Annexe no. 3 : Éléments du jeu expérimental

III. FICHIER SETUP / JEU EXPÉRIMENTAL- VERSION GARÇONS

{ Language is case insensitive }
{ All documentation is in this file, don't delete it }
{ Oh I forgot, some information is in the }
{ language.txt file but it's a bit outdated :( }
{ Tokens are numbers, names, and strings. Strings can }
{ contain underscores representing spaces }
{ Comments, like this, can be nested }
{ Thomas Wehrle, Dec. 1995 }

Setup ofraH { Name can be changed }

Parameters
Globals
8   { Number of levels }
8   { Number of lives (max 10) }
2   { Border width }
6   { Foodsize }
1   { Show Top Ten }
1   { Use different shapes for standard enemies if possible }
0   { Save starting MGBs as defaults (leave 0 here) }
6   { Container size (in items) }
3   { Delay in milliseconds for board show and hide }
800   { Time of power in pixel distance }
1200   { Time of faster speed in pixel distance }
800   { Time of shield effect in pixel distance }
1200   { Time of bonus in pixel distance }
0   { Type of pictures for helper (0=bgi 1=pcx 2=no helper) }
4   { Number of potions for full magic }
2   { Magic units per potion }
2   { Magic units critical threshold => blinking }
0   { Cycles after which helper asks for help (1. time) }
0   { Cycles after which helper asks again (0= never) }
{ Same thing again for the level on which magic fails }
1000   { Cycles after which helper asks for help (1. time) }
3500   { Cycles after which helper asks again (0= never) }
7   { Level when magic fails (0=never fails) }
1   { Magic units peg potion on magic fail level }
4   { Fast level that needs helpers intervention (0=never) }
5   { Speed of fast period in fast starting level }
4800   { Cycles after which helper slows down the fast level }
15000   { Cycles after which helper gives a life (mode 3) }
10000   { Score for getting a new life }
3   { Lifes that the hunter (fast enemy) has got }
{ Put a zero for never die but also never go to prison }
1   { Allow programmer mode }
1   { Display level done message }
0   { Global decrease of speed for all levels }
{ Only change this value here, if you play on a faster }
{ platform. Use /sin for individual adjustment }
0   { Horrible style (???????) }
2   { Default user control (1 = keyboard, 2 = joystick) }
1   { Keyboard mode default (1 or 2) }
1   { Joystick mode default (1 or 2) }
10   { Volume of left speaker (0..15) }
10   { Volume of right speaker (0..15) }
{ Only works when there is a mixer chip on the soundcard }
25   { Lightchange in creepy mode }
1   { High graphics resolution (0 = low resolution) (*) }
1   { Try to correct the character set (for Topten) (*) }
0   { Write-cache-disabled mode (*) }
{ This function adds the message_announce_snd because }
{ saving the message window interrupts the game a little }
{ bit. Only set this flag if there is a significant }
{ pause }
5   { Time out (secs) of confirmation dialog }
5   { Number of deaths on a level until the helper }
{ eliminates all enemies if called (and bell effect 1) }
10   { Number of deaths on a level until the helper }
{ eliminates all enemies anyway }
3   { min. enemy distance (in cells) for bell effect }
30   { Longest duration of a game pause (secs) (pause tool) }
0   { Countdown in pause above time out is hearable }
1   { alternate joystick logic for mode 1 (old logic) }
1000   { Time t until fast enemy appears. 0 = after power mode. }
{ 0 is the old behavior. If the player has power at time }
{ t it will wait until this mode ends. But if the player }
{ gets and looses power before time t, this will also }
{ trigger the appearance of the fast enemy }
1   { Control randomness (experimentally reproducible) }
{ This should be 1 except if one wants a _real_ game }
1   { Allow tool selection during a pause (pause tool) }
{ In this case only the red button terminates the pause }
Annexe no. 3 : Eléments du jeu expérimental

End Globals

( Note concerning truth (boolean) values:

( 0 = false / no

( 1 = true / yes

( (*) These are technical parameters. Do not change them if you do

( not understand them )

( Note concerning pcx helper pictures:

( a) The palette of the pcx file is NOT loaded because that could

( interfere with the rest of the rgb setting. Therefore choose

( colors not yet used or fixed and ok. Change the RGB values

( with the color editor. The pcx files are standard 256 color

( . PCX format files.

( b) All pictures should be of the same size. Picture 0 is taken

( as reference.

( c) Nothing happens if a picture does not exist

( d) Picture numbers are magic units. A potion can have more than

( one magic unit. On the level on which the magic fails a potion

( can have a different number of magic units.

( Note concerning cycles:

( Is a time measure. Is the number of updates a character gets.

