

ITB



Report Project

Nintendo DS game development
“GuideMii : ITB”



Jonathan Vacherat
Sebastien Peres
Sep – Dec 09

Project Supervisor: Arnold Hensman

Gratitude

First, we would like to thanks all the ITB staff and teachers who help us during our stay in ITB.

We also would like to thanks particularly Mr. Arnold Hensman for supervising this project, even if it's not something that he is used to do, regarding the nature of this project (C code). Thanks to him again because we chose a project that was interesting for us and he supported us.

We had a lot of fun doing this project and we are also proud of what we could achieve in 5 weeks.

Summary

Gratitude	1
Abstract	6
Introduction.....	7
Objective	7
Preface.....	8
1. Hardware presentation	9
1.1 Nintendo DS.....	9
1.2 Linker	10
2. Game Concept of “GuideMii : ITB”	11
2.1 What is “GuideMii : ITB”?	11
2.2 What kind of game is “GuideMii : ITB”?	11
2.3 What is a mini interactive guide?	11
2.4 What is the average length of a game?	11
2.5 Why a guide for the students in ITB?	11
2.6 How will the game be playable?	12
2.7 On which platform the game is planned?	12
2.8 Working time?	12
2.9 GuideMii : ITB – The general concept.	13
2.10 GuideMii : ITB – When the game starts.	13
2.11 GuideMii : ITB – The Mii avatar.	13
2.12 GuideMii : ITB – Help!.....	15
2.13 GuideMii : ITB - The sequence of a game session.	15
2.14 The in game interface!	15
2.15 GuideMii : ITB - The gameplay.	16
1) What can the user do?	16

2)	The Icons.....	16
3)	How to play.....	16
2.16	GuideMii : ITB – Graphic Design	18
1)	ANIMAL CROSSING (NINTENDO):.....	18
2)	HARVEST MOON (UBISOFT, NATSUME):.....	18
3)	Nintendo Mii Channel (NINTENDO):	19
2.17	Establishment of humorous scripted animation:	19
2.18	GuideMii : ITB – Sound Design.	20
2.19	GuideMii : ITB – Level Design.	20
2.19.1	Sub Map:	21
3.	Game Design of “GuideMii : ITB”	22
3.1	Menu	22
3.1.1	How menus works.	22
3.1.2	Menu Descriptions.	23
3.2	The Mii – Interactive Guide Avatar	26
3.2.1	Actions:	26
3.2.2	Movements:	27
3.2.3	Specifications:.....	28
3.2.4	Non-Player Characters (NPC):	28
3.2.5	Artificial Intelligence (AI):	28
3.3	Map – Levels	29
3.4	Interface	32
3.4.1	Main Interface.	32
3.4.2	Stylus	33
3.5	Sound.....	33
4.	Technical Design of “GuideMii : ITB”	34
4.1	Tools	34

4.1.1	DevkitPro	34
4.1.2	PAlib.....	34
4.1.3	PC NDS Emulator: no\$gba	34
4.1.4	Important Functions	35
4.1.4.1	Basic code, game loop.	35
4.1.4.2	Display a background.	36
4.1.4.3	Display text.	37
4.1.4.4	Display a sprite.	38
4.1.4.4	Scrolling and Parallax.	39
4.1.4.5	Collision detection.....	41
4.1.4.5.1	Color collision.	41
4.1.4.5.1	Circular collision.	42
	Further work.....	43
	Conclusion	44
	References.....	45

Abstract

When students arrive for the first time in ITB, they are lost and they spend a few days to know all about the institute (class room, restaurant, gym etc...). So we try to think about a funny way to help them to discover ITB buildings.

The idea of this project is to create a game where a player can move on a virtual map of ITB. The game will also give him some information about buildings.

To do this project, we chose to develop on Nintendo DS because of its features (double screen, touch screen ...). We will also use a C/C++ library (Palib) and a linker (R4) to communicate with the DS.

Introduction

“Students need at least four weeks of classes to know where their classrooms are!”^[1]
Starting from this simple observation, we had to find a way to educate a student on the ITB campus and buildings location, in an entertaining and easy way.

How can we teach students in an attractive way? Learning through play? What is more entertaining than a game? This program has to be interactive!

And this program is called: “GuideMii : ITB”, an interactive guide for the ITB Students.

Objective

The purpose of this project is to provide a video game on the Nintendo DS, whose goal is to guide the users through a map of ITB campus in an entertaining and easy way. The game provides advice and information following the movements of players.

Preface

As we really wanted to do a video game, this document will be written accordingly, following his own kind of structure. The usual Information Technology Computing project structure does not suit, so it will be written as follow:

- **1 – The Game Concept: The Idea!**

It summarizes the important elements of the game, it doesn't have to be precise and it's just an idea.

- **2 – The Game Design: What we really want!**

It refines the elements expressed in the game concept and details them in order to have a very precise description of the game.

- **3 – The Technical Design: How we are going to do?**

It is the technical aspect. How are we going to do what we have said in the Game Design. We have to find solution for every need.

