ESCAPE!
Player Manual and Game Specifications

By Chris Eng and Ken Rice
CSS450 Fall 2008

Contents

Player Manual

Object of Escape!

You control the fate of the Hero, who has found himself lost inside of a haunted mansion. Guide the Hero through each labyrinthine floor to *Escape!* to safety!

How to Play
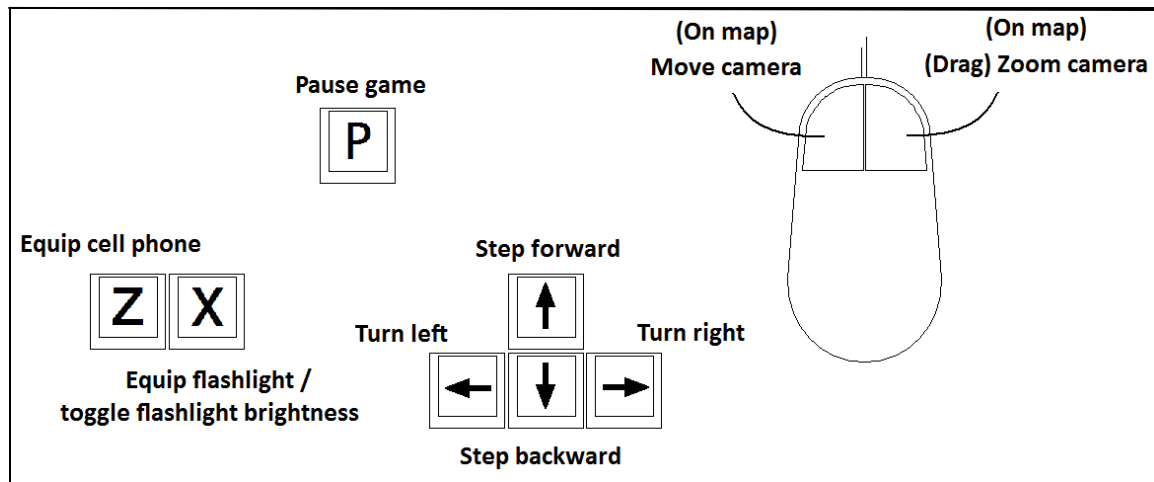
1. Controls

Figure 1. Game controls

Figure 1 shows the game controls. Guide the Hero forward, backward, left, or right using the arrow keys. Have the Hero equip the cell phone with Z, or equip the flashlight with X. If the flashlight is already equipped pressing X will toggle the beam intensity between low and high. Pause the game with P. Left mouse clicking on the map will reposition the camera position. Right mouse dragging on the map adjusts the camera zoom.

2. Game Display

Figure 2 shows the game display. The game display consists of six components: the *Main View*, the *Floor Map*, the *Hero Health* bar, the *Flashlight Battery Life* bar, the *Collected Keys* box, and the *Game Menu*.

The Main View is where primary game play occurs. Here is where you guide the Hero through each floor, collecting items and avoiding threats.

The Floor Map is initially obscured by several tiles. When you guide the Hero to a location hidden by a tile on the Floor Map, that tile will be lifted. As you explore, you will be able to see more of the Floor Map!

