Final Project: Asteroid Planning

Robot Intelligence: Planning 4649/7649
Group 2: Kyle Volle, Luke Tornquist, Steffan Slater, Xinyan Yan

***Abstract*—In this paper, planning and machine learning concepts are applied to the design of an autonomous agent to play an ”asteroids” type computer game.**

I. Introduction

The ”asteroids” genre of games poses several interesting planning problems. In this game, the player controls a spaceship in an environment that is filled with fast moving obstacles called asteroids. Collision with an asteroid causes the player to lose a life. Points are scored by shooting the asteroids and breaking them into two smaller, faster obstacles. If an obstacle is shot twice it is destroyed. As the game goes on more and more obstacles are spawned. The goal for the player then is to stay alive as long as possible to keep shooting asteroids.

This game is interesting for several reasons. For starters there is no final goal, but rather the planner seeks to stay alive for as long as possible. This differentiates this problem from more traditional motion planning problems. Additionally, in this game there are no static states. Conservation of momentum keeps the ship moving at a constant velocity if there is no player input. Combined with the dynamic nature of the obstacles, the game state can change rapidly requiring frequent replanning.

The planner utilizes several processes in order to maximize the potential future score. An array of safe trajectories is first calculated and from this list another process selects one of these trajectories based on the anticipated rewards and costs. Finally, a third process determines the game inputs corresponding to following this trajectory and places these inputs in an executor queue to be performed as the replanning occurs. Factors for selecting a trajectory include, but are not limited to, time required to achieve that trajectory, density of asteroids along the trajectory, distance to obstacles along the trajectory and the ability to destroy asteroids along the way. The Methods section will go into greater detail about the techniques and strategies employed.

II. Related Work

III. Methods

A. Game and Planner Interface

B. Navigation Planning

C. Machine Learning

IV. Experiments

V. Analysis

VI. Discussion

References

[1] Mike Stilman. Task constrained motion planning in robot joint space. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3074–3081. IEEE, 2007.

[2] Tobias Kunz and Mike Stilman. Manipulation planning with soft task constraints. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1937–1942. IEEE, 2012.