

```

setze "Kolonnmax 12 setze "Liniemax 12
setze "xmin -(23 * ganzzahl (:Kolonnmax / 2)) setze "ymin (23* ganzzahl (:Liniemax /2))
setze "Wahlliste []
startezufall 0 2
wiederhole (:Liniemax * :Kolonnmax) [
    setze "Wähl zufallzahl
    wenn :Wähl = 0 setze "Wahlliste Satz :Wahlliste "SCH
    wenn :Wähl = 1 setze "Wahlliste Satz :Wahlliste "ST
    wenn :Wähl = 2 setze "Wahlliste Satz :Wahlliste "B
]

```

```

lerne Kolonnlinie :v
setze "Linie ganzzahl (:v / :Kolonnmax) +1
wenn (Rest :v :Kolonnmax) = 0 setze "Linie :Linie -1
setze "Kolonn :v - (:Kolonnmax * (:Linie -1))
rückgabe Satz :Linie :Kolonn
ende

```

```

bild löschttext beschleunige vi fi schwarz
schreiblinie [ Scheren: Farbe geld]
schreiblinie [ Steine: Farbe grau]
schreiblinie [ Blatt: Farbe weiss]
schreiblinie []
schreiblinie [ Der Stein bricht die Scheren.]
schreiblinie [ Die Scheren zerschneiden das Blatt.]
schreiblinie [ Das Blatt wickelt den Stein ein.]
wiederhole (:Liniemax * :Kolonnmax) [
    setze "case Kolonnlinie whzahl
    prüfe ((:Liniemax <=21) und (:Kolonnmax <=21))
    wennwahr [
        setze "Position Satz :xmin + ((erstes :case)*23) :ymin - ((letztes :case)*23)
        ellipse [ erstes :Position letztes :Position 10 10 ]
        wenn (Element :Wahlliste whzahl) = "SCH füllfarbe :Position geld
        wenn (Element :Wahlliste whzahl) = "ST füllfarbe :Position grau
        wenn (Element :Wahlliste whzahl) = "B füllfarbe :Position weiss
    ]
    wennfalsch [
        wenn (Element :Wahlliste whzahl) = "SCH fi geld
        wenn (Element :Wahlliste whzahl) = "ST fi grau
        wenn (Element :Wahlliste whzahl) = "B fi weiss
        setze "Position Satz :xmin + ((erstes :case)*2) :ymin - ((letztes :case)*2)
        Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*2)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
    ]
]

```

```

startezufall 1 (:Liniemax * :Kolonnmax) setze "Grenze 0 setze "Zahler 0
solange 0<1 [ // Es ist "solange" unendlich.//
    setze "Zahler :Zahler +1
    setze "Austausche 0
    setze "p1 zufallzahl setze "p2 zufallzahl
    prüfe (Element :Wahlliste :p1) = "SCH
    wennwahr [
        prüfe (Element :Wahlliste :p2) = "ST
        wennwahr [
            setze "p :p1
            wandle :p1 wandle :p1
            ersetze :Wahlliste Element :Wahlliste :p2 :p1
            setze "p1 :p
            setze "case Kolonnlinie :p
            prüfe ((:Liniemax <=21) und (:Kolonnmax <=21))
            wennwahr [
                setze "Position Satz :xmin + ((erstes :case)*23) :ymin - ((letztes :case)*23)
            ]
        ]
    ]
]

```

```

        füllfarbe :Position grau
    ]
    wennfalsch [
        fi grau
        setze "Position Satz :xmin + ((erstes :case)*2) :ymin - ((letztes :case)*2)
        Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*2)-1) - (:Liniemax *2))
        Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
    ]
        setze "Austausche :Austausche +1
    ]
    wennfalsch [
        wenn (Element :Wahlliste :p2) = "B [
            setze "p :p2
            wandle :p2 wandle :p2
            ersetz :Wahlliste Element :Wahlliste :p1 :p2
            setze "p2 :p
            setze "case Kolonnlinie :p
            prüfe (:Liniemax <=21) und (:Kolonnmax <=21))
                wennwahr [
                    setze "Position Satz :xmin + ((erstes :case)*23) :ymin - ((letztes :case)*23)
                    füllfarbe :Position geld
                ]
            wennfalsch [
                fi Gelb
                setze "Position Satz :xmin + ((erstes :case)*2) :ymin - ((letztes :case)*2)
                Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*2)-1) - (:Liniemax *2))
                Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
            ]
            setze "Austausche :Austausche +1
        ]
    ]
    ]
    wennfalsch [
        prüfe (Element :Wahlliste :p1) = "ST
        wennwahr [
            wenn (Element :Wahlliste :p2) = "B [
                setze "p :p1
                wandle :p1 wandle :p1
                ersetz :Wahlliste Element :Wahlliste :p2 :p1
                setze "p1 :p
                setze "case Kolonnlinie :p
                prüfe (:Liniemax <=21) und (:Kolonnmax <=21))
                    wennwahr [
                        setze "Position Satz :xmin + ((erstes :case)*23) :ymin - ((letztes :case)*23)
                        füllfarbe :Position weiss
                    ]
                wennfalsch [
                    fi weiss
                    setze "Position Satz :xmin + ((erstes :case)*2) :ymin - ((letztes :case)*2)
                    Punkt Satz (((erstes :case)*4) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-2) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-1) - (:Liniemax *2))
                    Punkt Satz (((erstes :case)*4)-1) - (:Kolonnmax *2)) (((letztes :case)*4)-2) - (:Liniemax *2))
                ]
            ]
        ]
    ]

```



```
wenn (:Grenze = (:Liniemax * :Kolonmax)) [  
wenn (Element :Wahlliste 1)="ST  
  schreiblinie Satz Satz [ Der Stein hat nach Ablauf von ] :Zahler - :Grenze [ Spielen gewonnen. ]  
wenn (Element :Wahlliste 1)="SCH  
  schreiblinie Satz Satz [ Die Scheren haben nach Ablauf von ] :Zahler - :Grenze [ Spielen gewonnen. ]  
wenn (Element :Wahlliste 1)="B  
  schreiblinie Satz Satz [ Das Blatt hat nach Ablauf von ] :Zahler - :Grenz [ Spielen gewonnen. ]  
                                stop  
                                ]  
                                ]  
                                wennfalsch setze "Grenze 0  
                                ]  
]
```