3

The Hero Health bar will be depleted if the Hero is attacked. Do not let the Hero's health reach zero, or else the game will be over!
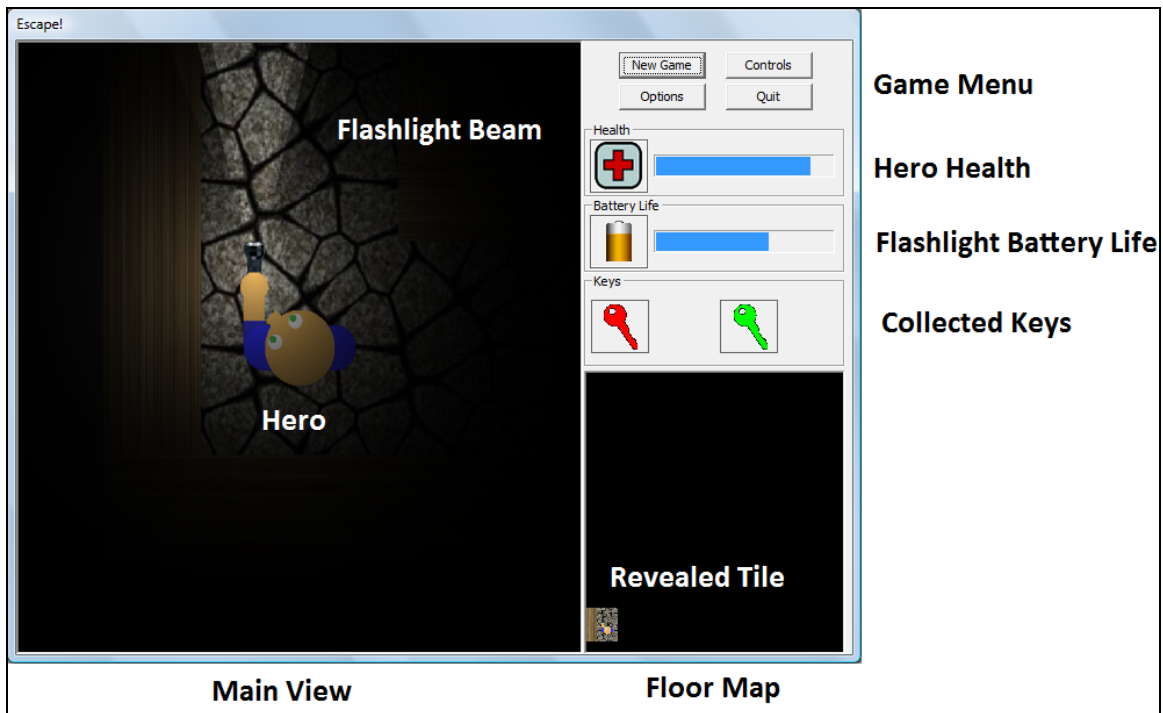

Figure 2. Game Display

The Flashlight Battery Life bar displays the flashlight's remaining power. The flashlight's battery will gradually deplete as long as the flashlight is equipped. If the flashlight's battery reaches zero, the Hero cannot use the flashlight anymore.

The Collected Keys box displays the key(s) that the Hero has picked up. If you have the Hero unlock a door, then the corresponding key will be removed from the Collected Keys box.

The Game Menu consists of four buttons: *New Game*, *Options*, *Controls*, and *Quit*. Clicking New Game will start a new game of *Escape!* Clicking Options will pause the game and display the Options window; here, you can adjust the game difficulty to your liking (note: if you adjust the difficulty, it will not be put into effect until you click New Game). Clicking Controls will pause the game and display the Controls window; here, you can see how to guide the Hero. Finally, clicking Quit will close the *Escape!* application.

3. Advancing Floors & Other Obstacles

On each floor, guide the Hero to the exit. An opening in the wall perimeter denotes an exit.

Along the way, a locked door may block the Hero's path. A door is colored red, green, blue, or yellow. To unlock a door, the Hero must pick up a key of the same color and bring it to the door as shown in Figure 3.
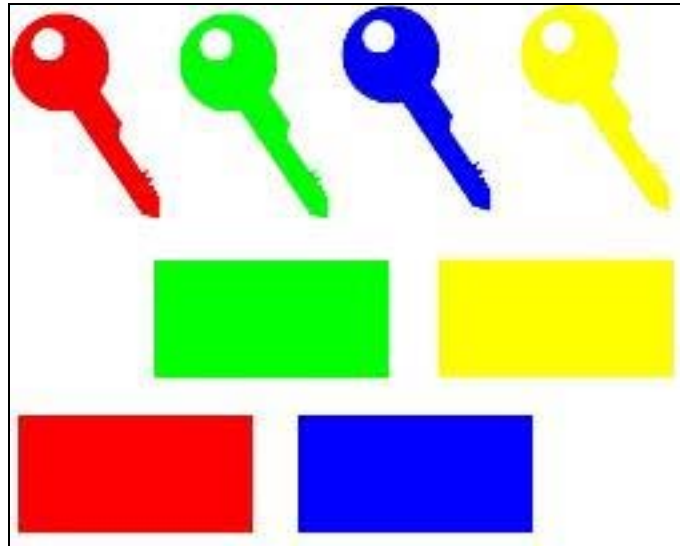
Figure 3. Colored Keys and Doors

