

Forum: [Hack Forums](#)

Thread: [Need a Script altered slightly](#)

Deeg - January 2, 2016, 12:28 pm

I have a bustabit script that I have completely done but what I pretty much want done on it is to make it take off a round every time I lose, then on the next round it comes back in and adjusts the bet so it regains the money I just lost. It already adjusts for the loss but it keeps going after I lose.

Ex what it does now: I win game 1 and go in for game 2 with the same bet. I lose game 2 so it goes in game 3 adjusting cashout and bet amount to regain what I lost game 2.

Pitfall to this example of current script. If there is a bust 5 games in a row that are REALLY low and lower than the script is set for it just constantly bets more so that at the end of game 5 then Im all out of bits if it loses.

Ex of what I want it to do after it is changed: I win game 1 and go in for game 2. I lose game 2 so it takes off for game 3 and goes back in for game 4 already adjusted to regain bits like it would do in current design.

<http://pastebin.com/7sjJ1YnB> <-- That is the link to the script. If you can edit it I would be greatly pleased! PM me Or Post below if you can do that!

Thanks if you are able to help me!

al5ina5 - January 3, 2016, 10:21 am

I can help, but it's gonna be much harder and cost you a lot more with that horrible obfuscation...

Sakuto - January 3, 2016, 12:26 pm

"I have completely done" + Provide obfuscated script. Hu, they are a problem here. And your username is not really Safixk.

Spoiler [\(Click to View\)](#)

```
Code var baseBet = 10;
var baseMultiplier = 1.13;
var variableBase = true;
var streakSecurity = 4;
var maximumBet = 999999;
var baseSatoshi = baseBet * 100;
var currentBet = baseSatoshi;
var currentMultiplier = baseMultiplier;
var currentGameID = -1;
var firstGame = true;
var lossStreak = 0;
var coolingDown = false;
console.log ("===== Safixk's BustaBit Bot =====");
console.log ("USERNAME: " + engine.getUsername ());
console.log ("STARTING BALANCE: " + (engine.getBalance () / 100).toFixed (2) + " bits");
var startingBalance = engine.getBalance ();
if (variableBase) {
```

```

    console.warn ("[WARN] THIS BOT IS EARLY ALPHA. Resilient to " + streakSecurity + "-loss
streaks.")
};
engine.on ("game_starting", function (_0x3493xe) {
    console.log ("===== New Game =====");
    console.log ("[BOT] Game ID:" + _0x3493xe["game_id"]);
    currentGameID = _0x3493xe["game_id"];
    if (coolingDown) {
        if (lossStreak == 0) {
            coolingDown = false
        } else {
            lossStreak--;
            console.log ("[BOT] Cooling down! Games remaining: " + lossStreak);
            return;
        }
    };
    if (!firstGame) {
        console.log ("[STATS] Session profit: " + ((engine.getBalance () - startingBalance) /
100).toFixed (2) + " bits");
        console.log ("[STATS] Profit percentage: " + (((engine.getBalance () / startingBalance) - 1) *
100).toFixed (2) + "%");
    };
    if (engine.lastGamePlay () == "LOST" && !firstGame) {
        lossStreak++;
        var _0x3493xf = 0;
        var _0x3493x10 = currentBet;
        while (_0x3493x10 >= baseSatoshi) {
            _0x3493xf += _0x3493x10;
            _0x3493x10 /= 4;
        };
        if (lossStreak > streakSecurity) {
            coolingDown = true;
            return;
        };
        currentBet *= 4;
        currentMultiplier = 1 + (_0x3493xf / currentBet);
    } else {
        lossStreak = 0;
        if (variableBase) {
            var _0x3493x11 = 100;
            for (i = 0; i < streakSecurity; i++) {
                _0x3493x11 += (100 * Math.pow (4, (i + 1)))
            };
            newBaseBet = Math.min (Math.max (1, Math.floor (engine.getBalance () / _0x3493x11)),
maximumBet * 100);
            newBaseSatoshi = newBaseBet * 100;
            if ((newBaseBet != baseBet) || (newBaseBet == 1)) {
                console.log ("[BOT] Variable mode has changed base bet to: " + newBaseBet + " bits");
                baseBet = newBaseBet;
                baseSatoshi = newBaseSatoshi;
            };
        };
        currentBet = baseSatoshi;
    }
}

