GSI: Gruff Scene Investigations

Trenton Gailey, Ben Glass, Bradley Kemp, Joseph Larsen, Peter Siburg, Braydon Spangler, Eric Szmutko, Yoshihiro Kobayashi

Arizona State University

University Drive and Mill Avenue, Tempe, AZ 85281, USA

tgailey@asu.edu, bglass1@asu.edu, bkemp1@asu.edu, jdlarse1@asu.edu, psiburg@asu.edu, bcspangl@asu.edu, eszmutko@asu.edu, ykobaya@asu.edu

Abstract

The goal of this project was to create a realistic review game for a High School level forensic science course hosted by ASU Prep Digital. We integrated information from the course material into a 3 stage game that takes the player through the 3 primary stages of crime scene investigation: evidence gathering, evidence analysis, and testimony in court. All assets were created by members of the team using a variety of tools.

Author Keywords

Forensic Science, ASU Prep Digital, Crime Scene Investigation

Introduction

The game itself comprises 3 separate phases: evidence gathering on the scene, evidence analysis in a lab, and final testimony in court. All phases of the game were designed based on ASU Prep Digital's Forensic Science 1 course as well as input from ASU Professor Kimberly Kobojek. The evidence-gathering phase takes place in an alleyway crime scene and consists of an initial examination of the scene, a detailed examination of the scene with marking and photographing of the evidence as well as the scene at self and a dumpster dive section to collect even more evidence. The evidence analysis phase takes place within a crime lab and consists of the player taking the various pieces of evidence through several different types of analysis including fingerprinting, blood typing, gun identification, bullet matching as well as cause of death analysis. The final stage of the game is a simulated expert witness testimony in which the player will be asked about key details of the crime and will be able to use notes generated after their completion of the different segments of the game.

Background

The following is a description of the hardware and software technologies used during the project.

Unity 2019.2.19f1

Unity is the world's leading real-time 3D development platform, providing a wide array of built-in tools to allow for the creation of games and deployment across a wide range of platforms including AR, VR, mobile, console, web, and pc. Unity 2019.2.19f1 was chosen for its known stability and shared knowledge of function amongst the team developers and coders.

Substance Painter 2019.2.3

Substance Painter by Adobe is the industry standard for non-destructive realtime 3D asset texturing and painting. Substance Painter was utilized to texture all of the 3D models in the game.

Blender 2.82a

Blender is a free and open-source 3D creation suite that supports the entirety of the 3D and 2D pipeline including modeling, rigging, animation, simulation, rendering, compositing, motion tracking, and video editing. Blender was used to model and animate all of the 3D models in the game.

Adobe Creative Cloud

The Adobe Creative Cloud is a collection of more than twenty desktop and mobile apps and services for photography, design, video, web, UX, and much more. Of the selection of programs within the Adobe

Creative Cloud, Photoshop and Illustrator were utilized to design and create the 2D assets in the game.

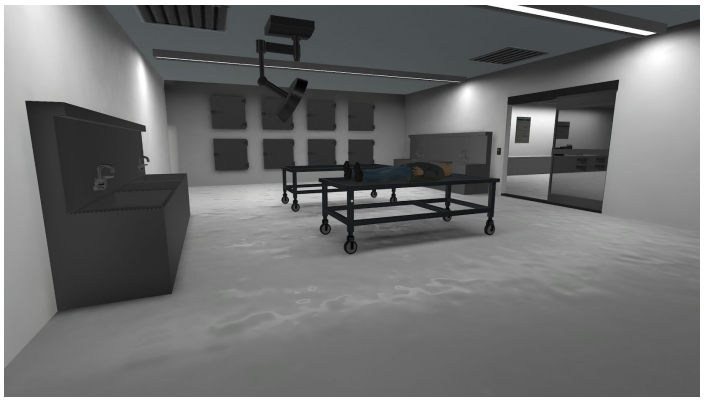
Logic Pro X

Logic Pro is a Digital Audio Workshop (DAW) used for creating audio files and/or manipulating digital audio files. Logic Pro offers a set of synthesizers, both mono, and poly, and effects to emulate any instruments that you can create. Logic Pro was used to create all of the music and sound effects used.

Game Design

Artistic Style

The artistic style for the game follows a realistic style approach that reflects a close resemblance to the actual items and tools that one would find in the real world. To accomplish the semblance of photorealism while maintaining an efficient runtime environment, low-poly models are combined with high-poly baked maps to preserve as much detail as possible. This combination of simplistic model design with high-quality textures allows for the creation of realistic environments that aid in replicating the actual feel of a true crime investigation.



