



Client: HKM Lab

Project name: Monopoly on Flash

Scope: Produce a Flash game based off a Taiwan Monopoly Games such as Sorich 2, 3, 4 or ドラえもん, Doraemon Monopoly.

Requirements:

- Map selections
- Custom characters, color, items
- AI players
- 3-4 building blocks
- Simple RPG battle scene
- Simple debug console

Technology: HTML, Actionscript, simple Flash animations, photoshop

Author: Heskeyo Kam

Project amended in 2005

```
// Action script...

// [Action in Frame 1]
stop ();

// [Action in Frame 2]
var my_xml = new XML();
my_xml.ignoreWhite = true;
var debug_;
var XMLLED = false;
my_xml.onLoad = function (success)
{
    if (success)
    {
        if (my_xml.status == 0)
        {
            x1 = my_xml.firstChild.childNodes;
            debug_ = "XML was loaded and parsed successfully";
            if (my_xml.loaded)
            {
                XMLLED = true;
            } // end if
        }
        else
        {
            debug_ = "XML was loaded successfully, but was unable to be parsed.";
        } // end else if
        var _loc1;
        switch (my_xml.status)
        {
            case 0:
            {
                _loc1 = "No error; parse was completed successfully.";
                break;
            }
            case -2:
            {
                _loc1 = "A CDATA section was not properly terminated.";
                break;
            }
            case -3:
            {
                _loc1 = "The XML declaration was not properly terminated.";
                break;
            }
            case -4:
            {
                _loc1 = "The DOCTYPE declaration was not properly terminated.";
                break;
            }
            case -5:
            {
```

```
        _loc1 = "A comment was not properly terminated.";
        break;
    }
    case -6:
    {
        _loc1 = "An XML element was malformed.";
        break;
    }
    case -7:
    {
        _loc1 = "Out of memory.";
        break;
    }
    case -8:
    {
        _loc1 = "An attribute value was not properly terminated.";
        break;
    }
    case -9:
    {
        _loc1 = "A start-tag was not matched with an end-tag.";
        break;
    }
    case -10:
    {
        _loc1 = "An end-tag was encountered without a matching start-tag.";
        break;
    }
    default:
    {
        _loc1 = "An unknown error has occurred.";
        break;
    }
} // End of switch
debug_ = "status: " + my_xml.status + " (" + _loc1 + ")";
delete my_xml;
}
else
{
    debug_ = "Unable to load/parse XML. (status: " + my_xml.status + ")";
} // end else if
};

// [Action in Frame 5]
function randomnumbers(lowest, highest, count, unique)
{
    var _loc5 = new Array();
    if (unique && count <= highest - lowest)
    {
        var _loc1 = new Array();
        for (var _loc3 = lowest; _loc3 <= highest; ++_loc3)
        {
```

```

        _loc1.push(_loc3);
    } // end of for
    for (var _loc3 = 1; _loc3 <= count; ++_loc3)
    {
        var _loc2 = Math.floor(Math.random() * _loc1.length);
        _loc5.push(_loc1[_loc2]);
        _loc1.splice(_loc2, 1);
    } // end of for
}
else
{
    for (var _loc3 = 1; _loc3 <= count; ++_loc3)
    {
        _loc5.push(lowest + Math.floor(Math.random() * (highest - lowest)));
    } // end of for
} // end else if
return (_loc5);
} // End of the function
function RRRandom(minVal, maxVal, nTimes)
{
    var _loc3 = new Array(nTimes - 1);
    for (var _loc1 = 0; _loc1 <= nTimes - 1; ++_loc1)
    {
        _loc3[_loc1] = minVal + Math.floor(Math.random() * (maxVal + 1 - minVal));
        for (var _loc2 = 0; _loc2 <= _loc1 - 1; ++_loc2)
        {
            if (_loc3[_loc1] == _loc3[_loc2])
            {
                _loc3[_loc1] = "";
                --_loc1;
            } // end if
        } // end of for
    } // end of for
    return (_loc3);
} // End of the function
function formatDecimals(num, digits)
{
    if (digits <= 0)
    {
        return (Math.round(num));
    } // end if
    var _loc4 = Math.pow(10, digits);
    var _loc2 = String(Math.round(num * _loc4) / _loc4);
    if (_loc2.indexOf(".") == -1)
    {
        _loc2 = _loc2 + ".0";
    } // end if
    var _loc6 = _loc2.split(".");
    var _loc3 = digits - _loc6[1].length;
    for (var _loc1 = 1; _loc1 <= _loc3; ++_loc1)
    {
        _loc2 = _loc2 + "0";
    }
}