```

```

        currentMultiplier = baseMultiplier;
    };
    console.log ("[BOT] Betting " + (currentBet / 100) + " bits, cashing out at " + currentMultiplier +
"x");
    firstGame = false;
    if (currentBet <= engine.getBalance ()) {
        if (currentBet > (maximumBet * 100)) {
            console.warn ("[WARNING] Bet size exceeds maximum bet, lowering bet to " +
(maximumBet * 100) + " bits");
            currentBet = maximumBet;
        };
        engine.placeBet (currentBet, Math.round (currentMultiplier * 100), false);
    } else {
        if (engine.getBalance () < 100) {
            console.error ("[BOT] Insufficient funds to do anything... stopping");
            engine.stop ();
        } else {
            console.warn ("[BOT] Insufficient funds to bet " + (currentBet / 100) + " bits.");
            console.warn ("[BOT] Resetting to 1 bit basebet");
            baseBet = 1;
            baseSatoshi = 100;
        }
    };
});
engine.on ("game_started", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game id" + currentGameID + " has started!")
    }
});
engine.on ("cached_out", function (_0x3493x12) {
    if (_0x3493x12["username"] == engine.getUsername ()) {
        console.log ("[BOT] Successfully cashed out at " + (_0x3493x12["stopped_at"] / 100) + "x")
    }
});
engine.on ("game_crash", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game crashed at " + (_0x3493x12["game_crash"] / 100) + "x")
    }
});

```

And I did not understand what you wanted to change on this script.

exploit.c - January 3, 2016, 2:30 pm

(Jan 3, 2016 01:26 PM)Sakuto Wrote: [CITE](#) "I have completely done" + Provide obfuscated script.
Hu, they are a problem here. And your username is not really Safixk.

Spoiler [\(Click to View\)](#)

```

Code var baseBet = 10;
var baseMultiplier = 1.13;
var variableBase = true;
var streakSecurity = 4;
var maximumBet = 999999;
var baseSatoshi = baseBet * 100;

```

```

var currentBet = baseSatoshi;
var currentMultiplier = baseMultiplier;
var currentGameID = -1;
var firstGame = true;
var lossStreak = 0;
var coolingDown = false;
console.log ("===== Safixk's BustaBit Bot =====");
console.log ("USERNAME: " + engine.getUsername ());
console.log ("STARTING BALANCE: " + (engine.getBalance () / 100).toFixed (2) + " bits");
var startingBalance = engine.getBalance ();
if (variableBase) {
    console.warn ("[WARN] THIS BOT IS EARLY ALPHA. Resillient to " + streakSecurity + "-loss
streaks.")
};
engine.on ("game_starting", function (_0x3493xe) {
    console.log ("===== New Game =====");
    console.log ("[BOT] Game ID:" + _0x3493xe["game_id"]);
    currentGameID = _0x3493xe["game_id"];
    if (coolingDown) {
        if (lossStreak == 0) {
            coolingDown = false
        } else {
            lossStreak--;
            console.log ("[BOT] Cooling down! Games remaining: " + lossStreak);
            return;
        }
    };
    if (!firstGame) {
        console.log ("[STATS] Session profit: " + ((engine.getBalance () - startingBalance) /
100).toFixed (2) + " bits");
        console.log ("[STATS] Profit percentage: " + (((engine.getBalance () / startingBalance) - 1) *
100).toFixed (2) + "%");
    };
    if (engine.lastGamePlay () == "LOST" && !firstGame) {
        lossStreak++;
        var _0x3493xf = 0;
        var _0x3493x10 = currentBet;
        while (_0x3493x10 >= baseSatoshi) {
            _0x3493xf += _0x3493x10;
            _0x3493x10 /= 4;
        };
        if (lossStreak > streakSecurity) {
            coolingDown = true;
            return;
        };
        currentBet *= 4;
        currentMultiplier = 1 + (_0x3493xf / currentBet);
    } else {
        lossStreak = 0;
        if (variableBase) {
            var _0x3493x11 = 100;
            for (i = 0; i < streakSecurity; i++) {
                _0x3493x11 += (100 * Math.pow (4, (i + 1)))
            }
        }
    }
}

