

Battles contract

GPK.Battles

Brief

- It is a game battle contract where:
 - ✓ pair 2 players
 - ✓ players transfer 5 WAX each
 - ✓ select cards manually/automatically
 - players choose cards manually within 30s time
 - After 30s, if cards not selected, then automatically contract selects the card
 - ✓ based on their cards selection, draw or nodraw happens
 - ✓ if draw, then players get 1 more chance. If 2 times draw successively for 1 game_id, then the round is dumped. And the card is available for selection in the cardwallet
 - ✓ if nodraw, then winner receives 4 cards with 1 as winning reward, whereas the loser receives 2 cards with 1 losing card (any by shuffle)

About

- contract name - gpkbattlesco
- contract's account name - gpkbattlesco
- action
 - pair2player
 - depositgfee [Payable action]
 - withdrawgfee
 - sel3card
 - sel3cardauto
 - play
 - receiverand [For WAX RNG Oracle service]
 - movegameinfo [Internal inline]
 - empifyplayer [External inline]
 - remplayer [External inline]
 - sendalert [Internal inline]

- table
 - ongamestat
 - usergamestat
 - players
 - gfeewallet

Compile

```
$ eosio-cpp gpkbattlesco.cpp -o gpkbattlesco.wasm
Warning, empty ricardian clause file
Warning, empty ricardian clause file
Warning, action <pair2player> does not have a ricardian contract
Warning, action <depositgfee> does not have a ricardian contract
Warning, action <withdrawgfee> does not have a ricardian contract
Warning, action <sel3card> does not have a ricardian contract
Warning, action <sel3cardauto> does not have a ricardian contract
Warning, action <play> does not have a ricardian contract
Warning, action <receiverand> does not have a ricardian contract
Warning, action <movegameinfo> does not have a ricardian contract
Warning, action <empifyplayer> does not have a ricardian contract
Warning, action <replayer> does not have a ricardian contract
Warning, action <sendalert> does not have a ricardian contract
Warning, action <pair2player> does not have a ricardian contract
Warning, action <depositgfee> does not have a ricardian contract
Warning, action <withdrawgfee> does not have a ricardian contract
Warning, action <sel3card> does not have a ricardian contract
Warning, action <sel3cardauto> does not have a ricardian contract
Warning, action <play> does not have a ricardian contract
Warning, action <receiverand> does not have a ricardian contract
Warning, action <movegameinfo> does not have a ricardian contract
Warning, action <empifyplayer> does not have a ricardian contract
Warning, action <replayer> does not have a ricardian contract
Warning, action <sendalert> does not have a ricardian contract
```

Deploy

- deploy contract

```
$ cleosw set contract gpkbattlesc1 ./
Reading WASM from
/mnt/f/Coding/github_repos/eosio_gpkbattles_contracts/gpkbattlesco/gpkbatt
lesco.wasm...
Publishing contract...
executed transaction:
66e19a80fda6d55889677820bf6371bab3858606c03c177c4b63842a21fd7e98  63848
bytes  10868 us
#      eosio <= eosio::setcode
{"account":"gpkbattlesc1","vmtype":0,"vmversion":0,"code":"0061736d0100000
001c9022f60000060037f7f7f0...
#      eosio <= eosio::setabi
{"account":"gpkbattlesc1","abi":"0e656f73696f3a3a6162692f312e3100100b64657
06f73697467666565000406706...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- Adding eosio.code to permissions (for inline actions)

```
$ cleosw set account permission gpkbattlesc1 active --add-code
executed transaction:
32dff3e5aba6607afbf1ea37f525f3a4094273818af81b72041d058a6a717208  184
bytes  162 us
#      eosio <= eosio::updateauth
{"account":"gpkbattlesc1","permission":"active","parent":"owner","auth":
{"threshold":1,"keys":[{"key...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

Testing

Action - pair2player

- pair any 2 player of simpleassets contract type

```
$ cleosw push action gpkbattlesc1 pair2player '["simpleassets"]' -p
gpkbattlesc1@active
Error 3050003: eosio_assert_message assertion failure
Error Details:
assertion failure with message: players_list must be min. 2 in size.
pending console output:
```

- So, you need min. 2 players to be added into the `players_list` of `players` table

1. pair 2 players

```
$ cleosw push action gpkbattlesc1 pair2player '["simpleassets"]' -p
gpkbattlesc1@active
executed transaction:
24057e1c790a89f89b7efb429e414f3d368c45451dc73d957f0db3e2f0373f96  104
bytes  299 us
# gpkbattlesc1 <= gpkbattlesc1::pair2player
{"asset_contract_ac":"simpleassets"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` of this contract