```

```

    } // end of for
    return (_loc2);
} // End of the function
function dec2(num)
{
    var _loc1 = String(Math.round(num * 100) / 100);
    var _loc4 = _loc1.indexOf(".");
    if (_loc4 == -1)
    {
        _loc1 = _loc1 + ".0";
    } // end if
    var _loc3 = _loc1.split(".");
    var _loc2 = 2 - _loc3[1].length;
    for (i = 1; i <= _loc2; i++)
    {
        _loc1 = _loc1 + "0";
    } // end of for
    return (Number(_loc1));
} // End of the function
function sortN(arr)
{
    var _loc2 = new Array();
    _loc2[0] = arr[0];
    var _loc3;
    for (var _loc5 = 1; _loc5 < arr.length; ++_loc5)
    {
        _loc3 = arr[_loc5];
        var _loc4 = false;
        for (var _loc1 = 0; _loc1 < _loc2.length; ++_loc1)
        {
            if (_loc3 > _loc2[_loc1] && _loc3 < _loc2[_loc1 + 1] || _loc3 == _loc2[_loc1])
            {
                _loc2.splice(_loc1 + 1, 0, _loc3);
                _loc4 = true;
                break;
            } // end if
        } // end of for
        if (!_loc4 && _loc3 < _loc2[0])
        {
            _loc2.unshift(_loc3);
            continue;
        } // end if
        if (!_loc4 && _loc3 > _loc2[_loc2.length - 1])
        {
            _loc2.push(_loc3);
        } // end if
    } // end of for
    return (_loc2);
} // End of the function
function arrJ2gether(x1, x2)
{
    if (x1.length >= x2.length)

```

```

{
    var _loc5 = x1.length;
}
else
{
    _loc5 = x2.length;
} // end else if
var _loc2 = [];
for (var _loc1 = 0; _loc1 < _loc5; ++_loc1)
{
    _loc2.push(x1[_loc1], x2[_loc1]);
} // end of for
return (_loc2);
} // End of the function
function math1(x)
{
    return (Math.sin(x * x) * x / 10);
} // End of the function
function math2(x)
{
    return (x * x / 50);
} // End of the function
function math3(x)
{
    if (actionTerminated)
    {
        return (x * x / 50);
    }
    else
    {
        return (0);
    } // end else if
} // End of the function
function GoSart(x)
{
    placement = [];
    for (var _loc2 = 0; _loc2 < x; ++_loc2)
    {
        ++pln[p][v_nowStep];
        var _loc1 = pln[p][v_nowStep];
        var _loc3 = oostep.length;
        _loc1 = _loc1 % _loc3;
        pln[p][v_nowStep] = _loc1;
        var _loc5 = oostep[_loc1][0];
        var _loc4 = oostep[_loc1][1];
        placement.push([_loc5, _loc4, _loc1]);
    } // end of for
    c[pln[p][v_name]].onEnterFrame = objrun;
} // End of the function
function upto()
{
    var _loc1 = placement[0][2];

```

```

if (placement.length > 1)
{
    if (obuilding[_loc1][0][5] == true)
    {
        temporary = true;
        eventToHappen(_loc1);
        c[pIn[p][0]].onEnterFrame = null;
    } // end if
    placement.shift();
}
else
{
    temporary = false;
    c[pIn[p][0]].onEnterFrame = null;
    eventToHappen(_loc1);
} // end else if
} // End of the function
function focus()
{
    var _loc4 = pIn[p][v_name];
    var _loc8 = c[_loc4]._x * _global.startScale / 100;
    var _loc7 = c[_loc4]._y * _global.startScale / 100;
    var _loc6 = -_loc8 - c._x + 275;
    var _loc5 = -_loc7 - c._y + 200;
    var _loc2 = Math.sqrt(Math.pow(_loc5, 2) + Math.pow(_loc6, 2));
    var _loc3 = Math.atan2(_loc5, _loc6);
    c._x = c._x + Math.cos(_loc3) * _loc2 * 0.200000;
    c._y = c._y + Math.sin(_loc3) * _loc2 * 0.200000;
} // End of the function
function dice(x)
{
    switch (x)
    {
        case "on":
        {
            dicnum = random(6);
            interval = setInterval(function ()
            {
                ++dicnum;
                dicnum = dicnum % 6;
                diceout = dicnum + 1;
            }, 50);
            break;
        }
        case "pick":
        {
            clearInterval(interval);
            break;
        }
    } // End of switch
} // End of the function
function playerturn(playNO)