```

```

    };
    newBaseBet = Math.min (Math.max (1, Math.floor (engine.getBalance () / _0x3493x11)),
maximumBet * 100);
    newBaseSatoshi = newBaseBet * 100;
    if ((newBaseBet != baseBet) || (newBaseBet == 1)) {
        console.log ("[BOT] Variable mode has changed base bet to: " + newBaseBet + " bits");
        baseBet = newBaseBet;
        baseSatoshi = newBaseSatoshi;
    };
};
currentBet = baseSatoshi;
currentMultiplier = baseMultiplier;
};
console.log ("[BOT] Betting " + (currentBet / 100) + " bits, cashing out at " + currentMultiplier +
"x");
firstGame = false;
if (currentBet <= engine.getBalance ()) {
    if (currentBet > (maximumBet * 100)) {
        console.warn ("[WARNING] Bet size exceeds maximum bet, lowering bet to " +
(maximumBet * 100) + " bits");
        currentBet = maximumBet;
    };
    engine.placeBet (currentBet, Math.round (currentMultiplier * 100), false);
} else {
    if (engine.getBalance () < 100) {
        console.error ("[BOT] Insufficient funds to do anything... stopping");
        engine.stop ();
    } else {
        console.warn ("[BOT] Insufficient funds to bet " + (currentBet / 100) + " bits.");
        console.warn ("[BOT] Resetting to 1 bit basebet");
        baseBet = 1;
        baseSatoshi = 100;
    }
};
});
engine.on ("game_started", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game id" + currentGameID + " has started!")
    }
});
engine.on ("cached_out", function (_0x3493x12) {
    if (_0x3493x12["username"] == engine.getUsername ()) {
        console.log ("[BOT] Successfully cashed out at " + (_0x3493x12["stopped_at"] / 100) + "x")
    }
});
engine.on ("game_crash", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game crashed at " + (_0x3493x12["game_crash"] / 100) + "x")
    }
});
});

```

And I did not understand what you wanted to change on this script.

From my understanding he wants it so when he loses, he wants to bet so he wins the money he

just lost back automatically, then go higher, bet, etc.

Deeg - January 3, 2016, 6:00 pm

(Jan 3, 2016 01:26 PM)Sakuto Wrote: [\[CITE\]](#) "I have completely done" + Provide obfuscated script. Hu, they are a problem here. And your username is not really Safixk.

Spoiler ([Click to View](#))

```
Code
var baseBet = 10;
var baseMultiplier = 1.13;
var variableBase = true;
var streakSecurity = 4;
var maximumBet = 999999;
var baseSatoshi = baseBet * 100;
var currentBet = baseSatoshi;
var currentMultiplier = baseMultiplier;
var currentGameID = -1;
var firstGame = true;
var lossStreak = 0;
var coolingDown = false;
console.log ("===== Safixk's BustaBit Bot =====");
console.log ("USERNAME: " + engine.getUsername ());
console.log ("STARTING BALANCE: " + (engine.getBalance () / 100).toFixed (2) + " bits");
var startingBalance = engine.getBalance ();
if (variableBase) {
    console.warn ("[WARN] THIS BOT IS EARLY ALPHA. Resilient to " + streakSecurity + "-loss streaks.");
};
engine.on ("game_starting", function (_0x3493xe) {
    console.log ("===== New Game =====");
    console.log ("[BOT] Game ID:" + _0x3493xe["game_id"]);
    currentGameID = _0x3493xe["game_id"];
    if (coolingDown) {
        if (lossStreak == 0) {
            coolingDown = false
        } else {
            lossStreak--;
            console.log ("[BOT] Cooling down! Games remaining: " + lossStreak);
            return;
        }
    };
    if (!firstGame) {
        console.log ("[STATS] Session profit: " + ((engine.getBalance () - startingBalance) / 100).toFixed (2) + " bits");
        console.log ("[STATS] Profit percentage: " + (((engine.getBalance () / startingBalance) - 1) * 100).toFixed (2) + "%");
    };
    if (engine.lastGamePlay () == "LOST" && !firstGame) {
        lossStreak++;
        var _0x3493xf = 0;
        var _0x3493x10 = currentBet;
        while (_0x3493x10 >= baseSatoshi) {
            _0x3493xf += _0x3493x10;
            _0x3493x10 /= 4;
        }
    }
}
```

```

};
if (lossStreak > streakSecurity) {
    coolingDown = true;
    return;
};
currentBet *= 4;
currentMultiplier = 1 + (_0x3493xf / currentBet);
} else {
    lossStreak = 0;
    if (variableBase) {
        var _0x3493x11 = 100;
        for (i = 0; i < streakSecurity; i++) {
            _0x3493x11 += (100 * Math.pow (4, (i + 1)))
        };
        newBaseBet = Math.min (Math.max (1, Math.floor (engine.getBalance () / _0x3493x11)),
maximumBet * 100);
        newBaseSatoshi = newBaseBet * 100;
        if ((newBaseBet != baseBet) || (newBaseBet == 1)) {
            console.log ("[BOT] Variable mode has changed base bet to: " + newBaseBet + " bits");
            baseBet = newBaseBet;
            baseSatoshi = newBaseSatoshi;
        };
    };
    currentBet = baseSatoshi;
    currentMultiplier = baseMultiplier;
};
console.log ("[BOT] Betting " + (currentBet / 100) + " bits, cashing out at " + currentMultiplier +
"x");
firstGame = false;
if (currentBet <= engine.getBalance ()) {
    if (currentBet > (maximumBet * 100)) {
        console.warn ("[WARNING] Bet size exceeds maximum bet, lowering bet to " +
(maximumBet * 100) + " bits");
        currentBet = maximumBet;
    };
    engine.placeBet (currentBet, Math.round (currentMultiplier * 100), false);
} else {
    if (engine.getBalance () < 100) {
        console.error ("[BOT] Insufficient funds to do anything... stopping");
        engine.stop ();
    } else {
        console.warn ("[BOT] Insufficient funds to bet " + (currentBet / 100) + " bits.");
        console.warn ("[BOT] Resetting to 1 bit basebet");
        baseBet = 1;
        baseSatoshi = 100;
    }
};
});
engine.on ("game_started", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game id" + currentGameID + " has started!")
    }
});
};

