Project Autonomous and Adaptive Systems 2021-22

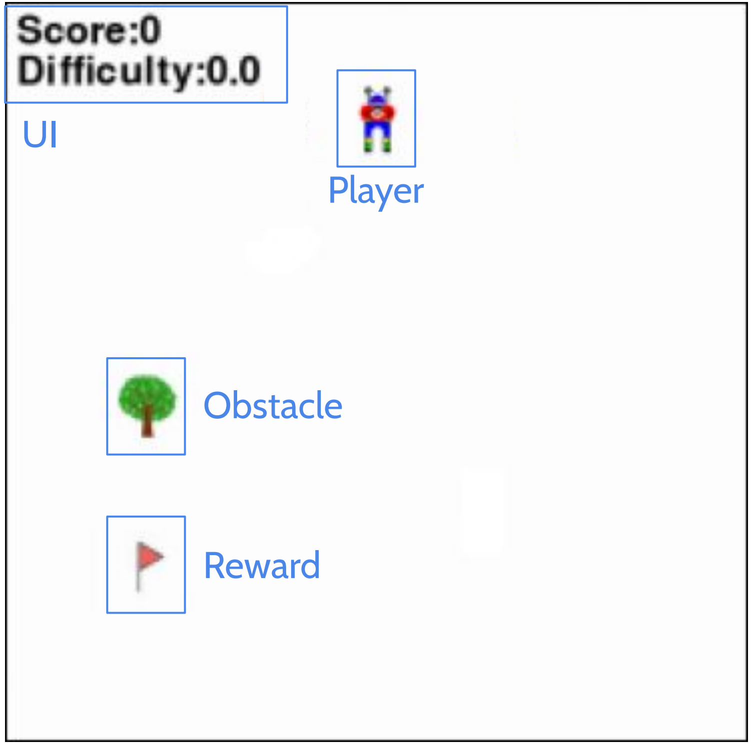
giovanni.minelli2@studio.unibo.it

Giovanni Minelli

Project objective: create a game environment parameterised by a difficulty value to enable the training of better autonomous player agents

[CoG 21: Adversarial Reinforcement Learning for Procedural Content Generation]

Project objective: create a game environment parameterised by a difficulty value to enable the training of better autonomous player agents



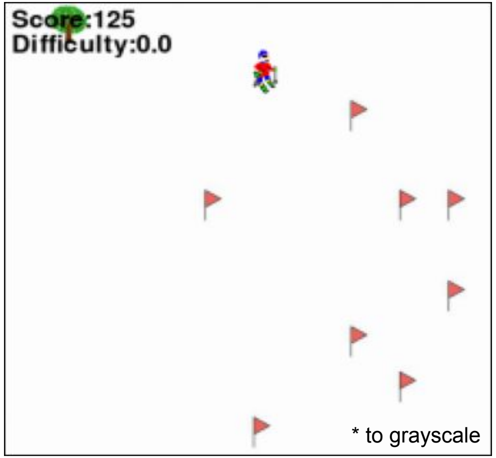
- Interface with game score and difficulty
- Vertical scrolling environment
- 5 degrees of action for the player
- -20 score for touching a tree
- +10 score for touching a flag
- Maps generated on the fly

Two agents:

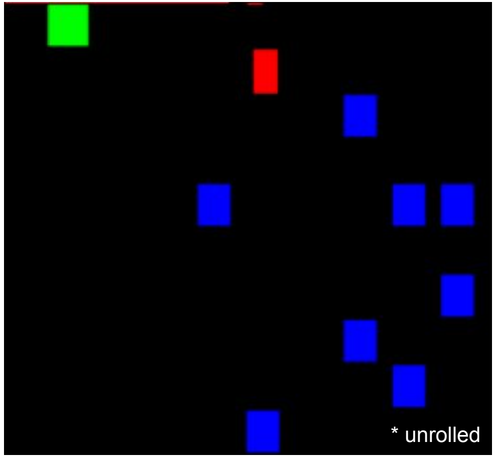
Solver: predict the player's action at every step
Generator: generates a new map of obstacles on demand