```

```

{
  playNO = playNO % pln.length;
  this.go._visible = false;
  this.bgwhite.gotoAndPlay(1);
  la = "";
  dice("on");
  showTextInfoOfplayer(playNO);
  var _loc3 = pln[playNO][v_AIplay];
  if (_loc3)
  {
    var pick = setInterval(function ()
    {
      sfx("e1");
      dice("pick");
      GoSart(diceout);
      clearInterval(pick);
    }, 1000 + random(1000));
  }
  else
  {
    this.go._visible = true;
  } // end else if
  p = playNO;
} // End of the function
function map2draw()
{
  var _loc18 = 5;
  var _loc19 = 14;
  var _loc3 = 0;
  var _loc17 = 0;
  for (var _loc2 in c)
  {
    if (typeof(c[_loc2]) == "movieclip")
    {
      if (_loc2.substr(0, 2) == "ws")
      {
        c.attachMovie("placec", "proper" + _loc3, _loc3 + 103);
        c["proper" + _loc3]._x = c["ws" + _loc3]._x;
        c["proper" + _loc3]._y = c["ws" + _loc3]._y;
        ++_loc3;
        continue;
      } // end if
      if (_loc2.substr(0, 1) == "w")
      {
        ++_loc17;
        continue;
      } // end if
      break;
      trace ("error");
    } // end if
  } // end of for...in
  for (var _loc2 = 0; _loc2 < _loc17; ++_loc2)

```



```

{
  oostep.push([c["w" + _loc2]._x, c["w" + _loc2]._y]);
} // end of for
for (var _loc2 = 0; _loc2 < _loc3; ++_loc2)
{
  if (_loc18 === _loc2)
  {
    c["proper" + _loc2].gotoAndStop("special");
    c["proper" + _loc2].attachMovie("shop0", "shop", 1);
    var _loc6 = ["Shop_ban_" + _loc2, c["ws" + _loc2]._x, c["ws" + _loc2]._y, "_", "_", true];
    var _loc14 = random(100) + 100;
    var _loc13 = random(1000) + 100;
    var _loc12 = random(5000) + 100;
    obulding.push([_loc6, _loc14, _loc13, _loc12]);
    continue;
  } // end if
  if (_loc19 === _loc2)
  {
    c["proper" + _loc2].gotoAndStop("special");
    c["proper" + _loc2].attachMovie("shop1", "shop", 1);
    _loc6 = ["Shop_wea_" + _loc2, c["ws" + _loc2]._x, c["ws" + _loc2]._y, "_", "_", false];
    _loc14 = random(100) + 100;
    _loc13 = random(1000) + 100;
    _loc12 = random(5000) + 100;
    obulding.push([_loc6, _loc14, _loc13, _loc12]);
    continue;
  } // end if
  _loc6 = ["proper" + _loc2, c["ws" + _loc2]._x, c["ws" + _loc2]._y];
  var _loc8 = [false, 0, false];
  _loc14 = random(100) + 100;
  _loc13 = random(1000) + 100;
  _loc12 = random(5000) + 100;
  var _loc11 = random(6000) + 100;
  var _loc10 = random(7000) + 100;
  var _loc9 = random(10000) + 100;
  var _loc7 = random(30000) + 100;
  var _loc5 = random(50000) + 100;
  var _loc4 = random(100000) + 100;
  var _loc15 = random(1000000) + 100;
  var _loc16 = _loc6.concat(_loc8);
  obulding.push([_loc16, _loc14, _loc13, _loc12, _loc11, _loc10, _loc9, _loc7, _loc5, _loc4,
    _loc15]);
} // end of for
for (var _loc2 = 0; _loc2 < pln.length; ++_loc2)
{
  c.attachMovie("CA" + _loc2, pln[_loc2][0], 7000 - _loc2);
  this.c[pln[_loc2][v_name]]._x = oostep[0][0];
  this.c[pln[_loc2][v_name]]._y = oostep[0][1];
  this.c[pln[_loc2][v_name]]._xscale = this.c[pln[_loc2][v_name]]._yscale = 50;
} // end of for
} // End of the function
function workingCompleted()