1. Hardware presentation

1.1 Nintendo DS

The Nintendo DS is a dual-screen handheld game console developed and manufactured by Nintendo. The console features a clamshell design with two LCD screens inside. The Nintendo DS also features a microphone and supports wireless. It's the handheld consoles most popular in the world.



Features:

The lower display of the Nintendo DS is overlaid with a touchscreen, designed to accept input from the included stylus or the user's fingers. The touchscreen allows users to interact with in-game elements more directly than by pressing buttons, for example, in the included chatting software, PictoChat, the stylus is used to write messages or draw.

A built-in microphone is located below the left side of the bottom screen. It has been used for a variety of purposes, including speech, chatting online between and during gameplay sessions and minigames that require the player to blow or shout into the microphone.

The Nintendo DS has compatibility with Wi-Fi which allowing players to interact with each other within short range (10–30 m, depending on conditions) or online with the Nintendo Wi-Fi Connection service.

1.2 Linker

A linker is able to run homebrew (home applications, listening to music (MP3), watch movies, run MSN, etc on his Nintendo DS. Linkers have been a controversial issue during a long period because it could be used as a way of hacking. This has led to many lawsuits to stop the sale of the linker. But they have not been won and the linkers remain legal thanks to a large community of amateur developers. There is several sort of linker. During our project time, we had problems with one of them. Our program was not compatible with the hardware DS because the linker was too old. To fix this problem, we chose to use another one called R4.

Released in 2007 it is the size of an original game and it's inserted in slot 1 of DS. It supports media cards Micro SD (2 GB or 4 GB).

Transferring games and homebrew is very simple, just use the USB Micro SD reader provided. Just simply connected to your PC and drop the files onto your Micro SD card by a simple « copy and paste ». It has already installed features like a browser to navigate into your files or a MP3 file player.



2. Game Concept of “GuideMii : ITB”

“Easy graphical method to learn more about the campus of the Institute of Technology Blanchardstown”

2.1 What is “GuideMii : ITB”?

GuideMii : ITB is a video game project whose goal is to guide the users through a map of ITB campus in an entertaining and easy way . The game features small avatars that will guide and provide advice and information following the movements of players.

2.2 What kind of game is “GuideMii : ITB”?

GuideMii : ITB is a mini interactive guide for the ITB Students. By providing a clear and intuitive interface, similar to a cartoon, and a very simple gameplay, GuideMii : ITB will be attractive for students or indeed anyone associated with ITB. A touch of humor will be integrated into the game.

2.3 What is a mini interactive guide?

The virtual guide will provides information and advice depending on where the user makes it moves on the map. The user can thus learn where this building, this room or this teacher is. Information on each site will be available if the user wants to learn more about it. The virtual guide will be designated as Mii.

2.4 What is the average length of a game?

Students are usually very impatient, they need information quickly, and if it’s in a funny way, it’s better. They probably use the game between classes or during their breaks. We can estimate a session of use to about 5 minutes, if they are in a rush, and maybe 20 minutes if they have time and want to learn a lot. That’s why we need a really clean and intuitive interface.

2.5 Why a guide for the students in ITB?

Motivation: The idea comes from a simple observation. Students, most of the time, do not know where to find the important spots, unless they already have been there several times. They need at least four weeks of classes to know where their classrooms are!

2.6 How will the game be playable?

The game will be playable exclusively off line and will be playable in single player. However, it will be possible to store a personal customized Mii, and access some resources online via Wi-Fi.

2.7 On which platform the game is planned?

Our goal is to release the game on the most suited platform and with a development time as short as possible.

We have chosen the Nintendo DS because:

- it is **handheld**, you can use it on the go.
- it has **double screen**, useful for display extra info.
- it has a **tactile screen**, the bottom one, more intuitive.
- it has **sound**, you can entertain the user.
- it is **powerful**, enough for what we want to do.
- it has a **microphone**, you can entertain the user.
- it has **Wi-Fi connection**, exchange some information.
- it is the **best selling handheld console** to date : 100 million in the world ! [2]

2.8 Working time?

The team is composed of two members. They will take the role of programmer, graphic artist and game designer. A little time is expected for music and sounds.

As this is a student project, there will be no financial aspect. The game is tailor-made for pedagogical purposes only within ITB.

The project must be completed within the allotted 5 weeks.

2.9 GuideMii : ITB – The general concept.

GuideMii : ITB comes from the concept of the virtual interactive guide which provides information and advices on an area visited virtually, like a real tourist guide (when you go to a museum or to a zoo ...).

The user can personalize his Mii avatar when he wants, with a Wii-like editor. The user can walk on a map of ITB campus and be warned by some text pop up telling him in which area he is.

2.10 GuideMii : ITB – When the game starts.

After a brief introduction, the user has to create his Mii. He can personalize his Mii completely via the editor, or if he doesn't want to, he can pick a random Mii. Once the choice is made, the user is free to move around the campus map. Some pop up will appear to guide the user.