```

```

engine.on ("cached_out", function (_0x3493x12) {
    if (_0x3493x12["username"] == engine.getUsername ()) {
        console.log ("[BOT] Successfully cashed out at " + (_0x3493x12["stopped_at"] / 100) + "x")
    }
});
engine.on ("game_crash", function (_0x3493x12) {
    if (!firstGame) {
        console.log ("[BOT] Game crashed at " + (_0x3493x12["game_crash"] / 100) + "x")
    }
});

```

And I did not understand what you wanted to change on this script.

Thanks man. This script is a friend's script. I did not create this. And the person above was correct in saying that I want to regain the money I just lost back. I do not want it to go back to basebet. I want it to be like it would if it were not going to skip a round. Thank you

Sakuto - January 3, 2016, 6:05 pm

I'm not really into this kind of game, but will try to edit this tomorrow. So when you lose a game, you don't want to stop playing, you just want to bet the same amount again?

Deeg - January 3, 2016, 6:08 pm

When I lose a game I want it to skip the next game. So sit out a game then on the NEXT game after it switches I want it to make the bet so it regains the money it just lost on the round that was lost causing it to skip a round. I want it to do what it already does now but literally skip a round if it loses then continue betting. But when it loses make the bet so it regains the money it lost just after the round it skipped.

Sakuto - January 4, 2016, 6:20 am

I tried your script, and I think I understand now, let's say for example:

Game 1, Bet 1000 Win.
 Game 2, Bet 1000 Lose.
 Game 3, Skip
 Game 4, Bet 4000 Win (at 1.25)
 Game 5, Bet 1000 ?

Something like this?

Deeg - January 4, 2016, 8:38 am

(Jan 4, 2016 07:20 AM)Sakuto Wrote: [\[CITE\]](#) I tried your script, and I think I understand now, let's say for example:

Game 1, Bet 1000 Win.
 Game 2, Bet 1000 Lose.
 Game 3, Skip
 Game 4, Bet 4000 Win (at 1.25)
 Game 5, Bet 1000 ?

Something like this?

That is exactly what I want! Just something so after the game it skips it adjusts the bet to win back the money it just lost! Exactly!