Env observation

View resized, converted to grayscale,
frame skipping 8 (max 6px), frame stacking 4



CNN: 224x224x4



MLP: 64*64*3

Actor Generator

Layer (type)	Kernel	Output Shape	Param #
input		224x224x1	0
input_aux		**1x1**	**0**
input_positions		**1x100**	**0**
conv2d	8x8 s 4	55x55x32	2080
conv2d_1	4x4 s 2	26x26x64	32832
conv2d_2	3x3 s 1	24x24x64	36928
flatten+**concat**		36864+**1**	0
dense		1x512	18875392
prediction		1x100	51300
filter_pred (Mult)		**1x100**	**0**

Actor Solver

Layer (type)	Kernel	Output Shape	Param #
input		224x224x4	0
conv2d	8x8 s 4	55x55x32	8224
conv2d_1	4x4 s 2	26x26x64	32832
conv2d_2	3x3 s 1	24x24x64	36928
flatten		36864	0
dense		1x512	18874880
prediction		1x5	2565

Networks

4 CNN networks for the estimation of a policy and a value function (solver and generator agent)

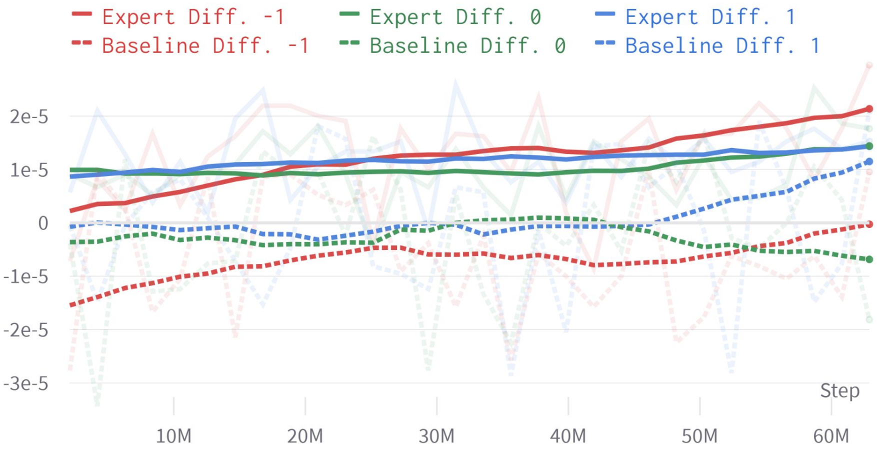
PPO Loss

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right], \quad (9)$$

where c_1, c_2 are coefficients, and S denotes an entropy bonus, and L_t^{VF} is a squared-error loss $(A_t - V_\theta(s_t))^2$

Proximal Policy Optimization Algorithms

Results



30 game episodes of baseline and expert with generator

Map	Difficulty	Game score		Total objects hit	
		Baseline	Expert	Baseline	Expert
Random map		697.5	1032.5	2792	2693
	1 (easy)	-37.5	642.5	2718	2293
Generator agent	0 (medium)	-87.5	528.3	2662	2196
	-1 (hard)	-290.8	554.2	2862	2294

Play Demo

Thank you

A2C PPO framework

PPO is a family of first-order methods (unlike TRPO) that use a few tricks to keep new policies close to old.

- PPO is a policy gradient method on-policy belongs to the family of Actor-Critic algorithms
- PPO can be used for environments with either discrete or continuous action spaces.
- and as well with environments episodic or sequential dependently by the advantage function

Loss in three terms:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right], \quad (9)$$

where c_1, c_2 are coefficients, and S denotes an entropy bonus, and L_t^{VF} is a squared-error loss $(A_t - V_\theta(s_t))^2$

Proximal Policy Optimization Algorithms

$$L^{CLIP}(\theta) = min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)$$

Advantage (note it's almost the same error of critic network)

$$A_t = r_{t+1} + V(s_{t+1}) + V(s_t)$$

Ratio for policy

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$