( Pixel distance time = cycles / stepwidth

( Note concerning pixel distance:

( This is a time measure for your convenience. It is the theoretical

( time interval that the player would need to walk n pixels. It is

( nevertheless a TIME measure. If the player does not walk time still

( elapses. This measure is independent of the current speed of step

( width of the player. Values must be > 0.

( Note concerning the graphics resolution:

( The game engine was developed for high resolutions. However, some

( notebooks, or older computers may not support this resolution.

( Therefore you can develop a game for lower resolution. Note that

( either the player and the enemies, or the boards have to be smaller

( to fit onto the screen, compared to higher resolutions. The same

( applies to the helper pictures, etc. Also note that even the low

( resolution is a SVGA resolution (non VGA standard, because it is

( also using 256 color display).

Names

LABCAT { World filename fixed part }
JAP { Player pattern filename fixed p. }
XYZ { The 3 different player bodies (state of damage) }
ANIM { Enemy pattern filename fixed p. }
HURTER { Hurting enemy pattern filename fixed p. }
HUTHER { Fast enemy pattern filename fixed p. }
JANUS { Janus pattern filename fixed p. }
HELPERS { Helper pattern filename fixed p. (special case) }
POMME { Name of bonus pattern(s) }
BOMB { Name of the skull pattern }
AUTO { Filename (3 letters) for recoding of events }
CAT99EX1 { Experiment signature }
INSTRCAT.FRC { First questionnaire }
ENDCAT.FRC { Final questionnaire }
DIVCAT.FRC { File of standard messages }
DEBCAT.FRC { File of level start messages }
FINCAT.FRC { File of level end messages }
ADVEIT.DAT { File of top ten list }

End Names

( Both messages files should exist. Missing messages in the standard

( file cause a warning, missing starting messages are ignored

( The different boards can be created with the pedit.exe editor.

( This editor can also be used to create the pattern for the player,

( the different enemies, tools etc.

( Call: pedit [filename]

Tools

TELEPORT { Bitmap of the teleporting tool }
ROLLERBLADES { Bitmap of the speed-up tool }
POWER { Bitmap of the power tool }
WALLTOOL { Bitmap of the set-home-block tool }
SHIELD { Bitmap of the shield tool }
RED CROSS { Bitmap of the repair/healing tool }
MAGIC { Bitmap of the magic potion (pseudo tool) }
SUPERS { Bitmap of the super bonus (life) (pseudo tool) }
TELEPHON { Bitmap of the help call tool }
KEY { Bitmap of the key tool }
PAUSE2 { Bitmap of the pause/hourglass tool }

End Tools

Modes { These numbers assume that the player color RGBs are 31 63 31 }
Annexe no. 3 : Eléments du jeu expérimental

{ red  green  blue = RGB values }
32  -32  -31  ( Powermode )
7   0    32  ( Fastmode  )
32  0    32  ( Shieldmode )

End Modes

{ Be careful when you make changes in above section! }
{ It is your responsibility that the RGB values are within the legal range }
{ The principle: 1) The RGB values define the "look" of a color }
{ 2) It's the additive variant (mix all ⇒ white), that is probably different from what you are used to }
{ 3) An RGB value MUST be within the interval [0,63] ! }
{ 4) Above numbers represent the CHANGES of the RGB values, }
{ 5) Modes that can be ON at the same time MUST not violate }
{ these principles. }
{ 6) Knowing the RGB values of the player's color is crucial }

Levels
( Level Tools global telep. no settlement ring player player player player enemy enemy enemy enemy enemy enemy level level questionnaire )
<table>
<thead>
<tr>
<th>delay</th>
<th>ring</th>
<th>death</th>
<th>effect</th>
<th>step</th>
<th>morph</th>
<th>number step</th>
<th>morph</th>
<th>lifes</th>
<th>start</th>
<th>stop</th>
<th>features</th>
<th>specials</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MARKED</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>MARKED</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>MARKED</td>
<td>19</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>MARKED</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>MARKED</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>MARKED</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>MARKED</td>
<td>17</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>MARKED</td>
<td>17</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

End Levels

{ Note concerning tools: ALL = Set all tools, randomly placed on fields with the tool marking color }
| NONE = No tools |
| MARKED = Set the tools according to the marked fields (tool colors) |
{ Or a string containing one or more of: }
| A = teleport |
| B = fast |
| C = power |
| D = walltool |
| E = shield |
| F = Repair |
| G = Magic 1 G = Enough Magic to help |
| H = Superbonus 1 H = Enough Super to refill |
| I = Bell |
| J = Key |
| K = Hourglass/Pause |

