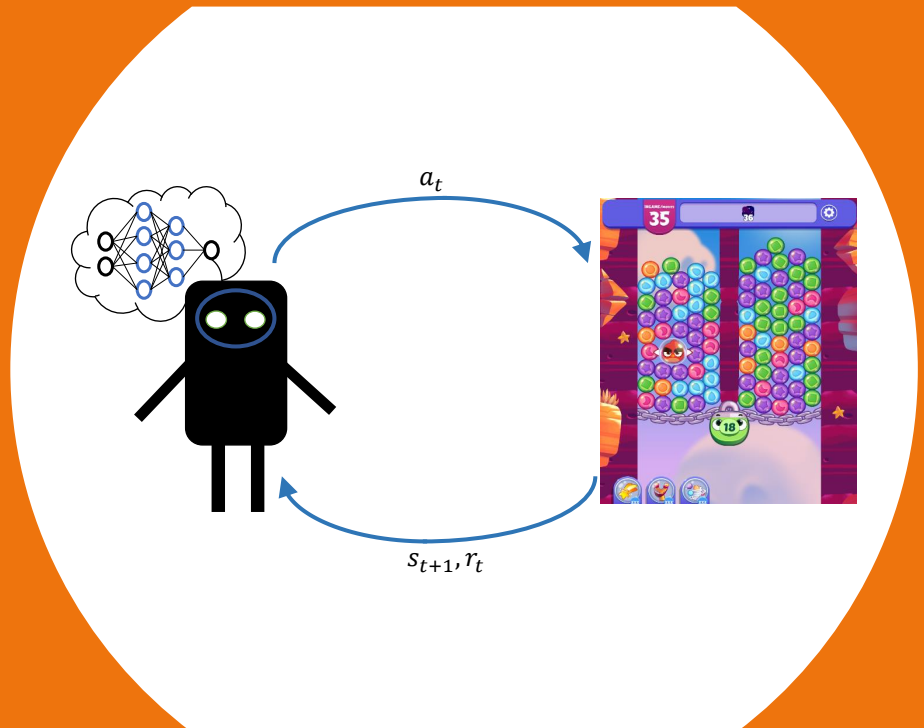


# Advances in AI-assisted Game Testing

Shaghayegh Roohi



# Advances in AI-assisted Game Testing

**Shaghayegh Roohi**

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at R001/U154 U1 lecture hall of the undergraduate center, on 7 March 2023 at 12 o'clock.

**Aalto University  
School of Science  
Department of Computer Science  
Aalto Game Research Group**

**Supervising professor**

Professor Perttu Hämäläinen, Aalto University, Finland

**Preliminary examiners**

Professor Paolo Burelli, IT University of Copenhagen, Denmark

Professor Pejman Mirza-Babaei, University of Ontario Institute of Technology (UOIT), Canada

**Opponent**

Professor Julian Togelius, New York University, United States of America

Aalto University publication series

**DOCTORAL THESES** 79/2022

© 2022 Shaghayegh Roohi

ISBN 978-952-64-0824-8 (printed)

ISBN 978-952-64-0825-5 (pdf)

ISSN 1799-4934 (printed)

ISSN 1799-4942 (pdf)

<http://urn.fi/URN:ISBN:978-952-64-0825-5>

Unigrafia Oy

Helsinki 2022

Finland



**Author**  
Shaghayegh Roohi**Name of the doctoral thesis**  
Advances in AI-assisted Game Testing**Publisher** School of Science**Unit** Department of Computer Science**Series** Aalto University publication series DOCTORAL THESES 79/2022**Field of research** Computer Science**Manuscript submitted** 29 May 2022**Date of the defence** 7 March 2023**Permission for public defence granted (date)** 18 November 2022**Language** English☐ **Monograph**☒ **Article thesis**☐ **Essay thesis****Abstract**

Game testing is an essential part of game development, in which developers try to select a game design that delivers a desirable experience for the players and engages them. However, the interactive nature of games makes the player experience and behavior unpredictable. Therefore, game testing requires collecting a large amount of playtest data in iterative sessions, which makes game testing time and money consuming.

Game testing includes a wide range of aspects from finding bugs and balancing game parameters to modeling player behavior and experience. This dissertation mostly concentrates on the player experience aspect. It proposes methods for (partially) automating and facilitating the game testing process. The first part of the dissertation focuses on player emotion analysis and proposes tools and methods for automatically processing and summarizing human playtesters' data. The second part of the dissertation concentrates on simulation-based approaches for modeling player experience and behavior to reduce the need for human playtesters.

In the first publication, we use deep neural networks for analyzing player facial expression data and provide a visualization tool for inspecting affect changes at game events, which replicates earlier results of physiological emotion analysis. Next, we extend this work by introducing a new dataset of game streamers' emotions in different granularities and considering other input signals like audio and speech for automatic emotion recognition. In the second part of the dissertation, simulation-based methods and reinforcement learning agents are used to predict game difficulty and engagement and capture the relation between these two game metrics.

In summary, this dissertation proposes and evaluates methods that advance automatic game testing by proposing approaches for automatic analysis of player emotions, which can be used for selecting specific segments of playtest videos for further inspection. In addition, we have provided accurate models of player experience and behavior using simulation-based methods that can be used to detect problematic game levels before releasing them to the actual players.

**Keywords** Game Testing, Player Experience, Emotion Recognition, AI Agents, Reinforcement Learning**ISBN (printed)** 978-952-64-0824-8**ISBN (pdf)** 978-952-64-0825-5**ISSN (printed)** 1799-4934**ISSN (pdf)** 1799-4942**Location of publisher** Helsinki**Location of printing** Helsinki **Year** 2022**Pages** 139**urn** <http://urn.fi/URN:ISBN:978-952-64-0825-5>



# Preface

This dissertation includes my Ph.D. work that has been done under the supervision of professor Perttu Hämäläinen in the department of computer science of Aalto university between 2017 and 2022.

First of all, I am grateful to professor Perttu Hämäläinen, who gave me the opportunity of continuing my studies and supported me during these years. This work was not possible to be accomplished without his guidance and support. He patiently helped me overcome the difficulties and led me toward completing this work by sharing his knowledge and constructive suggestions.

I acknowledge Helsinki Doctoral Education Network in Information and Communications Technology (HICT) and Academy of Finland, which financially supported this work.

I am thankful to professor Paolo Burelli and professor Pejman Mirza-Babaei, the examiners of this dissertation, for their valuable and insightful comments and reviews. In addition, I appreciate professor Julian Togelius, who accepted to be my opponent.

I appreciate my colleagues, Joose Rajamäki, Amin Babadi, Juuso Toikka, Maximus Kaos, and Noshaba Cheema, who created a friendly place to share ideas and get technical and emotional assistance whenever it was needed. Furthermore, I would like to thank my co-authors, Jari Takatalo, Christian Guckelsberger, J. Matias Kivikangas, Elisa D. Mekler, Mikke Tavast, and Tatu Blomqvist.

During my internship at Rovio Entertainment company, I had the opportunity to collaborate with amazing people, Asko Relas and Henri Heiskanen. I enjoyed my time there, where I got the chance to work on a commercial game and learned new technologies and concerns in a game industry environment.

I am thankful to my family and friends for their unconditional support during my Ph.D. studies. My parents and sister always encouraged me to do my best and supported me through good and bad days. Most of all, I am grateful to my husband, Kourosh Naderi, who was a great colleague and the best companion during this journey.

Preface

Helsinki, January 18, 2023,

Shaghayegh Roohi

# Contents

<b>Preface</b>	<b>1</b>
<b>Contents</b>	<b>3</b>
<b>List of Publications</b>	<b>5</b>
<b>Author’s Contribution</b>	<b>7</b>
<b>List of Figures</b>	<b>9</b>
<b>List of Tables</b>	<b>11</b>
<b>Abbreviations</b>	<b>13</b>
<b>Symbols</b>	<b>15</b>
<b>1. Introduction</b>	<b>17</b>
1.1 Publications . . . . .	18
1.2 Outline of the Thesis . . . . .	20
<b>2. Psychological Background</b>	<b>21</b>
2.1 Human Motivation . . . . .	21
2.1.1 Motivation in Games . . . . .	22
2.2 Human Emotion . . . . .	22
2.2.1 Discrete Emotion Models . . . . .	23
2.2.2 Continuous Emotion Models . . . . .	23
2.2.3 Emotion in Games . . . . .	23
2.3 Player Experience and Game Engagement . . . . .	24
2.3.1 Game Engagement and Difficulty . . . . .	25
<b>3. Technical Background</b>	<b>27</b>
3.1 AI-assisted Playtest Data Exploration . . . . .	27
3.2 Automatic Gameplay Highlight Detection . . . . .	28
3.2.1 Multi-modal Emotionally Salient Event Detection . . . . .	28
3.2.2 Facial Expression Recognition . . . . .	30



3.2.3	Audio Emotion Recognition . . . . .	32
3.2.4	Speech Sentiment Analysis . . . . .	35
3.3	Game Evaluation Metrics: A Computational View . . . . .	37
3.3.1	Game Difficulty . . . . .	37
3.3.2	Game Engagement . . . . .	37
3.4	Game AI Agents . . . . .	38
3.4.1	Markov Decision Process . . . . .	38
3.4.2	Deep Reinforcement Learning . . . . .	38
3.4.3	Monte Carlo Tree Search . . . . .	42
3.4.4	Simulation-based Game Testing Applications . . . . .	43
<b>4.</b>	<b>Developed Methods</b>	<b>45</b>
4.1	Publication I . . . . .	45
4.1.1	Facial Expression Analysis . . . . .	46
4.1.2	Affect Gradient . . . . .	46
4.1.3	Summary of Results . . . . .	47
4.1.4	Discussion and Future Work . . . . .	47
4.2	Publication II . . . . .	48
4.2.1	Dataset Creation . . . . .	48
4.2.2	Automatic Emotion Recognition . . . . .	48
4.2.3	Summary of Results . . . . .	50
4.2.4	Discussion and Future Work . . . . .	51
4.3	Publication III . . . . .	51
4.3.1	Game Description . . . . .	51
4.3.2	Game-playing Agent . . . . .	54
4.3.3	Game Level Difficulty and Engagement Prediction . . . . .	55
4.3.4	Extended Model . . . . .	55
4.3.5	Summary of Results . . . . .	56
4.3.6	Discussion and Future Work . . . . .	57
4.4	Publication IV . . . . .	57
4.4.1	Feature Selection . . . . .	57
4.4.2	Summary of Results . . . . .	58
4.4.3	Discussion . . . . .	58
<b>5.</b>	<b>Conclusion</b>	<b>61</b>
	<b>References</b>	<b>63</b>
	<b>Errata</b>	<b>75</b>
	<b>Publications</b>	<b>77</b>

# List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

- I** Roohi, Shaghayegh and Takatalo, Jari and Kivikangas, J. Matias and Hämäläinen, Perttu. Neural Network Based Facial Expression Analysis of Game Events: a Cautionary Tale. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, Melbourne, VIC, Australia, pp. 429–437, 10 2018.
- II** Roohi, Shaghayegh and Mekler, Elisa D. and Tavast, Mikke and Blomqvist, Tatu and Hämäläinen, Perttu. Recognizing Emotional Expression in Game Streams. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, Barcelona, Spain, pp. 301–311, 10 2019.
- III** Roohi, Shaghayegh and Relas, Asko and Takatalo, Jari and Heiskanen, Henri and Hämäläinen, Perttu. Predicting Game Difficulty and Churn Without Players. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, Virtual Event, Canada, pp. 585–593, 11 2020.
- IV** Roohi, Shaghayegh and Guckelsberger, Christian and Relas, Asko and Takatalo, Jari and Heiskanen, Henri and Hämäläinen, Perttu. Predicting Game Difficulty and Engagement Using AI Players. *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CHI PLAY, pp. 1-17, 9 2021.



# Author's Contribution

## **Publication I: “Neural Network Based Facial Expression Analysis of Game Events: a Cautionary Tale”**

The author developed the system, collected and analyzed the data, and wrote the majority of the paper. Dr. Kivikangas contributed in writing the emotion-related parts. Dr. Takatalo, Dr. Kivikangas, and Prof. Hämäläinen contributed through commenting, discussions, and revising the paper.

## **Publication II: “Recognizing Emotional Expression in Game Streams”**

The author developed the system, collected and analyzed the data, and wrote the majority of the paper. In addition, the author helped as one of the human annotators of the prepared dataset. Tavast and Blomqvist helped as human annotators. Prof. Mekler defined the dataset creation approach and helped as one of the human annotators. Prof. Hämäläinen contributed through commenting, discussions, and revising the paper.

## **Publication III: “Predicting Game Difficulty and Churn Without Players”**

The author developed the system and collected and analyzed the AI game-play data. Relas assisted with the software infrastructure and in collecting the ground truth human dataset. Relas, Dr. Takatalo, Heiskanen, and Prof. Hämäläinen contributed through commenting, discussions, and revising the paper.

## **Publication IV: “Predicting Game Difficulty and Engagement Using AI Players”**

The author developed the system and collected and analyzed the AI game-play data. Dr. Guckelsberger assisted in writing and contextualizing the paper. Dr. Takatalo, Relas, Heiskanen, and Prof. Hämäläinen contributed through commenting and discussions.

# List of Figures

2.1	Basic emotions mapped to a valence-arousal 2D map [112].	23
2.2	Flow theory emphasizes the balance between challenge and skill [28]. . . . .	25
2.3	Achieving game enjoyment does not always depend on the challenge-skill balance [90]. . . . .	26
3.1	A game streamer sharing her facial video ( <a href="https://www.kotaku.com.au/2018/01/how-to-stream-pc-games-on-twitch/">https://www.kotaku.com.au/2018/01/how-to-stream-pc-games-on-twitch/</a> ). . .	29
3.2	Top: Early fusion, in which extracted features from different modality modules are combined to output the final predictions. Bottom: Late fusion, in which predictions of different modality modules are combined to output the final predictions. . . . .	30
3.3	An example of a convolutional neural network. Convolutional layers extract features from the image, and pooling layers aggregate features over rectangular regions by computing averages or finding maximum values, and subsampling the results to decrease output resolution. The number of extracted features at each layer is equal to the number of orange boxes at that layer. Finally, a fully-connected layer outputs the final classifications. . . . .	32
3.4	The VGG16 architecture. FC denotes a fully connected layer.	32
3.5	An audio spectrogram. . . . .	34
3.6	Recurrent neural network architecture. $h_t$ is the hidden state at time $t$ . . . . .	34
3.7	Long short-term memory architecture [47]. $h_t$ and $c_t$ are the hidden and cell states at time $t$ , respectively. $\sigma$ is sigmoid and $\tanh$ is the hyperbolic tangent activation function. . .	35
3.8	A reinforcement learning agent. . . . .	39

3.9	An intrinsically motivated reinforcement learning agent adapted from [127]. Such an agent, in addition to the reward from the environment ( $r_{e_t}$ ), receives an intrinsic reward ( $r_{i_t}$ ) based on its internal model. . . . .	41
3.10	A Monte Carlo tree search agent. . . . .	43
4.1	Happiness signal for an individual player on three instances of <i>getting killed by enemy</i> event. a) the raw signal. b) the median-filtered signal. c) the median-filtered and mean-normalized signal. d) a line is fitted to the signal, and its slope shows the affect gradient. . . . .	46
4.2	Top: summary plots of means and standard deviations of all happiness signal segments and the summary affect gradients (i.e., the slopes of the lines fitted to the mean signals). Bottom: histograms of affect gradients computed from the segmented signals. . . . .	47
4.3	Overview of our automatic emotional event detection system.	49
4.4	A screenshot of Angry Birds Dream Blast. . . . .	54
4.5	The overview of our extended pass and churn rate prediction method. Level difficulty and current player population are passed to the pass and churn rate simulation block, which at each level outputs the pass and churn rate of that level and the remaining player population for the next level.	56
4.6	The scatter plots of the real players' and our simulations' pass and churn rates over 168 Angry Birds Dream Blast levels. The baseline method uses simple regression models to predict the pass and churn rates. Our Extended method improves it by considering player population changes over the levels. Scatter plot colors and shapes correspond to different game levels. . . . .	57
4.7	The correlation of agent's statistics calculated over different percentages of agent's best runs and human pass rates. . .	58
4.8	Box plot of churn rate mean squared errors. . . . .	59
4.9	Box plot of pass rate mean squared errors. . . . .	59

# List of Tables

4.1	Event congestion and inter-rater agreement in different window lengths and granularity levels. . . . .	50
4.2	Accuracy of classification using different window lengths and granularity levels. In each column, different input features are enabled for the final neural network. FE, S, AE, and AF indicate facial expressions, speech sentiment, audio expression analysis, and audio features, respectively. "full" column uses all 4 types of input features. . . . .	52
4.3	$F_1$ -score of classification using different window lengths and granularity levels. In each column, different input features are enabled for the final neural network. FE, S, AE, and AF indicate facial expressions, speech sentiment, audio expression analysis, and audio features, respectively. "full" column uses all 4 types of input features. . . . .	53
4.4	Mean squared errors and mean absolute errors of pass and churn rate predictions. . . . .	56





