20(W )\*15(H) 1=32px

stage1

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0

1. 0 0 0 0 0 0 0 0 4 4 4 4 0 0 0 0 0 0 0 0 0 //画面切り替え
2. 0 2 2 2 2 2 3 0 //3=Treasure
3. 0 2 2 2 2 2 2 0
4. 0 2 2 2 2 2 2 0
5. 0 0
6. 0 0
7. 0 0
8. 0 2 2 2 2 2 2 2 2 2 0
9. 0 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0
10. 0 2 2 2 2 2 2 2 2 2 0
11. 0 0
12. 0 0
13. 0 S 0
14. 0 0
15. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9

stage2

stage\_x = 20 //stage width(\*interval)

stage\_y = 15 //stage height(\*interval)

interval = 32 //pixel

chara\_x = 17 //charaのx座標(初期:17)

chara\_y = 12 //charaのy座標(初期:12)

current\_stage; //現在のステージ配列

関数

//ステージを描画

drawStage( ctx, stage ){  
//描画処理  
}

//次のステージまでの遅延の際表示する黒画面

drawBlack( ctx );

document.onkeyboard = keydown;

//キー操作

function keydown(){

if enter(){  
上が3 or横が３なら{

宝箱open;

stage [0][9-12] --> 4;

}

}

if up(){  
上が1or2なら{ y--;(up) }

上が４なら {

current\_stage = stage2;

chara\_x = 9;

chara\_y = 18;  
}

}

if down(){  
下が1or2なら{ y++;(down) }

下が4なら {

current\_stage = stage;

chara\_y = 1; //xは同じ  
}

}

if right(){  
右が1or2なら { x++; }

右が3なら{

宝箱open;

stage [0][9-12] --> 4;

}

}

if left(){  
左が1or2なら { x--; }

}

}

Boss

背景画像

C B

(4,9) (15,9)

ターン制

HP 50 100

スキルゲージ (init)1->2->3

------------------------------------------------------------

戦う　　10 8

ためる　スキル +1

回復　 +30

スキル 50

function charaData(hp, power, skill){

this.hp = hp;

this.power = power;

this .skill = skill;  
}

function bossData(hp, power){

this.hp = hp;

this.power = power;  
}

//init

var chara\_data= new charaData(50, 10, 1);

var boss\_data = new bossData(100,8);

chara画像描画

boss画像描画

chara\_data.hp描画

boss\_data.hp描画

ゲージ描画

選択画面描画

while( boss\_data.hp > 0)

攻撃選択{  
　　boss\_HP -= chara\_data.hp;  
}

ためる選択{

if( chara\_data.skill < 3 ){  
 chara\_data.skill++;

ゲージ描画

}

回復選択{  
chara\_data.hp +=30;  
}

スキル選択{

if( chara\_data.skill == 3 )  
chara\_data.hp +=30;  
}

敵攻撃

chara\_data.hp -= boss\_data.power;