{ Note concerning global speed: Since the new version of GAME we }
{ dump quite a bit of data to the disk. This slows a running game }
{ down compared to a game which does not record data. Therefore }
{ it might be advisable to "tune" a game in dump mode. }

{ Note concerning no death: }
| If no_death is bigger than 0 the helper will give a life to the }
| player: 1 = whenever the player has only got one life left }
| 2 = like 1, but only works once }
| 3 = after n cycles, to be specified in the global number sec. }
| 4 = 1 + 3 |

{ Note concerning bell effect: }
| 0 = default help: a) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| b) heal if hurted }
| c) protect (shield mode) otherwise }
| 1 = a) eliminate all enemies (also hidden ones) if }
| the subject has been killed at least }
| x times on this level }
| (x specified in global numbers)
| b) protect (shield mode) otherwise }
| 2 = a) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| and if there are no enemies in the range of }
| 3 cells (note that Janus is excluded)
| b) protect (shield mode) otherwise }
| 3 = a) heal if hurted }
| b) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| and if there are no enemies in the range of }
| 3 cells (note that Janus is excluded)
| c) protect (shield mode) otherwise }

{ Note concerning tools: ALL = Set all tools, randomly placed on fields with the tool marking color }
| NONE = No tools |
| MARKED = Set the tools according to the marked fields (tool colors) |
{ Or a string containing one or more of: }
| A = teleport |
| B = fast |
| C = power |
| D = walltool |
| E = shield |
| F = Repair |
| G = Magic 1 G = Enough Magic to help |
| H = Superbonus 1 H = Enough Super to refill |
| I = Bell |
| J = Key |
| K = Hourglass/Pause |

{ Note concerning global speed: Since the new version of GAME we }
{ dump quite a bit of data to the disk. This slows a running game }
{ down compared to a game which does not record data. Therefore }
{ it might be advisable to "tune" a game in dump mode. }

{ Note concerning no death: }
| If no_death is bigger than 0 the helper will give a life to the }
| player: 1 = whenever the player has only got one life left }
| 2 = like 1, but only works once }
| 3 = after n cycles, to be specified in the global number sec. }
| 4 = 1 + 3 |

{ Note concerning bell effect: }
| 0 = default help: a) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| b) heal if hurted }
| c) protect (shield mode) otherwise }
| 1 = a) eliminate all enemies (also hidden ones) if }
| the subject has been killed at least }
| x times on this level }
| (x specified in global numbers)
| b) protect (shield mode) otherwise }
| 2 = a) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| and if there are no enemies in the range of }
| 3 cells (note that Janus is excluded)
| b) protect (shield mode) otherwise }
| 3 = a) heal if hurted }
| b) give a life if only two lifes left (i.e. if }
| there is only one life on the right)
| and if there are no enemies in the range of }
| 3 cells (note that Janus is excluded)
| c) protect (shield mode) otherwise }
Annexe no. 3 : Eléments du jeu expérimental

( 4 =
  a) give power
  and if there are any misbehaving Janus characters then put them in jail
  b) eliminate all enemies (also hidden ones) if
  the subject has been killed at least
  x times on this level
  (x specified in global numbers)
  c) protect (shield mode) otherwise
  d) protect (shield mode) (for training)

(Note concerning level specials: NONE = no specials)

Or a string containing one or more of:

- F = Fast enemy
- J1 = Janus, first face
- J2 = Janus, second face
- H = Hurting enemy

(Note concerning teleport ring: 0 = default
1 = random entry
2 = random teleporter
>2 = random place)

(Note concerning level features:
The feature number represents a bit set. I am aware that
this is not extremely user friendly, but it is very flexible
once you got used to it. Coding:

- Bit 0 (value 1) : no power points on this level
- Bit 1 (value 2) : tools cannot disappear in creepy mode
- Bit 2 (value 4) : Amigo makes no big fuzz about helping
   This is intended for training versions
- Bit 3 (value 8) : Home ejects player in creepy mode
- Bit 4 (value 16) : Put food under enemy when marked
- Bit 5 (value 32) : Start level with no tools in container

(Note concerning sounds:
Digital sounds, all with the same sampling rate
22050 Hz, 8 bit standard (no compression), mono WAVE files
Conversion to raw sound files with my WAV2RAW utility in the
sounds directory. All sounds are loaded in XMS memory.
The necessary memory can be seen with the debug flag on (see
program arguments in the read.me file). It is currently
with the standard sounds around 650k.