```

```

{
  var _loc2 = function ()
  {
    _global.mmu.stop();
    _global.mmc.stop();
    setbgmVol(_global.fullvol);
    if (bank._visible == true)
    {
      bank.gotoAndStop("empty");
      bank._visible = false;
    } // end if
    if (paper._visible == true)
    {
      paper.gotoAndStop("empty");
      paper._visible = false;
    } // end if
  };
  if (temporary)
  {
    _loc2();
    c[p1n[p][0]].onEnterFrame = objrun;
  }
  else
  {
    _loc2();
    ++days;
    playerturn(days);
  } // end else if
} // End of the function
function upgrateHome()
{
  var _loc1 = obuilding[p1n[p][4]];
  var _loc3 = _loc1[0][4];
  if (p1n[p][1] >= _loc1[_loc3 + 1])
  {
    for (var _loc2 in c[_loc1[0][0]])
    {
      if (typeof(_loc2) == "movieclip" && _loc2 != "building")
      {
        c[_loc1[0][0]][_loc2].removeMovieClip();
      } // end if
    } // end of for...in
    c[_loc1[0][0]].attachMovie("p1" + Number(_loc3 + 1), "building", 200 + p);
    c[_loc1[0][0]].fg._x = 4;
    c[_loc1[0][0]].fg._y = 5;
    c[_loc1[0][0]].fg._yscale = c[_loc1[0][0]].fg._xscale = 75;
    p1n[p][1] = p1n[p][1] - _loc1[_loc3 + 1];
    ++_loc1[0][4];
    showTextInfoOfplayer(p);
  } // end if
} // End of the function
function payForRent(mclose)

```

```

{
    var _loc2 = obuilding[pIn[p][4]];
    if (pIn[p][1] >= 0)
    {
        pIn[p][1] = pIn[p][1] - rent;
        for (var _loc1 = 0; _loc1 < pIn.length; ++_loc1)
        {
            if (pIn[_loc1][v_name] == _loc2[0][v_hp])
            {
                pIn[_loc1][v_cash] = pIn[_loc1][v_cash] + rent;
            } // end if
        } // end of for
    }
    else
    {
        for (var _loc1 = 0; _loc1 < pIn.length; ++_loc1)
        {
            if (pIn[_loc1][v_name] == _loc2[0][v_hp])
            {
                pIn[_loc1][v_cash] = pIn[_loc1][v_cash] + rent;
            } // end if
        } // end of for
        pIn[p][2] = pIn[p][2] - rent;
    } // end else if
    showTextInfoOfplayer(p);
    paper._visible = false;
    if (mclose)
    {
        workingCompleted();
    } // end if
} // End of the function
function fightForRent()
{
    var _loc1 = obuilding[pIn[p][4]];
} // End of the function
function buyhome(mclose)
{
    var _loc1 = obuilding[pIn[p][4]];
    var _loc3 = pIn[p][1];
    var _loc2 = _loc1[1] * 10;
    if (_loc3 >= _loc2)
    {
        c[_loc1[0][0]].gotoAndStop("owed");
        c[_loc1[0][0]].attachMovie("flag" + p, "fg", 100);
        _loc1[0][3] = pIn[p][0];
        pIn[p][1] = pIn[p][1] - Math.floor(_loc2);
        paper._visible = false;
        if (mclose)
        {
            workingCompleted();
        } // end if
        showTextInfoOfplayer(p);
    }
}