# Abbreviations

**AI** Artificial Intelligence

**LLM** Large Language Model

**SDT** Self-determination Theory

**PENS** Player Experience of Need Satisfaction

**EMG** Electromyography

**GSR** Galvanic Skin Response

**ECG** Electrocardiography

**NPC** Non-player Character

**GEQ** Game Engagement Questionnaire

**UPEQ** Ubisoft Perceived Experience Questionnaire

**LSTM** Long Short-Term Memory

**HOG** Histogram of Gradients

**LBP** Local Binary Patterns

**SVM** Support Vector Machine

**KNN** K-nearest Neighbours

**CNN** Convolutional Neural Network

**AER** Audio Emotion Recognition

**MFCC** Mel-Frequency Cepstrum Coefficients

**HMM** Hidden Markov model

**STFT** Short-Time Fourier Transform

**ME** Maximum Entropy  
**NLP** Natural Language Processing  
**GVGAI** General Video Game AI  
**DRL** Deep Reinforcement Learning  
**RL** Reinforcement Learning  
**MCTS** Monte Carlo Tree Search  
**MDP** Markov Decision Process  
**GAE** Generalized Advantage Estimation  
**PPO** Proximal Policy Optimization  
**RND** Random Network Distillation  
**UCT** Upper Confidence bounds for Trees  
**MSE** Mean Squared Error  
**MAE** Mean Absolute Error  
**PED** Platformer Experience Dataset  
**FER** Facial Expression Recognition

# Symbols

$h_t$	Hidden state at time $t$ in RNN
$c_t$	Cell state at time $t$ in LSTM
$\sigma$	The sigmoid activation function in LSTM
$\tanh$	The hyperbolic tangent activation function in LSTM
$S$	The set of MDP states
$s_t$	State at time $t$
$A$	The set of MDP actions
$a_t$	Action at time $t$
$R$	The set of MDP rewards
$r_t$	The reward that the agent receives by taking action $a_t$ at state $s_t$
$\mathbb{R}$	The set of real numbers
$p(s_{t+1}, r_t   s_t, a_t)$	The transition probability of the agent observing state $s_{t+1}$ and receiving reward $r_t$ by taking action $a_t$ at state $s_t$
$\pi(a_t   s_t)$	The probability density of the action given the state at time $t$
$\sum$	Summation
$\mathbb{E}$	Expectation
$\gamma$	Discount factor
$G_t$	Return at time $t$
$V^\pi(s_t)$	on-policy state value function
$Q^\pi(s_t, a_t)$	on-policy state-action value function

## Symbols

$\theta$	Parameter used for the policy function
$\pi_\theta$	Policy function parametrized by $\theta$
$\nabla$	Gradient
$A^\pi(s_t, a_t)$	Advantage function under the policy $\pi$
$\phi$	Parameter used for the value function
$\hat{V}_\phi^\pi(s_t)$	Approximated value function parametrized by $\phi$
$\lambda$	The bias-variance trade-off parameter in generalized advantage estimation function
$r_{i_t}$	Intrinsic reward at time $t$
$r_{e_t}$	Extrinsic reward at time $t$
$I(a; s' s)$	Mutual information between the agent's action and next state given the current state
$H(a s)$	The entropy of the agent's action given the current state
$H(a s', s)$	The entropy of the agent's action given current and next states
$s_d$	A tree node at depth $d$
$C(s_d)$	The set of children of node $s_d$
$N_{s_d}$	The visit number of node $s_d$
$V(s_d)$	The value of node $s_d$
$c$	Exploration-exploitation trade-off in MCTS
$\rho_p$	Pass rate predictions
$\rho_c$	Churn rate predictions
$w_{churn}$	Churn rate prediction weight

# 1. Introduction

The task of game testing is repetitive and expensive. Automatic analysis of human player data or replacing playtesters with artificial intelligence (AI) could save game companies time and money. For instance, AI-based analysis of human playtesters' data could help with accelerating the assessment of large-scale playtest datasets or modeling player behavior [130]. In addition, employing AI agents can assist developers in selecting the best game design or finding bugs in a current design without hiring human playtesters with varying playing styles. This empowers game developers to test small design alterations fast and at the early stages of development [130].

This dissertation advances game testing through the lens of player experience. Since emotion is considered essential for understanding player experience [151], first, we investigate machine learning for analyzing playtesters' emotional expressions. Second, simulation-based methods are developed to predict the game difficulty and engagement experienced by players. In other words, this dissertation aims to solve two main problems: 1) How to better collect and analyze data from real humans, and 2) How to augment expensive human data with synthetic data.

The first part of this dissertation focuses on analyzing human player emotion, which can be used to examine the game events or select highlights of the game for further inspection. We investigate the role of multi-modality and emotional event granularity levels in detecting human emotion accuracy. This is made possible by modern deep neural network tools that allow flexible and accurate analysis of multiple data types such as facial images, audio signals, and speech transcripts [79].

The focus of the second part of the dissertation is on simulation-based player experience and behavior modeling. We employ AI agents for predicting human player data such as pass and churn rates, which helps identify problematic game levels and optimize them before presenting them to real players. Here, we build on recent advances in deep learning and reinforcement learning that have empowered automatic game testing by making it possible to create AI agents that can play the games as well as human

players [95, 96, 125]. Simulation-based testing research ranges from generating human-like agents [57, 2] and agents with diverse behaviors [121] to creating AI agent gameplay visualization tools [1]. Furthermore, AI agents have been employed in various areas such as game parameter tuning, finding bugs, and player experience and behavior modeling [9, 55, 45, 105, 49]. In this dissertation, we concentrate on using AI agents in predicting game difficulty and engagement.

Although the first and second parts of the dissertation are both applicable and beneficial in game testing as they are, there is another important connection between the two parts. Since emotion is a regulator of decision-making and memory processes, AI agents with emotion would be desirable in game testing because they could behave more human-like. In this regard, Publication I’s Affect Gradient method could be used to evaluate and visualize computational models of emotions by comparing synthetic affect gradient data generated by AI agents to real affect gradient data from human players. For recent advances in augmenting AI agents with computational emotion models, see the review by Ojha et al. [101]. For example, the Affect Gradient reduction at a particular game event as time passes could be an indicator of an event gradually losing its novelty for the players. One might also generate game-testing agents with virtual facial expressions that produce similar Affect Gradient measures as human players.

Initially, the Affect Gradient was built on top of facial expression analysis, however, as shown by Publication II, adding extra modalities helps emotion detection, so using other modalities would likely also improve Affect Gradient measurements, and help to generate better emotional models. In the future, it might be possible to develop and evaluate agents that not only express facial expressions but also produce emotional think-aloud narratives. Recent research on Large Language Models (LLMs) has demonstrated the ability of LLMs to precisely reason using chain-of-thought prompting [146] and express human-like emotion with respect to a particular prompt [134]. Based on this, it might be possible to automatically generate textual descriptions of game events and have an LLM like GPT-3 [20] to generate human-like think-aloud narratives of the subjective player experience.

## 1.1 Publications

**Publication I** employed a convolutional neural network (CNN) to classify human facial expressions into seven predefined emotion classes. Then, the affect changes at game events were investigated to explore the players’ reactions to different game events.

Our main finding was that players usually react to the *getting killed* event by smiling, which is detected as happiness. On the other

hand, players' concentration at events like *killing enemies* could be detected as mild anger. We hypothesized that players might laugh at the *getting killed* event because of the irony of the event or social signaling regarding being observed by the researchers. Our findings warn researchers that a direct interpretation of players' facial expressions might be misleading. In addition, we urged further inspection of other expression signal modalities like voice and speech in player emotion detection.

**Publication II** provided a dataset of emotional events (e.g., startle, happy) in streams of a puzzle game. The annotated events were also classified into pleasant/unpleasant/neutral and top5/not-top5 highlight categories. The automatic detection of these emotional events was investigated. We exploited the combination of facial, voice, and speech expressions for better detection of emotional events. For this matter, we used a novel neural network architecture that integrates the low-level audio features and emotional probability outputs of multiple networks pretrained with large datasets of facial images, voice data, and review text. Our results suggest that extra input modalities could help detect emotionally salient events.

**Publication III** proposed a pass and churn rate prediction method for a mass-market mobile puzzle game. Our two-level simulation approach combined Deep Reinforcement Learning (DRL) game-playing agents and simulation of player population evolution over game levels. Game level difficulties predicted by DRL agents' performance were used to model how the player population with simulated attributes like skill, persistence, and boredom changes over game levels.

We demonstrated that the simulation of player population changes over game levels is able to model individual player differences and capture the relation between the pass and churn rates over the levels without requiring training multiple AI agents per game level. However, improving the pass and churn rate predictions, for example, using Monte Carlo tree search agents, remained as future work.

**Publication IV** extended the previous paper by combining Monte Carlo Tree Search (MCTS) with the pretrained DRL agents and proposing a better feature selection approach for the pass and churn rate prediction models.

According to our results, combining MCTS and DRL improves prediction results. We also confirmed the finding of Kristensen et al. [75] that calculating AI agent performance based on its best runs can be a better predictor of human data rather than the AI agent's average performance.



## **1.2 Outline of the Thesis**

In chapter two, psychological theories behind human motivation, emotion, and experience are briefly overviewed. Chapter three elaborates on automatic and AI-assisted game testing approaches, including playtest data exploration and game-playing agents. We summarize our developed methods, results, and possible future improvements in chapter four. Finally, the main findings of this dissertation and possible future works are discussed in chapter five.

## 2. Psychological Background

The following chapter explains the psychological background necessary for understanding the motivation and reasoning behind the technology developed in this dissertation for player behavior and experience analysis and modeling. First, human motivation and emotion are discussed, then player experience and its relation to game engagement are explored. In short, player experience can be defined as how the player perceives the interaction with the game [152].

### 2.1 Human Motivation

As defined by Fiske and Taylor [42], "motivations provide the motor for behavior". In other words, the reason humans do certain things in specific ways resides in their motivations. There are multiple motivation theories that attempt to explain human behavior and propose different hierarchies and categories. For instance, Forbes [43] proposes a motivational model constructed by a  $3 \times 3$  table with the focus of aspiration (self, material world, social world) as columns and the level of aspiration (expectations, experiences, outcomes) as rows.

Human motivation is often classified into extrinsic and intrinsic subcategories [31]. Motives initiated by external rewards or punishments are called extrinsic. External drives, such as shame, fame, and wealth, come from an individual's surrounding environment. In contrast, intrinsic motivations arise from the inner satisfying characteristics of activities rather than external and detachable outcomes and are a result of satisfying basic psychological needs [32]. Both extrinsic and intrinsic motivations direct human behavior, but in different ways. Self-determination Theory (SDT) [114] as one of the popular motivation theories posits competence, autonomy, and relatedness as core psychological needs. Satisfying these needs, which are common among diverse cultures, facilitates intrinsic motivations and causes self-motivation. The needs can be defined as [115]:

**Autonomy** The need to act willingly

**Competence** The need to be challenged and overcome the challenges

**Relatedness** The need to connect with the other individuals

SDT has been widely applied in different domains like sport, healthcare, education [92], as well as video games [113] to predict the effective outcome. On the other hand, other sources discuss and identify a wider palette of needs, e.g., self-actualization [120] and novelty [5].

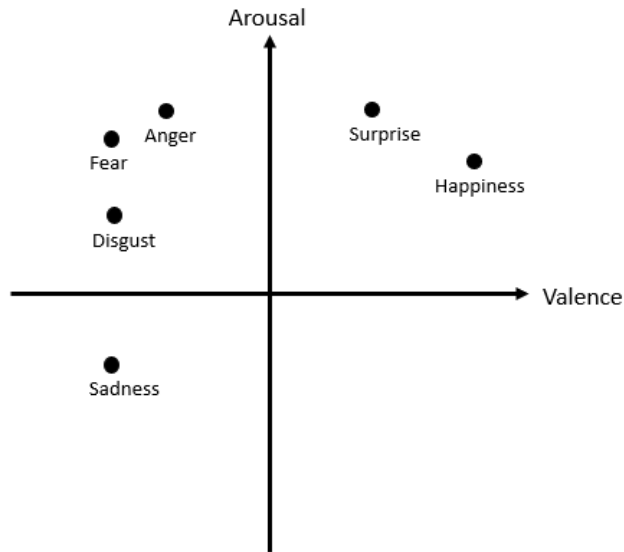
### 2.1.1 Motivation in Games

Player type identification classifies players based on their motivations to play the game. In an early work, Bartle [7] introduces four player types, including killer, achiever, explorer, and socializer, in a multiplayer Dungeon game. Bartle states that an engaging game should satisfy all types of players. In order to validate the player types introduced by Bartle, Yee [153] performs an empirical study of player motivations in an online multiplayer game. After performing factor analysis on a questionnaire with 40 questions, achievement, social, and immersion motivations with ten subcategories are suggested. Recently, Hamari and Tuunanen [53] have proposed achievement, exploration, sociability, immersion, and domination as five key dimensions of player type and motivation modeling.

For the sake of a general player motivation model and independence to game genres, Ryan et al. [115] experimented with the application of SDT in video games. They used the Player Experience of Need Satisfaction (PENS) questionnaire to measure the effect of different game features on satisfying basic player needs and two additional factors of presence and intuitive controls. Based on their findings, satisfying autonomy, competence, and relatedness is a predictor of game enjoyment and the tendency for future play. Furthermore, they found that perceived autonomy and competence are correlated with the feeling of presence.

## 2.2 Human Emotion

Several studies define *emotion* as a process in which humans show feelings and emotions in response to their environments. For instance, appraisal theory [116] posits that the way humans evaluate an event based on variables like novelty, goal relevance, social norms, and agency results in different emotions. The procedural nature of appraisal theory makes it a popular computational model of emotion in human-computer interaction and artificial intelligence fields[29].



**Figure 2.1.** Basic emotions mapped to a valence-arousal 2D map [112].

### 2.2.1 Discrete Emotion Models

Some emotion theories introduce fundamental or basic emotions like fear, anger, and joy [136, 38], which have been developed through human evolution, and each has an adaptation functionality. Each emotion can activate a particular neural circuit and result in a specific response and action. For example, neural activity related to fear has been developed with the purpose of survival [98]. As evidence of basic emotions, Ekman et al. [38] propose emotion-specific facial expression patterns and show their universality.

### 2.2.2 Continuous Emotion Models

According to continuous emotion models, emotions are characterized by continuous dimensions like valence and arousal [111]. Figure 2.1 illustrates basic emotions in the valence-arousal 2D map. Later, the dominance component was added to the valence and arousal dimensions for better discrimination of emotions [18]. Valence denotes the positivity (pleasantness) of emotion; the activation (responsiveness) of emotion is denoted by the arousal dimension, and dominance indicates the controllability (power) of emotion [112].

### 2.2.3 Emotion in Games

Emotion and games are intertwined, and one cannot easily separate them from each other [151]. In defining fun in games, Lazzaro [78] identifies

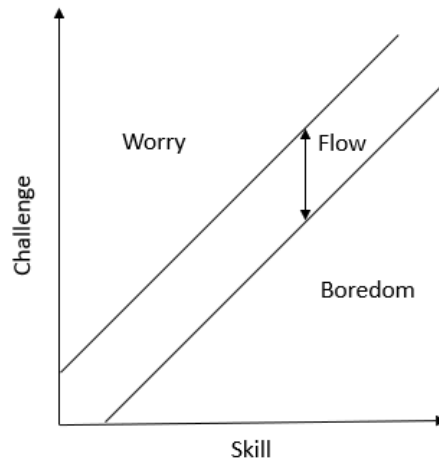
emotions as the key factors and shows that each type of fun elicits different emotions. For example, frustration and fiero feelings are expressed in hard fun, and the feeling of wonder is expressed in easy fun. Furthermore, several studies associate (positive) emotions with positive player experience [90]. On the other hand, there exists a growing body of research on how negative emotions and emotional challenge can also be central to games that players appreciate [14].

Emotions can be expressed as physiological reactions. Therefore multiple researchers have studied physiological data like electromyography (EMG), galvanic skin response (GSR), and electrocardiography (ECG) signals in detecting emotions in games [73]. In a study by Mirza-Babaei et al. [93], a user research tool is introduced, in which EMG signal is combined with game logs. Tan et al. [133] combine physiological signals and think-aloud data and show that they have a complementary effect on player experience understanding. Other signals like gaze, head and body pose, and facial expressions have also been studied in detecting the affective state of players [119, 132]. Researchers should be aware of the complementary effect of different signals. For instance, [87] states that as GSR is more sensitive to arousal, EMG is more responsive to valence.

Emotion in games has mainly been applied for developing adaptive games [11, 13]. Interaction with games triggers player emotions. An adaptive game adjusts itself to respond to players' emotional changes. Game content, non-player characters (NPCs), and game difficulty can be adjusted to improve the player experience. For instance, Blom et al. [11] exploit facial expressions to decrease the game difficulty when the player looks angry and increase it when the player seems neutral.