Sounds

- power (0 power point)
- food (1 food point)
- warning (2 mode end warning)
- normal (3 mode end)
- level (4 level done)
- bonusapp (5 bonus appears)
- earbon2 (6 bonus eaten)
- bonusdis (7 bonus disappears)
- tock (8 counter tick)
- newlife (9 gain a new life)
- go (10 level start / continue)
- victory (11 player kills enemy)
- crynth (12 player hurted nth time)
- whine (13 player hurted 1st time)
- bravo (14 player eats fast enemy)
- message (15 message from helper)
- msgcomes (16 announce message)
- roar (17 fast enemy appears) { Internally sometimes HUNTER }
- grin (18 janus grins) { Took the power away }
- helps (19 janus smiles) { Gave power }
- die (20 player dies)
- gameover (21 game over)
- alldone (22 game over and survived)
- toolfail (23 use tool failed) { 1. try to repair unnecessarily }
- tooldeny (24 refuse usage of tool) { 2. try to use a tool, but there is none }
- gettele (25 get teleporter tool)
- apptele (26 apply teleporter tool)
- getfast (27 get wing tool)
- appfast (28 apply wing tool)
- getpower (29 get power tool)
- power (30 apply power tool)
- getwall (31 get wall tool)
- appwall (32 apply wall tool)
- getshield (33 get shield tool)
- apshield (34 apply shield tool)
- scontact (35 shield contact = eject)
Annexe no. 3 : Eléments du jeu expérimental

getheal { 36 get healing tool }
apheal { 37 apply healing tool }
getmagic { 38 get magic tool }
apmagic { 39 apply magic }
fllmagic { 40 full magic }
decmagic { 41 magic decreases }
getsuper { 42 get super bonus }
getbell { 43 get bell tool }
getbell { 44 apply bell tool }
interupt { 45 game interruption }
backgame { 46 back to game }
(* ) homein { 47 home in }
(* ) homeout { 48 home out }
nukestrt { 49 post war starts }
nukebkgr { 50 post war background }
nukebomb { 51 post war dematerialize }
nukedrum { 52 post war drum }
nukecrepy { 53 post war creepy sound }
getkey { 54 get key tool }
apkey { 55 apply key tool }
telexport { 56 player passively telep }
yes { 57 player answered "yes" }
no { 58 player answered "no" }
droptool { 59 drop teleport tool }
droptool { 60 drop wing tool }
droptool { 61 drop power tool }
droptool { 62 drop wall tool }
droptool { 63 drop shield tool }
droptool { 64 drop repair tool }
droptool { 65 drop bell tool }
droptool { 66 drop key tool }
nodrop { 67 drop refuse sound }
pop { 68 change dir to up }
pop { 69 change dir to down }
pop { 70 change dir to left }
pop { 71 change dir to right }
confirm { 72 confirm dialog }
cntdown { 73 " " count down }
getpause { 74 get hourglass tool }
apppause { 75 apply hourglass tool }
droptool { 76 drop hourglass tool }
endpause { 77 end of pause }
select { 78 preselect tool }
selerror { 79 manip. err. in above }
deythcyr { 80 last cry in showdown }
nukestop { 81 armageddon stops }
tockbig { 82 loud counter sound }
normal { 83 fast mode ends }
normal { 84 shield mode ends }
nukestop { 85 helper kills enemies }
appshield { 86 helper protects }
getsuper { 87 helper gives new body }
apheal { 88 helper repairs body }
power { 89 helper: power + janus }
tock { 90 silent newlife }
lastcry { 91 hurted and dies }
fastcry { 92 hit by fast e and dies }
scontact { 93 fast enemy hides } [ after kill ]
healeff { 94 heal effect (2nd leg) }
scontact { 95 hurting encounter }
scontact { 96 janus deflected }
scontact { 97 prison is full }
telexport { 98 home ejection }
nukestop { 99 enemies go away }

End Sounds

Labels { max 12 characters currently, see maxpanelmessage in params }
Superccristal{ When super bonus is encountered but not necessary }
Fin_niveau { A level was successfully completed }
Bonus { A red bonus was encountered }
Nouvelle_vie{ Player got a new life }
Victoire { Player killed an enemy }
Merci { Player rescued his friend }
Game_over { Player finished all levels and survived }
Select_Tool { In keyboard mode 2: Tool selection mode }
Game_Over { Game over panel message }
Bonus_Vie { Remaining lifes => points (first entry) }

) Non panel labels }
Oui { Yes label used by dialog module (no panel mess.) }
Non { No label used by dialog module (no panel message) }

End Labels

End Parameters

Filters { Not used currently }
End Filters
End Setup
Annexe no. 3 : Eléments du jeu expérimental

