Ultra NIM

Rules

- Standard (Remove last element)

- Misere (Force opponent to remove last element)

- (+?) Remove N elements from all stacks

Features:

- LCD display (stacks, player, (+?) Pool

- Buttons (Selection)

<IO Design>------------------------------------------------------------------------------------------

|No UP| No DOWN | SELECT |

<Pile Select>

SELECT - Enters selection

No DOWN - Reduces current selection, cycles

No UP - Incremenets current selection, cycles 1, 2... All, 1 ...

<Take Amount Select>

SELECT - Enter selection

No DOWN - Reduces current selection, cycles

No UP - Increments current selection, cycles 1, 2... Max

<Start Game>

| 1 | 2 | 3 |

3 - GameMode3 (UltraNIM)

2 - GameMode2 (Misere)

1 - GameMode1 (Classic)

<Game Config> -> <Stack Config> -> <Piles Config>

SELECT - Confirm Selection, move to next screen

No UP - Increment Number

No DOWN - Decrement Number

<System Design>------------------------------------------------------------------------------------------

void game()

enum gameSet {

ULTRANIM,

MISERE,

CLASSIC

};

void startGame(gameSet setting);

void gameConfig(int stackAmt, int stackSizeMax)

void game() {

//game start

while(true) {

alloc gameSet setting;

alloc stackAmt, stackSizeMax;

startGame(setting);

gameConfig(stack­Amt, stackSizeMax);

playGame(setting, stackAmt, stackSizeMax);

free(everything);

}

}

void playGame(gameSet setting, int stackAmt, int stackSizeMax) {

alloc int stackArray[stackAmt];

setStacks(stackArray);

bool gameNotEnded;

while (gameNotEnded) {

runGame;

}­

}

void setStacks(int stackArray[], int stackSizeMax) {

//goes through array and set each stack to random 1 <= value <= stackSizeMax

}

void printToScreen(char \* textToDisplay);

int getButtonInput();