```

```

}
else
{
    paper.notok._visible = true;
} // end else if
} // End of the function
function showTextInfoOfplayer(player)
{
    playerName.text = pln[player][v_name];
    if (pln[player][v_cash] > 0)
    {
        Money.text = "Cash: " + pln[player][v_cash] + "G";
        Money.setTextFormat(tx_3);
    }
    else
    {
        Money.text = "Cash: " + pln[player][v_cash] + "G";
        Money.setTextFormat(tx_4);
    } // end else if
    Deposit.text = "Deposit: " + pln[player][v_bankVar] + "G";
} // End of the function
function DE_display()
{
    return (Math.floor(pln[p][2]));
} // End of the function
function CA_display()
{
    return (Math.floor(pln[p][1]));
} // End of the function
function ca_percent(e)
{
    var _loc1 = pln[p][1];
    return (Math.floor(e / 100 * Number(_loc1)));
} // End of the function
function de_percent(e)
{
    var _loc1 = pln[p][2];
    return (Math.floor(e / 100 * Number(_loc1)));
} // End of the function
function depositM(e)
{
    if (e < 1)
    {
    }
    else
    {
        pln[p][1] = pln[p][1] - Math.floor(e);
        pln[p][2] = pln[p][2] + Math.floor(e);
        showTextInfoOfplayer(p);
    } // end else if
} // End of the function
function depositG(e)

```

```

{
    if (e < 1)
    {
    }
    else
    {
        pln[p][1] = pln[p][1] + Math.floor(e);
        pln[p][2] = pln[p][2] - Math.floor(e);
        showTextInfoOfplayer(p);
    } // end else if
} // End of the function
function sfx(x)
{
    if (sndearable)
    {
        _global[x].start(0, 1);
    } // end if
} // End of the function
function setbgmVol(vol)
{
    _global.e3.setVolume(vol);
} // End of the function
function copymovieclipactor(how)
{
    if (how == "on")
    {
        this.attachMovie("CA" + p, "displayedpeople", 700);
        this.displayedpeople._xscale = this.displayedpeople._yscale = 800;
        this.displayedpeople._x = 400;
        this.displayedpeople._y = 250;
        this.panelcontrol.areatee.text = "Name: " + pln[p][v_name] + "\nCash: " + pln[p][v_cash] + "G
        \nDeposit: " + pln[p][v_bankVar] + "G";
    }
    else
    {
        this.displayedpeople.removeMovieClip();
        this.panelcontrol.areatee.text = "";
    } // end else if
} // End of the function
Array.prototype.Contains = function (value)
{
    var _loc3 = 0;
    var _loc2 = 0;
    for (var _loc2 = 0; _loc2 < this.length; ++_loc2)
    {
        if (this[_loc2] == value)
        {
            ++_loc3;
        } // end if
    } // end of for
    return (_loc3);
};

```

```

Array.prototype.shuffle = function ()
{
    var _loc2 = this.slice();
    var _loc6 = _loc2.length;
    var _loc5 = new Array();
    for (var _loc4 = 0; _loc4 < _loc6; ++_loc4)
    {
        var _loc3 = random(_loc2.length);
        _loc5[_loc5.length] = _loc2[_loc3];
        _loc2[_loc3] = _loc2[_loc2.length - 1];
        _loc2.pop();
    } // end of for
    return (_loc5);
};
Array.prototype.statistics = function ()
{
    function sortNumbers(element1, element2)
    {
        return (element1 - element2);
    } // End of the function
    var _loc9;
    var _loc8 = 0;
    var _loc6;
    var _loc10;
    var _loc7;
    var _loc5;
    var _loc4;
    this.sort(sortNumbers);
    for (var _loc3 = 0; _loc3 < this.length - 1; ++_loc3)
    {
        _loc8 = _loc8 + this[_loc3];
        _loc6 = _loc8 / this.length;
        _loc6 = Math.round(_loc6);
    } // end of for
    if (this.length / 2 != Math.floor(this.length / 2))
    {
        _loc9 = "odd";
    }
    else
    {
        _loc9 = "even";
    } // end else if
    if (_loc9 == "odd")
    {
        _loc10 = Math.round(this[(this.length - 1) / 2]);
    }
    else if (_loc9 == "even")
    {
        _loc10 = (this[this.length / 2] + this[this.length / 2 - 1]) / 2;
    } // end else if
    _loc4 = 1;
    _loc5 = 0;