```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer
{
  "rows": [{
    "data": {
      "game_id": "10001723968966",
      "player_1": "gbuser111115",
      "player_2": "gbuser111113",
      "game_fee": "0 ",
      "asset_contract_ac": "",
      "player1_cards": [],
      "player2_cards": [],
      "player1_cards_combo": "",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 0,
      "nodraw_count": 0,
      "total_play_count": 0
    },
    "payer": "gpkbattlesc1"
  }
],
"more": false,
"next_key": ""
}
```

- view the table `players` of this contract

```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 players --show-payer
{
  "rows": [{
    "data": {
      "asset_contract_ac": "simpleassets",
      "players_list": [
        "gbuser111112",
        "gbuser111114",
        "gbuser111111",
        "gbuser111121"
      ]
    },
    "payer": "gpkbattlesc1"
  }],
  "more": false,
  "next_key": ""
}
```

- Observation:
 - A row created with game_id & paired players
 - paired players are removed from the players_list

2. pair 2 players

```
$ cleosw push action gpkbattlesc1 pair2player '["simpleassets"]' -p
gpkbattlesc1@active
executed transaction:
5007f3dd69dc7a25003ee8e1ee1dcad7bdd09054ea28c220474ed9f96f86390b  104
bytes  289 us
# gpkbattlesc1 <= gpkbattlesc1::pair2player
{"asset_contract_ac":"simpleassets"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` of this contract

```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer
{
  "rows": [{
    "data": {
      "game_id": "10001723970651",
      "player_1": "gbuser111111",
      "player_2": "gbuser111113",
      "game_fee": "0 ",
      "asset_contract_ac": "",
      "player1_cards": [],
      "player2_cards": [],
      "player1_cards_combo": "",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 0,
      "nodraw_count": 0,
      "total_play_count": 0
    },
    "payer": "gpkbattlesc1"
  }],
  "more": false,
  "next_key": ""
}

```

- view the table `players` of this contract

```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 players --show-payer
{
  "rows": [{
    "data": {
      "asset_contract_ac": "simpleassets",
      "players_list": [
        "gbuser111121",
        "gbuser111115",
        "gbuser111112",
        "gbuser111114"
      ]
    },
    "payer": "gpkbattlesc1"
  }
],
"more": false,
"next_key": ""
}
```

- Observation:
 - A row created with game_id & paired players
 - paired players are removed from the players_list

3. pair 2 players

```
$ cleosw push action gpkbattlesc1 pair2player '["simpleassets"]' -p
gpkbattlesc1@active
executed transaction:
58d2c8f4808a666a6bffa2008cd258e17a84912ae7676c62410454323bddb361e  104
bytes  300 us
# gpkbattlesc1 <= gpkbattlesc1::pair2player
{"asset_contract_ac":"simpleassets"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` of this contract

```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer
{
  "rows": [{
    "data": {
      "game_id": "10001723987390",
      "player_1": "gbuser111112",
      "player_2": "gbuser111114",
      "game_fee": "0 ",
      "asset_contract_ac": "",
      "player1_cards": [],
      "player2_cards": [],
      "player1_cards_combo": "",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 0,
      "nodraw_count": 0,
      "total_play_count": 0
    },
    "payer": "gpkbattlesc1"
  }],
  "more": false,
  "next_key": ""
}

```

- view the table `players` of this contract


```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 players --show-payer
{
  "rows": [{
    "data": {
      "asset_contract_ac": "simpleassets",
      "players_list": [
        "gbuser111111",
        "gbuser111113"
      ]
    },
    "payer": "gpkbattlesc1"
  }
],
"more": false,
"next_key": ""
}
```

- Observation:
 - A row created with game_id & paired players
 - paired players are removed from the players_list

Action - depositgfee

- Player gbuser111112 wants to deposit game fee for playing game

```
$ cleosw push action eosio.token transfer '["gbuser111112",
"gpkbattlesc1", "5.00000000 WAX", "transfer 5 WAX fee for playing game"]'
-p gbuser111112@active
executed transaction:
50515aef04448f05af28f01aa4b1020e3df8fe3fed813a971f5638900a1acb2 160
bytes 277 us
# eosio.token <= eosio.token::transfer
{"from":"gbuser111112","to":"gpkbattlesc1","quantity":"5.00000000
WAX","memo":"transfer 5 WAX fee fo...
# gbuser111112 <= eosio.token::transfer
{"from":"gbuser111112","to":"gpkbattlesc1","quantity":"5.00000000
WAX","memo":"transfer 5 WAX fee fo...
# gpkbattlesc1 <= eosio.token::transfer
{"from":"gbuser111112","to":"gpkbattlesc1","quantity":"5.00000000
WAX","memo":"transfer 5 WAX fee fo...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table gfeewallet of this contract

```
$ cleosw get table gpkbattlesc1 gbuser111112 gfeewallet --show-payer
{
  "rows": [{
    "data": {
      "balance": "5.00000000 WAX"
    },
    "payer": "gpkbattlesc1"
  }
],
"more": false,
"next_key": ""
}
```