## 2.3 Player Experience and Game Engagement

Player experience can be measured through popular game questionnaires like Player Experience of Need Satisfaction (PENS) [115], Game Engagement Questionnaire (GEQ) [19], and Ubisoft Perceived Experience Questionnaire (UPEQ) [4]. Game engagement, which has an intricate and multidimensional essence, is usually approached with respect to the subjective experience of enjoyment and motivations behind playing games [17]. Csíkszentmihályi [27] identifies an activity to be engaging when it creates the Flow state characterized by challenges and the required skills to overcome them, immersive experience, high amount of concentration, sense of control, clear goals, and immediate feedback.



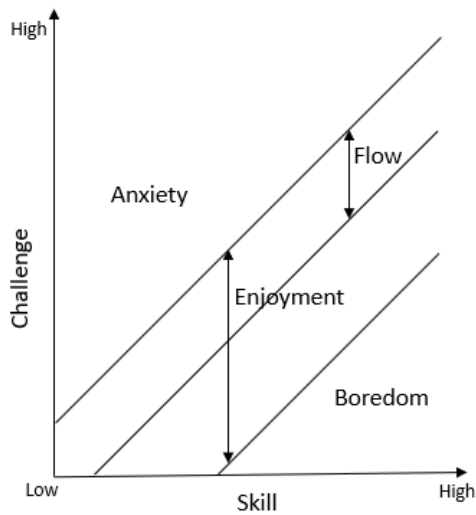
**Figure 2.2.** Flow theory emphasizes the balance between challenge and skill [28].

### 2.3.1 Game Engagement and Difficulty

According to the Flow theory [27], there should be a balance between game difficulty and player skill level to make an engaging game (Figure 2.2). Otherwise, excessive challenge yields anxiety, while a very easy game causes boredom. In both situations, the state of flow cannot be achieved.

In addition, many studies acknowledge *challenge* as one of the key motivations for playing the games [17]. Lucas and Sherry [83] applied six factors of uses and gratification theory (challenge, competition, diversion, fantasy, social interaction, and arousal) to playing games and found *challenge* as the main reason for playing games for both males and females. Similarly, competence in self-determination theory is one of the basic human needs and is satisfied when players encounter challenges in balance with their skill level so that they feel capable of overcoming the challenges [115, 109].

Nevertheless, the relation between game engagement and difficulty is not always straightforward, and it might depend on the type of game. This has been shown by Lomas et al. [82] who observed that lower difficulty leads to higher player engagement in their educational game. Considering game enjoyment as a key contributor to game engagement [17], Mekler et al. [90] distinguish enjoyment (i.e., the valence of player experience) from engagement (i.e., the intensity of player experience) and state that players might experience enjoyment even in the lack of challenge-skill balance, while player's skill level is higher than the challenges (Figure 2.3).



**Figure 2.3.** Achieving game enjoyment does not always depend on the challenge-skill balance [90].

## 3. Technical Background

What if one could provide methods to facilitate more efficient inspection of collected game testing data, exploit the enormous available data in game streams, or replace human playtesters with AI agents? This chapter discusses prior research on these topics to provide background and contextualize the technical contributions of this dissertation.

### 3.1 AI-assisted Playtest Data Exploration

Game companies perform large-scale game testing to validate their game design or tune their game parameters. Beyond game analytics data logged by game code, it is also possible to collect playtest video data at scale, through online services such as PlayTestCloud. A good visualization tool for summarizing and further exploration of playtest videos can accelerate the playtesting process.

Gameplay data can be visualized using different visualization techniques such as charts, movement trajectories, heatmaps, and node-link graphs [145]. Charts come in various types and are useful for comparison or to show aggregated statistics. Movement trajectories illustrate the exploration in the game environment. Heatmaps are useful to show the intensity of a variable at specific positions in the game map. Node-link graphs are able to demonstrate the data variables and the relations between them. Each visualization technique has its own weakness and strength and can coexist together to convey more information [145].

Mirza-Babaei et al. [93] have developed biometric storyboards, a visualization method that illustrates player experience graphs along with player physiological signals collected during playing the game. However, graphs might make the comparison between players difficult because they are gameplay-dependent. In many games, players have various game-playing styles and spend different amounts of time interacting with different parts of the game, which results in very different gameplays. As a good alternative to graphs, one can inspect game events.



Previous work has studied game events with a focus on player behavioral data analysis [72, 89]. Medler et al. [89] record and use graphs to visualize game events in a multiplayer online game, such as when players die or kill other players, when a weapon is obtained, or at the winning/losing the game. Visualizing game events assists in monitoring player performance and balancing the game. Drenikow and Mirza-Babaei [36] have created a visualization tool for game playtesting in which players’ facial data, movement trajectories, and in-game events are recorded. To illustrate where the events have happened, in-game events are shown as 2D sprites inside the game environment.

In Publication I, like the methods in [89, 36], we also study game events, but we focus on the effect of different events on affect. Error bar and histogram charts are used to visualize and summarize the affect changes around each game event of a platformer game [71].

### 3.2 Automatic Gameplay Highlight Detection

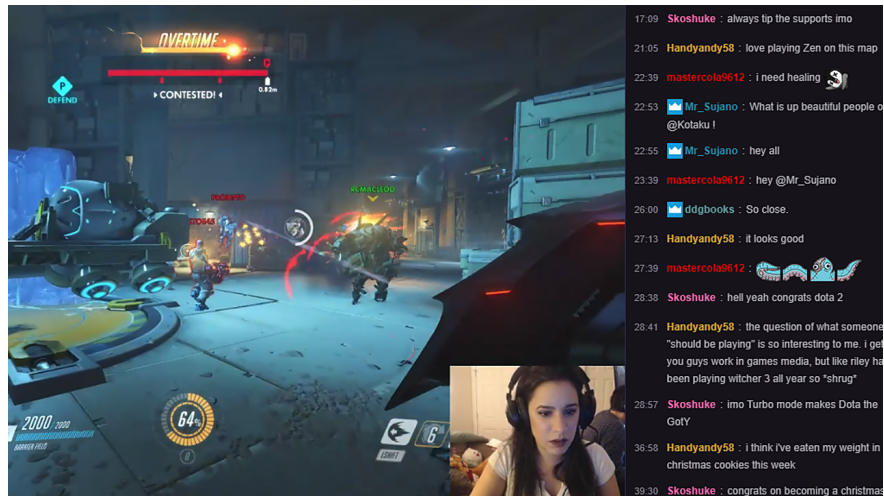
Sometimes there are no predetermined game events like in game streams, and sometimes the game events cannot encompass all the valuable information like the aesthetic aspects of the game. In these cases, detecting gameplay highlights and further investigating them would be helpful for game designers.

Ringer and Nicolaou [110] propose a novelty-based highlight detection. They extract features from audio and visual data of game streams, including reconstruction errors of auto-encoders [138] applied to the game streamer’s facial images and gameplay images. Then, extracted features are fed into Long Short-Term Memory (LSTM) [56] layers to predict the next frame features. 0.01% of the frames with the highest LSTM prediction errors are considered as the highlights. After manually labeling the detected highlights via different modalities, they found that the facial data is mostly good for detecting social-interaction events, the gameplay is good for detecting action events, and all modalities together have the best highlight detection performance.

#### 3.2.1 Multi-modal Emotionally Salient Event Detection

Game events provoke player emotions which are expressed through the player’s face, voice, and speech. Therefore, player emotional expressions can be used as an indicator of emotionally valuable game highlights. Recognizing player emotional expressions allows us to identify game highlights and classify such events into various categories like pleasant or unpleasant, which will be advantageous for player experience estimation.

Services like Youtube and Twitch construct the required foundation for

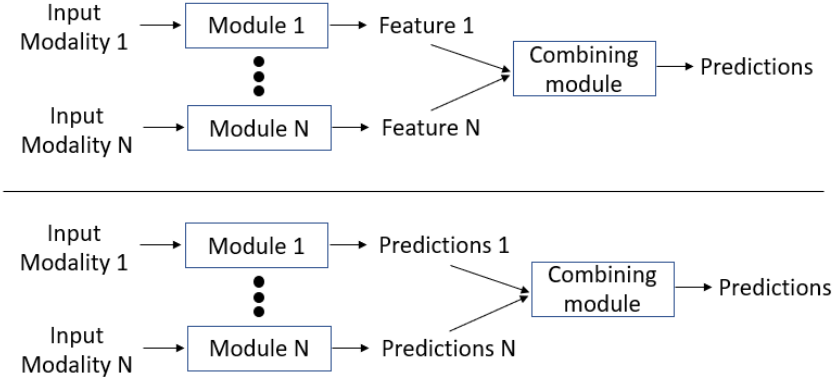


**Figure 3.1.** A game streamer sharing her facial video ( <https://www.kotaku.com.au/2018/01/how-to-stream-pc-games-on-twitch/>).

consumers to act as content creators and let them interact with their audience through the fast-growing online broadcasting industry known as streaming. Interestingly, more and more users prefer watching other people’s gameplays, which motivates game streamers to create even more content [128]. Streamers usually narrate their game experience through the gameplay in a non-controlled environment, and some of them also share their facial video. This makes game streams one of the perfect resources for exploring player emotional expressions using different modalities, especially as when playing and streaming a game as a performance, one is motivated to be expressive, which avoids the problem of players becoming expressionless with a high degree of concentration on the game.

Using various modalities would be helpful in capturing an individual’s true feelings. In a hypothetical situation where a subject is nervously laughing, speech or physiological signals might help detect the laughter as not signaling joy. Even though physiological signals are not available in game stream data, facial images, vocal data, and linguistic aspects of speech are other modalities that could be used.

As shown in Figure 3.2, different modalities can be combined in feature level (i.e., early fusion) or decision level (i.e., late fusion) [123]. In the first case, extracted features from all the resources are fused together, and machine learning methods are applied to the combined feature vector. In the decision-level fusion, each resource outputs its own predictions, which are combined to produce the final predictions.



**Figure 3.2.** Top: Early fusion, in which extracted features from different modality modules are combined to output the final predictions. Bottom: Late fusion, in which predictions of different modality modules are combined to output the final predictions.

### 3.2.2 Facial Expression Recognition

The task of recognizing emotional expressions from facial images is called Facial Expression Recognition (FER). Facial expressions can be classified into categorical classes of emotions like happiness, sadness, and anger. In addition, they can represent player experience. Tan et al. [132] show that player facial expressions are correlated with Game Experience Questionnaire [62] dimensions, including competence, immersion, flow, tension, challenge, negative affect, and positive affect.

**Datasets** Since automatic facial expression recognition is a well-explored problem, several datasets have been developed for it. For exploring these datasets, we refer to [80], but here we briefly introduce the ones that have been used in this dissertation.

A Kaggle competition<sup>1</sup> provides a public dataset containing approximately 28K training images, 7K test images, and corresponding emotional labels. Facial images are in grayscale and have a size of  $48 \times 48$ . Labels include six basic emotions: happiness, sadness, anger, disgust, surprise, fear, and an additional neutral class.

AffectNet [97] contains over 500K images from the Internet with their manually annotated labels of six basic emotions and additional classes like neutral, contempt, uncertain, non-face, and none. The uncertain label is assigned to the images that the annotator had not been sure about the facial expression. Non-face label encompasses images that are very distorted. Expressions like sleepy, tired, and

<sup>1</sup><https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>

shame that cannot be described by the six basic emotions, contempt or neutral, are annotated with the *none* label. In addition, AffectNet provides the valence and arousal values for all the images except those with non-face and uncertain labels.

**Methods** To prepare the recorded videos of players for facial expression analysis, first, one needs to perform face detection and crop the faces from the video frames. Next, face alignment should be applied to the cropped images [80]. Pitaloka et al. [106] showed that techniques like data augmentation by adding noise to the images and global contrast normalization, in which the image is normalized by its mean and standard deviation could improve the facial emotion recognition performance.

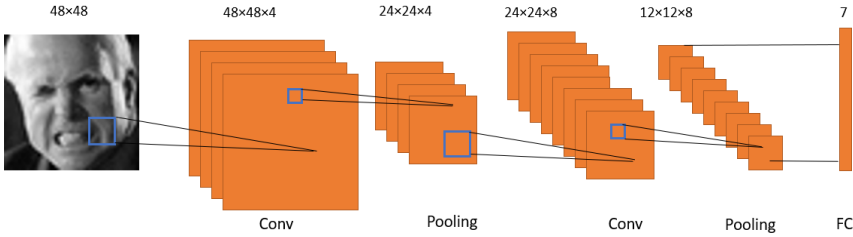
After preparing and preprocessing the data, feature extraction and classification steps should be done. Early methods used Histogram of Gradients (HOG) or Local Binary Patterns (LBP) for feature extraction and logistic regression, Support Vector Machine (SVM), or K-nearest Neighbours (KNN) for the classification part [77].

Feature extraction and classification steps are unified in deep learning methods, which have attracted much attention in various machine learning domains, including computer vision. Convolutional Neural Networks (CNNs) [149], as a subset of deep learning, are popular in the field of computer vision and have been employed for the facial expression recognition task as well [80].

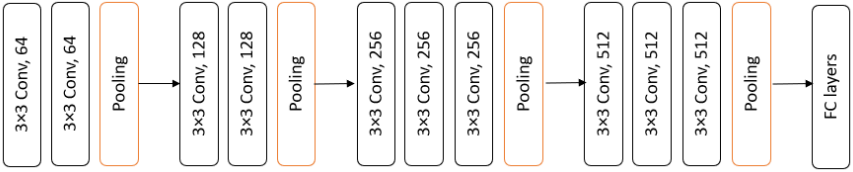
A convolutional layer scans the whole image with a window called filter, to quantify the local spatial dependencies between pixels, e.g., whether there's a corner or edge. In addition, using a shared window across an entire image reduces the number of parameters one needs to learn. A convolutional neural network (Figure 3.3) stacks multiple convolutional layers to extract a hierarchy of features from low level to high level.

Pooling layers are also popular in convolutional networks. Pooling is usually applied after convolutional layers to aggregate features locally, reduce the computational complexity by reducing the size of features for the next layer, and make the model almost invariant to small local translations in the input [46].

In the experiments of this dissertation, we deploy the VGG16 [126] architecture, one of the popular CNN models, for facial expression classification. Figure 3.4 illustrates the VGG16 architecture. The categorical cross-entropy loss function is used for training the network. The neural network outputs the probability of basic emotion classes, which later is used as one set of emotional features for detecting the emotional events in game streams.



**Figure 3.3.** An example of a convolutional neural network. Convolutional layers extract features from the image, and pooling layers aggregate features over rectangular regions by computing averages or finding maximum values, and subsampling the results to decrease output resolution. The number of extracted features at each layer is equal to the number of orange boxes at that layer. Finally, a fully-connected layer outputs the final classifications.



**Figure 3.4.** The VGG16 architecture. FC denotes a fully connected layer.

### 3.2.3 Audio Emotion Recognition

In addition to the face, humans usually express their emotions through their voice. For instance, they change their voice pitch regarding their emotion about an event. Acoustic characteristics of voice like energy, pitch, and tone can convey information about the emotional state of a subject [26].

**Datasets** There are many databases for the Audio Emotion Recognition (AER) task, such as IEMOCAP [23], TESS [37], RAVDESS [81], SAVEE [68], and CREMA-D [24]. IEMOCAP [23] includes audio-visual data of 10 actors reading pre-written scripts or having improvised dialogues in different emotional states. In this dataset, discrete emotion labels (i.e., happiness, sadness, anger, surprise, fear, disgust, frustration, excitement, neutral) and continuous emotion values of valence, arousal, and dominance ranked from one to five are provided by multiple human annotators.

RAVDESS [81] is another audio-visual dataset containing 7356 videos recorded from 12 male and 12 female actors in two strong and normal emotional intensities. It contains emotional labels, including calm, happiness, sadness, anger, fear, surprise, and disgust.

CREMA-D [24] includes 7442 videos from 91 actors and actresses of different ethnicities and ages. Actors say 12 sentences in the neutral state and basic classes of emotion except for surprise, with

low, medium, high, and unspecified emotional intensities.

SAVEE [68] contains the recording videos from 4 English male actors reading 15 sentences in 6 basic emotional states and the neutral state. In addition, TESS [37] dataset includes voices of 24 and 64 years old actresses saying the sentence "Say the word ..." filled with a word from a set of 200 words, in 6 basic emotion and neutral states.

**Methods** Traditional audio emotion recognition methods first extract features like pitch, energy, and Mel-Frequency Cepstrum Coefficients (MFCC) and then add a classification method like Hidden Markov Model (HMM) or SVM on top of the extracted features [39]. However, like many other machine learning tasks, we can leave the feature extraction to the neural networks and have an end-to-end recognition system [137, 139]. However, in addition to the raw audio signals, audio spectrograms are also widely used as the input of deep neural networks applied to the AER task [100, 60].

