The Impact of Dynamic Difficulty Adjustment on Player Experience in Video Games

Chineng “Cookie” Vang

University of Minnesota Morris - November 13, 2021

Introduction

- What makes video games fun to play?
- Dynamic Difficulty Adjustment (DDA) is a process used to alter the challenge of a video game in real time
- Not a new concept, but a controversial one

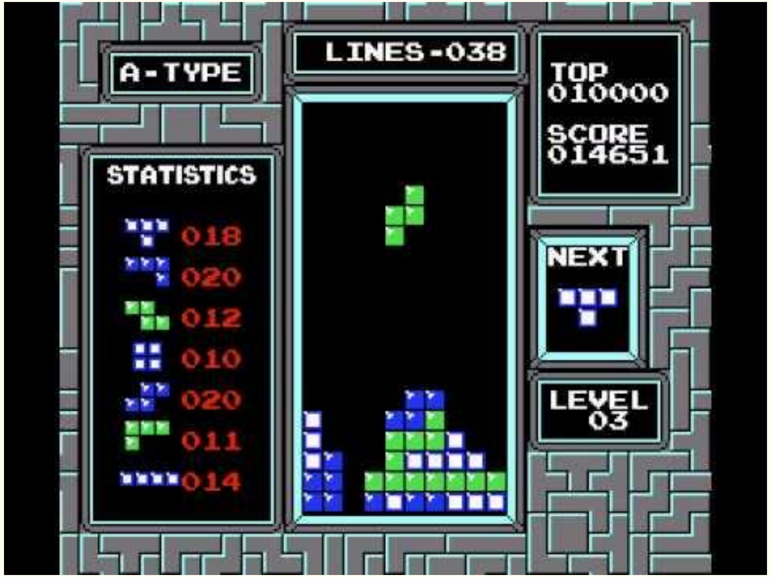
https://www.kotaku.com.au/2011/03/the-maker-of-mario-kart-justifies-the-blue-shell/

Outline

- **Background Information**
    - Context of DDA
    - Flow Theory
    - Controversy of DDA usage
- Implementations of DDA and player feedback
    - Monte Carlo Tree Search (MCTS)
    - Reinforcement Learning
    - Multiplayer DDA
- Conclusion

Background Information

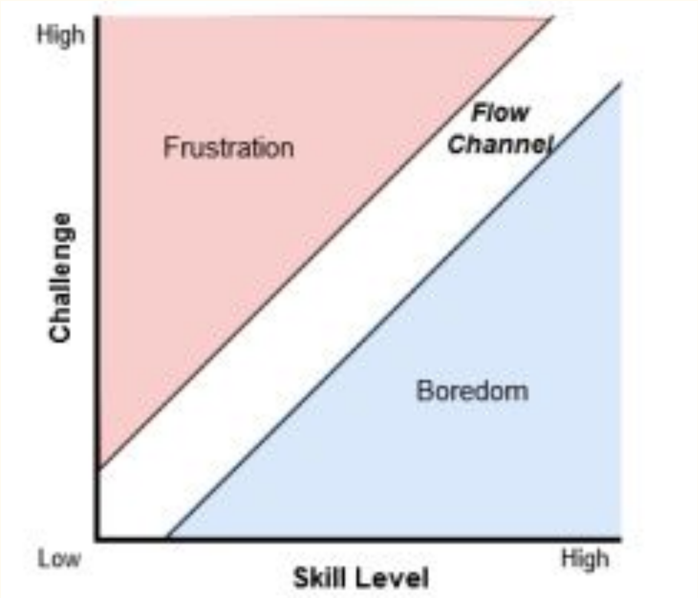
- What does DDA look like?
    - Changing the environment/enemies
    - Altering player ability
- What are other methods of implementing challenge without DDA?



https://www.youtube.com/watch?v=rUCN84SviLc

Background Information

- Goal of DDA
- Flow Theory
  - Flow: A mental state of energetic focus where a person is entirely immersed in completing a task
  - Mihaly Csikszentmihalyi (1970’s)
  - Flow Channel



Source: Pagalyte (2020)