Action - withdrawgfee

- View the gbuser111112 balance in the gfeewallet table

```
$ cleosw get table gpkbattlesc1 gbuser111112 gfeewallet
{
  "rows": [{
    "balance": "5.00000000 WAX"
  }
],
"more": false,
"next_key": ""
}
```

- gbuser111112 withdraws qty from gfeewallet table

```
$ cleosw get table gpkbattlesc1 gbuser111112 gfeewallet
$ cleosw push action gpkbattlesc1 withdrawgfee '["gbuser111112",
"5.00000000 WAX"]' -p gbuser111112@active
executed transaction:
644eec4c5862ad6ce6ed29a6702d594a4c47b200300f577dc7c421fe23d4ce9f  120
bytes  290 us
# gpkbattlesc1 <= gpkbattlesc1::withdrawgfee
{"player":"gbuser111112","qty":"5.00000000 WAX"}
# eosio.token <= eosio.token::transfer
{"from":"gpkbattlesc1","to":"gbuser111112","quantity":"5.00000000
WAX","memo":"player withdraws 5.00...
# gpkbattlesc1 <= eosio.token::transfer
{"from":"gpkbattlesc1","to":"gbuser111112","quantity":"5.00000000
WAX","memo":"player withdraws 5.00...
# gbuser111112 <= eosio.token::transfer
{"from":"gpkbattlesc1","to":"gbuser111112","quantity":"5.00000000
WAX","memo":"player withdraws 5.00...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table gfeewallet

```
$ cleosw get table gpkbattlesc1 gbuser111112 gfeewallet
{
  "rows": [],
  "more": false,
  "next_key": ""
}
```

Action - sel3card

Game id:10001723987390

- player gbuser111111 selects 3 cards

```
$ cleosw push action gpkbattlesc1 sel3card '["gbuser111112",
"simpleassets", "100000000007693", "100000000007694", "100000000007695"]'
-p gbuser111112@active
Error 3050003: eosio_assert_message assertion failure
Error Details:
assertion failure with message: the player is not in the wallet table.
pending console output:
```

- here, it is giving error, because there is no token transferred by player. View the table gfeewallet :

```
$ cleosw get table gpkbattlesc1 gbuser111112 gfeewallet --show-payer
{
  "rows": [],
  "more": false,
  "next_key": ""
}
```

- let's see the cards available

```
$ cleosw get table gpkbatescrow gbuser111112 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007693",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007694",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007695",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

- o Now, successfully selected 3 cards

```
$ cleosw push action gpkbattlesc1 sel3card '["gbuser111112",
"simpleassets", "100000000007693", "100000000007694",
"100000000007695"]' -p gbuser111112@active
executed transaction:
3baa1c2fd3023c9d22f6319b4459fed156dafaeb810cd7d5e7995848ca01efb0  136
bytes  517 us
# gpkbattlesc1 <= gpkbattlesc1::sel3card
{"player":"gbuser111112","asset_contract_ac":"simpleassets","card1_id":
"100000000007693","card2_id":...
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007693","status":"selecte
d"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007694","status":"selecte
d"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007695","status":"selecte
d"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- o view the card's status in table cardwallet

```
$ cleosw get table gpkbatescrow gbuser111112 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007693",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007694",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007695",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

- o view the card's status in table ongamestat

```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer -
-lower 10001723987390 --limit 1
{
  "rows": [{
    "data": {
      "game_id": "10001723987390",
      "player_1": "gbuser111112",
      "player_2": "gbuser111114",
      "game_fee": "5.00000000 WAX",
      "asset_contract_ac": "simpleassets",
      "player1_cards": [
        "100000000007693",
        "100000000007694",
        "100000000007695"
      ],
      "player2_cards": [],
      "player1_cards_combo": "1a2b",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 0,
      "nodraw_count": 0,
      "total_play_count": 0
    },
    "payer": "gpkbattlesc1"
  }
],
  "more": false,
  "next_key": ""
}

```

- player gbuser111114 selects 3 cards
 - let's see the cards available

```
$ cleosw get table gpkbatescrow gbuser111114 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007702",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007722",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007727",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

- o Now, successfully selected 3 cards

