# 基于分层强化学习算法(HRL)的游戏内bossAI生成

## 背景：

目前强化学习(Reinforce Learning,RL)已经涉及到众多游戏领域，在PVE游戏，比如Atari2600中，Mnih(2015,2016)和Hessel(2018)通过实际测验发现强化学习(RL)算法在其中绝大部分游戏的表现远超于人类玩家。而在传统PVP棋类游戏上，比如在围棋领域，由Deepmind团队(2016)研发使用MCTS算法的AlphaGO围棋AI成功战胜了人类围棋最强棋手，证明了AI可以在高复杂度和高维度的游戏环境。同时在战略游戏方面，OpenAI(2018)和Vinyals(2017)在Dota2和StarCraft2上也训练出足够优秀的AI。

## 先行研究：

关于游戏内AI的设计目前以强化学习(Reinforce learning,RL)为主。然而，目前关于使用RL的主流方向都在于如何设计一个强大的AI用来击败强大的人类玩家(比如职业玩家)。比如在近似我研究方向的格斗游戏中，Ishihara等人(2018)成功在2D的FightingICE(FICE)平台上应用MCTS算法。同时Tang等人(2020)也提出了一种新的改进后的RHEA算法在FICE平台上获得了很好的表现。而在3D领域，Inseok等人(2022)成功在3D格斗游戏Blade&Soul使用Actor-Critic算法制作了一个1V1人工智能代理，并在与职业选手对抗中达到了66%的胜率。我认为这是最接近我们研究方向的研究成果。然而其仍然存在众多问题，比如其只是一个强大的AI而并非一个优秀的游戏AI。同时3D游戏环境中的奖励十分稀疏，所以很容易出现延迟奖励(Reward delay)问题，并使算法陷入局部最优解。因此我希望引入分层强化学习(Hierarchical Reinforcement Learning,HRL)来解决该问题。

HRL最早起源Peter等人(1993)提出的Fedual算法，随着DRL的发展，我们实现了更优秀的HRL网络，例如Bacon等(2016)基于Actor-Critic算法结合Option理念的Option-Critic算法以及Bacon(2018)等对Option-Critic算法改进后的A2OC算法，其在吃豆人游戏中表现远超于普通强化学习网络。同时HRL算法在复杂游戏AI方面也有优秀的成果，比如在高复杂高维度的MineCraft游戏中由Chen(2016)提出的H-DRL终身学习模型。以及Tejas(2016)使用分层DQN算法(H-DQN)在DQN算法得分为0的Atari-Montezuma's Revenge游戏中获得了极好的表现。这些都充分证明了HRL有潜力可以处理存在稀疏奖励，高动作空间的3D游戏环境。

## 研究目标：

我希望实现如下的游戏BossAI，其可以达到。

1. 能很好的对抗人类玩家，并且有着多样的攻击方式。
2. 能尽可能像人类玩家一样和player进行交互，使Boss更人性化。(Human-like action)

想要实现目标1，其目标可以分成以下小目标：

* 1. 如何处理State，Action和复杂的稀疏奖励。
  2. 使得boss的存在多种模式(Selectable mode)。
  3. 如何训练模型。

想要实现目标二同样可以分为以下几个目标

* 1. 如何实现agent的类人行为。
  2. 如何衡量agent的类人行为。

我将逐条解释我实现的方法。

## 研究方法：

1.

首先，Niels(2019)提出一下观点：如果想要创造一个存在类人行为的agent(在本研究中指的是像人类玩家的bossAI)，我们首先需要训练一个能很好玩游戏的agent(即能力足够高)，然后寻找某些方法来使其更近似人类风格。因此我们的目标1可以看作是训练一个水平较高的agent。

关于模型方面，目前存在着众多的HRL模型，每一个模型都有着其独特的优点，所以在本文中我只讲解一下如何构建AI框架，在该文章中我将使用Bacon(2016)创造的Option-Critic模型来作为样板模型解释我的框架内容。

**Answer 1.1.:**

3D游戏环境中我们不仅仅需要考虑agent的技能(skill)方面，同时我们还要考虑agent的移动和与target之间的距离，因为我们需要根据距离来判断使用何种攻击最具有效率。因此我借鉴了Inseok(2022)的文章中关于导入LSTM网络来处理state和action的部分，关于state，我们每0.1秒便更新一次state，其是基于之前所有的观测，并且我们的具体来说包括了所有的人类玩家能观测到的信息，比如，双方的HP，MP，对手的技能动作，player距离墙壁的距离，所有技能的冷却时间，所有技能的伤害等等。同时关于我们的action，我们的agent会在每0.1秒决定出相应的动作，并且包含了两个方面，。