Background Information

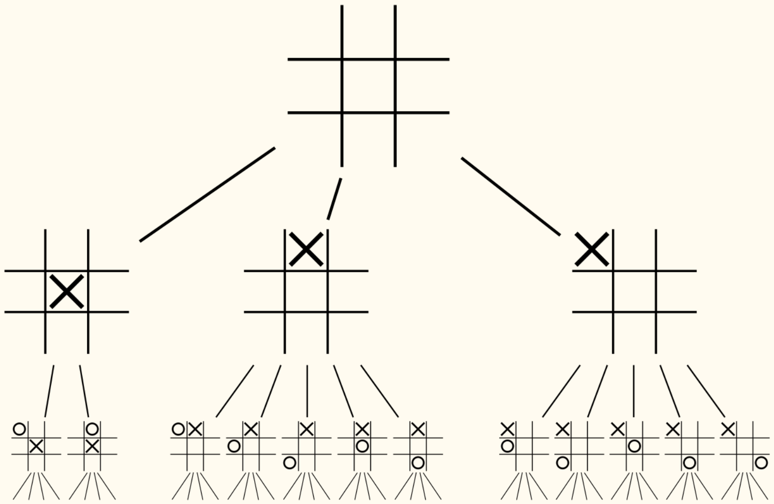
- The argument for DDA
  - Helps create balance in a video game
  - Challenge longevity
- The argument against DDA
  - System interference overshadows player skill
  - Perceived bias towards in multiplayer video games (a.k.a cheating!!)

Outline

- Background Information
  - Context of DDA
  - Flow Theory
  - Controversy of DDA usage
- **Implementations of DDA and player feedback**
  - Monte Carlo Tree Search (MCTS)
  - Reinforcement Learning
  - Multiplayer DDA
- Conclusion

DDA and Monte Carlo Tree Search

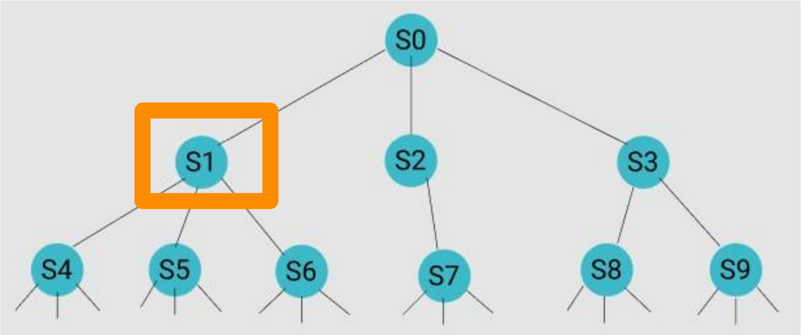
- Experimental fighting game
- What is a game tree?
    - States (nodes of tree)
    - Actions (branches of tree)



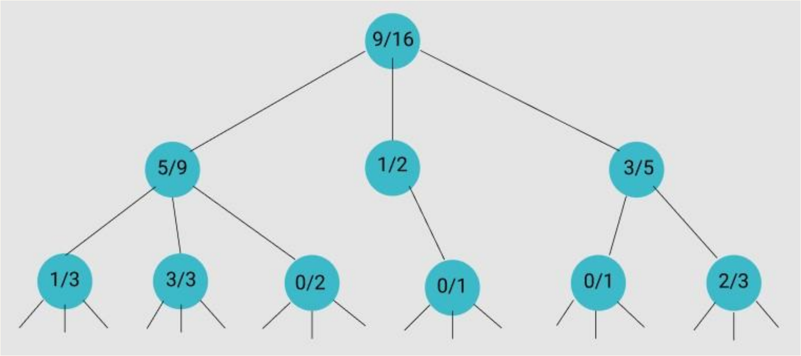
https://commons.wikimedia.org/wiki/File:Tic-tac-toe-game-tree.svg

DDA and Monte Carlo Tree Search

- Monte Carlo Tree Search (MCTS)
  - Builds statistics tree based off of game tree
  - Uses statistics tree to help the computer make best decisions available
  - 3 versions of MCTS to implement DDA