```

```

for (var _loc3 = 0; _loc3 <= this.length - 1; ++_loc3)
{
    for (var _loc2 = _loc3 + 1; _loc2 <= this.length - 1; ++_loc2)
    {
        if (this[_loc3] == this[_loc2])
        {
            _loc4 = _loc4 + 1;
            if (_loc4 > _loc5)
            {
                _loc5 = _loc4;
                _loc4 = 1;
                _loc7 = this[_loc3];
            } // end if
        } // end if
    } // end of for
} // end of for
trace ("mean = " + _loc6);
trace ("median = " + _loc10);
trace ("mode = " + _loc7);
};
getMax = function (tmpArray)
{
    var _loc3 = 0;
    for (var _loc1 = 0; _loc1 < tmpArray.length; ++_loc1)
    {
        if (tmpArray[_loc1] > _loc3)
        {
            _loc3 = tmpArray[_loc1];
        } // end if
    } // end of for
    return (_loc3);
};
getMin = function (tmpArray)
{
    var _loc3 = getMax(tmpArray);
    for (var _loc1 = 0; _loc1 < tmpArray.length; ++_loc1)
    {
        if (tmpArray[_loc1] < _loc3)
        {
            _loc3 = tmpArray[_loc1];
        } // end if
    } // end of for
    return (_loc3);
};
Array.prototype.bubble = function ()
{
    var _loc6 = new Array();
    var _loc5 = this.length;
    var _loc4 = 0;
    for (var _loc3 = _loc5 - 1; _loc3 > 0; --_loc3)
    {
        for (var _loc2 = 0; _loc2 < _loc3; ++_loc2)

```

```

    {
        if (this[_loc2] > this[_loc2 + 1])
        {
            _loc4 = this[_loc2];
            this[_loc2] = this[_loc2 + 1];
            this[_loc2 + 1] = _loc4;
        } // end if
    } // end of for
} // end of for
return (this);
};
MovieClip.prototype.objrun = function (x)
{
    distx = placement[0][0] - this._x;
    disty = placement[0][1] - this._y;
    var _loc4 = Math.sqrt(Math.pow(disty, 2) + Math.pow(distx, 2));
    var _loc3 = Math.atan2(disty, distx);
    var _loc2 = _loc3 * 180 / 3.141593;
    if (_loc2 > 360)
    {
        _loc2 = _loc2 - 360;
    } // end if
    if (_loc2 < 0)
    {
        _loc2 = _loc2 + 360;
    } // end if
    this.telly._rotation = _loc2;
    if (_loc2 > 45 && _loc2 < 135)
    {
        this.gotoAndStop("M3");
    }
    else if (_loc2 > 135 && _loc2 < 225)
    {
        this.gotoAndStop("M2");
    }
    else if (_loc2 > 225 && _loc2 < 315)
    {
        this.gotoAndStop("M4");
    }
    else if (_loc2 > 315 || _loc2 < 45)
    {
        this.gotoAndStop("M1");
    }
    else
    {
        this.gotoAndStop("M1");
    } // end else if
    this._x = this._x + Math.cos(_loc3) * 2;
    this._y = this._y + Math.sin(_loc3) * 2;
    if (Math.floor(Math.abs(_loc4)) == 0)
    {
        upto();
    }
}

```



```

    } // end if
};
var tx_3 = new TextFormat();
tx_3.bold = false;
tx_3.color = 65280;
tx_3.size = 12;
tx_3.bullet = false;
var tx_4 = new TextFormat();
tx_4.bold = true;
tx_4.color = 16711680;
tx_4.size = 12;
tx_4.bullet = false;
this.onEnterFrame = function ()
{
    focus();
    gdp.text = pln[p];
    c._xscale = c._yscale = _global.startScale;
};
this.go.onRelease = function ()
{
    this._visible = false;
    sfx("e1");
    dice("pick");
    GoSart(diceout);
};
this.seag0.onRelease = function ()
{
    this._visible = false;
    panelcontrol.play();
};
var obulding = [];
var oostep = [];
var placement = [];
var v_name = 0;
var v_cash = 1;
var v_bankVar = 2;
var v_hp = 3;
var v_nowStep = 4;
var v_AIplay = 5;
var v_cards = 6;
var v_stockown = 7;
var v_mosterown = 8;
p11 = ["Heskeyo", 4000, 10000, 0, 0, false, [], [], []];
p12 = ["John", 4000, 10000, 0, 0, true, [], [], []];
p13 = ["Mans", 4000, 10000, 0, 0, true, [], [], []];
p14 = ["Yoki", 4000, 10000, 0, 0, true, [], [], []];
pln = [p11, p14];
var diceout = 0;
var dicnum = 0;
var interval = 0;
var p = 0;
var days = 0;