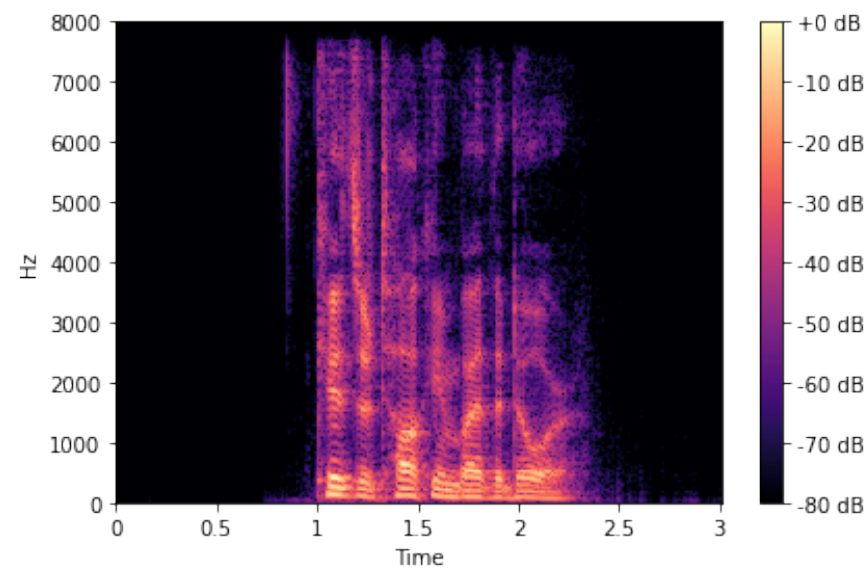
For converting the raw audio signal to the spectrogram, the Short-Time Fourier Transform (STFT) as a time-frequency representation of the signal is computed by applying discrete Fourier transforms in overlapping windows over the signal. The Fourier spectra are complex-valued vectors, so magnitude spectra or power spectra are calculated. Then, spectra are converted to decibels, which gives us log spectra. The heatmap visualization of the log spectra is called the spectrogram<sup>2</sup> (Figure 3.5).

Yafeng et al. [100] perform data augmentation using the retinal imaging principle algorithm to get spectrograms in various sizes and apply AlexNet [76], a convolutional neural network, to the spectrogram inputs. Another type of deep neural network that has been utilized for the AER task is Recurrent Neural Networks (RNNs) (Figure 3.6), which are specialized for sequential data learning. Long Short-Term Memory (LSTM) [47] is a recurrent neural network that can utilize information from longer sequences by regulating the amount of information passing through the network using its sigmoid activation gates (Figure 3.7). Huang and Narayanan [60] use a combination of convolutional and LSTM [60] layers to capture the temporal aspect of audio signals.

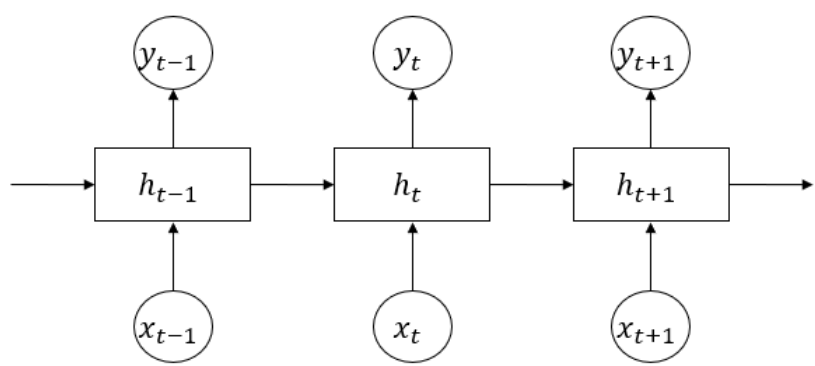
In Publication II, we convert the stream's audio signal to spectrograms, which later are mapped to 7 classes of emotions by the VGG16 model [126]. The last fully-connected layer of the network uses softmax activation to output the probability of each emotion class. These probabilities are another set of emotional features extracted from the streams.

---

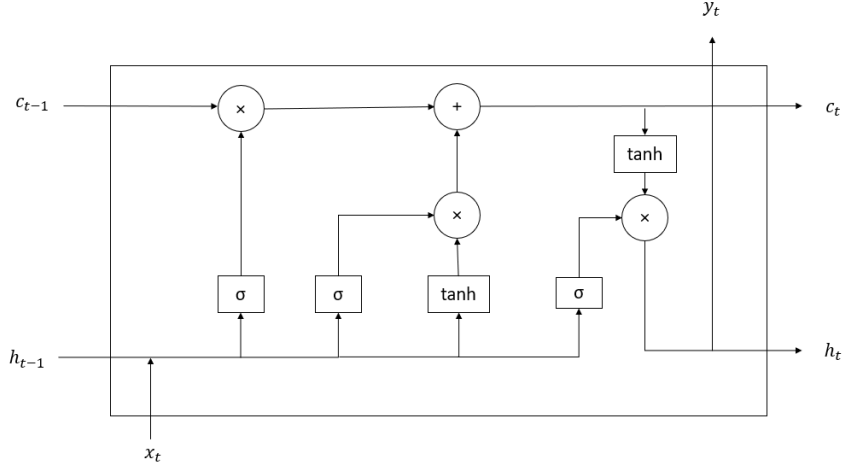
<sup>2</sup><https://wiki.aalto.fi/display/ITSP/Spectrogram+and+the+STFT>



**Figure 3.5.** An audio spectrogram.



**Figure 3.6.** Recurrent neural network architecture.  $h_t$  is the hidden state at time  $t$ .



**Figure 3.7.** Long short-term memory architecture [47].  $h_t$  and  $c_t$  are the hidden and cell states at time  $t$ , respectively.  $\sigma$  is sigmoid and  $\tanh$  is the hyperbolic tangent activation function.

### 3.2.4 Speech Sentiment Analysis

Alternative equipment for humans to express their emotions is the linguistic part of speech. Therefore, the positivity or negativity of the speech is another modality that can be used in detecting human feelings. To automatically classify the speech into positive or negative, first, we need to convert speech to text. This task can be done by humans or automatic speech recognition modules [154]. Nowadays, services like Youtube provide automatic speech recognition for many videos, including game streams.

**Datasets** Databases like Amazon review dataset<sup>3</sup> or IMDB movie reviews [84] are available for the sentiment analysis task. IMDB dataset includes 50K highly polarized reviews with an equal number of positive and negative reviews. Amazon review dataset contains around 3600K train and 400K test data. This dataset leaves out 3-star (i.e., neutral) reviews and labels 1-star and 2-star reviews as negative and 4-star and 5-star reviews as positive sentiment.

**Methods** First, the input text is standardized by removing special characters and HTML tags, lower-casing, and expanding contractions. Second, each data point is split into smaller tokens. For example, sentences are tokenized into words. In some methods, the input data should be converted to equal-length sequences. This can be done by truncating the longer sequences and zero-padding the shorter ones.

Traditional methods used hand-crafted features such as part-of-speech (i.e., labeling each word with its role in the sentence as noun,

<sup>3</sup><https://www.kaggle.com/bittlingmayer/amazonreviews/discussion/33444>



verb, etc.), word n-grams (i.e., the combination of  $n$  consecutive words in the text), and sentiment lexicon (i.e., having a list of positive and negative words, and computing the number of positive and negative words in the text) [150]. Then, classifiers like Maximum Entropy (ME) or SVM were trained on extracted features to predict the sentiments [150].

Although classic machine learning methods have produced good results, deep neural networks simplify the natural language processing tasks by providing a good representation of the text without requiring hand-crafted features. Recurrent neural networks like LSTM or temporal convolutional networks (1D convolution) are typical choices for the sequential and time-series data type like textual data.

In a text classification task, Zhang and Wallace [155] first tokenize each sentence into words and then convert the data into  $n \times d$  matrix, with  $n$  length of the sentence and  $d$  the word vector size. They apply 1D convolutional layers to the matrix by moving the kernel along the sentence dimension. Furthermore, researchers have combined CNN and LSTM to capture both local and long-term dependencies in the data [61].

Another popular type of deep neural network that has been widely used in the field of Natural Language Processing (NLP) is transformers [141]. Transformers employ a self-attention mechanism to relate different parts of the input sequence to each other. Unlike RNNs, transformers do not process the sequence in order; therefore, they can utilize parallel processing to make the training faster. In order to use the sequential ordering of tokens, transformers employ positional encoding, which can be learned or fixed vectors. Vaswani et al. [141] use sine and cosine functions with different frequencies for this matter.

To prepare the text data for the deep neural networks, we need to represent the text as numbers. One can use the one-hot encoding of words, an extremely sparse representation, or assign a unique integer value to each word of the vocabulary, which is a dense representation but still cannot encode the similarity of the words.

An efficient substitute for the above representations is word embeddings which are trainable real-valued vectors. They provide a dense representation space where semantically similar words have similar encodings. For instance, Word2Vec [91] learns word embeddings by skip-gram or continuous bag-of-words models in an unsupervised manner. Unlike the skip-gram model, which predicts the surrounding words from the current word, the continuous bag-of-words model predicts the current word given the surrounding words.

In Publication II's stream emotion recognition task, we used Keras trainable embedding layer [25] and 1D convolutional layers for sentiment analysis of the streamers' speech. The positivity probability of their speech played as another feature for our final game stream emotional event recognizer.

### 3.3 Game Evaluation Metrics: A Computational View

#### 3.3.1 Game Difficulty

Game difficulty and challenge are intertwined concepts and are used interchangeably by researchers [34]. There are different types of challenge: performative, cognitive, and emotional challenge. While performative challenge describes the challenges related to the player's physical performance, reaction, and accuracy, cognitive challenge encompasses difficulties related to the player's memory, strategy, and decision-making abilities [34]. Emotional challenge occurs when the player faces a difficult subject, has to make a hard decision, confronts an ambiguity or emotional narrative, or feels identification with the game characters [34, 15].

Game difficulty has been measured by game element characteristics like the gap size in a platform game [129], number of NPCs, the existence of pathfinding/traps in a General Video Game AI (GVGAI) game [58], or the board layout in a puzzle game [140]. Although calculating such metrics is not computationally expensive, they are game-specific and do not generalize well to all game genres. Furthermore, they do not cover the game difficulty caused by the player's interaction with the game mechanics.

Other methods have operationalized game difficulty by the simulated agents' performance like the average time required for solving a puzzle [85], the number of moves or problem decomposition capability [69], obtained game scores [99, 67], and success rate [3, 107, 51]. Success rate, which is practical even in case the game score is not available, is applicable to many types of challenge, especially the performative and cognitive challenge.

#### 3.3.2 Game Engagement

Game usage and time spent playing games can be seen as game engagement measurements [17]. Game companies attempt to reduce the churn rate, which is defined as the percentage of players leaving the game, e.g., per day or per played level. Some work predicts an individual player churn probability [10, 12], and others estimate the average churn ratio per game level [108]. The former is used to offer extra personalized stimulus to the players that are likely to leave the game. The latter can indicate problem-

atic and non-engaging game levels, allowing game designers to resolve the issues by game parameter tuning like game difficulty adjustment or presenting more novel and engaging game content/mechanics.

### 3.4 Game AI Agents

Advances in deep learning and access to more computational resources have expedited the creation of artificial intelligent agents and motivated game companies to replace and/or augment human playtesters with AI agents. Deep Reinforcement Learning (DRL), Monte Carlo Tree Search (MCTS), and their combination have successfully implemented game-playing agents capable of outperforming human players [95, 125, 144].

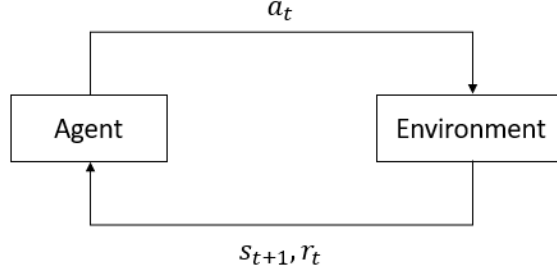
#### 3.4.1 Markov Decision Process

Markov Decision Process (MDP) is a sequential decision-making formalism in which an agent at state  $s_t \in S$  interacts with an environment by taking action  $a_t \in A$  based on policy  $\pi$  and receives the reward  $r_t \in R \subset \mathbb{R}$  and the new state  $s_{t+1} \in S$  with the transition probability of  $p(s_{t+1}, r_t | s_t, a_t)$ . The Markov property denotes that the current state should encapsulate all the information required for predicting the future state of the environment. Therefore, an agent will be able to decide independently of the past states and just based on the current state [131].

#### 3.4.2 Deep Reinforcement Learning

Reinforcement learning (RL) is a common approach for learning (approximately) optimal actions for an MDP. In an MDP environment, a reinforcement learning agent (Figure 3.8) acts based on its current learned policy to gather experience in the form of  $(s_t, a_t, r_t, s_{t+1})$  tuples. The policy  $\pi(a_t | s_t)$  maps each state to a specific action distribution. Using the collected experience, the policy will be optimized to maximize the expected cumulative discounted future reward  $G_t = \mathbb{E}_\pi \left[ \sum_{k=t}^T \gamma^{k-t} r_k \right]$ , where  $T$  is the termination time of an agent-environment interaction episode and  $\gamma$  is the discount factor, which determines the value of future rewards [131]. Typically, the policy is initialized to produce random actions, and the action noise gradually decreases during training as the policy transitions from exploration to exploitation.

Other important concepts in RL are on-policy state value  $V^\pi(s_t) \in \mathbb{R}$  and on-policy state-action value  $Q^\pi(s_t, a_t) \in \mathbb{R}$  functions.  $V^\pi(s_t)$  defines how good a state is in terms of the expectation of the cumulative future reward under the policy  $\pi$ .  $Q^\pi(s_t, a_t)$  defines a similar expectation to show the value of taking a specific action in a particular state [131]. In problems with large



**Figure 3.8.** A reinforcement learning agent.

state and action spaces, where simple lookup tables are not sufficient, deep reinforcement learning utilizes deep neural networks for approximating the value and policy functions.

### *Policy Gradient*

This dissertation utilizes policy gradient, a type of RL that directly optimizes the policy  $\pi_\theta$  parameterized by  $\theta$ , using gradient ascent to maximize the objective function  $J(\theta)$  with respect to policy parameters:

$$\theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t). \quad (3.1)$$

REINFORCE algorithm [147] is an ancestor of the most policy gradient methods, which uses the expected total return as the objective function:

$$J(\theta_t) = G_t = \mathbb{E}_\pi \left[ \sum_{k=t}^T \gamma^{k-t} r_k \right] \quad (3.2)$$

The gradient of the objective can be derived as Equation 3.3 [131]. For positive  $G_t$ , following the gradient will increase the probability of sampling  $a_t$  in state  $s_t$ . For negative  $G_t$ , following the gradient will make taking  $a_t$  in state  $s_t$  less likely. Thus, the updated policy will produce more actions with high returns.

$$\nabla J(\theta_t) = \mathbb{E}_\pi [G_t \nabla \log \pi_\theta(a_t | s_t)] \quad (3.3)$$

REINFORCE is simple and quite intuitive, but the gradient has high variance. However, the variance can be minimized by replacing the return  $G_t$  with advantage  $A^\pi(s_t, a_t) = r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$  [94]. Intuitively, this means that gradient ascent increases the probability of actions that produced higher returns than expected using the policy. In practice, the values need to be approximated using a value function predictor network, i.e., value function  $\hat{V}_\phi^\pi(s_t)$  parametrized by  $\phi$ , which can produce bias and make learning unstable. This is why Generalized Advantage Estimation (GAE) [117] was proposed.

To reduce the bias, the n-step return can be used to calculate the advantage  $\hat{A}^\pi(s_t, a_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} r_{t'} + \gamma \hat{V}_\phi^\pi(s_{t+n}) - \hat{V}_\phi^\pi(s_t)$ , but it causes variance. GAE

generalizes n-step advantage, so instead of cutting off the trajectory at an arbitrary n, it calculates the weighted sum of advantages  $\sum_n \omega_n \hat{A}_n^\pi$  with the exponential decay weights  $\omega_n \propto \lambda^{n-1}$ , which results in:

$$\hat{A}^\pi(s_t, a_t) = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \quad (3.4)$$

Where  $\delta_t = r_{t+1} + \gamma \hat{V}_\phi^\pi(s_{t+1}) - \hat{V}_\phi^\pi(s_t)$ , and  $\lambda$  determines the bias-variance trade-off.