```
$ cleosw push action gpkbattlesc1 sel3card '["gbuser111114",
"simpleassets", "100000000007702", "100000000007722",
"100000000007727"]' -p gbuser111114@active
executed transaction:
e4b9a0ee52ea75ed24566e29eca4b76127eb1757b30b965c18aa0796da8dcbf3  136
bytes  439 us
# gpkbattlesc1 <= gpkbattlesc1::sel3card
{"player":"gbuser111114","asset_contract_ac":"simpleassets","card1_id":
"100000000007702","card2_id":...
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007702","status":"selecte
d"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007722","status":"selecte
d"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007727","status":"selecte
d"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- o view the card's status in table cardwallet

```
$ cleosw get table gpkbatescrow gbuser111114 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007702",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007722",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007727",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

- o view the card's status in table ongamestat


```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer -
-lower 10001723987390 --limit 1
{
  "rows": [{
    "data": {
      "game_id": "10001723987390",
      "player_1": "gbuser111112",
      "player_2": "gbuser111114",
      "game_fee": "5.00000000 WAX",
      "asset_contract_ac": "simpleassets",
      "player1_cards": [
        "100000000007693",
        "100000000007694",
        "100000000007695"
      ],
      "player2_cards": [
        "100000000007702",
        "100000000007722",
        "100000000007727"
      ],
      "player1_cards_combo": "1a2b",
      "player2_cards_combo": "1a2b",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 0,
      "nodraw_count": 0,
      "total_play_count": 0
    },
    "payer": "gpkbattlesc1"
  }
],
"more": false,
"next_key": ""
}

```

Action - sel3cardauto

Game id:10001723968966

1. auto select cards for player - gbuser111115

```

$ cleosw push action gpkbattlesc1 sel3cardauto '["gbuser111115",
"simpleassets"]' -p gpkbattlesc1@active
executed transaction:
3d227f8b1f3a62ab67344bc17b98aabb30b28302ff996a41a7d83df7036c789e  112
bytes  390 us
#  gpkbattlesc1 <= gpkbattlesc1::sel3cardauto
{"player":"gbuser111115","asset_contract_ac":"simpleassets"}
#  gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007705","status":"selected"}
#  gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007706","status":"selected"}
#  gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007707","status":"selected"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]

```

- view the table `ongamestat` if the cards selected


```
$ cleosw get table gpkbatescrow gbuser111115 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007705",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007706",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007707",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

2. auto select cards for player - gbuser111113

```
$ cleosw push action gpkbattlesc1 sel3cardauto '["gbuser111113",
"simpleassets"]' -p gpkbattlesc1@active
executed transaction:
a146bdc0c05a20e54148f5967a0861fec569a2b1ac4e03cf86c3e339780bd973  112
bytes  373 us
# gpkbattlesc1 <= gpkbattlesc1::sel3cardauto
{"player":"gbuser111113","asset_contract_ac":"simpleassets"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007697","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007710","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007721","status":"selected"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table ongamestat if the cards selected

```
$ cleosw get table gpkbbattlesc1 gpkbbattlesc1 ongamestat --show-payer  
{  
  "rows": [{  
    "data": {  
      "game_id": "10001723968966",  
      "player_1": "gbuser111115",  
      "player_2": "gbuser111113",  
      "game_fee": "5.00000000 WAX",  
      "asset_contract_ac": "simpleassets",  
      "player1_cards": [  
        "100000000007705",  
        "100000000007706",  
        "100000000007707"  
      ],  
      "player2_cards": [  
        "100000000007697",  
        "100000000007710",  
        "100000000007721"  
      ],  
      "player1_cards_combo": "2a1b",  
      "player2_cards_combo": "2a1b",  
      "start_timestamp": 0,  
      "end_timestamp": 0,  
      "result": "",  
      "winner": "",  
      "loser": "",  
      "winner_transfer_cards": [],  
      "loser_transfer_cards": [],  
      "card_won": 0,  
      "status": "",  
      "random_value":  
"0000000000000000000000000000000000000000000000000000000000000000",  
      "draw_count": 0,  
      "nodraw_count": 0,  
      "total_play_count": 0  
    },  
    "payer": "gpkbbattlesc1"  
  }  
],  
  "more": false,  
  "next_key": ""  
}
```

- view the table `cardwallet`

```
$ cleosw get table gpkbatescrow gbuster111113 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007697",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007710",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  },{
    "data": {
      "card_id": "100000000007721",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