```

```

var actionTerminated = true;
map2draw();
paper._visible = false;
_global.startScale = 800;
var sndeable = true;
_global.fullvol = 100;
_global.halfvol = 25;
_global.e3.start(0, 9999);
var temporary;
playerturn(0);
stop ();
var rent = 0;
var tx_1 = new TextFormat();
tx_1.bold = true;
tx_1.color = 16777215;
tx_1.size = 12;
tx_1.bullet = false;
var tx_2 = new TextFormat();
tx_2.bold = true;
tx_2.color = 0;
tx_2.size = 20;
tx_2.bullet = false;
var buildupLv = function (lv, mo, need)
{
    setbgmVol(0);
    tellTarget("paper")
    {
        gotoAndStop("q3");
        setProperty("", _visible, true);
        canBeUpgrate_1 = true;
        canBeUpgrate_2 = true;
        canBeUpgrate_3 = true;
        canBeUpgrate_4 = true;
        od.text = "";
        od.text = od.text + ("If you wanna to extend you achitecture to " + lv + "LV for " + mo + "G\n");
        od.text = od.text + ("Appromixately that would use " + Math.floor(need / mo * 100) + "%\n");
        od.text = od.text + "Are you confident on building it up?";
        od.setTextFormat(tx_2);
        tellTarget("")
        {
        } // End of TellTarget
    } // End of TellTarget
};
var eventToHappen = function (now_step)
{
    var _loc18 = function ()
    {
        tellTarget("bank")
        {
            gotoAndStop("a");
            setProperty("", _visible, true);
            tellTarget("")

```

```

    {
    } // End of TellTarget
} // End of TellTarget
};
var _loc19 = function ()
{
    tellTarget("bank")
    {
        gotoAndStop("k0");
        setProperty("", _visible, true);
        tellTarget("")
        {
        } // End of TellTarget
    } // End of TellTarget
};
var _loc16 = function ()
{
    setbgmVol(_global.halfvol);
    tellTarget("paper")
    {
        gotoAndStop("q1");
        notok._visible = false;
        setProperty("", _visible, true);
        od.text = "";
        od.text = od.text + ("???" + _root.dec2(Number(o2 / nowmoney * 100)) + "% ??????????\n");
        od.text = od.text + ("?? " + placename + "\n");
        od.text = od.text + ("?? " + o2 + "G\n");
        od.text = od.text + "??? 100G\n";
        od.text = od.text + ("stage 1 " + o3 + "G\n");
        od.text = od.text + ("stage 2 " + o4 + "G\n");
        od.text = od.text + ("stage 3 " + o5 + "G\n");
        od.text = od.text + ("stage 4 " + o6 + "G\n");
        od.text = od.text + ("stage 5 " + o7 + "G\n");
        od.text = od.text + ("stage 6 " + o8 + "G\n");
        od.text = od.text + ("stage 7 " + o9 + "G\n");
        od.text = od.text + ("stage 8 " + o10 + "G\n");
        od.text = od.text + ("stage 9 " + o11 + "G\n");
        od.setTextFormat(tx_1);
        tellTarget("")
        {
        } // End of TellTarget
    } // End of TellTarget
};
var _loc14 = function ()
{
    _global.mmc.start(0, 30);
    setbgmVol(0);
    tellTarget("paper")
    {
        gotoAndStop("q2");
        setProperty("", _visible, true);
        od.text = "";

