1. Levels and goals

IV. Evidenced outcomes

Giacomo Bergami

September 10, 2021

Newcaslte University

Objectives

- Provide real use-case contexts in videogames exploiting some of the previously suggested technologies.

Goal Oriented Action Planning

The Sims 4 (2014)

The Sims 4 (2014)

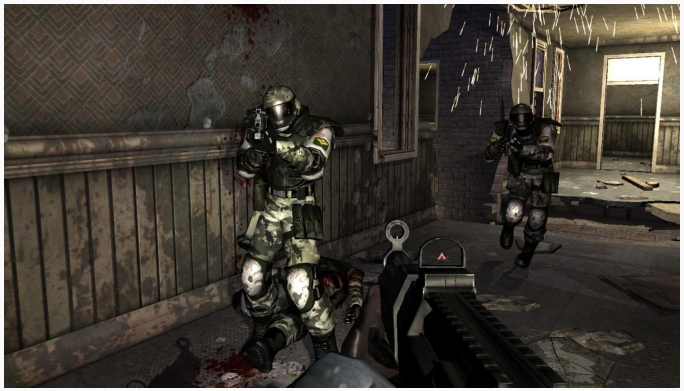
- Characters' behaviour is influenced by *character values*.
- The characters' well-being and their interaction is heavily influenced by those.
- Specific actions (e.g., *studying*) might also influence the character values (e.g., *responsibility*).

$\Rightarrow$ The rewards $\mathcal{R}$ are multi-objective functions, where the weights associated to each value might change through time (see the rightmost map).

$\Rightarrow$ In the game, there is no explicit goal.

$\Rightarrow$ Still, state values $\mathcal{V}$ might be pre-trained, and used for giving the player some suggestiongs throughout the game (training objective: health and well-being).

F.E.A.R. (2005)



- It exploits "logical" *Automated Planning*, where Horn clauses $H \overset{act}{\Rightarrow} T$ are exploited to represent preconditions (H) and effects (T) of a given action act.
- AI might have multiple goals (e.g., *Patrol* and *KillEnemy*).

- To reach the goal, the AI might exploit one of the 120 actions available (not at the same time and for all type of characters).
- The combination of all the possible states is reduced by updating state values $\mathcal{V}$ with the situation within the perceived environment (decoupling states from actions).
- Contrariwise to Reinforcement Learning, where we get the best set of actions for all the possible state, the compressed nature of the states requires some re-planning (*ReplanRequired*) when the actions of one user invalidate the plan.

Procedural Generation

Darkest Dungeon (2016)

- Dungeons' layouts are "randomly" generated,
- as well as both the fight and curio shop placements in them.
- There might be more sophisticated rules related to difficulty:
  1. You must have at least min fights (per XP).
  2. You can't have more than max fights (per XP).
  3. τ long quests have *Secret Rooms*.

No Man's Sky (2016)

- In their approach, they exploited **fractal** generation via a mathematical formula to generate worlds.
- It contains $> 1.8 \times 10^{19}$ planets, so they could not be all implemented!
- It comes with complete solar systems, varied weather systems, flora and fauna:
  1. creatures are aware of specific objects,
  2. might have preferences/friends,
  3. and emergent behavior emerges from simple actions.
- Night/day cycles as well as the sky that you percieve are dictated by game physics.
- They exploit seeds to make generation repeatable.

Reinforcement Learning

Civilization IV & Dota 2

Dota 2 (2013)

Civilization VI (2016-2018)

- **Dota 2**: The AI played 10,000 years of games against itself, then used this knowledge to defeat its opponents.
- **Civilization**: The mod *Smoother Difficulty 2.0* the game could make predictions about the player's play-style and then learn to counter accordingly.

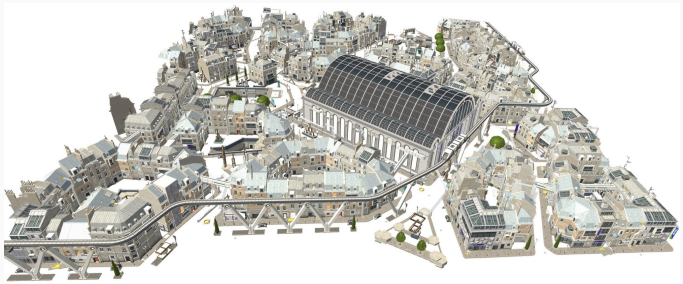
These technologies exploited (deep) learning, but the lack of modelling required to actually learn from actual players' moves.

Ubisoft: Reinforcement Learning for Car Driving

A screenshot from the demo.

- In a recent paper by Ubisoft, the company showed how reinforcement learning can be used to car racing.
- The objective of the car is to *follow a given path as fast as possible*.
- Actions are modelled as follows:
  1. two continuous actions (acceleration and steering) and
  2. one binary discrete action (hand brake).
- Their model is slightly more complicated, as it also requires a neural network. Still, the mathematical pre-requirements to understand the concepts are introduced in the previous lecture.

Ubisoft: RL for Navigation in AAA Video Games



A map from "Hyper Scape" (2020).

- This scenario is more complicated, as the agent has no hard-coded path to follow.
- Similarly to our previous examples, an agent wants to reach a goal by moving within a real-world environment.
- Actions are modelled as follows:
  1. Forward/Backward
  2. Left/Right
  3. Rotate
  4. Jump and Double jump
- Even in this scenario, Reinforcement Learning is applied to Deep Learning (mainly to summarize the whole set of states).