Tailoring Horror Games by Integrating Bodily Sensors

Student Name: Steven Lowes

Supervisor Name: Magnus Bordewich

Submitted as part of the degree of BSc Computer Science to the

Board of Examiners in the School of Engineering and Computing Sciences, Durham University

2019-01-25

Abstract —

A Context/Background

Context/Background

- Horror games are a large industry
- People become thrilled by being scared
- Games currently are a one-size fits all approach - Compare to non-game horror things? Haunted house etc? Do they tailor it to the individual
- People are different, and respond differently to being scared

Aims

- Implement and test a system which alters the frequency (and type?) of scary events in a horror game based on feedback from bodily sensors

Method

- A simple horror environment will be created in the game minecraft.
- Jump scares can be triggered programmatically.
- Create an adaptive method for scaring users which uses information from GSR sensor
- Compare this ”adaptive” method to two naive solutions (random scare placement and even spread)

Proposed Solution

- After a jump scare, data from the GSR sensor can be examined to determine how scared the player was
- If the player is not as scared, allow more time between jump scares
- If the player is very scared, increase frequency of scares

Keywords — Galvanic Skin Response, Electrodermal Activity, Human-Computer Interaction, Video Games

I INTRODUCTION

- Horror industry
  - Examine horror industry, size, especially gaming
  - Comment on lack of tailoring to the individual
  - Compare with tailored scare experiences
  - Maybe there is some research on this
- GSR
  - Explain the mechanism of GSR
  - Response time, average curve + recovery
  - Explain cause, size of effect that is needed to cause a response
  - Talk about other uses
  - What kind of things is it best at measuring

A Project Aim

- Do users enjoy playing a horror game more when its scares are tailored based on feedback from bodily sensors? Is galvanic skin response sufficient to understand a user's response to a scare?

B Deliverables

1. Minimum Deliverables
   - Create an immersive game environment in which you can control events such as the arrival of new enemies and their location.
   - Create a system for tracking the user EDA measurements and game events simultaneously.
   - Create a standardised game setting for users to play through and record EDA measurements as they progress. Record some user experiences.
   - Determine what game events trigger responses and select events to use in next deliverables.
2. Intermediate Deliverables
   - Analyse data from users and try to determine: susceptibility to expected new shock events, e.g. underlying tension or delay since last event.
   - Create one or more adaptive systems for triggering events at moments of maximum impact.
   - Create a (null hypothesis) system for random generation of events.
3. Advanced Deliverables

- Conduct a user study to determine whether the adaptive systems give a better user experience than the random system.
- Revise the adaptive system based upon empirical evidence obtained.

II DESIGN

A Requirements

- Functional Requirements
- Nonfunctional Requirements

B In-Game Environment

The environment will be a haunted house. Users will be given the goal of finding a number of items throughout the house, but they will not be able to reasonably complete the task within the time. This just ensures that the users have a purpose and continue to explore the environment

C Jump Scares

Describe the jump scare briefly, show a screenshot from the game and explain the sound
Show some GSR data from a jump scare?

D Implementation

D.1 Choice of Game

The game Minecraft was chosen due to its comprehensive modding api, cross-platform compatibility, customisability, and ease of use when it comes to creating environments.

D.2 Hardware

What arduino, what sensor
What code is the arduino running?
How is communication happenning? (serial + jrxtx library)

D.3 Choice of Language

The language Kotlin was chosen due to my familiarity with it, and its interoperability with java, which Minecraft and its modding api use. Kotlin also allows code to be written more succinctly with less boilerplate to improve productivity and expressiveness.

E Evaluation

Success will be measured based on a combination of data from the sensors, and players self reported enjoyment + scaredness on a 10-point scale.

E.1 Controls

Each user will play the game once. Repeat playthroughs are not reasonable as it is expected that the game will become less scary after one playthrough. Each adaptive-method playthrough will be matched with some naive method playthroughs to control for the number of scares and total playtime. (Adaptive algorithm will have a differing number of scares each time, and for each adaptive run-through we'll do one of each naive method pre-set to have the same number of scares)

Playthroughs are anticipated to last around 10 minutes (will be predetermined) and contain around 10 scares (will be determined based on the adaptive algorithm).

F References

-

References

Budgen, D. (2003), *Software Design*, 2nd edn, Addison Wesley.

Euther, K. (2006), Title of paper. unpublished.

Futher, R. (2006), Title of paper 2. submitted for publication.