```

```

od.text = od.text + ("???? " + r + "G\n");
od.text = od.text + ("?????????" + _root.dec2(Number(r / nowmoney * 100)) + "%\n");
od.text = od.text + ("????? " + ownership + " " + r + "G");
od.setTextFormat(tx_2);
tellTarget("")
{
} // End of TellTarget
} // End of TellTarget
};
var _loc15 = function (x)
{
tellTarget("paper")
{
gotoAndPlay(mess);
setProperty("", _visible, true);
od.text = x;
od.setTextFormat(tx_1);
tellTarget("")
{
} // End of TellTarget
} // End of TellTarget
};
var _loc12 = pln[p][5];
var _loc3 = obulding[now_step];
var nowmoney = pln[p][1];
var _loc9 = pln[p][2];
var placename = String(_loc3[0][0]);
var _loc7 = placename.substring(5, 8);
var _loc5 = pln[p][0];
var ownership = _loc3[0][3];
var o2 = Number(parseInt(_loc3[1]) * 10);
var o3 = parseInt(_loc3[2]);
var o4 = parseInt(_loc3[3]);
var o5 = parseInt(_loc3[4]);
var o6 = parseInt(_loc3[5]);
var o7 = parseInt(_loc3[6]);
var o8 = parseInt(_loc3[7]);
var o9 = parseInt(_loc3[8]);
var o10 = parseInt(_loc3[9]);
var o11 = parseInt(_loc3[10]);
var _loc6 = parseInt(_loc3[0][4]);
var _loc8 = _loc6 + 1;
var _loc10 = _loc3[_loc8];
if (!_loc12)
{
if (placename.substr(0, 4) == "Shop")
{
_global.mmu.start(0, 30);
setbgmVol(0);
if (_loc7 == "ban")
{
_loc18();
}
}
}
}

```

```
    } // end if
    if (_loc7 == "wea")
    {
        _loc19();
    } // end if
}
else if (placename.substr(0, 6) == "proper")
{
    if (_loc6 == 0)
    {
        rent = 100 + o2 / 2;
    }
    else
    {
        rent = int(_loc3[_loc6] / 2 + _loc6 * _loc3[_loc6] / 10);
    } // end else if
    var r = rent;
    if (!ownership)
    {
        _loc16();
    }
    else if (ownership != _loc5)
    {
        _loc14();
    }
    else if (ownership == _loc5)
    {
        buildupLv(nowmoney, _loc8, _loc10);
    }
    else
    {
        workingCompleted();
    } // end else if
} // end else if
}
else if (_loc12 == true)
{
    var _loc4 = "";
    if (placename.substr(0, 4) == "Shop")
    {
        if (_loc7 == "ban")
        {
            if (nowmoney > 50000)
            {
                var _loc11 = random(40000) + 10000;
                _loc4 = _loc5 + " has deposited " + _loc11 + " G.";
                depositM(_loc11);
            }
            else if (nowmoney < 0)
            {
                var _loc17 = -nowmoney;
                _loc11 = random(_loc9 * 0.800000) + _loc17 + _loc9 * 0.200000;
            }
        }
    }
}
```

```

        _loc4 = _loc5 + " has withdrawn " + _loc11 + " G.";
        depositG(_loc11);
    }
    else if (nowmoney > 0 && nowmoney < 5000)
    {
        _loc11 = _loc9 * 0.300000;
        _loc4 = _loc5 + " got " + _loc11 + " G.";
        depositG(_loc11);
    }
    else if (nowmoney < 50000 && nowmoney > 5000)
    {
        _loc4 = "Nothing to do... \n Wait until next turn for luck.";
    } // end else if
}
else if (_loc7 == "wea")
{
    _loc4 = "Nothing to do... \n Wait until next turn for luck.";
} // end else if
}
else if (placename.substr(0, 6) == "proper")
{
    if (_loc6 == 0)
    {
        rent = 100 + o2 / 2;
    }
    else
    {
        rent = int(_loc3[_loc6] / 2 + _loc6 * _loc3[_loc6] / 10);
    } // end else if
    if (ownership == false)
    {
        var _loc13 = Number(parseInt(_loc3[1]) * 10);
        if (nowmoney >= _loc13)
        {
            _loc4 = _loc5 + " purchased " + placename + ".\n " + o2 + " was used to pay the
            place.";
            buyhome(false);
        }
        else
        {
            _loc4 = "Nothing to do....";
        } // end else if
    }
    else if (ownership != _loc5)
    {
        _loc4 = _loc5 + " give " + ownership + " for " + rent + " G as rent \n" + ownership + "
        gain the rent " + rent + " G.";
        payForRent(false);
    }
    else if (ownership == _loc5)
    {
        if (nowmoney >= _loc10)

```

```
{
    _loc4 = _loc5 + "The square was used for building extention.\n " + ownership + "
    upgrated it to " + _loc8 + "LV \nIt involed with " + _loc10 + "G on the extention.";
    upgrateHome();
}
else
{
    _loc4 = "Nothing to do....";
} // end else if
}
else
{
    _loc4 = "Nothing to do....";
} // end else if
} // end else if
_loc15(String(_loc4));
} // end else if
};
```