# CARD

**private** String code;

**private** String title;

**private** String type;

**private** String typeCode;

**private** String subTypeCode;

**private** String text;

**private** **int** cost;

**private** String faction;

**private** String factionCode;

**private** String factionLetter;

**private** **int** factionCost;

**private** String flavor;

**private** String illustrator;

**private** String number;

**private** String quantity;

**private** String setName;

**private** String setCode;

**private** **boolean** uniqueness;

**private** **boolean** limited;

**private** String cyclenumber;

**private** String ancurLink;

**private** String url;

**private** String imagesrc;

Code måste vara positivt tal bestående av 5 siffror.

Alla ints måste vara positiva tal.

Faction måste vara ”Anarch”, ”Criminal” eller ”Shaper” för Runnerkort. Faction måste vara ”Haas-Bioroid”, ”Weyland”, ”Jinteki” eller ”NBN” för Corpkort.

FactionCost måste vara mellan 0 och 5.

Quantity måste vara mellan 1 och 3.

Corp

**Identity**

**Operation**

**Agenda**

**Ice**

**Upgrades**

**Asset**

Runner

**Identity**

## GabrielSantiago

**private** **boolean** succesfulHQRun = false;

När GabrielSantiago börjar varje tur, skall succesfulHQRun sättas till false. När en succesful run på HQ har gjorts, skall credits öka med 2, och succesfulHQRun sättas till true.

**Program**

**Hardware**

**Resource**

**Event**

# SPELARE

**private** **int** clicks;

**private** **int** credits;

**private** **int** agendaPoints;

**private** **int** maxHandSize;

**private** **boolean** winner = **false**;

**private** **boolean** turnOver = **false**;

**private** Deck deckList = **new** ArrayList<>();

Clicks måste vara mellan 0 och 3 för Corporation, och mellan 0 och 4 för Runner. Skall alltid börja på det högsta vid start av varje tur. Vid varje action som Spelare gör, skall clicks minskas med 1. Vid 0 click är turen slut.

Credits måste vara ett positivt tal. Skall alltid börja som 5 vid start av en match. Credits skall kunna ökas och minskas.

agendaPoints måste vara positivt tal. Skall alltid börja som 0 vid start av en match. agendaPoints skall endast kunna ökas, vid 7 eller mer agendaPoints har spelaren vunnit.

Winner börjar som false. turnOver börjar som false.

Corporation

**private** ArrayList<Card> researchAndDevelopment = **new** ArrayList<>();

**private** ArrayList<Card> HQ = **new** ArrayList<>();

**private** ArrayList<Card> archives = **new** ArrayList<>();

Vid början av en match skall alla korten i deckList finnas i researchAndDevelopment. När kort dras från researchAndDevelopment skall kortet läggas i HQ och raderas ur researchAndDevelopment.

Archives är en tom ArrayList i början av varje match. När kort läggs i Archives från HQ eller researchAndDevelopment så skall de raderas ur sin tidigare ArrayList.

HQ, R&D och Archives är Central Servers. Remote Servers är kort som installeras under spelets gång, det finns inga Remote Servers i början av en match. Ett kort som installeras i en Remote Server installeras som unrezzed, men kan när som helst rezzas utan att det kostar ett click. Corporation betalar då rezkostnaden som står på kortet.

Ice skall kunna installeras framför både Central och Remote Servers, och skyddar då den specifika servern.

Runner

**private** ArrayList<Card> stack = **new** ArrayList<>();

**private** ArrayList<Card> grip = **new** ArrayList<>();

**private** ArrayList<Card> heap = **new** ArrayList<>();

Vid början av en match skall alla korten i deckList finnas i stack. När kort dras från stack skall kortet läggas i grip och raderas ur stack.

heap är en tom ArrayList i början av varje match. När kort läggs i heap från grip eller stack så skall de raderas ur sin tidigare ArrayList.

Runner installerar kort i sin Rig. Den består av tre rader, en för Hardware, en för Program, och en för Resources.

När Runnern installerar kort i sin Rig installeras de direkt aktiva (rezzed), och Runnern betalar direkt installationskostnaden för kortet.