Game id:10001723970651

1. auto select cards for player - gbuster111111

```
$ cleosw push action gpkbattlesc1 sel3cardauto '['gbuster111111',
"simpleassets"]' -p gpkbattlesc1@active
executed transaction:
88b7894a2c035ee30897c051360323092bb364e0d736e1ced58b0b222d7fe956  112
bytes  447 us
# gpkbattlesc1 <= gpkbattlesc1::sel3cardauto
{"player":"gbuster111111","asset_contract_ac":"simpleassets"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuster111111","card_id":"100000000007690","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuster111111","card_id":"100000000007691","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuster111111","card_id":"100000000007692","status":"selected"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` if the cards selected


```
$ cleosw get table gpkbatescrow gbuser111111 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007690",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007691",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007692",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

2. auto select cards for player - gbuser111113

```
$ cleosw push action gpkbattlesc1 sel3cardauto '["gbuser111113",
"simpleassets"]' -p gpkbattlesc1@active
executed transaction:
d3d916e92e6e1a81f1ebaf92086cb20f93ecac1d1d26b0500707aaef2ace1cb7  112
bytes  494 us
# gpkbattlesc1 <= gpkbattlesc1::sel3cardauto
{"player":"gbuser111113","asset_contract_ac":"simpleassets"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007697","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007710","status":"selected"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007721","status":"selected"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table ongamestat if the cards selected

[illegible]

- view the table `cardwallet`

```
$ cleosw get table gpkbatescrow gbuser111113 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007697",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007710",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007721",
      "contract_ac": "simpleassets",
      "usage_status": "selected"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}
```

Action - play

Case-1: 2 times Draw

1. play the game_id **10001723968966**

```

$ cleosw push action gpkbattlesc1 play '['10001723968966']' -p
gpkbattlesc1@active
executed transaction:
20c26da600da56414a85fe35aa20f34ff01a7a55def4f1a8c6aab3e034a3892e  104
bytes  490 us
# gpkbattlesc1 <= gpkbattlesc1::play
{"game_id":"10001723968966"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007705","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007706","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007707","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007697","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007710","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007721","status":"available"}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"gbuser111115 has one last chance to
select card, as game with id: ...
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":"gbuser111113 has one last chance to
select card, as game with id: ...
# gbuser111115 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"gbuser111115 has one last chance to
select card, as game with id: ...
# gbuser111113 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":"gbuser111113 has one last chance to
select card, as game with id: ...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]

```

- view the table `ongamestat` of this contract

```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer -
-lower 10001723968966 --limit 1
{
  "rows": [{
    "data": {
      "game_id": "10001723968966",
      "player_1": "gbuser111115",
      "player_2": "gbuser111113",
      "game_fee": "5.00000000 WAX",
      "asset_contract_ac": "simpleassets",
      "player1_cards": [],
      "player2_cards": [],
      "player1_cards_combo": "",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "draw",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "waitdue1draw",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 1,
      "nodraw_count": 0,
      "total_play_count": 1
    },
    "payer": "gpkbattlesc1"
  }
],
  "more": false,
  "next_key": ""
}

```

- view the card's status in table cardwallet of gbuser111115

```

$ cleosw get table gpkbatescrow gbuser111115 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007705",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007706",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007707",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}

```

- view the card's status in table cardwallet of gbuser111113

```

$ cleosw get table gpkbatescrow gbuser111113 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007697",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007710",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007721",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}

```

○ Observations:

- As the game is draw for 1st time. So, the game row is still present
- The players have been alerted to select cards again
- the cards are marked as “available” in the `cardwallet` table 2. Draw again

```

$ cleosw push action gpkbattlesc1 play '["10001723968966"]' -p
gpkbattlesc1@active
executed transaction:
49ef55f7fd44970af4244a847dfa0dcd01db1c1cca991e1eeea06560e5e008fb
104 bytes  503 us
# gpkbattlesc1 <= gpkbattlesc1::play
{"game_id":"10001723968966"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007705","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007706","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111115","card_id":"100000000007707","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007697","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007710","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111113","card_id":"100000000007721","status":"available"}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":" The game with id:
'10001723968966' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":" The game with id:
'10001723968966' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::moergameinfo
{"game_id":"10001723968966","message":"your game with id:
'10001723968966' is moved to 'usergamestat..."}
# gbuser111115 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":" The game with id:
'10001723968966' is draw for 2 times."}
# gbuser111113 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":" The game with id:
'10001723968966' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"your game with id:
'10001723968966' is moved to 'usergamestat' tab..."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":"your game with id:
'10001723968966' is moved to 'usergamestat' tab..."}
# gbuser111115 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"your game with id:
'10001723968966' is moved to 'usergamestat' tab..."}
# gbuser111113 <= gpkbattlesc1::sendalert
{"user":"gbuser111113","message":"your game with id:
'10001723968966' is moved to 'usergamestat' tab..."}
warning: transaction executed locally, but may not be confirmed by
the network yet ]

```

2. play the game_id 10001723987390