IV. FICHIER SETUP / JEU EXPÉRIMENTAL-VERSION FILLES

{ Language is case insensitive )
{ All documentation is in this file, don't delete it )
{ Oh I forgot, some information is in the )
{ language.txt file but it's a bit outdated :( )
{ contain underscores representing spaces )
{ Comments, like this, can be nested )
{ Thomas Wehrle, Dec. 1995 )

Setup ofraH { Name can be changed }
Parameters

Globals
8 ( Number of levels )
8 ( Number of lives (max 10) )
2 ( Border width )
6 ( Foodsize )
1 ( Show Top Ten )
1 ( Use different shapes for standard enemies if possible )
0 ( Save starting RGBs as defaults ( leave 0 here )
6 ( Container size (in items) )
3 ( Delay in milliseconds for board show and hide )
800 ( Time of power in pixel distance )
1200 ( Time of faster speed in pixel distance )
800 ( Time of shield effect in pixel distance )
1200 ( Time of bonus in pixel distance )
0 ( Type of pictures for helper (0=bgl 1=pcx 2=no helper )
4 ( Number of potions f...
Annexe no. 3 : Eléments du jeu expérimental

(Also note that this only works in joystick mode 1)
(or keyboard mode 1 (the other modes seem to be obs.))

End Globals

( Note concerning truth (boolean) values:)
( 0 = false / no
( 1 = true  / yes

(*) These are technical parameters. Do not change them if you do
( not understand them ;)

( Note concerning pcx helper pictures:)
( 1) The palette of the pcx file is NOT loaded because that could
( interfere with the rest of the rgb setting. Therefore choose
( colors not yet used or fixed and ok. Change the RGB values
( with the color editor. The pcx files are standard 256 color
( PCX format files.
( 2) All pictures should be of the same size. Picture 0 is taken
( as reference.
( 3) Nothing happens if a picture does not exist
( 4) Picture numbers are magic units. A potion can have more than
( one magic unit. On the level on which the magic fails a potion
( can have a different number of magic units.

( Note concerning cycles:)
( 1 Is a time measure. Is the number of updates a character gets.
( 2 Pixel distance time = cycles / stepwidth

( Note concerning pixel distance:)
( 1 This is a time measure for your convenience. It is the theoretical
( time interval that the player would need to walk n pixels. It is
( nevertheless a TIME measure. If the player does not walk time still
( elapses. This measure is independent of the current speed of step
( width of the player. Values must be > 0.

( Note concerning the graphics resolution:)
( The game engine was developped for high resolutions. However, some
( notebooks, or older computers may not support this resolution.
( Therefore you can develop a game for lower resolution. Note that
( either the player, and the enemies, or the boards have to be smaller
( to fit onto the screen, compared to higher resolutions. The same
( applies to the helper pictures, etc. Also note that even the low
( resolution is a SVGA resolution (non VGA standard, because it is
( also using 256 color display).

Names
LABCAT    { World filename fixed part }
JAP       { Player pattern filename fixed p. }
XYZ       { The 3 different player bodies (state of damage) }
ANIM      { Enemy pattern filename fixed p. }
HURTER    { Hurting enemy pattern filename fixed p. }
HUNTER    { Fast enemy pattern filename fixed p. }
JANUS     { Janus pattern filename fixed p. }
HELPERS   { Helper pattern filename fixed p. (special case) }
POEME     { Name of bonus pattern(s) }
BOMB      { Name of the skull pattern }
AUTO      { Filename (3 letters) for recoding of events
( or AUTO, or NONE
CAT99EX1  { Experiment signature }
INSTRCTF.FRC { First questionnaire }
ENDCTF.FRC { Final questionnaire }
DIVCTF.FRC { File of standard messages }
DEBCTF.FRC { File of level start messages }
FINCTF.FRC { File of level end messages }
ADOPITT.DAT { File of top ten list }

End Names

( Both messages files should exist. Missing messages in the standard
( file cause a warning, missing starting messages are ignored
( The different boards can be created with the pedit.exe editor.
( This editor can also be used to create the pattern for the player,
( the different enemies, tools etc.
( Call: pedit [filename]

Tools
TELEPORT  { Bitmap of the teleporting tool }
ROLLERBLADES { Bitmap of the speed-up tool }
POWER     { Bitmap of the power tool }
WALLTOOL  { Bitmap of the set-home-block tool }
SHIELD    { Bitmap of the shield tool }
REDCROSS  { Bitmap of the repair/healing tool }
MAGIC     { Bitmap of the magic potion (pseudo tool) }
SUPERS    { Bitmap of the super bonus (life) (pseudo tool) }
TELEPHON  { Bitmap of the help call tool }
KEY       { Bitmap of the key tool }
PAUSE2    { Bitmap of the pause/hourglass tool }

