

Prakhar Saxena

Professor Jeffrey Popyack

CS 481 – 001

May 14, 2020

Honours Project: Dice Wars

Risk is a strategy board game, extremely popular among families. The motive of the game is to conquest the entire political map on the board.

How is this related to this Honours project?

Dice Wars

Dice Wars, the game that I have implemented, is very much alike risk, but it's a much shorter game, although that can change very quickly if you have numerous players.

Each region contains a stack of 6-sided dice which equate to relative power of the player. In this implementation, each player starts with just 1 die in their regions.

On a player's turn, the player has the choice to challenge any other player, who has a stack of dice in a neighbouring region. This challenge is resolved when each player rolls their dice, and the player with the higher total wins, that round. If there is a tie, then it is settled in the favour of the defender, not the attacker. The objective of the game, like Risk, is total global domination, or in this case total board domination.

Implementation

My initial thoughts were to create a superficial CUI on terminal, but I was not convinced that it was the way to go, because all of the web examples that I saw online had many elements of colours and animation in them.

I did not implement any animations, except for changing of the colours in the map. There only a few basic actions I let a user make: Left-click, Right-click and Esc. The start up screen does give you the quick instructions as to what these actions are and do. Like left click to select and right click to pass.

System Requirements

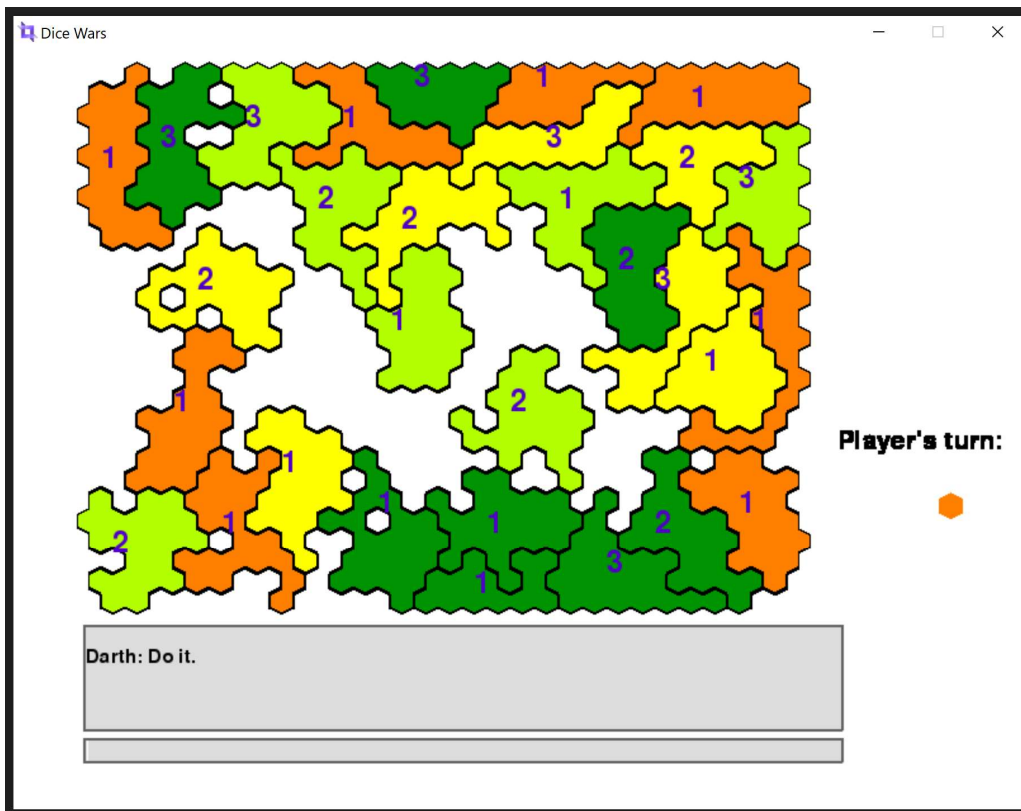
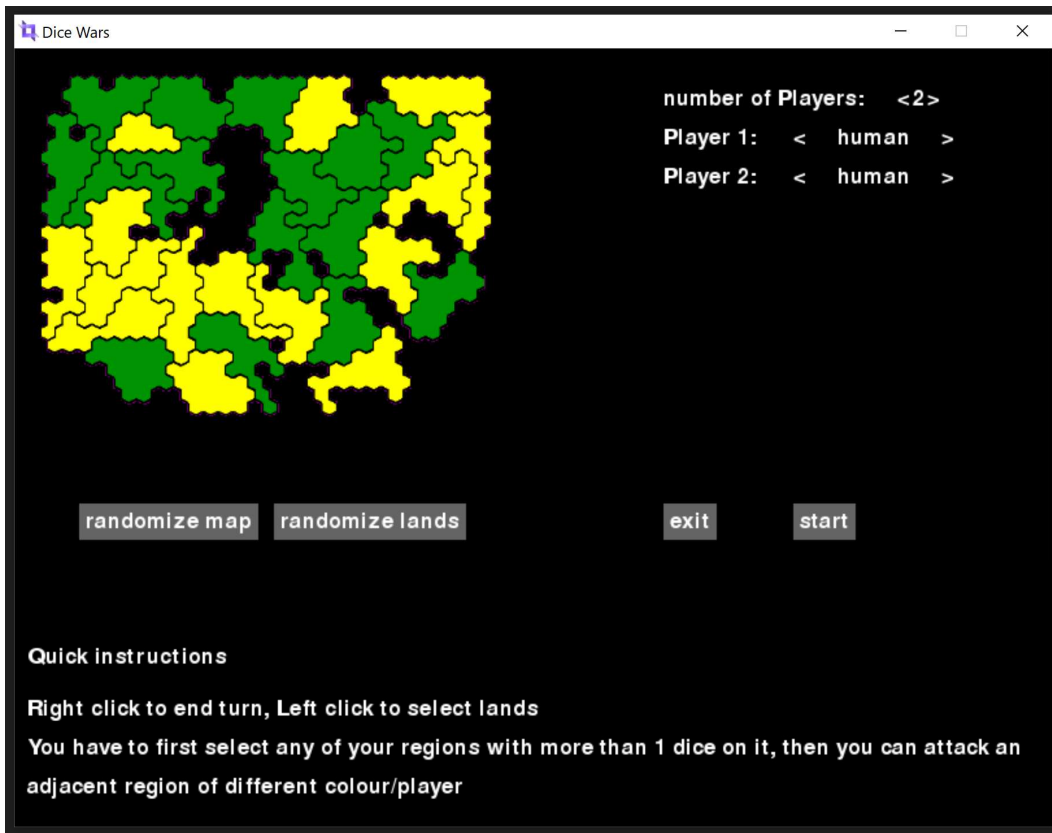
- Python 2
- Python pip package: pygame //you can do pip install pygame