Mini-Games

The use of mini-games are used throughout the game in order to reinforce and test the player's knowledge of certain forensic science skills that are taught over the course of the forensic science course.

A list of the various mini-games that the player will play through in the game:

1) Dumpster Diving
    - The player will have to sift through a dumpster at the crime scene in order to ascertain extraneous trash items from possible evidence of the crime scene.
2) Photography
    - The player must show their skills at identifying proper camera angles and photography locations when initially documenting a crime scene.
3) Matching
    - The player will have to use evidence from the crime scene to match against DNA, fingerprints, bullets, or guns documented in a computer database in order to determine suspects, identify weapons, and piece together a narrative for the crime scene.
4) Body analysis
    - The player will have to examine wounds located on the body from the crime scene in order to determine the cause of death.

Game Flow

The following is a description of the game flow that the player will experience as they progress through the game.

1) Document scene and collect evidence
2) Analyze evidence in the lab
    a) Compare evidence to control samples from suspects
3) Determine culprit and testify in court

Systems Implementation

The majority of the interactive portion of the game's code-base was structured around the concept of **Interactables**. This generic concept allowed us a framework around which to build the tasks items needed to be able to perform. Each different type of interaction was an interactable script which allowed us to easily group them, check for completions for the narrative controller, and even have multiple interactables on the same object. Dialogue and Journal interactions were also triggered through an interactable and then passed on to their respective systems.

[Example of dumpster minigame - utilizing many different interactions]

This point leads to the second major system implementation: the Journal. The journal is a constant element through the game, which does not destroy after the scene ends. Since the journal is guaranteed to be one constant element, we access its commands through static functions. Through static calls, we can add images and text to the journal, which can be accessed anytime using the **[TAB]** key. This allows us to use the journal as a tracker for everything the player does. This means that the player does not have to remember everything that happens in the game, and can refer to the journal for name and object specifics.

[Example of the Journal in the game]

The third major aspect of the code-base was the dialogue system we created. Since this game was designed to be narrative / dialogue heavy, it was important to develop a system that was flexible enough to allow quick changes to dialogue and powerful enough to cause actions and create events from within the dialogue system. This was done through a dialogue reader, which has a list of commands and parsings that afford the writing of complex dialogue systems from within a single text file. The following block shows an example of a text file that we used:

> Police Officer "Victim has yet to be ID'd. Unsure what he was doing here, but the body seems cold. Could've been here for a long time."
> "The police officer's initial report has been added to the journal" AddToJournal(First responder officer found dead body in alleyway with no witnesses. Body was cold upon arrival implying the victim has been dead for a while.)
> Det. Gruff "Alright, first thing's first. Let's walk around and take a look with what I'm working with."

This dialogue file shows how flexible the system is at incorporating different people talking and using commands such as AddToJournal. This dialogue system was modeled after an open-source visual novel system tutorial, which also facilitated the control of dialogue and the unity scene through text files (Stellar Studios, 2018).

[Example of in game dialogue with branching choice]

The final system we built was the matching minigame system. It is designed to facilitate different variations on a matching game. The designer can choose 4 images to be attached to the minigame and then give them a position in world space. The player can then scroll through them and compare each image to a sample/comparison image shown on the user's interface.

[Example of matching minigame (fingerprints)]

Conclusion and Future Work

In conclusion, we have succeeded in creating a game that effectively presents the major points of crime scene investigation, evidence analysis and how the evidence is presented in a court of law. The game is feature and content complete with all art, sound and system assets being created by members of the team. The Dumpster diving, matching, body analysis and photography minigames are all implemented in such a way to be re-usable and extendable. The game has received positive feedback for the sponsors and others involved in the supervision and evaluation of its development.

Much can be done with all the game's art, sound and systems assets. The systems created in this project can be re-used to create other crime scene scenarios to further test a students knowledge and educate them about the functions of crime scene investigation. The game's systems can also be built upon to add more minigames like possible witness interviewing and interrogation. The art and sound assets are all high quality and could be used in future iterations of the project as well as other projects.

References

Stellar Studio. (2018, January 10). Making a Visual Novel for Free [Video Playlist]. YouTube. https://www.youtube.com/playlist?list=PLGSox0FgA5B7mApF1vhbspLj5NpzKedU6