The Hero is not alone in the mansion. Be careful to keep the Hero away from Ghosts and Blobs shown in Figure 4 that haunt the residence! They will attack and harm the Hero when found. Fortunately, you can have the Hero toggle the flashlight beam to high mode: if a Ghost or Blob is caught in the high beam, they will be temporarily blinded, allowing you an opportunity to guide the Hero to safety.
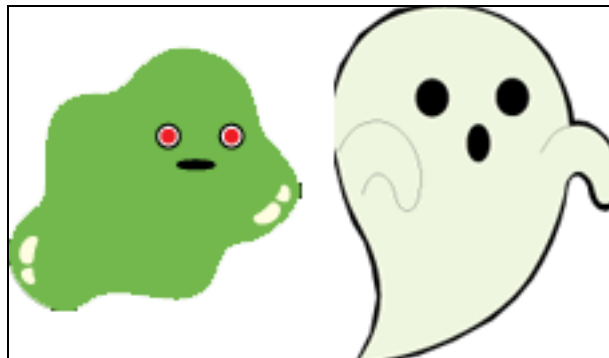

Figure 4. Blobs and Ghosts

4. Gameplay Tips

The flashlight, though a very important tool for guiding and protecting the Hero, is not an unlimited resource. The flashlight battery will slowly deplete as long as it is on, and it will deplete even faster if the flashlight is in high beam mode! If the battery life falls to zero, then you cannot have the Hero use the flashlight.

So what can you do to maintain battery life? Obviously, keeping the flashlight in normal beam mode will help. In addition, you may consider having the Hero equip the cell phone, which does not use any of the flashlight's battery! The downside of the cell phone is that its illumination is very poor.

If you are not into conservation, you can replenish the flashlight's battery if you find other batteries hidden somewhere in the mansion. Each new battery will completely recharge the flashlight!

If a Ghost or Blob attacks the Hero, the Hero's health will be reduced. When this happens, the Hero will be granted a brief period of invincibility as indicated when the Hero begins blinking. Use this opportunity to guide the Hero to safety!

The tiles in the Floor Map will gradually fade back in, obscuring parts of the map that you had revealed. The rate that the tiles fade back in depends on what the Hero has equipped: the tiles fade in very slowly if the Hero is has the flashlight on high beam mode, less slowly if the flashlight is on normal beam mode, or very quickly if the Hero is holding the cell phone!