```
$ cleosw push action gpkbattlesc1 play '['10001723987390']' -p
gpkbattlesc1@active
executed transaction:
ce752d6fdc84c1766d0d15c715ecd234371aab7e832ac741badbd24bffc018a3  104
bytes  496 us
# gpkbattlesc1 <= gpkbattlesc1::play
{"game_id":"10001723987390"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007693","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007694","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007695","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007702","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007722","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007727","status":"available"}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":"gbuser111112 has one last chance to
select card, as game with id: ...
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":"gbuser111114 has one last chance to
select card, as game with id: ...
# gbuser111112 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":"gbuser111112 has one last chance to
select card, as game with id: ...
# gbuser111114 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":"gbuser111114 has one last chance to
select card, as game with id: ...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` of this contract


```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer -
-lower 10001723987390 --limit 1
{
  "rows": [{
    "data": {
      "game_id": "10001723987390",
      "player_1": "gbuser111112",
      "player_2": "gbuser111114",
      "game_fee": "5.00000000 WAX",
      "asset_contract_ac": "simpleassets",
      "player1_cards": [],
      "player2_cards": [],
      "player1_cards_combo": "",
      "player2_cards_combo": "",
      "start_timestamp": 0,
      "end_timestamp": 0,
      "result": "draw",
      "winner": "",
      "loser": "",
      "winner_transfer_cards": [],
      "loser_transfer_cards": [],
      "card_won": 0,
      "status": "waitdue1draw",
      "random_value":
"0000000000000000000000000000000000000000000000000000000000000000",
      "draw_count": 1,
      "nodraw_count": 0,
      "total_play_count": 1
    },
    "payer": "gpkbattlesc1"
  }
],
  "more": false,
  "next_key": ""
}

```

- view the card's status in table cardwallet of gbuser111112

```

$ cleosw get table gpkbatescrow gbuser111112 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007693",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007694",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007695",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}

```

- view the card's status in table cardwallet of gbuser111114

```

$ cleosw get table gpkbatescrow gbuser111114 cardwallet --show-payer
{
  "rows": [{
    "data": {
      "card_id": "100000000007702",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007722",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }, {
    "data": {
      "card_id": "100000000007727",
      "contract_ac": "simpleassets",
      "usage_status": "available"
    },
    "payer": "gpkbatescrow"
  }
],
  "more": false,
  "next_key": ""
}

```

○ Observations:

- As the game is draw for 1st time. So, the game row is still present
- The players have been alerted to select cards again
- the cards are marked as “available” in the `cardwallet` table 2. Draw again

```

$ cleosw push action gpkbattlesc1 play '["10001723987390"]' -p
gpkbattlesc1@active
executed transaction:
b496df7f3123d63ff0f429398500cb52229d234c8a361a6f6bd028ac422e0f86
104 bytes  506 us
# gpkbattlesc1 <= gpkbattlesc1::play
{"game_id":"10001723987390"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007693","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007694","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111112","card_id":"100000000007695","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007702","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007722","status":"available"}
# gpkbatescrow <= gpkbatescrow::setgstatus
{"player":"gbuser111114","card_id":"100000000007727","status":"available"}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":" The game with id:
'10001723987390' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":" The game with id:
'10001723987390' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::moergameinfo
{"game_id":"10001723987390","message":"your game with id:
'10001723987390' is moved to 'usergamestat..."}
# gbuser111112 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":" The game with id:
'10001723987390' is draw for 2 times."}
# gbuser111114 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":" The game with id:
'10001723987390' is draw for 2 times."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":"your game with id:
'10001723987390' is moved to 'usergamestat' tab..."}
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":"your game with id:
'10001723987390' is moved to 'usergamestat' tab..."}
# gbuser111112 <= gpkbattlesc1::sendalert
{"user":"gbuser111112","message":"your game with id:
'10001723987390' is moved to 'usergamestat' tab..."}
# gbuser111114 <= gpkbattlesc1::sendalert
{"user":"gbuser111114","message":"your game with id:
'10001723987390' is moved to 'usergamestat' tab..."}
warning: transaction executed locally, but may not be confirmed by
the network yet ]

