**Instructions**

The game can be started by running Game\_main.py.

The objective of the game is to fulfill your individual assignment by collecting resources on the island with the colored tiles and shipping them to the island with the white tiles.

Resources can be collected by clicking a pawn (circle in the player color; the active player’s pawns are highlighted in pink) and then pressing the DIG sign which is presented. Alternatively, a pawn can be moved to an empty tile by clicking on any of the tiles which are highlighted in red. Each pawn can perform two actions per round. All collected resources are stored in your harbor (the triangle on the colored island). To see which resources you collected, you can click the harbor.

Resources can be moved from your harbor to any of your ships adjacent to it. To do this, click on the harbor, select the resources which need to be moved and press the button corresponding to the ship you want to move them too. Each ship can hold six resource units.

Ships need a pawn to man them in order to be able to move. In order to occupy a boat, select a pawn which is in moveable range of a ship and click on the ship. A white downward arrow will appear on any reachable ship if a pawn is selected. You will be able to move the ship during your next turn.

When a ship with an occupying pawn is selected and the ship has moves left, a ring of red highlighted tiles appears indicating all hexes the ship can move to. This is always a ring shape. The move range can be influenced by burning fuel, which is one of the resource types which can be collected. If a ship is selected, two panels appear. One contains the all resources on the ship (left), the other contains all fuel resources and the “row” option. Selecting row means that you do not burn fuel but it also means your ship moves slowly. If you select any fuel resource, the ring of reachable tiles gets wider. When you click any of the highlighted tiles, the selected fuel is burned and the ship moves.

You can steal resources from enemy ships. To do this, you need to move your ship to a position adjacent to an enemy ship. You can now click the enemy ship and select the resource you want to steal. Note that you ship needs room to carry the resource. Each ship can only steal one resource per turn.

When you move your ship to your home town (triangle on the white land) you can move your resources from the ship to the town by clicking the boat, selecting the resources in the popup and pressing the button of the home town below the resource list.

When you click your home town, a list of stored resources is shown as well as an assignment description. Each assignment consists of two stages. The first one consists of building something and requires a lot of resources. The second stage consists of adding special collectible resources. Each of these will gain you 1 point. If sufficient resources get selected in the leftmost panel to satisfy the current assignment, the fulfill button in the assignment pane is activated and can be pressed to fulfill the assignment. The third panel all the way to the right is an unfinished beta feature.

The resource types in the game are earth, wood, metal, stone, fuel and collectible. Each resource item has two properties which add up to a value of 4. Existing combinations are earth/stone, earth/fuel, fuel/stone, fuel/wood and stone/metal. Each in a 1/3,2/2 and 3/1 version. In addition, for every resource type 10 collectibles are present in the game. These have a resource value of 2 and the label collectible.

To end your turn, press the “end turn” button.

**Program architecture**

The program consists of three major chunks:

* the game class, which manages the turns, the points, the ownership of player pieces and the assignments
* the grid class, which manages which object is located where
* the visualizer which handles input and output.

The visualizer is separated from the rest of the program in order to allow fancier visualization later on without having to redevelop the whole game. The split also will make it easier to split up the program in a client and a server application for multiplayer. Because of the way TKinter works, the visualizer is currently controlling the game and grid classes. I think this is very unelegant, but this can be fixed later when the program is split in a client and a server.