273
Annexe no. 3 : Eléments du jeu expérimental

End Tools

Modes { These numbers assume that the player color RGBs are 31 63 31 }
{ red = red + blue = RGH values
32 -32 -31 { Powermode
0 -31 32 { Fastmode
32 0 32 { Shieldmode
End Modes

Note concerning tools:
ALL = Set all tools, randomly placed on fields with the tool marking color
NONE = No tools
MARKED = Set the tools according to the marked fields (tool colors)
Or a string containing one or more of:
A = teleport
B = fast
C = power
D = walltool
E = shield
F = Repair
G = Magic 1 G = Enough Magic to help
H = Superbonus 1 H = Enough Super to refill
I = Bell
J = Key
K = Hourglass/Pause

Note concerning global speed: Since the new version of GAME we
(dump quite a bit of data to the disk. This slows a running game)
down compared to a game which does not record data. Therefore
it might be advisable to "tune" a game in dump mode.

Note concerning no death:
(If no death is bigger than 0 the helper will give a life to the
player if he has only got one life left)
 formulas: 1 = whenever the player has only got one life left
 2 = like 1, but only works once
 3 = after n cycles, to be specified in the global number sec.

Note concerning bell effect:
0 = default help: a) give a life if only two lifes left (i.e. if
there is only one life on the right)
b) heal if hurted

1 = a) eliminate all enemies (also hidden ones) if
the subject has been killed at least
x times on this level
(x specified in global numbers)
b) protect (shield mode) otherwise

2 = a) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
b) protect (shield mode) otherwise

3 = a) heal if hurted
b) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
c) protect (shield mode) otherwise

Levels
(Level Tools global telep. no bell player player enemy enemy enemy enemy  enemy creepy mode level level questionnaire)

Delay ring death effect step morph number step morph lifes start stop features specials

Level Tools
goal teleport no bell player player player enemy enemy enemy enemy enemy enemy

Standards

End Levels

Note concerning the RGB values of the player's color is crucial

End Modes

Note concerning modes:
(1) The RGB values define the "look" of a color
2) It's the additive variant (mix all => white), that is probably different from what you are used to (= subtractive variant, mix all => black).
3) An RGB value MUST be within the interval [0,63]
4) Above numbers represent the CHANGES of the RGB values,
NOT the RGB values.
5) Modes that can be ON at the same time MUST not violate these principles.
6) Knowing the RGB values of the player's color is crucial

End Tools

Note concerning tools:
ALL = Set all tools, randomly placed on fields with the tool marking color
NONE = No tools
MARKED = Set the tools according to the marked fields (tool colors)
Or a string containing one or more of:
A = teleport
B = fast
C = power
D = walltool
E = shield
F = Repair
G = Magic 1 G = Enough Magic to help
H = Superbonus 1 H = Enough Super to refill
I = Bell
J = Key
K = Hourglass/Pause

Note concerning global speed: Since the new version of GAME we
dump quite a bit of data to the disk. This slows a running game
down compared to a game which does not record data. Therefore
it might be advisable to "tune" a game in dump mode.

Note concerning no death:
(If no death is bigger than 0 the helper will give a life to the
player if he has only got one life left)
 formulas: 1 = whenever the player has only got one life left
 2 = like 1, but only works once
 3 = after n cycles, to be specified in the global number sec.

Note concerning bell effect:
0 = default help: a) give a life if only two lifes left (i.e. if
there is only one life on the right)
b) heal if hurted
c) protect (shield mode) otherwise

1 = a) eliminate all enemies (also hidden ones) if
the subject has been killed at least
x times on this level
(x specified in global numbers)
b) protect (shield mode) otherwise

2 = a) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
b) protect (shield mode) otherwise

3 = a) heal if hurted
b) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
c) protect (shield mode) otherwise

Annexe no. 3 : Eléments du jeu expérimental

End Tools

Modes { These numbers assume that the player color RGBs are 31 63 31 }
{ red = red + blue = RGH values
32 -32 -31 { Powermode
0 -31 32 { Fastmode
32 0 32 { Shieldmode
End Modes

Note concerning tools:
ALL = Set all tools, randomly placed on fields with the tool marking color
NONE = No tools
MARKED = Set the tools according to the marked fields (tool colors)
Or a string containing one or more of:
A = teleport
B = fast
C = power
D = walltool
E = shield
F = Repair
G = Magic 1 G = Enough Magic to help
H = Superbonus 1 H = Enough Super to refill
I = Bell
J = Key
K = Hourglass/Pause

Note concerning global speed: Since the new version of GAME we
dump quite a bit of data to the disk. This slows a running game
down compared to a game which does not record data. Therefore
it might be advisable to "tune" a game in dump mode.