```

Case-2: 1 time Draw & 1 time Nodraw

Case-3: Nodraw

- play the game_id **10001723970651**

```
$ cleosw push action gpkbattlesc1 play '["10001723970651"]' -p
gpkbattlesc1@active
executed transaction:
862fe8039460ee3e3a6c7ef5abac3ed88d909e205dc9edd3e698c2f68f74299e  104
bytes  283 us
# gpkbattlesc1 <= gpkbattlesc1::play
{"game_id":"10001723970651"}
#      orng.wax <= orng.wax::requestrand
{"assoc_id":"10001723970651","signing_value":"14370121078412468780","caller":"gpkbattlesc1"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

- view the table `ongamestat` of this contract

```

$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer
{
  "rows": [{
    "data": {
      "game_id": "10001723970651",
      "player_1": "gbuser111111",
      "player_2": "gbuser111113",
      "game_fee": "5.00000000 WAX",
      "asset_contract_ac": "simpleassets",
      "player1_cards": [
        "100000000007690",
        "100000000007691",
        "100000000007692"
      ],
      "player2_cards": [
        "100000000007697",
        "100000000007710",
        "100000000007721"
      ],
      "player1_cards_combo": "1a2b",
      "player2_cards_combo": "2a1b",
      "start_timestamp": 1600514002,
      "end_timestamp": 1600514003,
      "result": "nodraw",
      "winner": "gbuser111111",
      "loser": "gbuser111113",
      "winner_transfer_cards": [
        "100000000007690",
        "100000000007691",
        "100000000007692",
        "100000000007710"
      ],
      "loser_transfer_cards": [
        "100000000007697",
        "100000000007721"
      ],
      "card_won": "100000000007710",
      "status": "over",
      "random_value":
"cc707bbd70e47595c2712fa6bb50c1c8d5a4f7282c1a03e406b12cf8d2629b47",
      "draw_count": 0,
      "nodraw_count": 1,
      "total_play_count": 1
    },
    "payer": "gpkbattlesc1"
  }
],
  "more": false,
  "next_key": ""
}

```

- disburse cards to winner & loser & move the game info

```

$ cleosw push action gpkbattlesc1 disndcards '['10001723972761']' -p
gpkbattlesc1@active
executed transaction:
ea1d6fd387b89e59195d8ac498537f8fca581ebb4a95d1db16ab22f39da2d35f  192
bytes  1227 us
# gpkbattlesc1 <= gpkbattlesc1::disndcards
{"game_id":"10001723972761"}
# gpkbatescrow <= gpkbatescrow::disburse
{"game_id":"10001723972761"}
# gpkbattlesc1 <= gpkbattlesc1::moergameinfo
{"game_id":"10001723972761","message":"your game with id: '10001723972761'
is moved to 'usergamestat...
# simpleassets <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111121","assetids":
["100000000007716","100000000007728","10000000...
# simpleassets <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111115","assetids":
["100000000007705","100000000007706"],"memo":"...
# gpkbatescrow <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111121","assetids":
["100000000007716","100000000007728","10000000...
# gbuser111121 <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111121","assetids":
["100000000007716","100000000007728","10000000...
# gpkbatescrow <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111115","assetids":
["100000000007705","100000000007706"],"memo":"...
# gbuser111115 <= simpleassets::transfer
{"from":"gpkbatescrow","to":"gbuser111115","assetids":
["100000000007705","100000000007706"],"memo":"...
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111121","message":"your game with id: '10001723972761' is
moved to 'usergamestat' tab...
# gpkbattlesc1 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"your game with id: '10001723972761' is
moved to 'usergamestat' tab...
# gbuser111121 <= gpkbattlesc1::sendalert
{"user":"gbuser111121","message":"your game with id: '10001723972761' is
moved to 'usergamestat' tab...
# gbuser111115 <= gpkbattlesc1::sendalert
{"user":"gbuser111115","message":"your game with id: '10001723972761' is
moved to 'usergamestat' tab...
warning: transaction executed locally, but may not be confirmed by the
network yet      ]

```

- view the table `ongamestat` of this contract & check that the game row is moved to `usergamestat` table

```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 ongamestat --show-payer
{
  "rows": [],
  "more": false,
  "next_key": ""
}
```

- view the table `cardwallet` of `gbuser111121` & check that their cards have been transferred back to winner & loser with 4 & 2 cards respectively

```
$ cleosw get table gpkbatescrow gbuser111121 cardwallet --show-payer
{
  "rows": [],
  "more": false,
  "next_key": ""
}
```

- view the table `cardwallet` of `gbuser111115` & check that their cards have been transferred back to winner & loser with 4 & 2 cards respectively

