
BDD-BASED FRAMEWORK WITH RL INTEGRATION: AN APPROACH FOR VIDEOGAMES AUTOMATED TESTING

Vincent Mastain
Graduate Student, CY Tech
Cergy, France
vincentmastain@gmail.com

Fabio Petrillo
Departement de génie logiciel
Ecole de Technologie Supérieure, ÉTS
Montreal, QC
fabio.petrillo@etsmtl.ca

November 8, 2023

ABSTRACT

Testing plays a vital role in software development, but in the realm of video games, the process differs from traditional software development practices. Game developers typically rely on human testers who are provided with checklists to evaluate various elements. While major game developers already employ automated testing using script-based bots, the increasing complexity of video games is pushing the limits of scripted solutions, necessitating the adoption of more advanced testing strategies. To assist game studios in enhancing the quality of their games through automated testing, we propose the integration of Behavior Driven Development (BDD) with Reinforcement Learning (RL). This positional paper summarizes our proposal and framework under development.

1 Introduction

Testing is an important part of software development, today we count numerous different software testing methods like TDD (Test Driven Development) or BDD (Behavior Driven Development). Those methods are based on the test case writing, allowing automated testing a lot of different situations.

Behavior-Driven Development (BDD) is an agile software development methodology that facilitates effective communication among individuals with varying levels of technical expertise (developers, quality assurance experts, etc) [1, 2]. Indeed, BDD employs natural language to write the test cases, providing detailed descriptions of behaviors and the expected outcomes [1].

In the world of video games, the process is typically different. Game developers rely on human testers who are provided with a list of elements to evaluate [3, 4]. These testers then communicate their findings and results back to the developers.

Reinforcement learning is a machine learning technique in which an autonomous agent learns to make decisions based on its experience, with the goal to maximize the cumulative reward [5]. This concept has found applications in various domains, including the gaming industry. For example, OpenAI has developed an agent capable of playing Dota 2 a multiplayer online battle arena (MOBA) [6].

Automated testing is already employed by major game developers, employing script-based bots but as more cases arise, the scripted solution's capacity is already being exceeded, necessitating the use of more advanced testing strategies [7]. Presently, research efforts are underway, such as the work of *Gillberg et al.*, who utilized RL with autoplayer bots in their study [7], and *Gordillo et al.*, who explored playtesting coverage using RL [8]. Nevertheless, numerous video games still contend with a substantial number of issues upon their initial release [3].

2 Proposal

Considering the challenges in ensuring the quality of video games upon their initial launch, our proposal outlines an automated testing approach for video games combining the BDD methodology and RL. With this proposal, we establish the following hypotheses:

Hypothesis 1

The methodology based on BDD for video game testing allows for faster and higher-quality testing.

Hypothesis 2

The methodology has the capability to automate various types of tests that are traditionally conducted by humans.

Hypothesis 3

The methodology enhances communication between game developers and designers.

Goal

Improve the quality of the games through a modification of the game testing methodology.

To accomplish this, we intend to create a framework and assess it using a game engine, various games, and different reinforcement learning methods as PPO [9] or DQN [10].

The framework consists of two main components: the “Test side” and the “Train side”. These components are implemented as separate modules within the framework. The figure 1 illustrates the architecture of the framework.

