Clear minimal/passing requirement: Prefabricated Maps
Because Smallworld has a board for 2, 3, 4, and 5 players respectively, we only need to make 4 static boards, and sort players into these by player amount.

Extra work: Procedurally Generated Maps
This would require either one algorithm, or four similar configurations of some sort of algorithm. I (Nicolai) will try to pick apart map characteristics/rules and write them below.

GENERAL:

Graphics note: If we create a basic map shape (times four) and divide it into fields, we can color a version for each of the 5 terrain types (not water, it's static), cut them out, and apply them to the appropriate fields

Terrain types:

Default colors & shapes:

Forest:	Dark green, Green. Trees.
Swamp:	Brown land patches and Grey water.
Field:	Yellow, Orange. Square fields, sometimes a tiny village.
Meadow:	Green, Light green. Various cattle.
Mountain:	White, Grey, Brown. Mountaintops.
Water:	Deep Blue, White. Waves.

BOARD:
<<Enum type>> (_forest, swamp, field, meadow, mountain, water_)

TERRAIN (class) (All terrain objects are fields)

type typeID (_type of terrain (forest, swamp, field, etc._)
*int unitID (_0 is default (none), 1 is tribals, beyond that, every race has their own number._)
int totalStrength (_value calculated only when changed, instead of once per potential battle._)
int baseStrength = 2 (_every field has a base strength of 2*_) *$^{\text{Except mountain terrain.}}$
(If typeID == mountain, baseStrength += 1) (make sure this is only run once)
*int unitAmount (_every unit on a field adds 1 strength_)
int debuff (_in the base game, debuff is always = -1, but it might as well be a variable value_)
int (or enumerator thing) modifier (arraylist) (_acts similar to type, but variable in location_)(_Hills, mine, and magic._)
boolean Immunity (_Field can only be accessed by the race that triggered it_)

*depends on how units work