line or stage direction. Users may optionally specify pre-recorded dialogue and/or background audio file(s) that will be heard during the animation.

After completing the script, an instructor or student may plan how to film the movie by creating a shot list that specifies the desired camera shot, camera movement, and lens effects for each beat. To add a shot list entry, the user chooses the beat number, camera number, and shot distance selected from extreme close-up (ECU), close-up (CU), medium (MS), medium close-up (MCU), medium long (ML), long shot (LS), and extreme long (EL). The user then selects the desired camera height relative to the action as either low, normal, or high. The user then chooses the camera move to indicate whether or not the camera is in motion during this shot and, if so, the type of motion, chosen from pan, tilt, track, crane, track to follow, or crane to follow. The user may specify a preferred type of lens by indicating its size in millimeters.

With the 3-D layout editor, users can block and animate movement of characters, props, and cameras to realize the actions and dialogue planned in the script, along with the camera shots and moves planned in the shot list. The 3-D layout editor presents a simplified animation timeline organized according to the beats defined in the script.

Figure 1 presented the 3-D layout editor configured to edit the scene consisting of eleven story beats indicated by the numbered circles. A user may create or delete beats by clicking on the beats menu icon. The window is configured to display two viewport windows, one for a top-down or overhead view of the virtual movie set containing three main characters and two cameras, and a second viewport that shows the view as seen through a selected movie camera. The top-down view is especially useful to visualize the blocking of the scene to see how the characters and cameras are set up. The lines extending from the two cameras indicate the triangular field of view regions visible to each camera.

To begin blocking a scene, a user first chooses which pre-built 3-D model will represent each character. To add an object to the scene, the user clicks on the Add object menu (+) icon, then chooses the type of object, such as camera or character from the pull-down menu. If the user chooses to add an object, the 3-D layout editor displays a tabbed gallery of thumbnail images which preview the available 3-D prop objects (Figure 6). Additional tabs will display similar palettes of available 3-D character models, vehicles, and backdrop scenes.

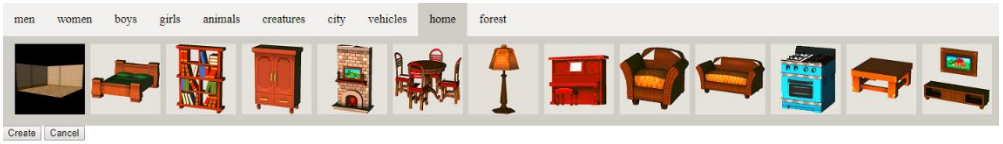



Fig. 6. Tabbed gallery of available 3-D object models.

To adjust the position or orientation of an object or camera in the top-down viewport, the user first clicks on that viewport to make it the current viewport (indicated by the yellow highlighted header bar). In the active viewport, the user can click the Pick Object button to indicate that subsequent mouse or touch inputs in that viewport will select or manipulate objects. In any viewport a user can simply click or touch (if a touchscreen is available) any character, prop, or camera. With an object selected, the user can drag the mouse to move an object. To rotate a selected object, they then click and drag the round rotation handle that extends from the selected object.

In a typical workflow, the user will add one or more movie cameras, each assigned a unique descriptive name. Each virtual movie camera sees the action from a different perspective. In editing, the user will select which camera's view is seen in the movie at each beat. The same direct-manipulation controls are used to move and turn cameras in the top-down view.

To make more precise adjustments to movie cameras, the user can configure a viewport to look through a selected movie camera. To do this, he or she would click on the Camera Tools button in a viewport header bar. This will open a palette of camera control tools including:

- Choose which camera view is seen through this viewport.
- Mark this movie camera to be the current edit, e.g., the movie camera whose view is output in the final film.
- Center the camera view on the currently selected object.
- Center the camera view on all objects in the current scene.
- Truck the camera position left, right, up, or down.

- Dolly the camera position forward or backward along its current aim direction vector.
- Pan the camera, rotating it left-right and/or up-down.
- Tilt the camera by rotating it clockwise or counter-clockwise about its aim direction vector.
- Lens zoom-in or out.

For example, suppose a user wishes to see a more dramatic canted angle shot for beat #7 from Figure 3. The user would click on the leftmost icon (Camera) to access a pull-down menu to select the beat #7's camera. This menu is also used to choose between standard orthographic top, front, and side views, editor perspective, a selected movie camera, or the edit camera that displays the director's desired camera view for each beat. The user would then click on the truck tool to raise the camera higher. Camera toolbar icons toggle yellow when clicked to indicate that subsequent mouse or touch gestures in that viewport will manipulate a specific camera property. Next, the user would employ the tilt tool to aim the camera down at the characters, then use the roll tool to introduce the canted angle. Figure 7 shows the resulting canted angle shot for beat #7.