Nowadays, various open-source implementations of RL algorithms exist [70, 35], and one typically only needs to define the learning problem as an MDP, i.e., compose the state observations, define actions and apply them in the environment, and implement the code needed to compute the rewards. Furthermore, most RL algorithms operate in an episodic fashion, which can be described in pseudo-code as:

---

**Algorithm 1** episodic reinforcement learning
 

---

```

for each iteration do
  for each episode do
    Sample an initial state
    until a terminal state encountered or time limit  $T$  do
      Sample an action  $a_t \sim \pi_\theta(a_t|s_t)$  and apply it in the environment
      Observe next state  $s_{t+1}$  and reward  $r_t$ 
      Store the experience tuple  $(s_t, a_t, r_t, s_{t+1})$ 
    end
  end
  Update  $\pi_\theta$  using the collected experience
end
    
```

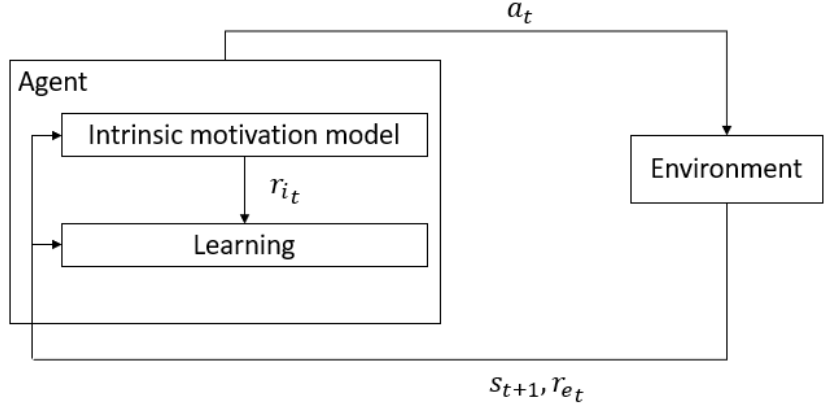
---

This means that one also needs to implement the initial state sampling and define which states are terminal ones. In games, the initial state is typically the game’s start, and player dying is considered a transition to a terminal state.

This dissertation utilizes Proximal Policy Optimization (PPO) [118], one of the most popular episodic RL algorithms. PPO combines GAE with a regularized gradient update to prevent large changes to the policy weights.

### *Intrinsically motivated Reinforcement Learning Agents*

One crucial component of reinforcement learning is the reward function. Different reward functions lead to different performances and problem-solving strategies. Additional to the rewards that come from the environment, another type of reward is usually defined based on the agent’s internal state, which is called intrinsic reward (Figure 3.9). Intrinsic reward ideas are borrowed from human intrinsic motivations and are often used for better exploration in case of sparse external rewards. Moreover, it has been shown that intrinsically motivated agents can produce diverse and human-like behaviors [86].



**Figure 3.9.** An intrinsically motivated reinforcement learning agent adapted from [127]. Such an agent, in addition to the reward from the environment ( $r_{e_t}$ ), receives an intrinsic reward ( $r_{i_t}$ ) based on its internal model.

Curiosity, novelty, information gain, and empowerment are popular intrinsic rewards in reinforcement learning methods. Curiosity encourages the agents to the transitions that yield high prediction error over a learned model [103]. Novelty-seeking methods give unseen states higher rewards [8]. Furthermore, information gain assigns higher rewards to the transitions that provide the agent with the most information about the environment. Information gain can be defined as the KL-divergence between  $p(z|y)$  and  $p(z)$  [59], which means the larger the KL-divergence, the more information about the latent variable  $z$ , is given by observing  $y$ . The latent variable  $z$  denotes the agent’s knowledge of the environment, such as the transition function  $p_\theta(s_{t+1}|s_t, a_t)$ , whereas the variable  $y$  could be the collected experience  $(s_t, a_t, s_{t+1})$  through interacting with the environment.

Another intrinsic reward slightly different from the ones above is empowerment, defined as having the maximum comprehensible effect on the environment. Empowerment is known as the mutual information between the agent’s action and next state given current state  $I(a; s'|s) = H(a|s) - H(a|s', s)$ . In multi-agent environments, coupled empowerment has been demonstrated to produce emergent cooperative and competitive behaviors [48, 50].

Pathak et al. [103] use the error of the learned forward dynamic model as a curiosity measure. Therefore, the agent is rewarded whenever it encounters novel and unseen transitions. Later, Burda et al. [21] show that Pathak et al.’s work is sensitive to randomness in the environment, and the agent receives rewards for spending time at the sources of randomness. For solving this problem, Burda et al. [22] propose Random Network Distillation (RND), in which the agent learns the output of a fixed random neural network. More recently, Pathak et al. [104] used the disagreement between the ensembles of deterministic neural networks as

the intrinsic reward so that the agent does not receive any reward due to the randomness within the environment.

In Publication IV, Unity ML-agents 0.10.0 [70] framework, which implements the PPO algorithm [118] and curiosity reward [103] is employed to implement a game-playing agent for playing a match-3 game.

### 3.4.3 Monte Carlo Tree Search

Monte Carlo Tree Search is an alternative to RL for solving MDP problems. In MCTS, for selecting the approximately best action at each state, four main steps (Figure 3.10) should be iteratively performed until the simulation budget is exhausted:

**Selection** Traverse the tree from the root according to the tree policy until reaching an under-explored node (i.e., a node that has at least one unexpanded child). Upper Confidence bounds for Trees (UCT) is used as the tree traversal policy:

$$\operatorname{argmax}_{s_{d+1} \in C(s_d)} \frac{V(s_{d+1})}{N_{s_{d+1}}} + c \sqrt{\frac{\ln N_{s_d}}{N_{s_{d+1}}}}, \quad (3.5)$$

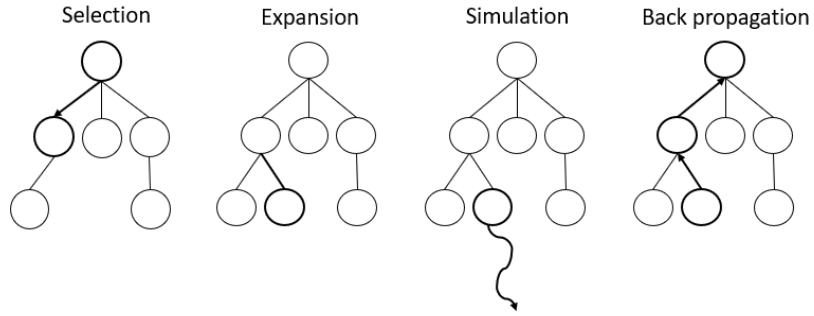
Where  $C(s_d)$  denotes the set of children and  $N_{s_d}$  is the visit number of the node  $s_d$  at depth  $d$ .  $V(s_{d+1})$  indicates the value of the node  $s_{d+1}$ , one of the node  $s_d$ 's children. Parameter  $c$  adjusts the trade-off between exploitation and exploration.

**Expansion** One of the unexplored actions of the selected node in the above step is randomly chosen to be expanded.

**Simulation** From the expanded node's state, the environment is forward simulated based on the rollout policy until it reaches a terminal state or the time horizon limitation is met. Rollout policy could be any policy, even a random one.

**Back-propagation** The return of the simulation is back-propagated towards the tree's root, and the visit numbers of all the nodes in this path are increased by one.

Unlike reinforcement learning, MCTS does not require lengthy training and can be run with an arbitrary computational budget. However, MCTS can be computationally expensive, which can be mitigated by employing a pre-trained DRL policy as the MCTS rollout policy. This can yield more reliable rollout returns. MCTS has been a very successful game-playing agent [52, 107, 57] and can even play the game of Go better than humans when combined with RL [124].



**Figure 3.10.** A Monte Carlo tree search agent.

### 3.4.4 Simulation-based Game Testing Applications

Nowadays, more and more game companies show interest in employing simulated agents for tuning game parameters, estimating the level difficulties, or detecting bugs [66, 51, 30] because this can reduce the cost of game testing substantially. In the context of game testing, winning the game is not necessarily the ultimate goal and AI agents which act like humans are preferable [16].

One can create human-like agents by imitation learning [51, 105]. Gudmundsson et al. [51] trained an agent in a supervised learning manner to learn the human actions in each game state of a match-3 game. Then, the agent performance was used to predict the average human pass rates of the game levels. Similarly, Pfau et al. [105] created different simulated agents by learning human players' actions in different character classes of a role-playing game. They tried to balance the game parameters through inter-class matches between the agents. Imitation learning is a data-driven approach, so it needs lots of human data, which is not always available, for instance, when new game mechanics or game contents are introduced. Moreover, it has difficulty generalizing to the states that are not available in the expert dataset.

According to computational rationality [44], an alternative to imitation learning, humans behave in the direction of maximizing the utility of actions under the restricted computational capabilities of their brains. Computational rationality can be implemented using DRL or MCTS if the reward function can be defined or inferred. Poromaa [107] showed that the performance of an MCTS agent could be mapped to the average human pass rates per game level. Bergdahl et al. [9] employed DRL agents to detect parts of the map that agents get trapped inside them or unexpected paths that agents could use to reach the goals. In a match-3 game, Kristensen et al. [75] predicted each level's pass rates using the number of moves a PPO agent uses to complete the level. They investigated the effect of curriculum learning. They found that training the agent on



a batch of preceding levels and fine-tuning it on the target level produces the best results.

Some work concentrates on predefined strategies and player types such as runner, explorer, and achiever [52, 30, 57, 121] to generate game-playing agents. Silva et al. [30] tested a board game using rule-based agents with different game-playing styles and found multiple defects in the game ruleset. Holmgaard et al. [57] replaced the UCT formula of MCTS with different utility functions corresponding to different game-playing styles. They investigated the interaction of each type of agent with the game elements and the way each of them traverses the dungeon maps. Such agents later could be employed to test whether an expected outcome from a particular game-playing style (e.g., the time each player type spends in various parts of the map) would be satisfied or not. Shun et al. [122] created more human-like agents with less training cost by using common strategies among human players as the action space of the DRL agents. The main limitation of the heuristic-based methods is that predefined heuristics created by game designers may not cover all human strategies.

There is a need for methods that are less dependent on manually defined heuristics. For example, Shen et al. [121] combined evolutionary algorithms, DRL reward shaping, and multi-objective optimization for creating various agent types in a combat game, e.g., defensive, neutral, and aggressive. Furthermore, intrinsic rewards can be used to create self-motivated agents with particular or diverse behaviors [48, 50, 41]. Recently, Matusch et al. [86] have shown that intrinsically motivated agents produce similar behavior to human players by measuring the overlap in their observations. However, the combination of different intrinsic rewards and their application in game testing remains under-explored.

## 4. Developed Methods

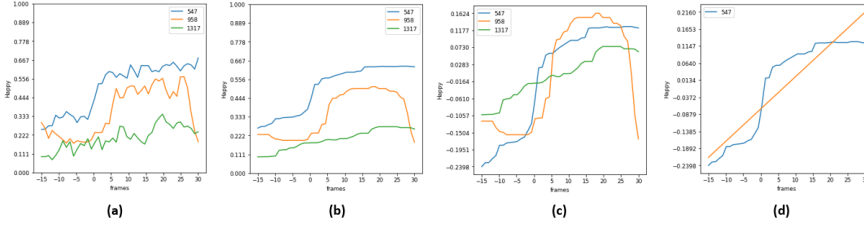
This chapter summarizes the methods developed for publications I-IV. On a high level, Publication I analyzes the players' facial expressions at game events. It raises caution for direct usage of facial expressions in guiding game development since they may not communicate real player emotions. Publication II extends Publication I by taking advantage of multi-modal expression signals, including facial, vocal, and speech data. This attempt shows that expression signals, especially facial and vocal data, can be used to detect game highlights. However, they are less successful in detecting fine-grained emotions, and this task appears hard even for human annotators.

Publications III and IV focus on employing artificial intelligence to model player experience and behavior. Publication III trains a deep reinforcement learning agent per game level, collects agent gameplay data, and uses the data to predict game level pass and churn rates, i.e., measures of difficulty and retention/engagement. The player population changes over the game levels are simulated, capturing how the relation between levels pass and churn rates evolves. In practical game production, the results could be used for identifying game levels that would cause high churn rates. This method is extended in Publication IV by utilizing Monte Carlo tree search agents and a better feature selection for the pass and churn rate prediction models. One of the main takeaways of Publication IV is that computing statistics from the best agent runs can be a better predictor of human player data than the average agent performance.

### 4.1 Publication I

We propose a vision-based system for analyzing players' facial expressions at game events, utilizing an existing platformer game dataset with event logs and facial videos [71]. Our approach computes changes in emotional expression intensities at each game event, which we call affect gradients.

Our motivations for this publication are: 1) facilitating the generation



**Figure 4.1.** Happiness signal for an individual player on three instances of *getting killed by enemy* event. a) the raw signal. b) the median-filtered signal. c) the median-filtered and mean-normalized signal. d) a line is fitted to the signal, and its slope shows the affect gradient.

and validation of computational models of players’ emotions and motivations, and 2) creating a tool for analyzing and summarizing playtest videos to reduce game testing costs.

#### 4.1.1 Facial Expression Analysis

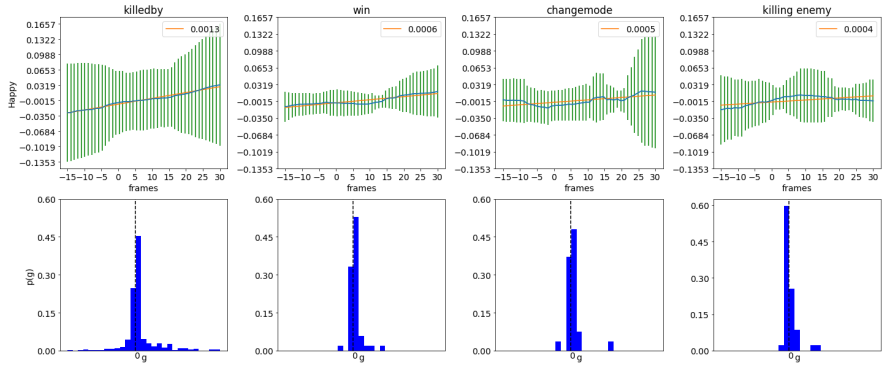
A VGG16-based deep neural network [126] is trained to map players’ facial expressions to 7 basic classes of emotions, e.g., happiness, anger, sadness, and surprise. We train the neural network on Kaggle facial expression recognition challenge dataset<sup>1</sup>.

#### 4.1.2 Affect Gradient

We use Platformer Experience Dataset (PED) [71] to evaluate our automatic facial expression analysis system. PED contains the game events and their corresponding timestamps, collected from 28 males and 30 females playing a clone of Super Mario Bros, while their facial data has been recorded.

Figure 4.1 illustrates our affect gradient method steps. First, the neural network trained on the facial expression recognition dataset is applied to the facial data extracted from the PED dataset to get the raw expression signals (i.e., the probability of emotion classes). Then, having videos of 30 frames per second, a median filter with a kernel size of 15 (half a second) is used to smooth the expression signals. For each game event, a time window of  $[-15, 30]$  frames ( $[-0.5, 1]$  seconds) around it is extracted, and the mean of the window is subtracted from the signal. Finally, the slope of a line fitted to the resulting expression signal is considered the affect gradient of that game event. Moreover, for each game event, we present the histogram of individual affect gradients and the summary plot of means and standard deviations of all segmented signals (Figure 4.2). The slope of a line fitted to the mean signal is considered as the summary affect gradient.

<sup>1</sup><https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>



**Figure 4.2.** Top: summary plots of means and standard deviations of all happiness signal segments and the summary affect gradients (i.e., the slopes of the lines fitted to the mean signals). Bottom: histograms of affect gradients computed from the segmented signals.

### 4.1.3 Summary of Results

Our key finding—replicating previous work using facial electromyography instead of vision [74]—is that *getting killed by enemy* often causes smiling, perhaps due to irony or social signaling related to the recording situation of being observed in a research setting. This is detected as increased happiness by the system. In contrast, events like *killing enemy* might get interpreted as anger since players usually frown during concentration. Our results warn of direct usage of player facial expressions to guide game development. For example, although *getting killed by enemy* has the highest happiness gradient, it does not seem reasonable to design a game that kills the player as often as possible.

### 4.1.4 Discussion and Future Work

Visual inspection of the Affect Gradient showed similar results to the previous physiological emotion analysis methods [74]. However, for the Affect Gradient to be used reliably, it would need further validation beyond the observations made in the paper. Such a quantitative analysis would require comparison with some ground truth measure (e.g., physiological signals), or could be done by evaluating the success of Affect Gradient in predicting game success or modeling player experience, which is beyond the scope of this research and remains future work.

We further emphasize that the paper’s contribution is to introduce a tool for visualizing and summarizing playtest videos by focusing on game events instead of gameplay graphs; it does not intend to compare the approach against some existing human emotion analysis approach. In the same vein as recent critical affect research [6, 135], this paper further highlights the limitation of facial expression analysis in game design. Naively, one

might expect to optimize the game experience by delivering situations to the players that make them happy. However, a direct interpretation of our happiness measurement is that dying makes the players happy, while it is obvious that maximizing the player death rate is not a viable design goal, except in rare cases where the extreme difficulty is used for comical and viral purposes, as with Flappy Bird and Trap Adventure II. The limitation of facial expression analysis motivates Publication II to investigate other modalities like player speech in player emotion analysis.

## 4.2 Publication II

Publication II investigates the emotional expressions of game streamers in two puzzle games, *Unravel* [63] and its sequel, *Unravel 2* [64]. Multi-modal expression signals of players, including facial expressions, voice expressions, speech sentiment, and low-level audio features like pitch and loudness, are integrated using multiple neural networks to detect the players' emotions labeled by human annotators.

