Group 6 Term Project III: Milestone Report #2

Tiny Islands (2019) https://dr-d-king.itch.io/tiny-islands

Github Repository: https://github.com/xuhongkang/tinyislands

Alex Ma, Andrew Sayegh, Diego Valdivia Cox, Hongkang Xu

Current Progress:

- Re-implemented game model with all scoring and moving features, including a game state representation, board representation, tile representation and tile type representation: https://github.com/xuhongkang/tinyislands/blob/bb11528f8d1d20be01dbd01c5b1227f73e7ffca1/codebase/minisland/model.py

- ^ Abstracted tile's entire scoring logic under current game rules in our code, supporting the potential addition of all complex tile types from the original game.

- ^ Designed the model to be fully scalable in terms of board size, turn limits and number of choices to choose from each turn.

- Decided as a group to implement breadth first search under a predetermined seed to find the optimal policy and absolute highest score. This feature will come in handy when we need to gauge the optimality of our reinforcement algorithms for milestone 3. The main motivation for reinforcement algorithms is to optimize agent gameplay without prior knowledge of a seed's choice tile distribution. https://github.com/xuhongkang/tinyislands/blob/bb11528f8d1d20be01dbd01c5b1227f73e7ffca1/codebase/minisland/search.py

- ^ Implemented basic pruning based on two invariants: successor state scores cannot be negative or smaller than that of its previous state. Via testing, we found that this method saved up to 45% execution time under current game configurations (5x5 board, 10 turns, 2 choices).

Proposed Changes:

https://github.com/xuhongkang/tinyislands/blob/bb11528f8d1d20be01dbd01c5b1227f73e7ffca1/codebase/minisland/specifications.md

- Smaller Board, 5x5 ~~or 7x7~~
- Less Number of Turns - 10 for 5x5 ~~or 16 for 7x7~~
- Simple Terrain Generator given a fixed number of randomized choices (limited positions (1 for each), two terrain types from which to choose)
- Change square tiles to hexagon tiles, increase the amount of adjacent and corner tiles.
- Create a relationship between scoring and number of tile sides?
- Abstract Tile (types of terrain tiles)

Proposed New Tile Types and Associated Scoring Logic:

1. Type A
    a. Gives +2 score if there's not any other Type A sharing its row and column
2. Type B
    a. Gives +1 score for each Type B ~~adjacent to~~ **near** it
3. Type C
    a. Gives +1 score for each Type A or Type B near it
    b. Gives -2 score for each Type C ~~adjacent/near~~ **adjacent to** it

Explanation: After careful consideration, we believe that the above 3 types of tile design best capture the many features of terrain tiles in Tiny Islands, and will be the candidates for the tiles in our simplified model. This simplified model, whose mechanics will be introduced below, will become the environment in which our agent will be trained for Milestone 2.

Proposed New Rules and Mechanics:

1. The **Board** will consist of 25 (initially empty) **Tiles** ordered in a 5x5 grid.
2. A **Tile** is a unit located on the grid with a unique 2d **Position**. Tiles are initially empty, but may be assigned specific **Types**. If a Tile has already been assigned a Type, it cannot be reassigned. (This feature might conflict with MDP principles since the Seed will have

to offer alternative Choices if the tile at the initial target position has already been assigned. This rule might be relaxed further in the future.)

3. **Position** is a unique identifier for Tiles denoting its 2D position on the **Board**. For a 5x5 Board, Positions may range from (0, 0) to (4, 4) where (x, y) represents the Tile in column x on row y on the Board.
4. A **Type** is a specific typing for a Tile that represents a unique piece of scoring logic. Scoring logic computes the individual score of a Tile, based on information about adjacent (directly connected in at most 4 directions) or near (directly connected in at most 4 directions) Tile to the reference Tile. Tiles with different Types score differently individually.
5. At the start of the **Turn**, the agent chooses from a **Choice**. A **Choice** consists of two sets of Tile **Type**-**Position** pairs. By selecting one of the sets, the state progresses to the next state by changing the **Type** of the **Tile** at the paired **Position** to its target **Type**.
6. A **Game** of ~~*Tiny*~~ *Mini Islands* consists of 10 turns. The final score is calculated by adding up all the individual scores of **Tiles** based on their **Type**.

Proposed Searching and Optimization Algorithms:

We aim to complete the objectives using the below searching and optimization algorithms:

1. Breadth First Search to generate full search space and evaluate performance (Completed).
2. Simple Pruning via devaluating negative rewards to optimize performance (Completed).
3. Depth Limited Search via our built in evaluation function for testing purposes (Will implement after completed task 4 and considering scaled models).
4. Prospects for model based reinforcement learning with scaled up models.
5. Training a separate island drawing agent using sample test data via supervised learning

For Milestone 3 we aim to work on tasks 4 and 5, complete the poster and present our work in a video.