

# Using Player's Emotional Input to Influence the Difficulty of a Game

## Introduction

The purpose of the project is to use the emotional input of the player to influence how a game is played. It can be difficult to correctly gauge how hard a game should be for a player, so having a way of influencing the difficulty from the user's emotions is a way of limiting frustration from video games.

### Objectives

- Research the literature on affective gaming
- Choose technology appropriate for creating a game
- Create a game with factors that can be changed based on emotional input
- Test the game with a sample of players

## Basic Concept of affective gaming (How to gauge - ACE)

Assist me - The game itself can measure how a player is doing given a task. If a player is struggling with a certain task (taking too long), hints can be given in the hope to alleviate some of the player's frustration.

Challenge me - This can be difficult to cater for, as every player has an individual skill level. A way game developers have tried to provide for this problem is having multiple difficulty levels - dependent on how the player perceives their own expertise.

Emote me - Game developers have tried to provide gameplay to make the player feel a certain way in a given moment. However this can be somewhat difficult to achieve as players may not feel the same way the developers hoped they would, causing a severe disconnect between developer and player.

How we can remedy this: "By measuring the user's actual emotional state, the game can modify its content to once again provoke the

['Gilleade05 Affective Videogames and Modes of Affective Gaming A...ge Me Emote Me'](#)

## Technology used

Unity - A game engine that allows for the development of games, with the use of scripts and game objects.



C# - The best language to use in conjunction with Unity, as it is a language that is very flexible with a short compilation time.

Microsoft Visual Studio - The IDE used to code the scripts in C# for Unity, it has connectivity with Unity that allows scripts to be opened easily through Unity into Visual Studio - improving workflow.



Example of game environment created in Unity

## Methods

The way the game works is to have the user carry out challenges that can change on the go, based on an emotional input from the user. This will make the game easier or harder.

Scripts are applied to in-game objects, which gives them different behaviour/interaction with the player. These factors can be changed based on how the user is feeling. For example, if they are angry, the game can become easier:

- Enemies move slower.
- Enemies deal less damage.
- Player does more damage/has more health.

The same can be done to make the game harder, for a higher incentive to get more points.

## Expected Results

- It is expected that the game difficulty can be changed by the user, creating a tailor made experience for the player - reducing frustration.
- The score system will give weighted score based on the player's difficulty level.
- The process of changing difficulty level should be easy for the player; to reduce frustration caused by the game.



Example of the game recording the score of the player

## Conclusion & Future work

In conclusion the game can change difficulty on the go, further testing still needs to be carried out on other students to see how well the game works in practice.

- Adding different forms of emotional gauge. For example, there could be controller input. How hard the player is handling their controller could be an indication of how angry they are.
- Incentives for choosing a higher difficulty. This could be in the form of a leader board, or getting more score from choosing a higher difficulty.

Author: Tom Kent Myers  
[tmk18fsd@bangor.ac.uk](mailto:tmk18fsd@bangor.ac.uk)

Supervisor: L.I. Luncheva  
[l.i.kuncheva@bangor.ac.uk](mailto:l.i.kuncheva@bangor.ac.uk)