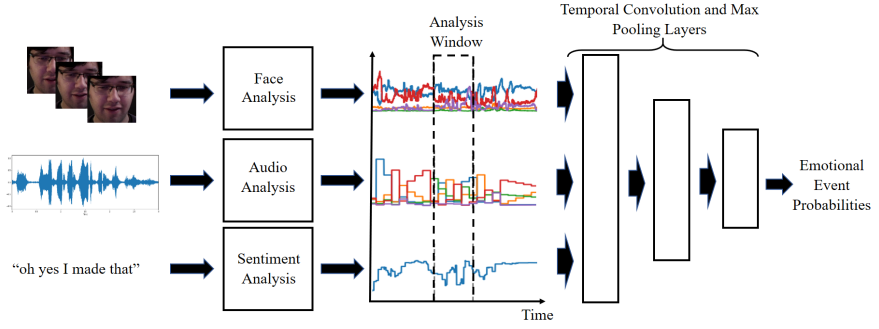
The main motivation of this publication is to create an automatic emotion recognition system to assist game testing by detecting gameplay highlights. Such a system could help analyze player experience by selecting parts of the playtest videos for further investigation.

### 4.2.1 Dataset Creation

The reason behind selecting *Unravel* and *Unravel 2* for our analysis is that their linear game design leads to an almost similar experience for all the players, and there were enough game streams available for them. Videos in which the players' face was visible, only one player was present, and an automatic English transcript was available were selected to be annotated. Instead of using basic emotions, 13 event codes related to the stream videos, such as startle, happiness, and surprise, were selected, and their expressive features were defined. *Not Applicable (NA)* label, one of the 13 codes, was assigned to the events that none of the 12 codes could describe. In addition, each event code was classified as pleasant, unpleasant, or neutral. Two human annotators labeled each video and marked the Top 5 most emotionally salient events for each video.

### 4.2.2 Automatic Emotion Recognition

Our dataset is too small to allow training a large neural network model that could map the raw video and audio directly to the target labels. Therefore, as shown in Figure 4.3, first, separate neural networks trained with existing larger datasets extract emotionally salient features. Then,



**Figure 4.3.** Overview of our automatic emotional event detection system.

a final small neural network trained on our custom dataset maps these features to our targets. Our system comprises the following building blocks:

**Facial expression analysis** VGG16 neural network [126] is trained for classifying facial data into seven emotion classes. This neural network is trained on Affectnet dataset [97], which contains approximately 500k manually annotated facial images. In order to mitigate the class imbalances, we down-sample the dataset into the final 73K train and 18K test images. Finally, the trained neural network is applied to the cropped facial images of streamers and outputs the emotion probability signals.

**Audio expression analysis** Streamer’s audio signal is resampled at 16 KHz. Audio’s power spectra calculated using Short-Time Fourier Transform (STFT) with a window size of 512 and stride of 128 are converted to decibel units and down-sampled by a factor of 4. Each 3 seconds of audio signal results in a  $64 \times 94$  spectrogram image that is mapped to 7 classes of emotions through a VGG16 neural network [126] trained with 9K and tested with 2K data points from the composition of various datasets [23, 37, 81, 68, 24, 23].

The trained neural network is used to extract features from the stream’s audio. However, besides the streamer’s voice, the stream’s audio also includes game music. In order to separate the vocal part, we use the Librosa library [88] to filter out the harmonic part of the audio. Then, the streamer’s audio in segments of 3 seconds is fed to the trained neural network, which outputs the emotion probability signals. The emotion probabilities of the blank parts of the audio signal are filled by linear interpolation of the neighbor values.

**Speech sentiment analysis** A neural network encompassing embedding, temporal convolution, and fully-connected layers is trained on the Amazon reviews dataset<sup>2</sup> to map the reviews into positive and nega-

<sup>2</sup><https://www.kaggle.com/bittlingmayer/amazonreviews/discussion/33444>

**Table 4.1.** Event congestion and inter-rater agreement in different window lengths and granularity levels.

Window length	Congestion	Inter-rater agreement			
		2-class	2-class-top events	4-class	14-class
1	0.0	59.6	53.4	34.9	18.8
2	0.1	64.3	56.8	39.3	24.0
3	0.3	67.0	59.1	41.8	27.3
4	0.5	68.3	61.0	43.1	29.4
5	1.1	68.7	60.3	43.7	30.8

tive classes. The dataset contains 3600K train and 400K test data. First, streamer speech is extracted using Youtube automatic transcript system. Then, we feed the 3-second segments of speech to the trained neural network to get their positivity probabilities. Like the audio signal, we have to handle blank speech segments where the streamer does not speak. The sentiments of these segments are interpolated linearly based on the nearest non-blank segments.

**Low-level audio features** Voice loudness and pitch often change with emotional events. Thus we include audio pitch, root-mean-square audio signal power, and perceptually weighted loudness as extra features.

**Emotionally salient event detection** In the final step, we combine all the extracted features from the above steps and feed them to the neural network with multiple temporal convolution layers and one fully-connected layer. We test multiple classification tasks with different granularities: event/no event binary classification, pleasant/unpleasant/neutral/no event classification, top 5 event/ no top 5 event binary classification, and full 13 event codes plus no event classification.

### 4.2.3 Summary of Results

Table 4.1 shows the congestion of events, i.e., the percentage of timestamps that are logged by the same annotator in the same window as an event, and human annotators’ inter-rater agreement in different window lengths and granularity levels. As granularity increases, the agreement between annotators decreases due to the difficulty of detecting subtle expressions [148] and discriminating between less-distinct emotions like startle and surprise. Increasing the window length results in a higher inter-human agreement, but it also increases the event congestion, so we limited our automatic emotion analysis to windows with a size of 1 to 5 seconds. Our automatic emotion analysis results (Table 4.2) indicate that our system is mainly successful in binary classification tasks, which could make it suit-

able for game stream highlight detection. Furthermore, facial expressions and audio features are the most effective modalities in our system.

#### 4.2.4 Discussion and Future Work

Using all the modalities does not typically produce the best results, which might be because of overfitting. All the input features should present significant additional information to prevent overfitting, but features like speech sentiments and audio emotion expressions have low impacts, as shown in Tables 4.2 and 4.3. Since we are using Youtube automatic transcript, the data quality might not be good. Moreover, the low performance of audio emotion analysis might be because of the game music and other sound effects degrading the audio quality.

Notably, our automatic emotion recognition system and the inter-rater agreement are not completely comparable, because the automatic system tries to predict the aggregation of both human annotators' data.

A major limitation of our work is the audio and video classifier training data, which has fewer emotion classes than our own data. Future work should investigate the use of new and more extensive datasets [33, 102].

For future work, one can improve our dataset by providing manually-extracted transcripts of the streams and augmenting the dataset with the duration of each event instead of just marking the event's occurrence timestamps. In addition, providing game events as an extra modality would be beneficial. Recent work has shown that game events can be extracted using supervised learning applied to gameplay videos [65].

### 4.3 Publication III

Publication III is motivated by the need to develop better models of player experience and behavior for testing a game's design before releasing it to the real players. This reduces the need for human playtesters and eases the repetitive and expensive task of game testing.

In Publication III, game-playing agents' gameplay statistics are used to predict the difficulty and engagement of match-3 game levels. In addition, we try to capture the relation between these two game metrics.

#### 4.3.1 Game Description

A non-deterministic physics-based free-to-play mobile game, Angry Birds Dream Blast [40], is used for evaluating our method (Figure 4.4). Angry Birds Dream Blast is a match-3 game in which players should collect game objects specified as the level goal to pass a level. They can collect adjacent bubbles with the same color or use boosters to collect a large number of

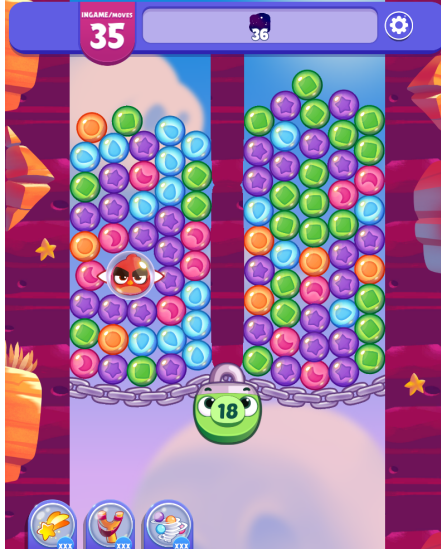


**Table 4.2.** Accuracy of classification using different window lengths and granularity levels. In each column, different input features are enabled for the final neural network. FE, S, AE, and AF indicate facial expressions, speech sentiment, audio expression analysis, and audio features, respectively. "full" column uses all 4 types of input features.

Granularity	Window length	Accuracy (%)				
		FE	FE+S	FE+S+AE	FE+S+AF	Full
2-class	1	63.0	63.5	63.6	<b>64.9</b>	<b>64.9</b>
	2	68.5	68.7	68.3	<b>70.7</b>	69.8
	3	67.9	67.5	67.0	<b>68.6</b>	67.5
	4	67.6	67.0	66.8	<b>68.7</b>	68.0
	5	68.1	67.6	67.1	<b>68.7</b>	68.0
2-class/ top events	1	70.2	68.7	67.3	<b>72.5</b>	71.3
	2	74.9	76.3	75.9	<b>80.4</b>	77.6
	3	74.3	73.4	73.4	75.6	<b>76.6</b>
	4	73.1	71.6	71.5	77.2	<b>78.0</b>
	5	71.2	69.0	71.3	74.2	<b>76.3</b>
4-class	1	41.9	<b>42.9</b>	42.2	40.3	39.5
	2	42.7	43.9	43.0	<b>44.5</b>	43.2
	3	<b>42.8</b>	42.3	40.3	42.5	41.6
	4	44.5	44.0	43.5	<b>45.4</b>	43.5
	5	41.0	41.7	41.9	<b>42.1</b>	41.7
14-class	1	19.8	<b>21.6</b>	21.0	19.4	20.7
	2	24.0	22.6	23.5	<b>26.4</b>	25.1
	3	18.3	<b>22.7</b>	21.3	21.7	21.3
	4	19.6	21.2	20.5	<b>23.8</b>	22.8
	5	19.7	19.1	19.7	<b>22.7</b>	20.8

**Table 4.3.**  $F_1$ -score of classification using different window lengths and granularity levels. In each column, different input features are enabled for the final neural network. FE, S, AE, and AF indicate facial expressions, speech sentiment, audio expression analysis, and audio features, respectively. "full" column uses all 4 types of input features.

Granularity	Window length	$F_1$ -score (%)				
		FE	FE+S	FE+S+AE	FE+S+AF	Full
2-class	1	60.0	60.3	61.4	<b>63.0</b>	62.3
	2	66.5	66.9	66.8	<b>69.9</b>	68.4
	3	65.8	65.2	64.3	<b>66.3</b>	64.3
	4	65.0	64.5	64.0	<b>66.4</b>	65.3
	5	65.1	65.0	64.0	<b>66.9</b>	65.1
2-class/ top events	1	69.1	66.8	64.6	<b>71.5</b>	69.3
	2	74.4	76.0	75.9	<b>80.7</b>	77.4
	3	73.2	71.7	71.6	74.1	<b>75.3</b>
	4	72.2	70.3	70.4	76.5	<b>78.0</b>
	5	69.0	65.8	69.1	72.6	<b>75.3</b>
4-class	1	51.8	<b>52.4</b>	51.9	49.5	49.1
	2	53.4	54.5	53.5	<b>55.4</b>	54.0
	3	<b>53.4</b>	53.1	50.6	53.3	52.1
	4	54.9	54.0	53.6	<b>55.5</b>	53.8
	5	51.1	51.8	<b>52.1</b>	52.0	51.8
14-class	1	29.5	<b>34.0</b>	33.1	29.2	32.0
	2	35.9	34.1	35.4	<b>39.4</b>	38.1
	3	28.1	<b>34.1</b>	32.2	32.9	31.6
	4	30.1	32.6	31.7	<b>34.9</b>	33.6
	5	29.5	29.3	30.7	<b>34.4</b>	32.2



**Figure 4.4.** A screenshot of Angry Birds Dream Blast.

objects in a row or column. There are obstacles and locks in the game which should be removed in order to move forward.

#### 4.3.2 Game-playing Agent

A Deep Reinforcement Learning (DRL), particularly Proximal Policy Optimization (PPO) [118], agent is trained per Angry Birds Dream Blast [40] level. In order to use DRL, as explained earlier in Section 3.4, the game should be framed in the form of a Markov Decision Process (MDP). The MDP is defined as follows:

**State observation** A combination of  $84 \times 84$  game screenshot images and a numerical vector encompassing game details like the number of moves left, the number of level goals to collect, lock information, and the camera position is used as the state observation. The images are encoded through a composition of convolution and fully-connected layers and then concatenated with the numerical vector observation.

**Action space** Each action specifies a position in the game environment that the agent can tap. The game screen is discretized into a  $32 \times 32$  grid with the center of each cell corresponding to one of the actions, leading to a discrete action space with the size of 1024.

**Reward function** The agent's reward function consists of extrinsic rewards from the game environment and the intrinsic reward calculated as the forward dynamic model prediction error [103], which indicates how surprised and curious the agent is about a transition.

Extrinsic reward components are: win bonus, lose penalty, level goal collection percentage, progress in unlocking the game locks, spatial movement of the agent toward the end of the game, a small constant negative reward for penalizing the agent for each tap, and a click reward which directs the agent towards tapping on the matches as close as possible and is computed as  $r_{distance} = c_0 \exp\left(-\frac{d}{c_1}\right)$ , where  $d$  is the distance to the closest match and  $c_0$  and  $c_1$  are tuning parameters.

### 4.3.3 Game Level Difficulty and Engagement Prediction

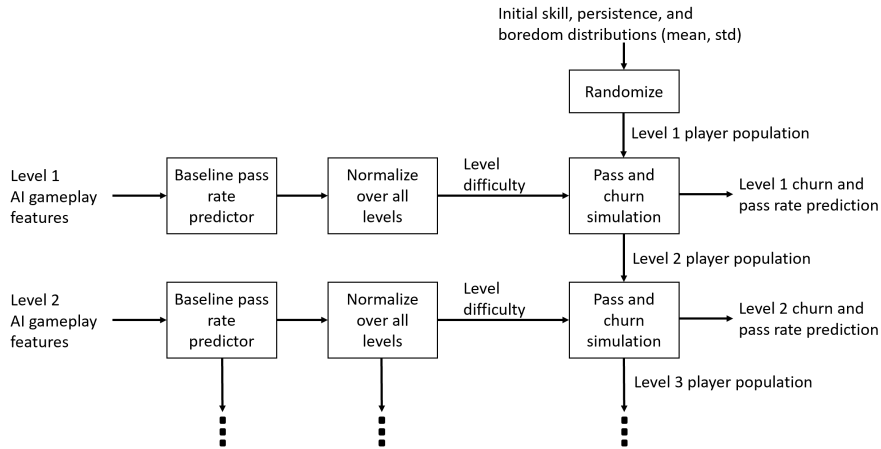
Game level difficulties are measured by the levels' average player pass rates. In addition, levels' churn rates are used as a measure of engagement. We propose two methods for predicting the pass and churn rates. The Baseline model uses simple regression models to map AI agents' gameplay statistics (e.g., min, max, mean, std, different percentiles of agent's pass rate, cleared goals percentage, and moves left ratio) to the pass and churn rates. The Extended model (Figure 4.5) considers how the player population evolves over the game levels as the churned players are removed from the population. This leads to better modeling of the relation between pass and churn rates.

### 4.3.4 Extended Model

This method considers a population of players with skill, persistence, and the inclination to get bored attributes. Initially, the attributes are sampled from normal distributions. In addition, game level difficulties are estimated with the normalized baseline model's pass rate predictions.

Given the initial player population distributions and level difficulties, the player population evolution over game levels is simulated using simple behavioral rules. At each game level, if a simulated player's skill level is higher than the level difficulty, the player passes the level. Otherwise, the player tries again and learns from its own mistakes until it passes the level or the number of attempts is higher than the player's persistence level; in the latter case, it leaves the game. This way, we can model how less persistent players leave the game earlier. Moreover, players who pass the level will leave the game with some probability according to their tendency to get bored. This way, we can model how game engagement usually reduces over time [143, 142]. Finally, the players who have not left the game advance to the next level, and the process is then repeated for that level.

We use CMA-ES [54] for optimizing our simulation parameters, including the means and standard deviations of the skill, persistence, and boredom normal distributions, learning rate, and the amount of noise added to skill and persistence. The objective function  $MSE(\rho_p) + w_{churn}MSE(\rho_c)$  is used,



**Figure 4.5.** The overview of our extended pass and churn rate prediction method. Level difficulty and current player population are passed to the pass and churn rate simulation block, which at each level outputs the pass and churn rate of that level and the remaining player population for the next level.