Note concerning no death:
(If no death is bigger than 0 the helper will give a life to the
player if he has only got one life left)
 formulas: 1 = whenever the player has only got one life left
 2 = like 1, but only works once
 3 = after n cycles, to be specified in the global number sec.

Note concerning bell effect:
0 = default help: a) give a life if only two lifes left (i.e. if
there is only one life on the right)
b) heal if hurted
c) protect (shield mode) otherwise

1 = a) eliminate all enemies (also hidden ones) if
the subject has been killed at least
x times on this level
(x specified in global numbers)
b) protect (shield mode) otherwise

2 = a) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
b) protect (shield mode) otherwise

3 = a) heal if hurted
b) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
c) protect (shield mode) otherwise

Annexe no. 3 : Eléments du jeu expérimental

End Tools

Modes { These numbers assume that the player color RGBs are 31 63 31 }
{ red = red + blue = RGH values
32 -32 -31 { Powermode
0 -31 32 { Fastmode
32 0 32 { Shieldmode
End Modes

Note concerning tools:
ALL = Set all tools, randomly placed on fields with the tool marking color
NONE = No tools
MARKED = Set the tools according to the marked fields (tool colors)
Or a string containing one or more of:
A = teleport
B = fast
C = power
D = walltool
E = shield
F = Repair
G = Magic 1 G = Enough Magic to help
H = Superbonus 1 H = Enough Super to refill
I = Bell
J = Key
K = Hourglass/Pause

Note concerning global speed: Since the new version of GAME we
dump quite a bit of data to the disk. This slows a running game
down compared to a game which does not record data. Therefore
it might be advisable to "tune" a game in dump mode.

Note concerning no death:
(If no death is bigger than 0 the helper will give a life to the
player if he has only got one life left)
 formulas: 1 = whenever the player has only got one life left
 2 = like 1, but only works once
 3 = after n cycles, to be specified in the global number sec.

Note concerning bell effect:
0 = default help: a) give a life if only two lifes left (i.e. if
there is only one life on the right)
b) heal if hurted
c) protect (shield mode) otherwise

1 = a) eliminate all enemies (also hidden ones) if
the subject has been killed at least
x times on this level
(x specified in global numbers)
b) protect (shield mode) otherwise

2 = a) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
b) protect (shield mode) otherwise

3 = a) heal if hurted
b) give a life if only two lifes left (i.e. if
there is only one life on the right)
and if there are no enemies in the range of
3 cells (note that Janus is excluded)
c) protect (shield mode) otherwise

Annexe no. 3 : Eléments du jeu expérimental
Annexe no. 3 : Eléments du jeu expérimental

{ 4 =               a) give power                                    }
{  and if there are any misbehaving Janus                 }
{  characters then put them in jail                        }
{  5 =               a) heal if hurted                        }
{  b) eliminate all enemies (also hidden ones) if          }
{  the subject has been killed at least                    }
{  x times on this level                                   }
{  (x specified in global numbers)                         }
{  c) protect (shield mode) otherwise                      }
{  6 =               a) protect (shield mode) (for training)  }

(Note concerning level specials: NONE = no specials)
(Or a string containing one or more of):
{  F = Fast enemy                                           }
{  J1 = Janus, first face                                   }
{  J2 = Janus, second face                                  }
{  H = Hurting enemy                                        }

(Note concerning teleport ring: 0 = default)
{  1 = random entry                                         }
{  2 = random teleporter                                    }
{  >2 = random place                                        }

(Note concerning level features):
(The feature number represents a bit set. I am aware that
this is not extremely user friendly, but it is very flexible
once you got used to it. Coding:

{  Bit 0 (value 1) : no power points on this level          }
{  Bit 1 (value 2) : tools cannot disappear in creepy mode  }
{  Bit 2 (value 4) : Amigo makes no big fuzz about helping  }
{  Bit 3 (value 8) : Home ejects player in creepy mode      }
{  Bit 4 (value 16) : Put food under enemy when marked      }
{  Bit 5 (value 32) : Start level with no tools in container}
{  Bit 6 (value 64) : Kill all enemies after n death        }

(Note concerning sounds:)
(Digital sounds, all with the same sampling rate)
(22050 Hz, 8 bit standard (no compression), mono WAVE files)
(Conversion to raw sound files with my WAV2RAW utility in the
sounds directory. All sounds are loaded in XMS memory.
(The necessary memory can be seen with the debug flag on (see
program arguments in the read.me file). It is currently
with the standard sounds around 650k.)