同时，因为我们没有办法实现像人类玩家那样在某一时间段内(比如1s内)做出多次移动行为，所以我们使用维持移动策略(Maintaining decisions)，即在选定的时间内(比如1s内)使我们的agent只进行一个动作，比如1s内一直向左走或者向右走。这样可以大幅度减少我们的计算量。

关于处理复杂的稀疏奖励，首先我认为3D游戏的环境是符合半马尔可夫链(Semi-MDPs)的，agent在该游戏环境中，可以在一段时间内执行一套复杂的攻击技能，或者是在特定的时间点在达到某个要求后执行特定的技能，这些都可以借由HRL算法来解决。不过HRL也存在着一些问题，以Option-Critic算法为例，我们需要人工设置Option的数量，所以在此我们不得不借助相关的游戏专家来确定哪个Option有用，同时我们也需要大量试错去验证Option的内部参数。

**Answer：1.2.&1.3.**

1.2.和1.3.的问题我们可以同时解决，首先是如何使boss存在多种模式。我目前的方法是，我们训练三种不同的agent，其分别为：Attacter，Defender，Balancer。我们当然可以由AI自主根据情况选择不同的模式，此时该问题将会变成一个多智能体强化学习(Muti-agent Reinforcement learning,MARL)问题，问题的复杂度会直线上升，同时可能不得不需要一些新的算法来处理，此处可以作为以后的研究方向。而在本计划书中，我们只探讨人为选择agent的情况。Attacter是一个极具攻击型的agent，我们可以在他的State中多添加一个时间惩罚，目的是使该Agent尽可能快的结束战斗。Defender则是一个防守型agent，该agent的特点是希望尽可能与target保持一定距离，通过远程消耗来击败敌人。Balancer则是二者的结合。

训练方法我们采用Deepmind(2016)训练AlphaGO的自对弈训练方法(Self-play)方法进行训练，同时我们借鉴了Inseok(2022)中的训练方法，该方法分为以下几个步骤：

1. 我们对数据进行分类，将数据分为Attact，Defend，Balance三种类型数据，然后我们用这三种类型的数据训练相对应的AI。
2. 同时我们在多训练一个agent，我们称其为opponent，该agent是由这三种数据组合训练出来的。
3. 我们让这些agents随机对抗，然后通过胜率改变自己的策略。

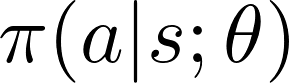
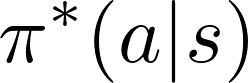
这种方法可以使不同风格的agent在面对到不同风格的对手时如何改变自己的策略，并学习如何应对他们。通过这种方法，我们成功达成了1.2.和1.3.的目标。

**2.**

类人代理(human-like agent)是目前AI游戏领域的一大难关。即使到现在也并没有一个足够好的解决方案来使得agent具有类人行为(human-like agent)。不过目前已经有很多出色的论文开始探讨这个方面，同时我也会基于我的理解来提出解决的方案。

**Answer：2.1.**

根据Niels(2019)的观点，作者认为比起强制降低agent的能力，更有用的办法是以更加精细的方式调整agent的性能水平(performance level)。例如，分别控制代理的反应速度(reaction-speed)和长期规划能力(long-term planning ability)。或者是限制agent的某些能力。在这里一种比较可行的方法是Liapis(2014)等人使用的procedural personal方法，即将agent的偏好编码为一种有效权重，以此来增加agent的多样化。该论文作者认为这种方法制作的智能体模型在心理学方面可以有效的模拟人类玩家的决策。

在我看来，逆强化学习(Inverse Reinforcement Learning, IRL)可能会是另一种解决方案。该方法由Ng(2000)提出，其思想是，将人类专家的行为近似于一种黑盒(black box)，我们的目的是学到一个策略网络，使其近似于人类专家的策略。这种方法可以很好的使agent的行为近似于人类专家。并且我认为，训练这样模型一定要先训练一个超人AI(Super-human AI)，然后再对这个模型进行逆强化学习(IRL)才能达到一个良好的效果。

**Answer：2.2.**

根据Yunqi(2020)的论文中表示，评价一个agent是否存在类人行为(human-like action)的标准为两种，一种为技能(skill)，另一种为风格(style)。这是评判是否为类人智能体(human-like agent)的重要标准。因此我们可以通过这两种基准来判断智能体是否已经达到类人标准。