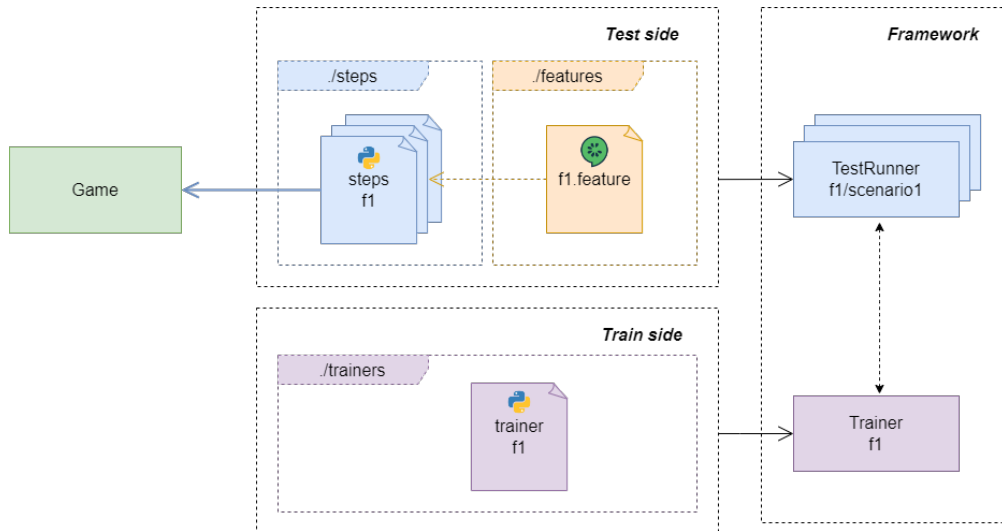


Figure 1: Framework architecture

By using this method, game designers will write the feature files in the Gherkin language, in our context, a scenario refers to a game situation where we perform the test. Therefore, the "when" keyword of the scenarios denotes a specific moment or a potential situation within the game as opposed to its typical usage where "when" symbolize an event performed.

The following code 2 and the figure 3 illustrate the usage of the Gherkin language adapted for video games testing. In this particular instance, we have employed the popular game "Flappy Bird" for demonstration. The objective of the game is to guide a bird safely through openings between pipes without allowing it to make contact with them. Therefore, an illustrative set of test scenarios would involve verifying the feasibility of the bird passing through various combinations of pipes. In this specific scenario, we are testing a limit case where the first pipe is at the lowest position and the second pipe is at the highest position.

```

1 Feature: Jumping
2   Scenario: Bottom and top
3     Given A game with a player
4     And A pipe generator
5     When The first pipe is at 0 %
6     And the next pipe is at 100 %
7     Then The player should have passed 2 pipes

```

Figure 2: Flappy bird test scenario

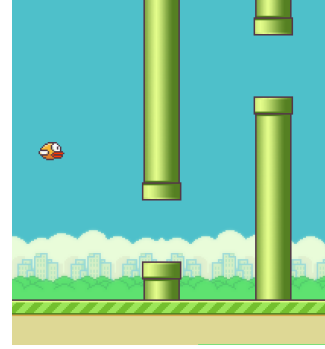


Figure 3: Flappy bird test illustration

As the usual BDD method the test developer will write the steps files corresponding to each scenario intended by the game designers. They will also write the trainer files which corresponds to the reinforcement learning implementation with a reward function and a feature extraction of the game's observation.

The framework has two different mode, the train mode which train a model based on the reward function and the game scenario and the test mode which use a trained model in order to pass assertions.

3 Conclusion

In this paper, we present our approach of Behavior Driver Game Development combined with Reinforcement Learning and our hypothesis.

In summary, combining the widely used BDD methodology with the growing capabilities of Reinforcement Learning holds great promise for improving the video game development process. This approach has the potential to reduce the workload of video game testers significantly. By leveraging the latest research in Reinforcement Learning [4, 7, 8], we aim to enhance game quality and minimize technical issues upon release, benefiting both developers and players.

References

- [1] Dan North. Introducing BDD, 9 2006.
- [2] Scott Bellware. Behavior-Driven development.
- [3] Cristiano Politowski, Fábio Petrillo, and Yann-Gaël Guéhéneuc. A Survey of Video Game Testing. *ACM International Conference on Automation of Software Test (AST)*, 5 2021.
- [4] Joakim Bergdahl, Camilo Gordillo, Konrad Tollmar, and Linus Gisslén. Augmenting automated game testing with deep reinforcement learning. *2020 IEEE Conference on Games (CoG)*, 8 2020.
- [5] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement Learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 5 1996.
- [6] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Jozefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, M. N. Petrov, Henrique Ponde De Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv (Cornell University)*, 1 2019.
- [7] Jonas Gillberg, Joakim Bergdahl, Alessandro Sestini, Andrew Eakins, and Linus Gisslén. Technical challenges of deploying reinforcement learning agents for game testing in AAA games. *arXiv (Cornell University)*, 7 2023.

- [8] Camilo Gordillo, Joakim Bergdahl, Konrad Tollmar, and Linus Gisslén. Improving playtesting coverage via curiosity driven reinforcement learning agents. *2021 IEEE Conference on Games (CoG)*, 8 2021.
- [9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy optimization Algorithms. *arXiv (Cornell University)*, 7 2017.
- [10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv (Cornell University)*, 12 2013.