

STARPLANNER

Technologies:
JAVA
STARCRAFT
BYAPI
JBIDGE

Supervisor:
PETROS
KEFALAS

Developer:
PANAGIOTIS
PEIKIDIS

AIM
 To demonstrate that planning meets the demands of modern video games.

Objectives

- Understand planning
- Understand Game AI
- Discover Game AI demands
- Create & integrate a planner
- Evaluate planner

StarPlanner v0.8

StarPlanner v0.8

Run StarCraft

Test StarPlanner

StarPlanner Manager

Use StarPlanner Enable Logging

Created plan: BuildBarrack,TrainMarine,BuildEngineeringBay
 PlanPlanner: Activating plan with 3 steps:BuildBarrack,TrainMarine
 PlanPlanner: Activating action: BuildBarrack.
 Adding supply depot
 (1,null)PlanPlanner: Activating action: TrainMarine
 PlanPlanner: Activating action: BuildEngineeringBay
 Created plan: BuildBarrack
 PlanPlanner: Activating plan with 1 steps:BuildBarrack
 PlanPlanner: Activating action: BuildBarrack.
 Created plan: BuildBarrack
 PlanPlanner: Activating plan with 1 steps:BuildBarrack
 PlanPlanner: Activating action: BuildBarrack.

Auto Scroll Debug Level: Game Speed 100

BuildOrder Manager

Goals

Terran BuildFirstWave
 Terran GoalTest

Actions

BuildBarrack
 TrainMarine
 BuildEngineeringBay

Terran BuildFirstWave

Generate

Strategic Manager

Goals

Actions

Generate

For my final year project I created a bot that plays StarCraft: Brood War named StarPlanner. The aim was to demonstrate that planning meets the demands of modern video games. To do so, I researched the area of Automated Planning and Game Artificial Intelligence to learn how to build a planning system and find out what are the demands of modern video games. Using this information I developed a planning system and integrated it to StarCraft as a player. For those of you who don't know, planning is an artificial intelligence technique that, given a set of actions and a goal, a plan is generated. A plan is a sequence of actions that, when followed, reaches the goal from the current state. The bot uses planning in two levels: the strategic and the construction level. For example, a strategic goal could be to build a second base. The plan generated could include a step that requires the construction of some units. This is fed to the construction planner as a goal in order to generate another plan that will construct the units needed. All steps generated by the planners are then executed by lower-level managers. Next, I evaluated the bot according to the

demands found during research and concluded that indeed, planning meets them. The demands of modern video games are: 1) it... must work... with the limited resources provided 2) it must be extensible (for future projects), 3) it must be flexible (to provide additional functionality) and, finally 4) it must provide both autonomy and the means to control this autonomy whenever necessary. The planning system was built separately and StarPlanner extends it into a working StarCraft bot. Flexibility is possible by the very nature of planning. Adding more functionality is as simple as adding more actions. Controlling the bot's behavior is done from a graphical user interface by enabling or disabling goals or actions. Finally, the user can also run the planner without StarCraft to generate a plan (for testing purposes). Thus, all functionalities of StarPlanner demonstrate that indeed, planning meets the demands of modern video games.