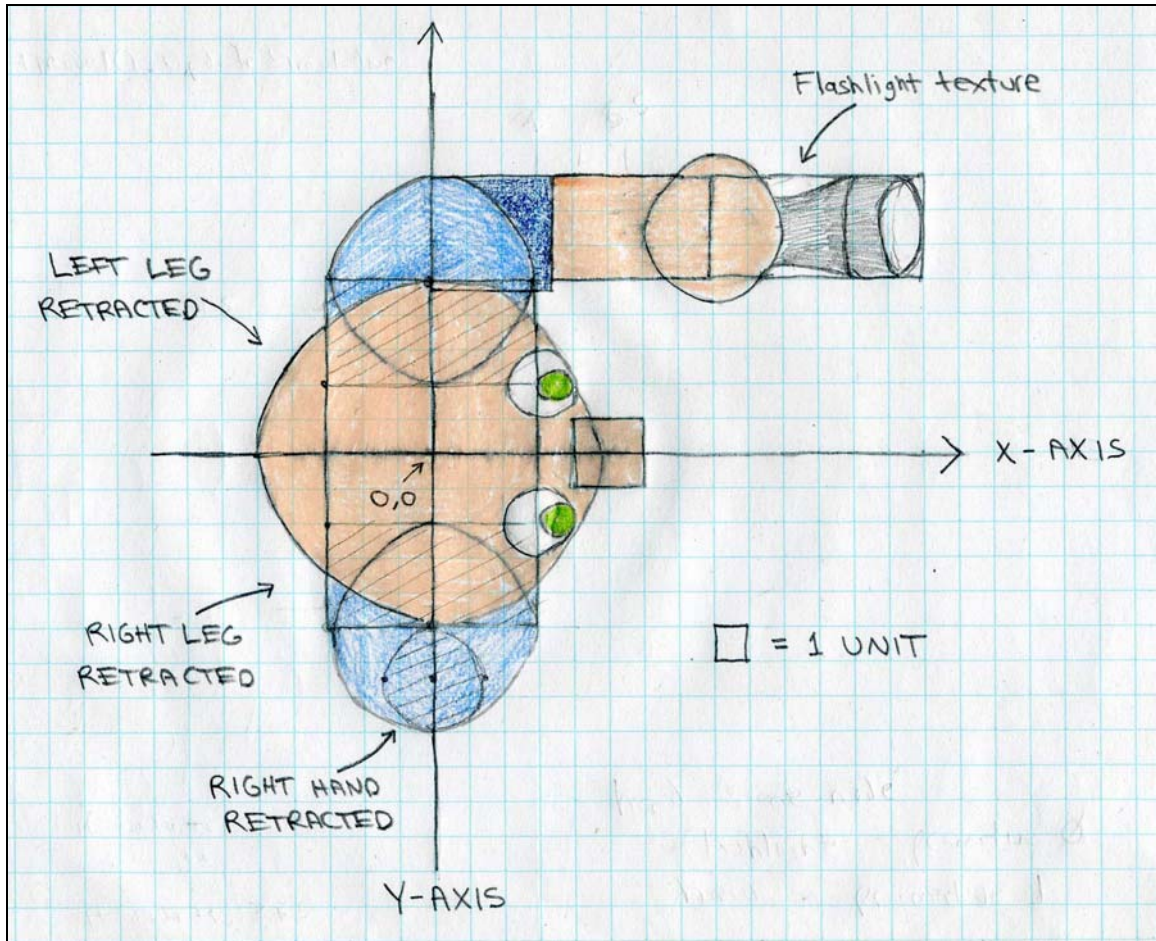# Hero Graphic Hierarchy

Geometry

Figure 5. Hero's geometry

The geometry shown in Figure 5 uses the drawing order of the primitives to achieve the hero's appearance. Also note that both legs and the right hand are initially drawn retracted into the body so that they are not seen. When the hero walks these three primitives animate to achieve the effect of walking.
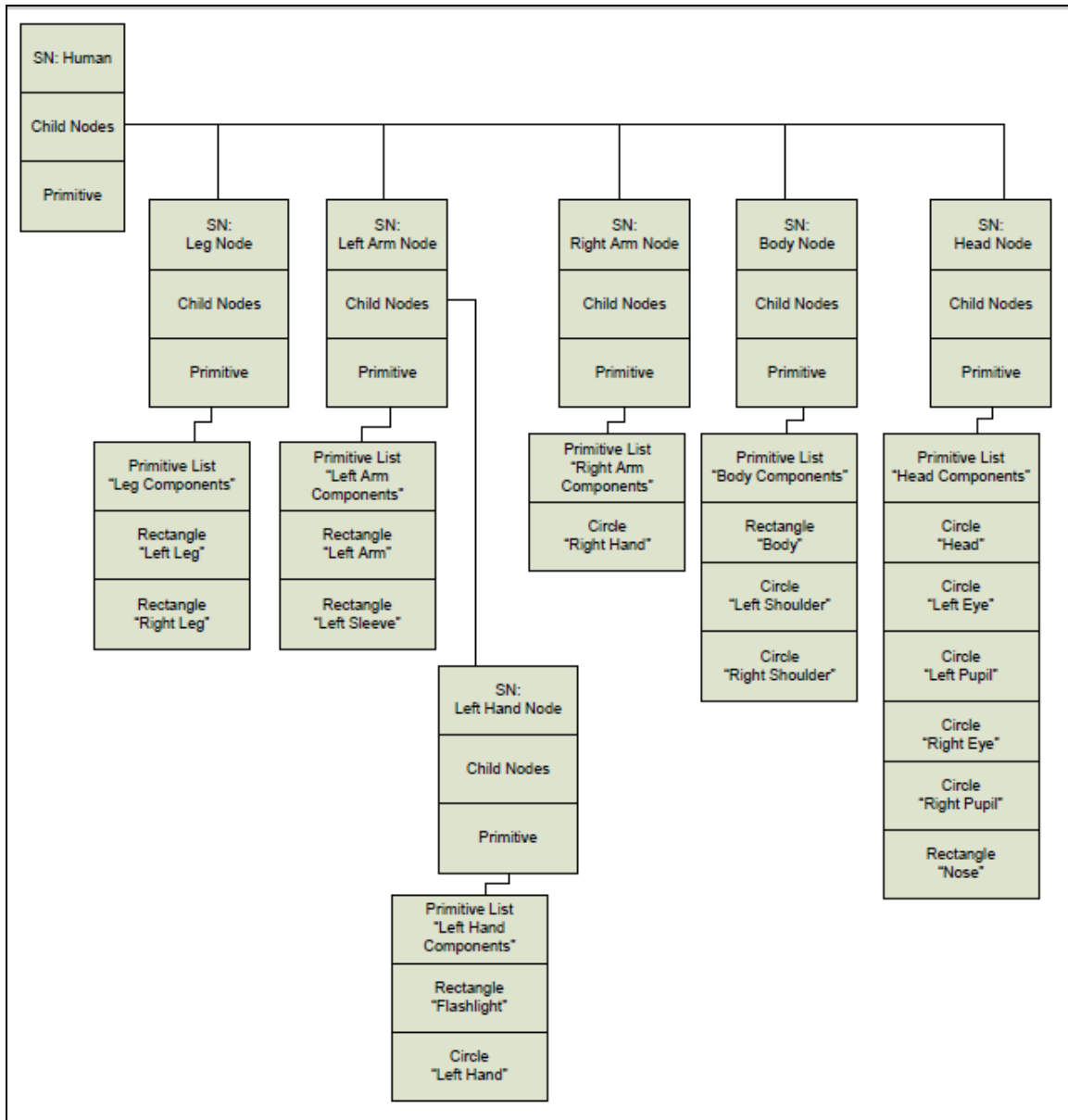
Data Structure

SN: Human
- Child Nodes
- Primitive

SN: Leg Node
- Child Nodes
- Primitive

Primitive List "Leg Components"
- Rectangle "Left Leg"
- Rectangle "Right Leg"

SN: Left Arm Node
- Child Nodes
- Primitive

Primitive List "Left Arm Components"
- Rectangle "Left Arm"
- Rectangle "Left Sleeve"

SN: Left Hand Node
- Child Nodes
- Primitive

Primitive List "Left Hand Components"
- Rectangle "Flashlight"
- Circle "Left Hand"

SN: Right Arm Node
- Child Nodes
- Primitive

Primitive List "Right Arm Components"
- Circle "Right Hand"