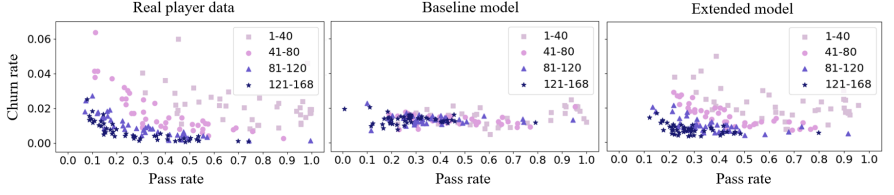
where MSE is the mean squared error,  $\rho_p$  and  $\rho_c$  are pass and churn rate predictions, and we use human pass rate variance divided by human churn rate variance as  $w_{churn}$ .

4.3.5 Summary of Results

Table 4.4 presents mean squared errors (MSEs) and mean absolute errors (MAEs) of the baseline and extended methods computed using 5-fold cross-validation. The extended model acts similarly to the baseline model in terms of pass rate prediction, but it improves churn rate prediction with an effect size of approximately two standard deviations. Figure 4.6 demonstrates our extended method’s ability to better model the relation between the pass and churn rates over game levels.

**Table 4.4.** Mean squared errors and mean absolute errors of pass and churn rate predictions.

Method	Validation MSE		Validation MAE	
	Pass rate	Churn rate	Pass rate	Churn rate
Baseline	$\mu = 0.02244$ $\sigma = 0.00803$	$\mu = 0.00013$ $\sigma = 0.00003$	$\mu = 0.11228$ $\sigma = 0.01663$	$\mu = 0.00866$ $\sigma = 0.00076$
Extended model	$\mu = 0.02320$ $\sigma = 0.00831$	$\mu = 0.00008$ $\sigma = 0.00002$	$\mu = 0.11467$ $\sigma = 0.01647$	$\mu = 0.00607$ $\sigma = 0.00073$



**Figure 4.6.** The scatter plots of the real players’ and our simulations’ pass and churn rates over 168 Angry Birds Dream Blast levels. The baseline method uses simple regression models to predict the pass and churn rates. Our Extended method improves it by considering player population changes over the levels. Scatter plot colors and shapes correspond to different game levels.

### 4.3.6 Discussion and Future Work

Evaluating our method on a fairly small dataset of game levels and the low training speed of the DRL agents are the main limitations of this work. Furthermore, our pass rate predictions for the harder levels are not very accurate (Figure 4.6). This issue is investigated in Publication IV, which directly builds on Publication III.

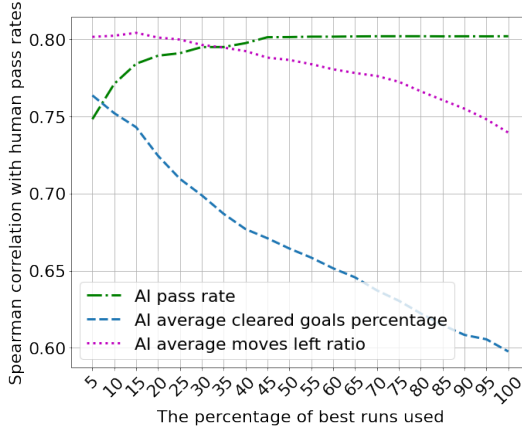
## 4.4 Publication IV

Like Publication III, this publication aims to improve AI agents for accurate player experience and behavior modeling, which could help game developers tune game parameters and select a more engaging game design without requiring human playtesters. In this publication, we extend Publication III with two improvements: 1) using Monte Carlo tree search and 2) better feature selection for the prediction models. We use the same dataset as Publication III to measure if these modifications yield more accurate predictions.

### 4.4.1 Feature Selection

Publication III extracted 16 features from the agent’s gameplay, including mean, standard deviation, min, max, and different percentiles of the agent’s pass rate, cleared goals percentage, and moves left ratio over multiple runs. Then, it predicted the average human pass and churn rates of each game level using these features.

In this work, we only use the three features which have the highest correlation with human pass rates: AI pass rate, average cleared goals percentage, and average moves left ratio. Furthermore, inspired by Kristensen et al. [75], we test computing the features over the percentage of agent’s best runs (i.e., runs with the highest moves left ratio) that yields the highest correlation with human pass rates. In the following, we denote



**Figure 4.7.** The correlation of agent’s statistics calculated over different percentages of agent’s best runs and human pass rates.

the full 16 features as F16, the 3 features as F3, and the 3 features computed from only the best runs as F3P. As shown in Figure 4.7, although AI pass rates calculated over all the runs have the highest correlation with human pass rates, using only 5% and 15% of the best runs is better for computing the average cleared goals percentages and the average moves left ratios.

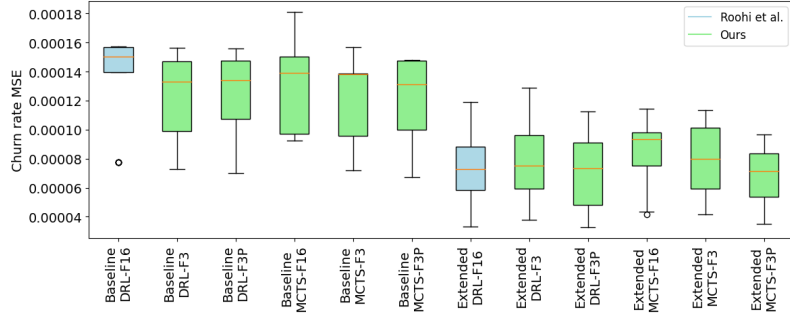
#### 4.4.2 Summary of Results

For our experiments, we use a total of 12 settings encompassing 2 prediction methods from Publication III (i.e., baseline and extended models), 3 feature selection strategies (i.e., F16, F3, and F3P), and 2 AI agents (i.e., DRL and MCTS). Since collecting MCTS statistics is computationally expensive, we only execute 20 runs, which makes applying F3P feature selection on MCTS statistics unreliable. Therefore, we combine F3 features from MCTS and F3P features from DRL for *MCTS-F3P* experiments. Furthermore, to make MCTS more sample-efficient, we use the DRL policy as the rollout policy.

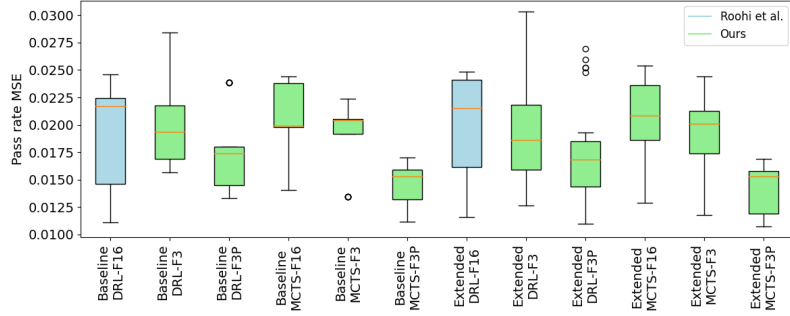
According to the box plots of pass and churn rate prediction mean squared errors in Figures 4.9 and 4.8, our F3P features yield the best pass rate predictions, and Extended-MCTS-F3P produces the best overall results. All the extensions in this work act similarly in terms of churn rate prediction.

#### 4.4.3 Discussion

Our experiments demonstrate that MCTS combined with DRL can yield better predictions of human pass rates. Additionally, we confirm that computing the statistics over the best agent’s runs rather than the agent’s average performance can improve the predictions. The main limitation of



**Figure 4.8.** Box plot of churn rate mean squared errors.



**Figure 4.9.** Box plot of pass rate mean squared errors.

our work is that our current dataset contains a limited number of early game levels, and we have not evaluated our method on late or new game levels. In addition, due to computational limitations, we could not assess the effect of more MCTS runs and different parameter values.





## 5. Conclusion

This dissertation advances AI-assisted playtesting in two ways. First, it proposes methods for the automatic inspection of player emotions and summarizing the analysis results. Our Publication I proposes a new approach for analyzing players' facial expressions. We demonstrate an ability to summarize player data and extract outlier game events. Inspection of the results shows that our emotion analysis approach produces similar results to previous physiological methods without requiring extra hardware with better scaling to large amounts of playtest data.

In addition, Publication II investigates multi-modal expression signals for inspecting game streamers' emotions in different granularities. In this paper, we provide a game stream emotion dataset, which could be utilized in the future game stream and player emotion research. Moreover, the paper demonstrates that extra expression modalities like player voice could improve the automatic detection of emotionally salient game events. Our method produces accuracies on par with the human inter-rater agreements and our results suggest that increasing emotion granularity level makes emotion recognition harder for both machine learning methods and human annotators. Publications I and II's player emotion analysis methods could be used in game testing and player experience research by selecting highlights of playtest videos for further inspection.

Second, this dissertation advances simulation methods and game-playing agents for player experience and behavior modeling. Such methods can assist game developers in detecting game design flaws and selecting the most engaging design from a variety of existing ones without requiring human playtesters. This way, games could be tested at a low cost before releasing to real human players. Our third publication collects AI agent gameplay statistics to predict game level difficulty and engagement. It shows simple simulation-based modeling of player population evolution can replace the need for training different agent types per game level and result in accurate level difficulty and churn predictions. Moreover, Publication IV extends Publication III, and demonstrates that a better selection of features out of AI agent gameplay statistics and combining

DRL with MCTS can improve the predictions.

Regarding limitations and future research, Publication I demonstrated that the direct application of player facial expression analysis for game design is not completely reliable. Publication II prepared a new emotion dataset and illustrated that having other expression signals like voice and speech improves emotion detection. However, augmenting Publication II's dataset with manually extracted video transcript, each emotional event's duration, and annotations of game events remain as future work. Adding game event annotations, similar to Publication I, could improve the accuracy of emotional event detection and help distinguish each expression signal's role in different game event types. The game events could be labeled by human annotators or even automatically extracted using machine learning. In addition to enhancing the current dataset, collecting data in a controlled environment, where the players do not get distracted by social interactions and players' voices can be recorded with higher quality, is another possible avenue for future study.

Publications III and IV model the relation between pass and churn rates in early game levels of a match-3 game. It would be interesting to check the generalizability of our approach in later game levels and other game genres. One could also investigate the changes required in Publication III's player population simulation module for modeling the relation between the pass and churn rates in other game types. Possible options would be a simulation module that includes a more sophisticated model of how consecutive passing/losing affects player behavior and a more complex learning curve model. In future work, it would also be interesting to deploy and test our methods in an actual game development process, to test new game levels before releasing them to real players.

In the long run, advancing the understanding of human emotion, experience and behavior may allow generating human-like and self-motivated artificially intelligent agents. Such agents could act more similarly to human players and be a better replacement or complement for human playtesters. For instance, it might be possible to empower game-testing agents with computational models of emotion. Furthermore, if the agents express their modeled emotions through synthetic facial and speech data, one could then exploit the emotion analysis approaches proposed in this dissertation to visualize the data, detect gameplay highlights, and validate the models by comparing them to real human data. This was one of our motivations for this dissertation from the beginning, but many aspects remain to be explored, e.g., modeling game event novelty, which might also require better models of both human perception and memory.

# References

- [1] Shivam Agarwal, Christian Herrmann, Günter Wallner, and Fabian Beck. Visualizing ai playtesting data of 2d side-scrolling games. In *2020 IEEE Conference on Games (CoG)*, pages 572–575. IEEE, 2020.
- [2] Sinan Ariyurek, Aysu Betin-Can, and Elif Surer. Automated video game testing using synthetic and humanlike agents. *IEEE Transactions on Games*, 13(1):50–67, 2021.
- [3] Daniel Ashlock and Justin Schonfeld. Evolution for automatic assessment of the difficulty of sokoban boards. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.
- [4] Ahmad Azadvar and Alessandro Canossa. Upeq: Ubisoft perceived experience questionnaire: A self-determination evaluation tool for video games. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, New York, NY, USA, 2018. Association for Computing Machinery.
- [5] Leyla Bagheri and Marina Milyavskaya. Novelty–variety as a candidate basic psychological need: New evidence across three studies. *Motivation and Emotion*, 44(1):32–53, 2020.
- [6] Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M Martinez, and Seth D Pollak. Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological science in the public interest*, 20(1):1–68, 2019.
- [7] Richard A Bartle. *Designing virtual worlds*. New Riders, 2004.
- [8] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, page 1479–1487, Red Hook, NY, USA, 2016. Curran Associates Inc.
- [9] Joakim Bergdahl, Camilo Gordillo, Konrad Tollmar, and Linus Gisslén. Augmenting automated game testing with deep reinforcement learning. In *2020 IEEE Conference on Games (CoG)*, pages 600–603. IEEE, 2020.
- [10] Paul Bertens, Anna Guitart, and África Periañez. Games and big data: A scalable multi-dimensional churn prediction model. In *2017 IEEE conference on computational intelligence and games (CIG)*, pages 33–36. IEEE, 2017.

- [11] Paris Mavromoustakos Blom, Sander Bakkes, Chek Tien Tan, Shimon Whiteson, Diederik Roijers, Roberto Valenti, and Theo Gevers. Towards personalised gaming via facial expression recognition. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE'14*, page 30–36. AAAI Press, 2014.
- [12] Valerio Bonometti, Charles Ringer, Mark Hall, Alex R Wade, and Anders Drachen. Modelling early user-game interactions for joint estimation of survival time and churn probability. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.
- [13] Boyan Bontchev and Dessislava Vassileva. Assessing engagement in an emotionally-adaptive applied game. In *Proceedings of the fourth international conference on technological ecosystems for enhancing multiculturality, TEEM '16*, pages 747–754, New York, NY, USA, 2016. Association for Computing Machinery.
- [14] Julia Ayumi Bopp, Elisa D. Mekler, and Klaus Opwis. Negative emotion, positive experience? emotionally moving moments in digital games. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16*, page 2996–3006, New York, NY, USA, 2016. Association for Computing Machinery.
- [15] Julia Ayumi Bopp, Klaus Opwis, and Elisa D Mekler. “an odd kind of pleasure” differentiating emotional challenge in digital games. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, New York, NY, USA, 2018. Association for Computing Machinery.
- [16] Igor Borovikov, Yunqi Zhao, Ahmad Beirami, Jesse Harder, John Kolen, James Pestrak, Jervis Pinto, Reza Pourabolghasem, Harold Chaput, Mohsen Sardari, et al. Winning isn’t everything: Training agents to playtest modern games. In *AAAI Workshop on Reinforcement Learning in Games*, 2019.
- [17] Elizabeth A Boyle, Thomas M Connolly, Thomas Hainey, and James M Boyle. Engagement in digital entertainment games: A systematic review. *Computers in human behavior*, 28(3):771–780, 2012.
- [18] Margaret M Bradley and Peter J Lang. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59, 1994.
- [19] Jeanne H Brockmyer, Christine M Fox, Kathleen A Curtiss, Evan McBroom, Kimberly M Burkhart, and Jacquelyn N Pidruzny. The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of experimental social psychology*, 45(4):624–634, 2009.
- [20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [21] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *ICLR*, 2019.
- [22] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, May 6-9, 2019*.

- [23] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.
- [24] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multi-modal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.
- [25] François Chollet et al. Keras. <https://keras.io>, 2015.
- [26] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80, 2001.
- [27] Mihaly Csikszentmihalyi and Mihaly Csikszentmihaly. *Flow: The psychology of optimal experience*, volume 1990. Harper & Row New York, 1990.
- [28] Mihaly Csikszentmihalyi and Reed Larson. *Flow and the foundations of positive psychology*, volume 10. Springer, 2014.
- [29] Celso M de Melo, Ana Paiva, and Jonathan Gratch. Emotion in games. *Handbook of Digital Games*, pages 573–592, 2014.
- [30] Fernando de Mesentier Silva, Scott Lee, Julian Togelius, and Andy Nealen. Ai-based playtesting of contemporary board games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG '17*, New York, NY, USA, 2017. Association for Computing Machinery.
- [31] Edward L Deci. *Intrinsic motivation*. New York: Plenum Press, 1975.
- [32] Edward L Deci and Richard M Ryan. The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior. *Psychological inquiry*, 11(4):227–268, 2000.
- [33] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*, 2020.
- [34] Alena Denisova, Paul Cairns, Christian Guckelsberger, and David Zendle. Measuring perceived challenge in digital games: Development & validation of the challenge originating from recent gameplay interaction scale (corgis). *International Journal of Human-Computer Studies*, 137:102383, 2020.
- [35] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
- [36] Brandon Drenikow and Pejman Mirza-Babaei. Vixen: Interactive visualization of gameplay experiences. In *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG '17*, New York, NY, USA, 2017. Association for Computing Machinery.
- [37] Kate Dupuis and M Kathleen Pichora-Fuller. Toronto emotional speech set (tess). 2010.
- [38] Paul Ekman, Wallace V Friesen, Maureen O'sullivan, Anthony Chan, Irene Diacoyanni-Tarlatzis, Karl Heider, Rainer Krause, William Ayhan LeCompte, Tom Pitcairn, Pio E Ricci-Bitti, et al. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53(4):712, 1987.