Fig. 7. Using camera tools to compose a crane shot with canted angle.

After creating and placing cameras for each beat, the user could edit the virtual movie by indicating which camera's view is seen at each beat. To mark a cut (or edit) to a camera, the user would click the mark edit button in the camera tool bar. When a user presses the play movie button, the software will automatically cut to marked camera at the start of a beat. Users create animation by placing characters or other objects in different locations or orientations at each beat. The system will automatically interpolate in-betweens position and orientation over the duration of each beat, which the user can adjust.

Users can access dialog boxes to customize the movie display including letterboxing to match the movie's frame aspect ratio, showing details about the movie film format, camera lens, and current movie time in minutes:seconds:frames format, sub-title text of character dialogue lines, critique comment texts, font sizes, and colors for text overlays.

While inside the blocking editor, the user can access the content menu (third icon from the left) in the main toolbar. The user can choose to open dialogs to see the script, shot list, critique, or shot breakdown displays. If the user chooses to view the current script entry, he or she will see a panel (Figure 8) that shows the contents of the script entry for the current beat. If the user is editing a movie project, this form is editable. Otherwise, the form is read-only. This script entry dialog allows users the ability to write or edit the script entry for the current beat while working in the 3-D layout.

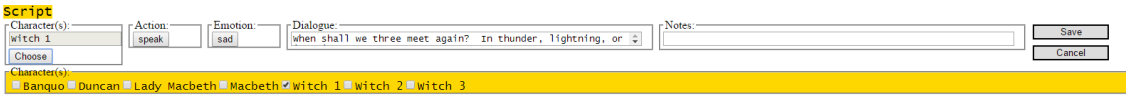



Fig. 8. View or edit script entry for the current selected story beat.

If the user chooses to view the current shot list entry, he or she will see a panel (Figure 9) showing the contents of the shot list entry for the current beat. If the user is editing a movie project, this form is editable. Otherwise, the form is read-only.

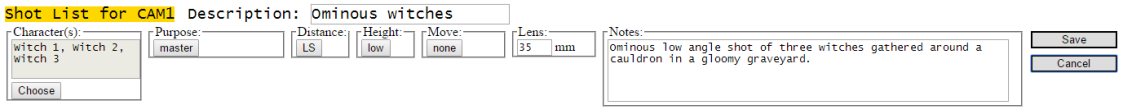



Fig. 9. View or edit shot list entry for the currently selected story beat.

If the user wishes to view or create a shot breakdown, the system presents a panel (Figure 10) that prompts the user to describe the cinematic properties of the current shot by choosing selections from multiple drop-down categories including:

- Purpose – choose the role of the shot as master, point-of-view, over-the-shoulder, shot-reverse-shot, cut-away, or cut-in.
- Continuity – choose editing continuity as respect the line of action, cross the line of action, or none.
- Distance – identify shot distance as extreme long shot, long shot, medium shot, medium-close shot, close-up shot, or extreme-close up shot.
- Height – identify camera height relative to the subject as low, normal, or high.
- Angle – identify the relative horizontal angle between the camera and the subject as one of frontal, left, etc.
- Move – identify the type of change in camera position as one of dolly-in, dolly-out, truck, crane, etc.
- Turn – identify the type of camera rotation as pan or tilt.
- Lens – identify type of lens effect as one of shallow focus, deep focus, etc.

Figure 10 shows the shot breakdown panel with the list of shot distance choices expanded. After the user selects a category, the expanded set of choices is collapsed.

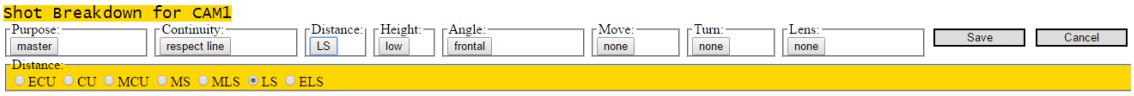



Fig. 10. Choose shot distance from expanded menu of multiple choice options for shot breakdown.

The critique commentary dialog visible along the bottom of Figure 1 prompts the user to write a deeper analysis of the current beat by describing the filmmaker's intent, weaknesses of the blocking or composition, strengths, and summary.