Spielbesprechung:

Muenzenanzahl:
- maximalanzahl pro spiel festlegen
- am spielbrett koennen maxima n muenzen liegen
- muenzen von wassertoten spielern werden zufeaellig am Feld verteilt, dies findet am Ende einer Runde statt
- Muenzen duerfen nur auf Feldern verteilt werden wo kein Wasser/Spieler ist
- Muenzen von wassertoten spielern koennten in einer Liste gespeichert werden, allerdings kann nur eine begrenzte Anzahl wieder aufs Feld verteilt werden.

Spieler:
- randomisiert pro Runde Reihenfolge festgelegt damit es fairer ist.
- muss sich bei jedem Zug bewegen
- Rueckgabe des Moves ans Spielbrett mit N, S, W, O (Nord, sued, west, ost)

Sterben:
- Verlust aller Muenzen und Speicherung in Liste zur spaeteren Verteilung am Ende der Runde

Codierung des Spielfelds:
- moegliche Felder der Spielfeldmatrix
~ = Wasser
^ = Berg
$ = Muenze
alphabetisch (gross/klein moeglich) = Spieler
space = leeres Feld

Speicherung des Spielstandes:
ein Text file mit:
1. derzeitigem Spielfeld
2. Trennungs Zeile
3. Status jedes Spielers Anzahl Muenze, wieviele Muenzen am Stack noch vorhanden, ZugriffsReihenfolge

Speicherung der History:
Pro Spielzug Matrix in Historyfile schreiben (wurde noch nicht fix gesagt ob das wirklich noetig ist)

Spielfeld Matrix:
- flexible Groesse, evtl. Abhaengig von Spieleranzahl
- Aufpassen das einzelne Teile des Spielfelds nicht z.b. durch Wasser abgeschottet sind
- Losung fuer Ringe/abgeschottete Flaechen Ring fuellen und Zugang verschaffen indem “Wassergraben” unterbrochen wurde
- Schwierigkeitsparameter beachten abhaengig vom Grad der Schwierigkeit mehr Wasser/Berge

Termination Condition:
- keine Muenzen am Stack

- keine Muenzen in wassertoter muenzliste

Kommunikation zwischen Spielbrett und Spieler:
- nach jedem Zug Spielfeld aktualisiert und in textfile geschrieben damit der naechste Spieler das File einlesen kann und Zug basierend auf diesem File entscheiden kann
- in jedem Spielzug wird der Spieler vom Spielbrett aufgerufen
z.B. Function call player der die Reihenfolge (Liste) einliest , den current spieler popped und dann
das dazugehoerige spieler python file aufruft.

Tasks:

- ➔ Zufallskarten generierung (Isolde)
- ➔ GUI (Gregor)
- ➔ Parameter einlesen: check ob alle Input parameter passen, Inputspieler files einlesen, Schwierigkeitsgrad einlesen, Spielfeld einlesen (Mario)
- ➔ Initialisierung: Spieler Platzierung, randomisierung rundenweise, Muenzen platzieren (Magdalena)
- ➔ Moves: Fileaufruf
  if schaut ob Feld ausserhab dann setze ihn an den anderen Spielrand oder tot wenn Wasser am Rand
  if Wasser – Muenzen in Liste, setze Spieler randomisiert auf ein moegliches Feld
  if Berg – no move
  if other player – verschiebe den anderen falls moeglich, falls berg stehenbleiben
  if Muenze – increase Muenzencount fuer den Spieler
  (Veronika, Moritz)
- ➔ Spieleraufruf:
  call playerfile das mit dem gepoppten spieler korrespondiert
  get answer of player
  nach Spielzug Karte und Spielstand in File schreiben
  Pruefen ob Termination Condition
  Wenn Spiel Zu Ende (keine Muenzen im Stack oder auf Spielfeld) schreibe Sieger und letzten Zustand raus
  (Christian)
- ➔ Robert Pseudocode

Naechstes Treffen naechste Woche. Vielleicht koennen wir da die Programmteile dann gemeinsam integrieren.