Sakuto - January 4, 2016, 9:10 am

Here you go:

```
Code var baseBet = 10;
var baseMultiplier = 1.13;
var variableBase = true;
var streakSecurity = 4;
var maximumBet = 999999;
var baseSatoshi = baseBet * 100;
var currentBet = baseSatoshi;
var currentMultiplier = baseMultiplier;
var currentGameID = -1;
var firstGame = true;
var lossStreak = 0;
var coolingDown = false;
console.log ("===== Safixk's BustaBit Bot =====");
console.log ("USERNAME: " + engine.getUsername ());
console.log ("STARTING BALANCE: " + (engine.getBalance () / 100).toFixed (2) + " bits");
var startingBalance = engine.getBalance ();
```

```
if (variableBase) {
    console.warn ("[WARN] THIS BOT IS EARLY ALPHA. Resilient to " + streakSecurity + "-loss streaks.");
};
```

```
engine.on ("game_starting", function (_0x3493xe) {
    var shouldPlayGame = true;
```

```
    console.log ("===== New Game =====");
    console.log ("[BOT] Game ID:" + _0x3493xe["game_id"]);
    currentGameID = _0x3493xe["game_id"];
    if (coolingDown) {
        if (lossStreak == 0) {
            coolingDown = false
        } else {
            lossStreak--;
            console.log ("[BOT] Cooling down! Games remaining: " + lossStreak);
            return;
        }
    };
```

```
    if (!firstGame) {
        console.log ("[STATS] Session profit: " + ((engine.getBalance () - startingBalance) / 100).toFixed (2) + " bits");
        console.log ("[STATS] Profit percentage: " + (((engine.getBalance () / startingBalance) - 1) * 100).toFixed (2) + "%");
    };
```

```
    if (engine.lastGamePlay () == "LOST" && !firstGame) {
        lossStreak++;
        var _0x3493xf = 0;
        var _0x3493x10 = currentBet;
        while (_0x3493x10 >= baseSatoshi) {
            _0x3493xf += _0x3493x10;
```

```

    _0x3493x10 /= 4;
};
if (lossStreak > streakSecurity) {
    coolingDown = true;
    return;
};
currentBet *= 4;
currentMultiplier = 1 + (_0x3493xf / currentBet);

shouldPlayGame = false;
} else {
    lossStreak = 0;
    if (variableBase) {
        var _0x3493x11 = 100;
        for (i = 0; i < streakSecurity; i++) {
            _0x3493x11 += (100 * Math.pow (4, (i + 1)))
        };
        newBaseBet = Math.min (Math.max (1, Math.floor (engine.getBalance () / _0x3493x11)),
maximumBet * 100);
        newBaseSatoshi = newBaseBet * 100;
        if ((newBaseBet != baseBet) || (newBaseBet == 1)) {
            console.log ("[BOT] Variable mode has changed base bet to: " + newBaseBet + " bits");
            baseBet = newBaseBet;
            baseSatoshi = newBaseSatoshi;
        };
    };
    currentBet = baseSatoshi;
    currentMultiplier = baseMultiplier;
};

console.log ("[BOT] Betting " + (currentBet / 100) + " bits, cashing out at " + currentMultiplier +
"x");

firstGame = false;

if (currentBet <= engine.getBalance () && shouldPlayGame) {
    if (currentBet > (maximumBet * 100)) {
        console.warn ("[WARNING] Bet size exceeds maximum bet, lowering bet to " +
(maximumBet * 100) + " bits");
        currentBet = maximumBet;
    };
    engine.placeBet (currentBet, Math.round (currentMultiplier * 100), false);
} else {
    if (engine.getBalance () < 100) {
        console.error ("[BOT] Insufficient funds to do anything... stopping");
        engine.stop ();
    } else {
        console.warn ("[BOT] Insufficient funds to bet " + (currentBet / 100) + " bits.");
        console.warn ("[BOT] Resetting to 1 bit basebet");
        baseBet = 1;
        baseSatoshi = 100;
    }
};
};

```

```
});  
  
engine.on ("game_started", function (_0x3493x12) {  
    if (!firstGame) {  
        console.log ("[BOT] Game id" + currentGameID + " has started!")  
    }  
});  
  
engine.on ("cached_out", function (_0x3493x12) {  
    if (_0x3493x12["username"] == engine.getUsername ()) {  
        console.log ("[BOT] Successfully cashed out at " + (_0x3493x12["stopped_at"] / 100) + "x")  
    }  
});  
  
engine.on ("game_crash", function (_0x3493x12) {  
    if (!firstGame) {  
        console.log ("[BOT] Game crashed at " + (_0x3493x12["game_crash"] / 100) + "x")  
    }  
});  
My BTC address: 1PVSNSmKjSDNdAQkqHAHFs4VkqwjXmvV62 :B
```