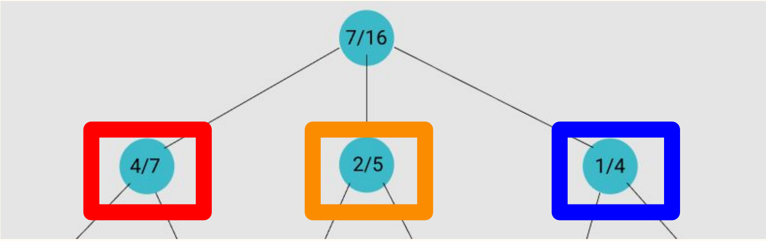
Game Tree



Statistics "Stats" Tree

DDA and Monte Carlo Tree Search

- Challenge Sensitive Action Selection (CSAS)
  - CSAS starts by picking the **mean valued** action
  - Reevaluate difficulty
  - Increase difficulty → Choose actions with a **higher value** than current one
  - Decrease difficulty → Choose actions with a **lower value** than current one



DDA and Monte Carlo Tree Search

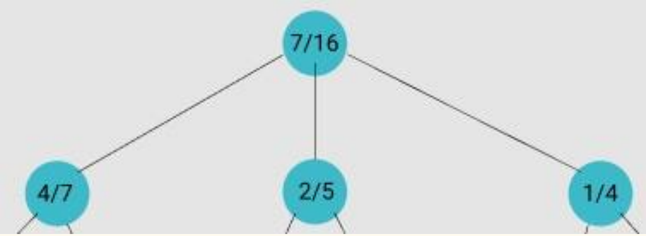
- Reactive Outcome Sensitive Action Selection (ROSAS)
    - ROSAS always picks an action trying to get the computer’s health bar to match the player’s

Player Computer

https://www.youtube.com/watch?v=0G53iYQ2ZHg

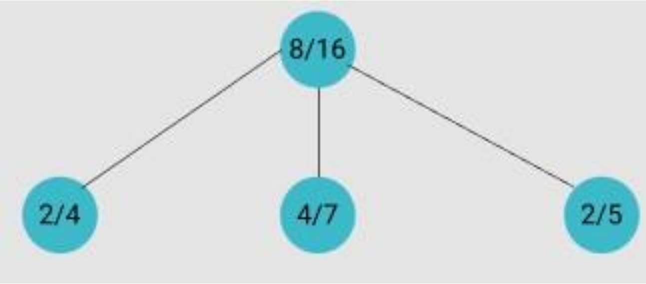
DDA and Monte Carlo Tree Search

- Adaptive True Reactive Outcome Action Selection (ATROSAS)
    - Wants computer’s health bar to match the player’s
    - Best available action chosen (like MCTS)
    - Alters how stats tree is built in MCTS



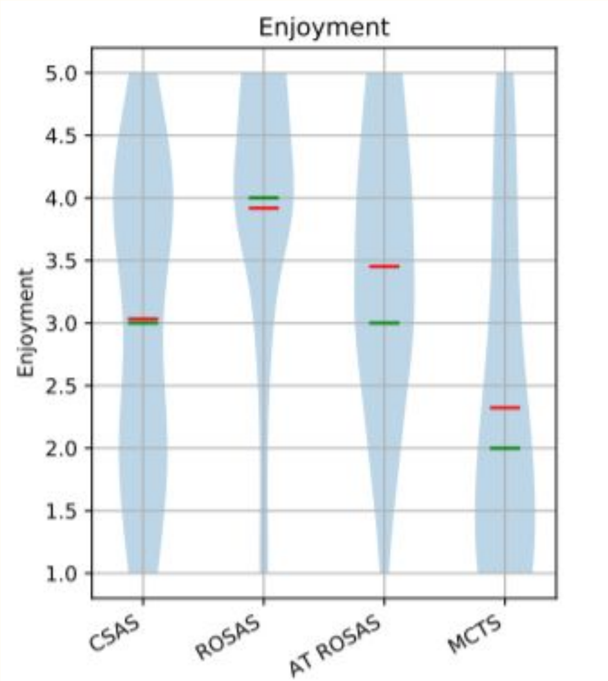
Likelihood of leading to victory

Likelihood of matching health bars



DDA and Monte Carlo Tree Search

- Hypothesis: DDA is best implemented when players win 50% of the time
- Results:
    - CSAS won 41% of the time
    - ROSAS won 46% of the time
    - ATROSAS won 51% of the time
- Enjoyment, Realism, Difficulty