Sounds
  power (0 power point)
  food (1 food point)
  warning (2 mode end warning)
(*) normal (3 mode end)
  level (4 level done)
  bonusapp (5 bonus appears)
  eatbon2 (6 bonus eaten)
  bonusdis (7 bonus disappears)
  tock (8 counter tick)
  newlife (9 gain a new life)
  go (10 level start / continue)
  victory (11 player kills enemy)
  crynth (12 player hurted nth time)
  whine (13 player hurted 1st time)
  bravo (14 player eats fast enemy)
  message (15 message from helper)
  msgcomes (16 announce message)
  roar (17 fast enemy appears)
  Internally sometimes HUNTER.
{  An alternativ is ROAR
  grin (18 janus grins)
  takes the power away)
  jhelps (19 janus smiles)
  (Gave power)
  die (20 player dies)
  gameover (21 game over)
  alldone (22 game over and survived)
  toolfail (23 use tool failed)
{  1. try to repair unnecessarily
{  2. try to set a wall on an illegal lace
  tooldeny (24 refuse usage of tool)
{  1. try to use a tool in shield mode
{  2. try to use a tool, but there is none
  gettele (25 get teleporter tool)
  aptele (26 apply teleporter tool)
  getfast (27 get wing tool)
  appfast (28 apply wing tool)
  getpower (29 get power tool)
  power (30 apply power tool)
  getwall (31 get wall tool)
  appwall (32 apply wall tool)
  getshield (33 get shield tool)
  appshield (34 apply shield tool)
  scontact (35 shield contact = eject)
  getheal (36 get healing tool)
  appeal (37 apply healing tool)
  getmagic (38 get magic tool)
  appmagic (39 apply magic)
Annexe no. 3 : Eléments du jeu expérimental

fllmagic { 40 full magic }
decmagic { 41 magic decreases }
getsuper { 42 get super bonus }
getbell { 43 get bell tool }
getbell { 44 apply bell tool }
interrupt { 45 game interruption }
backgame { 46 back to game }
(* ) homein { 47 home in }
(* ) homeout { 48 home out }
nukestart { 49 post war starts }
nukebkgr { 50 post war background }
nukedetonate { 51 post war dematerialize }
nukedrum { 52 post war drum }
nukecrepy { 53 post war creepy sound }
getkey { 54 get key tool }
appkey { 55 apply key tool }
telease { 56 player passively teleport }
yes { 57 player answered "yes" }
no { 58 player answered "no" }
droppool { 59 drop teleport pool }
droppool { 60 drop wing pool }
droppool { 61 drop power pool }
droppool { 62 drop wall pool }
droppool { 63 drop shield pool }
droppool { 64 drop repair pool }
droppool { 65 drop bell pool }
droppool { 66 drop key pool }
nodrop { 67 drop refuse sound }
up { 68 change dir to up }
down { 69 change dir to down }
left { 70 change dir to left }
right { 71 change dir to right }
confirm { 72 confirm dialog }
cntdown { 73 " count down }
getpause { 74 get hourglass tool }
apppause { 75 apply hourglass tool }
droppool { 76 drop hourglass pool }
endpause { 77 end of pause }
select { 78 preselect tool }
selected { 79 manip. err. in above }
deathcry { 80 last cry in showdown }
nukestop { 81 armageddon stops }
tockbig { 82 loud counter sound }
normal { 83 fast mode ends }
normal { 84 shield mode ends }
nukestop { 85 helper kills enemies }
appshields { 86 helper protects }
getsuper { 87 helper gives new body }
appheal { 88 helper repairs body }
power { 89 helper: power + janus }
tock { 90 silent newlife }
lastcry { 91 hurted and dies }
fastcry { 92 hit by fast e and dies }
scontact { 93 fast enemy hides } { after kill }
haleff { 94 heal effect (2nd leg) }
scontact { 95 hurting encounter }
scontact { 96 janus deflected }
scontact { 97 prison is full }
teleport { 98 home ejection }
nukestop { 99 enemies go away }

End Sounds

Labels { max 12 characters currently, see maxpanelmessage in params }
Supercristal( When super bonus is encountered but not necessary )
Fin_niveau { A level was successfully completed }
Bonus { A red bonus was encountered }
Nouvelle_vie{ Player got a new life }
Victoire { Player killed an enemy }
Merci { Player rescued his friend }
Game_over { Player finished all levels and survived }
Select_Tool { In keyboard mode 2: Tool selection mode }
Game_Over { Game over panel message }
Bonus_Vie { Remaining lifes => points (first entry) }

[ Non panel labels ]
Oui { Yes label used by dialog module (no panel mess.) }
Non { No label used by dialog module (no panel message) }

End Labels

End Parameters

Filters { Not used currently }
End Filters

End Setup