- [39] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern recognition*, 44(3):572–587, 2011.
- [40] Rovio Entertainment. *Angry Birds Dream Blast*. Game [Android, iOS], November 2018.
- [41] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, May 6-9, 2019*.
- [42] SP Fisk and SE Taylor. *Social cognition*. New York: Random House, 1984.
- [43] David L Forbes. Toward a unified model of human motivation. *Review of general psychology*, 15(2):85–98, 2011.
- [44] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278, 2015.
- [45] Miguel González-Duque, Rasmus Berg Palm, David Ha, and Sebastian Risi. Finding game levels with the right difficulty in a few trials through intelligent trial-and-error. In *2020 IEEE Conference on Games (CoG)*, pages 503–510. IEEE, 2020.
- [46] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. <http://www.deeplearningbook.org>.
- [47] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.
- [48] Christian Guckelsberger, Christoph Salge, and Simon Colton. Intrinsically motivated general companion npcs via coupled empowerment maximisation. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.
- [49] Christian Guckelsberger, Christoph Salge, Jeremy Gow, and Paul Cairns. Predicting player experience without the player.: An exploratory study. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '17*, page 305–315, New York, NY, USA, 2017. Association for Computing Machinery.
- [50] Christian Guckelsberger, Christoph Salge, and Julian Togelius. New and surprising ways to be mean. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2018.
- [51] Stefan Freyr Gudmundsson, Philipp Eisen, Erik Poromaa, Alex Nodet, Sami Purmonen, Bartłomiej Kozakowski, Richard Meurling, and Lele Cao. Human-like playtesting with deep learning. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.
- [52] Cristina Guerrero-Romero, Annie Louis, and Diego Perez-Liebana. Beyond playing to win: Diversifying heuristics for gvgai. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 118–125. IEEE, 2017.
- [53] Juho Hamari and Janne Tuunanen. Player types: A meta-synthesis. *Transactions of the Digital Games Research Association*, 1(2), 2014.
- [54] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

- [55] Daniel Hernandez, Charles Takashi Toyin Gbadamosi, James Goodman, and James Alfred Walker. Metagame autobalancing for competitive multi-player games. In *2020 IEEE Conference on Games (CoG)*, pages 275–282. IEEE, 2020.
- [56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [57] Christoffer Holmgård, Michael Cerny Green, Antonios Liapis, and Julian Togelius. Automated playtesting with procedural personas through mcts with evolved heuristics. *IEEE Transactions on Games*, 11(4):352–362, 2018.
- [58] Hendrik Horn, Vanessa Volz, Diego Pérez-Liébaná, and Mike Preuss. Mct-s/ea hybrid gvgai players and game difficulty estimation. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.
- [59] Rein Houthoofd, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, page 1117–1125, Red Hook, NY, USA, 2016. Curran Associates Inc.
- [60] Che-Wei Huang, Shrikanth Narayanan, et al. Characterizing types of convolution in deep convolutional recurrent neural networks for robust speech emotion recognition. *arXiv preprint arXiv:1706.02901*, 2017.
- [61] Qiongxia Huang, Riqing Chen, Xianghan Zheng, and Zhenxing Dong. Deep sentiment representation based on cnn and lstm. In *2017 International Conference on Green Informatics (ICGI)*, pages 30–33. IEEE, 2017.
- [62] Wijnand IJsselstein, Karolien Poels, and Yvonne AW De Kort. The game experience questionnaire: Development of a self-report measure to assess player experiences of digital games. *TU Eindhoven, Eindhoven, The Netherlands*, page 42, 2008.
- [63] Coldwood Interactive. *Unravel*. Game [PC, PlayStation 4, Xbox One], February 2016. Electronic Arts.
- [64] Coldwood Interactive. *Unravel Two*. Game [PC, PlayStation 4, Xbox One], June 2018. Electronic Arts.
- [65] Thanapong Intharah and Gabriel J. Brostow. Deeplogger: Extracting user input logs from 2d gameplay videos. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY ’18*, page 221–230, New York, NY, USA, 2018. Association for Computing Machinery.
- [66] Aaron Isaksen, Daniel Gopstein, and Andrew Nealen. Exploring game space using survival analysis. In *Proceedings of the 10th International Conference on the Foundations of Digital Games, FDG 2015, Pacific Grove, CA, USA, June 22-25, 2015*. Society for the Advancement of the Science of Digital Games, 2015.
- [67] Aaron Isaksen, Drew Wallace, Adam Finkelstein, and Andy Nealen. Simulating strategy and dexterity for puzzle games. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 142–149. IEEE, 2017.
- [68] Philip Jackson and SJUoSG Haq. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, 2014.
- [69] Petr Jarušek and Radek Pelánek. Difficulty rating of sokoban puzzle. In *Proceedings of the 2010 Conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers’ Symposium*, page 140–150, NLD, 2010. IOS Press.



- [70] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.
- [71] Kostas Karpouzis, Georgios N Yannakakis, Noor Shaker, and Stylianos Asteriadis. The platformer experience dataset. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 712–718. IEEE, 2015.
- [72] Jun H. Kim, Daniel V. Gunn, Eric Schuh, Bruce Phillips, Randy J. Pagulayan, and Dennis Wixon. Tracking real-time user experience (true): A comprehensive instrumentation solution for complex systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 443–452, New York, NY, USA, 2008. Association for Computing Machinery.
- [73] J Matias Kivikangas, Guillaume Chanel, Ben Cowley, Inger Ekman, Mikko Salminen, Simo Järvelä, and Niklas Ravaja. A review of the use of psychophysiological methods in game research. *journal of gaming & virtual worlds*, 3(3):181–199, 2011.
- [74] J Matias Kivikangas et al. Emotion and social context in a digital game experience. *Institute of Behavioural Sciences, Studies in Psychology*, 2015.
- [75] Jeppe Theiss Kristensen, Arturo Valdivia, and Paolo Burelli. Estimating player completion rate in mobile puzzle games using reinforcement learning. In *2020 IEEE Conference on Games (CoG)*, pages 636–639. IEEE, 2020.
- [76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [77] Jyoti Kumari, Reghunadhan Rajesh, and KM Pooja. Facial expression recognition: A survey. *Procedia computer science*, 58:486–491, 2015.
- [78] Nicole Lazzaro. Why we play: affect and the fun of games. *Human-computer interaction: Designing for diverse users and domains*, 155:679–700, 2009.
- [79] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [80] Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, 2020.
- [81] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.
- [82] Derek Lomas, Kishan Patel, Jodi L. Forlizzi, and Kenneth R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 89–98, New York, NY, USA, 2013. Association for Computing Machinery.
- [83] Kristen Lucas and John L Sherry. Sex differences in video game play: A communication-based explanation. *Communication research*, 31(5):499–523, 2004.

- [84] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- [85] Timo Mantere and Janne Koljonen. Solving, rating and generating sudoku puzzles with ga. In *2007 IEEE congress on evolutionary computation*, pages 1382–1389. IEEE, 2007.
- [86] Brendon Matusch, Jimmy Ba, and Danijar Hafner. Evaluating agents without rewards. *arXiv preprint arXiv:2012.11538*, 2020.
- [87] Iris B Mauss and Michael D Robinson. Measures of emotion: A review. *Cognition and emotion*, 23(2):209–237, 2009.
- [88] Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.
- [89] Ben Medler, Michael John, and Jeff Lane. Data cracker: Developing a visual game analytic tool for analyzing online gameplay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, page 2365–2374, New York, NY, USA, 2011. Association for Computing Machinery.
- [90] Elisa D. Mekler, Julia Ayumi Bopp, Alexandre N. Tuch, and Klaus Opwis. A systematic review of quantitative studies on the enjoyment of digital entertainment games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’14, page 927–936, New York, NY, USA, 2014. Association for Computing Machinery.
- [91] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.
- [92] Marina Milyavskaya and Richard Koestner. Psychological needs, motivation, and well-being: A test of self-determination theory across multiple domains. *Personality and individual differences*, 50(3):387–391, 2011.
- [93] Pejman Mirza-Babaei, Lennart E. Nacke, John Gregory, Nick Collins, and Geraldine Fitzpatrick. How does it play better? exploring user testing and biometric storyboards in games user research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, page 1499–1508, New York, NY, USA, 2013. Association for Computing Machinery.
- [94] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [95] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

- [96] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [97] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2019.
- [98] Agnes Moors. Theories of emotion causation: A review. *Cognition and emotion*, 23(4):625–662, 2009.
- [99] Thorbjørn S Nielsen, Gabriella AB Barros, Julian Togelius, and Mark J Nelson. General video game evaluation using relative algorithm performance profiles. In *European Conference on the Applications of Evolutionary Computation*, pages 369–380. Springer, 2015.
- [100] Yafeng Niu, Dongsheng Zou, Yadong Niu, Zhongshi He, and Hua Tan. A breakthrough in speech emotion recognition using deep retinal convolution neural networks. *arXiv preprint arXiv:1707.09917*, 2017.
- [101] Suman Ojha, Jonathan Vitale, and Mary-Anne Williams. Computational emotion models: a thematic review. *International Journal of Social Robotics*, 13(6):1253–1279, 2021.
- [102] Cheul Young Park, Narae Cha, Soowon Kang, Auk Kim, Ahsan Habib Khandoker, Leontios Hadjileontiadis, Alice Oh, Yong Jeong, and Uichin Lee. K-emocon, a multimodal sensor dataset for continuous emotion recognition in naturalistic conversations. *Scientific Data*, 7(1):1–16, 2020.
- [103] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML17*, page 2778–2787. JMLR.org, 2017.
- [104] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, pages 5062–5071. PMLR, 2019.
- [105] Johannes Pfau, Antonios Liapis, Georg Volkmar, Georgios N Yannakakis, and Rainer Malaka. Dungeons & replicants: automated game balancing via deep player behavior modeling. In *2020 IEEE Conference on Games (CoG)*, pages 431–438. IEEE, 2020.
- [106] Diah Anggraeni Pitaloka, Ajeng Wulandari, T Basaruddin, and Dewi Yanti Liliana. Enhancing cnn with preprocessing stage in automatic emotion recognition. *Procedia computer science*, 116:523–529, 2017.
- [107] Erik Ragnar Poromaa. Crushing candy crush: predicting human success rate in a mobile game using monte-carlo tree search, 2017.
- [108] David Reguera, Pol Colomer-de Simón, Iván Encinas, Manel Sort, Jan Wedekind, and Marián Boguná. The physics of fun: Quantifying human engagement into playful activities. *arXiv preprint arXiv:1911.01864*, 2019.
- [109] Scott Rigby and Richard Ryan. The player experience of need satisfaction (pens): An applied model and methodology for understanding key components of the player experience. *Retrieved from immersyve.com/PENS\_Sept07.pdf*, 2007.
- [110] Charles Ringer and Mihalis A. Nicolaou. Deep unsupervised multi-view detection of video game stream highlights. In *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG '18*, New York, NY, USA, 2018. Association for Computing Machinery.

- [111] James A Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003.
- [112] James A Russell and Albert Mehrabian. Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3):273–294, 1977.
- [113] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.
- [114] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68–78, 2000.
- [115] Richard M Ryan, C Scott Rigby, and Andrew Przybylski. The motivational pull of video games: A self-determination theory approach. *Motivation and emotion*, 30(4):344–360, 2006.
- [116] Klaus R Scherer. Appraisal theory. *Handbook of Cognition and Emotion*, 1999.
- [117] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.
- [118] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [119] Noor Shaker, Stylianos Asteriadis, Georgios N Yannakakis, and Kostas Karpouzis. A game-based corpus for analysing the interplay between game context and player experience. In *International Conference on Affective Computing and Intelligent Interaction*, pages 547–556, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [120] Kennon M Sheldon, Andrew J Elliot, Youngmee Kim, and Tim Kasser. What is satisfying about satisfying events? testing 10 candidate psychological needs. *Journal of personality and social psychology*, 80(2):325, 2001.
- [121] Ruimin Shen, Yan Zheng, Jianye Hao, Zhaopeng Meng, Yingfeng Chen, Changjie Fan, and Yang Liu. Generating behavior-diverse game ais with evolutionary multi-objective deep reinforcement learning. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3371–3377. International Joint Conferences on Artificial Intelligence Organization, 7 2020.
- [122] Yuchul Shin, Jaewon Kim, Kyohoon Jin, and Young Bin Kim. Playtesting in match 3 game using strategic plays via reinforcement learning. *IEEE Access*, 8:51593–51600, 2020.
- [123] Mohammad Faridul Haque Siddiqui and Ahmad Y Javaid. A multimodal facial emotion recognition framework through the fusion of speech with visible and infrared images. *Multimodal Technologies and Interaction*, 4(3):46, 2020.
- [124] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

- [125] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharmashan Kumar, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [126] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [127] Satinder Singh, Andrew G Barto, and Nuttapon Chentanez. Intrinsically motivated reinforcement learning. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.
- [128] Max Sjöblom and Juho Hamari. Why do people watch others play video games? an empirical study on the motivations of twitch users. *Computers in human behavior*, 75:985–996, 2017.
- [129] Nathan Sorenson, Philippe Pasquier, and Steve DiPaola. A generic approach to challenge modeling for the procedural creation of video game levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):229–244, 2011.
- [130] Samantha Stahlke, Atiya Nova, and Pejman Mirza-Babaei. Artificial players in the design process: Developing an automated testing tool for game level and world design. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '20*, page 267–280, New York, NY, USA, 2020. Association for Computing Machinery.
- [131] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [132] Chek Tien Tan, Sander Bakkes, and Yusuf Pisan. Correlation between facial expressions and the game experience questionnaire. *13th International Conference Entertainment Computing (ICEC)*, LNCS-8770:229–231, 2014.
- [133] Chek Tien Tan, Tuck Wah Leong, and Songjia Shen. Combining think-aloud and physiological data to understand video game experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 381–390, New York, NY, USA, 2014. Association for Computing Machinery.
- [134] Mikke Tavast, Anton Kunnari, and Perttu Hämäläinen. Language models can generate human-like self-reports of emotion. In *27th International Conference on Intelligent User Interfaces*, pages 69–72, 2022.
- [135] Anna Tcherkassof and Damien Dupré. The emotion–facial expression link: evidence from human and automatic expression recognition. *Psychological Research*, 85(8):2954–2969, 2021.
- [136] Silvan Tomkins. *Affect imagery consciousness: Volume I: The positive affects*. Springer publishing company, 1962.
- [137] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalios A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

- [138] Volodymyr Turchenko, Eric Chalmers, and Artur Luczak. A deep convolutional auto-encoder with pooling-unpooling layers in caffe. *arXiv preprint arXiv:1701.04949*, 2017.
- [139] Panagiotis Tzirakis, George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou. End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1301–1309, 2017.
- [140] Marc Van Kreveld, Maarten Löffler, and Paul Mutser. Automated puzzle difficulty estimation. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 415–422. IEEE, 2015.
- [141] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30 of *NIPS’17*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [142] Markus Viljanen, Antti Airola, Jukka Heikkonen, and Tapio Pahikkala. Playtime measurement with survival analysis. *IEEE Transactions on Games*, 10(2):128–138, 2017.
- [143] Markus Viljanen, Antti Airola, Tapio Pahikkala, and Jukka Heikkonen. Modelling user retention in mobile games. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.
- [144] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>, 2019.
- [145] Günter Wallner and Simone Kriglstein. *An Introduction to Gameplay Data Visualization*, page 231–250. ETC Press, Pittsburgh, PA, USA, 2015.
- [146] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.
- [147] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- [148] Tanja SH Wingenbach, Chris Ashwin, and Mark Brosnan. Validation of the amsterdam dynamic facial expression set–bath intensity variations (adfes-biv): A set of videos expressing low, intermediate, and high intensity emotions. *PloS one*, 11(1):e0147112, 2016.
- [149] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495, 2017.
- [150] Ashima Yadav and Dinesh Kumar Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, 2020.

- [151] Georgios N Yannakakis and Ana Paiva. Emotion in games. *Handbook on affective computing*, 2014:459–471, 2014.
- [152] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. Player Modeling. In Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius, editors, *Artificial and Computational Intelligence in Games*, volume 6 of *Dagstuhl Follow-Ups*, pages 45–59. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013.
- [153] Nick Yee. Motivations for play in online games. *CyberPsychology & behavior*, 9(6):772–775, 2006.
- [154] Dong Yu and Li Deng. *Automatic Speech Recognition*. Springer, 2016.
- [155] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

# Errata

## Publication IV

In section 6, we mentioned that "The Baseline-DRL-F16 and Extended-DRL-F16 combinations are the original approaches analyzed previously". However, while Publication III collects AI agent gameplay data from the last training iterations, Publication IV collects AI agent gameplay data in inference mode when the agent is not trained anymore. This, together with the inherent randomness of RL training, explains the slight differences between the numerical values reported in the publications. Nevertheless, the conclusions remain the same.







ISBN 978-952-64-0824-8 (printed)  
ISBN 978-952-64-0825-5 (pdf)  
ISSN 1799-4934 (printed)  
ISSN 1799-4942 (pdf)

Aalto University  
School of Science  
Department of Computer Science  
[www.aalto.fi](http://www.aalto.fi)

**BUSINESS +  
ECONOMY**

**ART +  
DESIGN +  
ARCHITECTURE**

**SCIENCE +  
TECHNOLOGY**

**CROSSOVER**

**DOCTORAL  
THESES**