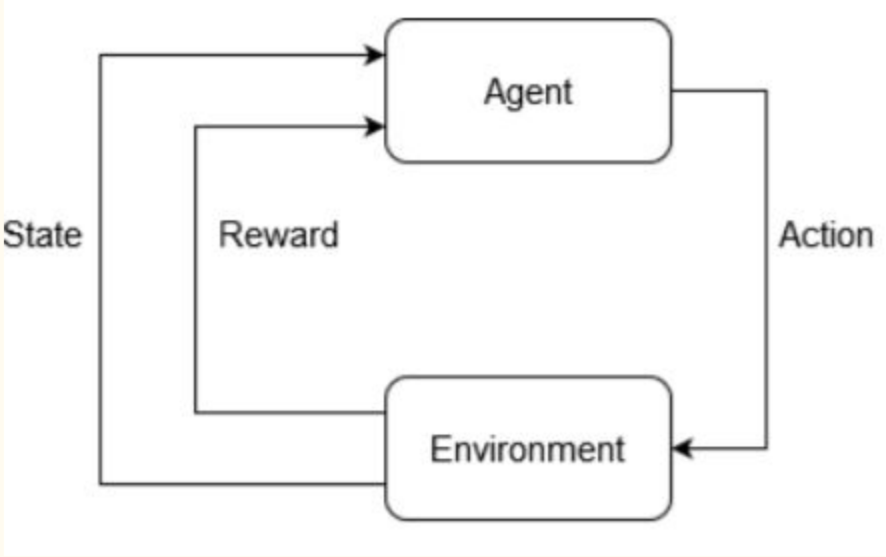
= Mean

= Median

Source: Demediuk (2019)

DDA and Reinforcement Learning

- Experimental turn-based fighting game
- Elements of a Markov decision process
    - Environment
    - Agent
    - State(s)
    - Actions
    - Rewards

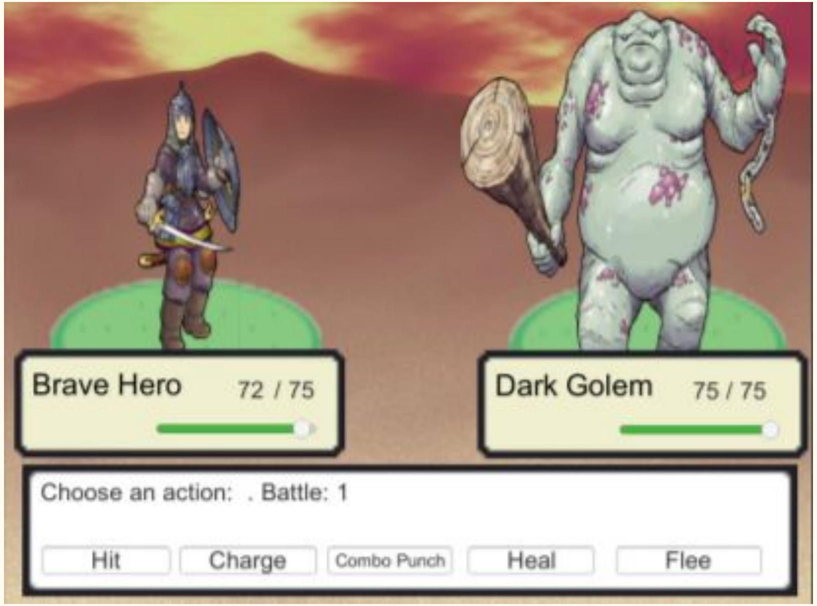


Source: Pagalyte (2020)

DDA and Reinforcement Learning

State	Player HP	Agent HP
0	75	==
1	51-74	==
2	31-50	==
3	16-30	==
4	1-15	==
5	<Agent HP	66-74
6	<Agent HP	46-65
7	<Agent HP	31-45
8	<Agent HP	16-30
9	<Agent HP	1-15
10	66-74	<Player HP
11	46-65	<Player HP
12	31-45	<Player HP
13	16-30	<Player HP
14	1-15	<Player HP
15	0	-
16	-	0