SN: Body Node
- Child Nodes
- Primitive

Primitive List "Body Components"
- Rectangle "Body"
- Circle "Left Shoulder"
- Circle "Right Shoulder"

SN: Head Node
- Child Nodes
- Primitive

Primitive List "Head Components"
- Circle "Head"
- Circle "Left Eye"
- Circle "Left Pupil"
- Circle "Right Eye"
- Circle "Right Pupil"
- Rectangle "Nose"

Figure 6. Hero's data structure

Escape! System Evaluation

Known Bugs

There are a few bugs remaining in the current version of *Escape!* On floor three, the blue and green keys are incorrectly swapped in color, but is ironically difficult to fix due to underlying code.

Occasionally the Ghost will disappear from the floor once there are no keys remaining.

The flashlight beam mode and battery depletion rate can become unsynchronized when advancing to the next floor. For instance, if the Hero has the flashlight beam on high mode when exiting a floor, the battery life will continue to drain at the high rate at the start of the next floor even though the flashlight is showing the normal beam.

If a player moves the camera position to a new location on the map, the current light source will be displayed there, disjointed from its source (for instance, the flashlight beam will be displayed in locations very far from the flashlight itself).

Limitations

Escape! was built with very ambitious goals in mind. While many of the target features were successfully implemented in the game, there are a few that did not. In addition, some target features that *did* make it into the game were unable to be fully polished.

First the monsters: three monster types were planned for the game. While the Ghost and Blob were implemented, the third type –the Zombie– did not. The Zombie was to wander each floor much like the Hero, except it would not be very smart and frequently walk into walls. Also, multiples of each monster type were in the works, but unfortunately did not make the final version.

Second, an exact collision box for the flashlight's high beam has not been implemented. Though the high beam will stun monsters, players do not need to turn the Hero to catch enemies with it. As long as a monster is onscreen, the high beam will stun it.

Third, the sounds in *Escape!* are played via Window's PlaySound(…) function call. While this provides a fairly lightweight avenue for audio effects, PlaySound(…) is not very sophisticated: only one sound may play at a time, meaning a new sound will stop any sounds playing previously. Though not a game play issue, the integrity of the game's spooky atmosphere is weakened when this happens.

Fourth, due to the tiles that obscure the Floor Map, the floor sizes could not exceed beyond about 27x27 tiles (729 tiles total) without incurring substantial game play slowdown. Since the tiles fade in by a small amount each frame (at approximately 40 frames per second), they are quite taxing on the processor.

Fifth, file saving/loading functionality was not implemented in this version of *Escape!* The related buttons in the Options dialog are disabled. On the other hand, the game is relatively short, so saving/loading would not be very useful, we reason.

Lastly, true illumination is not implemented here. That is, instead of brightening dark surfaces and objects with the flashlight or cell phone, bright surfaces and objects are merely darkened in a manner that looks like illumination. Because of this, a few unavoidable graphical inconsistencies can be seen in *Escape!*, especially when scrolling the camera.

Possible Extensions in Future Versions

In future versions of *Escape!*, each of the limitations listed above should be addressed. In addition, several more levels may be implemented now that the core game engine has been implemented, as level creation is the easiest game component to handle.

Future versions of *Escape!* may be rewritten to enable multiple doors of the same color, opening up design space for more sophisticated floor patterns. More monster types may be implemented, allowing for more diverse Hero-monster game play interaction as well.

The textures (and in general, the graphics) of *Escape!* would benefit from more specialized artwork. The current version reuses a few textures many times, causing several unpleasant seams where walls connect. Elsewhere, untextured primitives are used for illustrations, but they unintentionally stand out in comparison to their textured neighbors. Similarly, the Hero could be better detailed to achieve a higher sense of realism.

Finally, a true illumination routine should be implemented in future versions of *Escape!* If this can be achieved, then several of the illumination-related bugs and patches in the current version may be eliminated due to streamlined ray-tracing techniques.