```
$ cleosw get table gpkbatescrow gbuser111115 cardwallet --show-payer
{
  "rows": [],
  "more": false,
  "next_key": ""
}
```

- view the table `players` & check that the players are not added back

```
$ cleosw get table gpkbattlesc1 gpkbattlesc1 players --show-payer
{
  "rows": [{
    "data": {
      "asset_contract_ac": "simpleassets",
      "players_list": [
        "gbuser111112",
        "gbuser111114",
        "gbuser111111",
        "gbuser111113"
      ]
    },
    "payer": "gpkbattlesc1"
  }],
  "more": false,
  "next_key": ""
}
```


- If there is a need to delete the game row after the stage `se13card` or `se13cardauto`

```
$ cleosw push action gpkbattlesc1 testaddplayr '["simpleassets",
"gbuser111111"]' -p gpkbattlesc1@active
executed transaction:
15419dcd509ca107e8345669e1d8ad9c837ca867980ead7a598836195446c53  112
bytes  225 us
# gpkbattlesc1 <= gpkbattlesc1::testaddplayr
{"asset_contract_ac":"simpleassets","player":"gbuser111112"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

```
$ cleosw push action gpkbattlesc1 testaddplayr '["simpleassets",
"gbuser111114"]' -p gpkbattlesc1@active
executed transaction:
ee5b545d6b83424376effc8c798d73f0ea417020d7dac5231eccb66fce2268c5  112
bytes  221 us
# gpkbattlesc1 <= gpkbattlesc1::testaddplayr
{"asset_contract_ac":"simpleassets","player":"gbuser111113"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

```
$ cleosw push action gpkbattlesc1 testdelongam '["10001723801103"]' -p
gpkbattlesc1@active
executed transaction:
65bc2a6a15781f12e391f967e53241c51b7b5b1bc9d7d189b1bc773dc2162d5a  104
bytes  297 us
# gpkbattlesc1 <= gpkbattlesc1::testdelongam
{"game_id":"10001723786534"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

```
$ cleosw push action gpkbatescrow testsetcstat '["gbuser111111",
"100000000007690", "available"]' -p gpkbatescrow@active
executed transaction:
67f9664fb7245e933d34c9617985d97d1358a55fa5d98980470eef0d6a0b3098  120
bytes  232 us
# gpkbatescrow <= gpkbatescrow::testsetcstat
{"player":"gbuser111112","card_id":"100000000007693","status":"available"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

```
$ cleosw push action gpkbatescrow testsetcstat '["gbuser111114",
"100000000007722", "available"]' -p gpkbatescrow@active
executed transaction:
c9ded880c5855177fe3422beada68c1a8f2dab86b2e72489cd1d31e8e619c986  120
bytes  163 us
# gpkbatescrow <= gpkbatescrow::testsetcstat
{"player":"gbuser111112","card_id":"100000000007694","status":"available"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

```
$ cleosw push action gpkbatescrow testsetcstat '["gbuser111112",
"100000000007695", "available"]' -p gpkbatescrow@active
executed transaction:
716d42bfe63f6277147bdfac9f65cfd2bd7421a30b23ab45e242258db9720ff9  120
```

```
bytes 166 us
# gpkbatescrow <= gpkbatescrow::testsetcstat
{"player": "gbuser111112", "card_id": "100000000007695", "status": "available"}
warning: transaction executed locally, but may not be confirmed by the
network yet      ]
```

TODO:

- ☐ Suppose the `pair2player` ACTION happens & the players don't transfer money >> `selectcards` forever.
 - In this case, we lose the RAM space in `gpkbattlesco` contract
 - Solution: We need to set some time (at `pair2player`) for the 1(only `selectcard`) or 2 (both `gfee` & `selectcard`) further steps

NOTES

- The steps followed in the game is as follows:
 - i. Pair 2 players using `pair2player` action
 - ii. Each player transfers fees using `token::transfer` action
 - iii. Each Player select 3 cards using `select3card` action
 - iv. if same card selected ==> Draw if different cards selected ==> RNG is generated & then the winner gets the any 1 of 3 cards from loser.
- Shuffle & Select: This method is applied for:
 - for a list of players, choose 2 for pairing them.
 - for a list of cards, choose 3 for auto-select of cards if the player doesn't select cards within 30 secs time.
- During game after result declaration:
 - draw:
 - 1 time: no transfer of cards, set cards available
 - 2 times: no transfer of cards, set cards available, put players back into the `players_list`
 - non-draw: disburse 4 to winner & 2 to loser
 - the players are not put back into `players_list` in `players` table.
- Post game
 - Only in case of non-draw, the `ongamestat` table's row data is moved to respective players' `usergamestat` table.