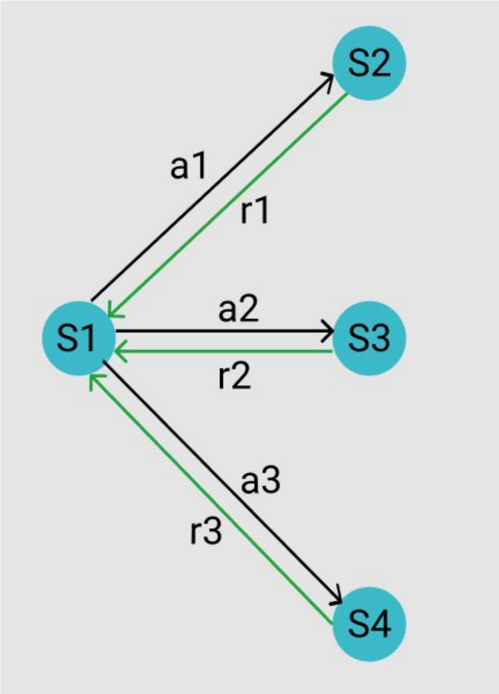
Source: Pagalyte (2020)



Source: Pagalyte (2020)

DDA and Reinforcement Learning

- Q-learning vs SARSA algorithm (state-action-reward-state-action)
- Exploration vs Exploitation
  - (50,50)
  - (30,70)
  - (70,30)
    - 70% exploring, 30% exploiting



*S1 = state 1
a1 = action 1
r1 = reward for action 1

DDA and Reinforcement Learning

- Versions of games experimented
  - Easy (no DDA)
  - Hard (no DDA)
  - 50-50
  - 30-70
  - 70-30
- A “balanced” version of the game: $1.5 \leq \mu(\text{average}) \leq 2.5$

Average Responses to Different Types of Games

	Success	Skill	Challenge	Effort
Easy	3	2.8	0	0.1
Hard	1.5	2.3	3.6	3.3
50-50	**2.5**	**2.5**	**2.2**	**2.3**
30-70	3.1	2.8	3	2.8
70-30	**2.2**	**2.5**	**2.5**	**2.5**

Source: Pagalyte (2020)

*Player feedback was based on the Likert scale where a 0 means “not at all” and a 4 means “extremely”

DDA and Multiplayer Video Games

- Trickier to implement DDA in multiplayer video games
- Multiplayer DDA (MDDA) “instance”
  - A gameplay feature in multiplayer games designed to reduce the difference in challenge experienced by all players

DDA and Multiplayer Video Games

- High performing perspective (HPP)
- Low performing perspective (LPP)

Components of an MDDA Instance

Component	Attributes
Determination	• Pre-gameplay • Gameplay
Automation	• Applied by system (automated) • Applied by player(s) (manual)
Recipient	• Individual • Team
Skill Dependency	• Skill dependent • Skill independent
User Action	• Action required • Action not required
Duration	• Single-use • Multi-use • Time-based
Visibility	• Visible to recipient • Visible to non-recipients • Not visible

Source: Baldwin (2016)

DDA and Multiplayer Video Games

- Player Control
  - HPP: More control to minimize DDA impact (single-use and skill dependent)
  - LPP: No particular preference
  - HPP and LPP: Automation
- Personal Benefit
  - What about the components made gameplay enjoyable?
  - LPP: Multi-use and skill independent
- Awareness of MDDA instances
  - Visibility of DDA
  - Players want to know when the game is being altered

Outline

- Background Information
  - Context of DDA
  - Flow Theory
  - Controversy of DDA usage
- Implementations of DDA and player feedback
  - Monte Carlo Tree Search (MCTS)
  - Reinforcement Learning
  - Multiplayer DDA
- Conclusion

Conclusion

- The research suggests favorable player feedback of DDA
  - More engaging than games without DDA
  - Can provide balanced gameplay
- When does DDA become a problem?
  - When DDA's presence is too strong/negates skill
- Future work

Acknowledgements

- I would like to thank Kristin Lamberty, Elena Machkasova, Nic McPhee, and Tim Snyder for their insight and wisdom over the research process. I would also like to thank my family and friends for their support and belief in my work

References

- Alexander Baldwin, Daniel Johnson, and Peta Wyeth. Crowd-Pleaser: Player Perspectives of Multiplayer Dynamic Difficulty Adjustment in Video Games. (October 2016)
- Elinga Pagalyte, Maurizio Mancini, and Laura Clement. Go with the Flow: Reinforcement Learning in Turn-based Battle Video Games. (October 2020)
- Simon Demediuk, Xiaodong Li, Marco Tamassia, and William Raffe. Challenging AI: Evaluating the Effect of MCTS-Driven Dynamic Difficulty Adjustment on Player Enjoyment. (January 2019)

Questions?