Running it..

The user is required to launch the game using: 'python run_game.py'. This should open a game window with the heading Dice War.

The user can change the number of players by quite simply clicking on the number itself. Now when playing it, in order to execute an action, you will first have to select any of your region with more than 1 die, then select a neighbouring region belonging to any other player. These are all left clicks. If you decide to select a different land instead, you will first have to deselect, the selected land first. At the same time, you can check all the logs on the terminal/cmd you ran the game from.

That is really it, if you know how Dice Wars works, these instructions should be enough for you to play this game.



Understanding Agents

There are a few agents in this game, Namely AI1, Expansive AI, and Passive AI.

They all are essentially implementations of the *AbstractImplementation*.

If one wants to add a new AI, they will need to implement this *AbstractImplementation* class. It shouldn't be too bad, it includes the following methods:

- update
- on_mouse_motion
- on_mouse_btn_down
- on_mouse_btn_up
- on_attack_result
- on_select_result
- on_newturn
- on_gamestate_update

Of course, you don't have to implement all of them, I certainly didn't. But I created this implementation, keeping a mouse driven UI in mind. The `on_gamestate_update` and `on_newturn` are certain important ones, because they're the actual action taking methods. You can take help of the already implemented AIs to code your own.

Behaviour:

The AIdumb is essentially a random choice, as an ironic joke, I call it Expansive in the UI.

The Passive AI is as the name suggests, passive, it would pass most of the times. It won't seek to expand.

The AI1 is the smartest one of all the AI's present in the game. Although it does have an element of randomness, but that was added to compute the dice probabilities.

The Human is you. It's just as smart as you are.