1. X. ：关于Offline RL和Online RL

在我看来，在游戏领域，我们可以实现无限次与环境交互，所以我们可以很简单的就得到大量的数据。因此我认为没有必要去可以使用Offline RL。我们完全可以使用Online RL达成我们的目标。

## 意义：

关于创造该AI意义，我认为可以分成两点来说明。

1. 如果能创造一个多样化的，有着随机动作且更人性化的游戏AI，这对电子游戏产业将会是一次巨大的革新。游戏AI的存在可以增强玩家的游戏体验，增强游戏的挑战性，丰富游戏内容。据我所知，日本的Sony，美国的EA，中国的Mihoyo等游戏公司都在进行着游戏AI的研发。一旦研发成功，将会给游戏公司带来巨大的经济效益。
2. 通过游戏我们可以验证我们的AI技术和AI算法，然后将该AI推广到其他领域。比如无人机技术，机器人技术等。这样有助于人工智能技术的发展。

**Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., *et al*. 2015. Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529.**

**Mnih, V., Badia, A. P., Mirza, M., Graves, A., Harley, T., Lillicrap, T. P., *et al*. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48:*1928- 1937.**

**Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, *et al*. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.**

**David Silver、Aja Huang，*et al，2016.*Mastering the game of Go with deep neural networks and tree search. In** *Nature volume 529*:**484–489.**

**OpenAI. 2018. OpenAI five, <https://blog.openai.com/openai-five>.**

**Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., *et al*. 2017. Starcraft II: A new challenge for reinforcement learning. In *arXiv preprint arXiv:1708.04782***

**Ishihara, M., Ito, S., Ishii, R., Harada, T., and Thawonmas, R. 2018. Monte-Carlo Tree Search for Implementation of Dynamic Difficulty Adjustment Fighting Game AIs Having Believable Behaviors. In *2018 IEEE Conference on Computational Intelligence and Games:* 1-8.**

**Zhentao Tang, Yuanheng Zhu, Dongbin Zhao,Simon M. Lucas. 2020. Enhanced Rolling Horizon Evolution Algorithm With Opponent Model Learning: Results for the Fighting Game AI Competition**. **In** *IEEE Transactions on Games ( Volume: 15, Issue: 1, March 2023)*:**5 - 15.**

**Inseok Oh, Seungeun Rho, Sangbin Moon, Seongho Son, Hyoil Lee, Jinyun Chung.2022. Creating Pro-Level AI for a Real-Time Fighting Game Using Deep Reinforcement Learning. In** *IEEE Transactions on Games ( Volume: 14, Issue: 2, June 2022)*:**212 - 220.**

**Peter Dayan, Geoffrey E. Hinton. 1993. Feudal Reinforcement Learning. In** *Advances in Neural Information Processing Systems 5*.

**Pierre-Luc Bacon, Jean Harb, Doina Precup. 2016. The Option-Critic Architecture. In** *Accepted to the Thirthy-first AAAI Conference On Artificial Intelligence (AAAI).*

**Jean Harb, Pierre-Luc Bacon, Martin Klissarov, Doina Precup. 2018. When Waiting is not an Option : Learning Options with a Deliberation Cost. In** *Accepted to the Thirthy-first AAAI Conference On Artificial Intelligence (AAAI).*

**Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J. Mankowitz, Shie Mannor. 2016.In** *AAAI Publications, Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).*

**Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, Joshua B. Tenenbaum. 2016. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In *arXiv preprint arXiv:1604.06057.***

**Niels Justesen, Philip Bontrager, Julian Togelius, Sebastian Risi. 2019. Deep Learning for Video Game Playing. In** *IEEE Transactions on Games ( Volume: 12, Issue: 1, March 2020):***1 - 20.**

**David Silver, Aja Huang, Chris J. Maddison, Arthur Guez,** *et al.* **2016*.* Mastering the game of Go with deep neural networks and tree search. In** *Nature volume 529:***484–489.**

**C. Holmg˚ard, A. Liapis, J. Togelius, and G. N. Yannakakis. 2014. Generative agents for player decision modeling in games. In** *FDG***.**

**Ng, A. Y., & Russell, S. 2000. Algorithms for inverse reinforcement learning. In** *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000).*

**Yunqi Zhao, Igor Borovikov, Fernando de Mesentier Silva,** *et al.* **2020.** Winning Is Not Everything: Enhancing Game Development With Intelligent Agents. In *IEEE Transactions on Games:*1-1.