2.11 GuideMii : ITB – The Mii avatar.

Three random Mii are available at the beginning of the game. One is a female, the second one is a male and the third one is a strange Mii. If the user wants to personalize his own Mii, an editor will pop up and allow to edit:

- **The name** -> Nickname of the Mii.
- **The sex** -> Female, Male, Other?
- **The color of the skin** -> A palette of color pop up.
- **The hair** -> Some preset hair cut.
- **The face** -> Some preset face.
- **The color of shirt** -> A palette of color pop up.

Example (inspiration):





2.12 GuideMii : ITB – Help!

Accessible at all times, a help! Icon is displayed on the bottom tactile screen.

Touching it will pop up a menu and some topics:

- Explanation of the game interface: what are the icons?
- How can I walk the map?
- How can I Zoom in/out?

2.13 GuideMii : ITB - The sequence of a game session.

After choosing his (or her) Mii avatar, the user can begin the exploration of the map.

The user is assisted at all times by some pop up text and can access the help touching the icon at the bottom. There is no specific goal for this game. The objective is to educate the user by telling him where he is at any particular point and giving relevant information about that location via text displays and sound effects.

2.14 The in game interface!

The interface must be **very simple, intuitive and friendly**. Users should be able to learn how to play the game at the first attempt.

We thus find:

- The **help!** Icon,
- The **menu!** Icon,
- The **zoomIn!** Icon,
- The **zoomOut!** Icon,
- The **name of the Mii**,
- The **name of the zone**,
- The **moreInfo!** Icon.

2.15 GuideMii : ITB - The gameplay.

The following features are available to GuideMii : ITB users:

- An **interactive virtual visit** to the ITB campus site,
- A sense of “**humor**”,
- **Information** and **replica layout of all ITB Buildings**,
- A **customizable Mii avatar**,
- **Learning through play**.

1) What can the user do?

The user can **walk in the map**, **ask for some extra info** about areas, **confirm and enter into a block or a room** that he is interested in, **teleport** to a specific location in the map and can also **drive a car** across the campus.

2) The Icons.

The **5 icons** are located on **the bottom screen**, because it is the tactile one. They are available at all times, except for the **moreInfo! Icon** which **will be animated accordingly**. They are located on the four corner of the screen (zoomIn!/zoomOut! Icons close to each other in a corner).

3) How to play.

- **Walk:** Touching with the **stylus** a direction and the **Mii will follow**. The user can also move the Mii with the **traditional Directional Pad**.
- **Zoom in:** Touching the **zoomIn! Icon** and the map will zoom at the center of the screen.
- **Zoom out:** Touching the **zoomOut! Icon** and the map will zoom out at the center of the screen.
- **View the help:** Touching the **help! Icon** will bring up a menu with some help topics.
- **View the menu:** Touching the **menu! Icon** will bring up the menu.
- **Ask for Info:** Touching the **moreInfo! Icon** will pop up a text with information in relation to the current zone. This icon will be **disabled in gray** if the user is **not in a particular area**. When there is **more info about a zone**, the icon will be **displayed properly and animated**, in order **to attract the user attention**.
- **Confirm an entrance into a block or a room:** Pressing the **button A** when near a block or a room (Question pop up).

- **Drive the car:** Bringing the **avatar on top of the car** and **press the A Button to get in. Left/Right to steer the car, Up/Down to go forward/backward. Press B to get out.**
- **Teleport:** **Double tap with the stylus** the location in the map will **teleport the avatar** to that location.

Example scenario:

- The user points to a location and the Mii follows the stylus (He can also use the D-Pad),
- The Mii enter a special area,
- The name of the area change on the screen, and the moreInfo! Icon start its animation,
- The user touches the moreInfo! Icon,
- A text pop up and display some info about the area,
- The user leave the area, the text pop down,
- The user takes the car by pressing A near it. He drives the car with the D-pad. He presses B to get out,
- The user double tap the a location on the map and the avatar is teleported to that specified location,
- The user is near a block entrance, a question pop up asking if he wants to get in,
- The user press A, fade out , fade in of the screen and the Mii is on a sub map of this block,
- The user touches the menu! Icon, the game pause, and the menu pop up.

For the Mii editor, all is done by **simply touching arrows to customize the Mii**. There will be a **virtual keyboard** displayed in order to enter the **Mii's name**.

2.16 GuideMii : ITB – Graphic Design

Graphics in the game are **simple and cartoon-type**. The graphics are important. They must be both **simple and cute**, but **not too much** because it is aimed at **students**.

All graphics are 2D.

Here are images of famous games for comparison:

1) ANIMAL CROSSING (NINTENDO):



2.16.1 Screenshots from Animal Crossing^[3]. We want that cartoon style.

2) HARVEST MOON (UBISOFT, NATSUME):



2.16.2 Screenshots from Harvest Moon^[4]. We want this icon style and overlay.

3) Nintendo Mii Channel (NINTENDO):



2.16.3 Screenshots from the Wii Channel Mii [5]. We want this style for the avatar.

2.17 Establishment of humorous scripted animation:

- When the user **does not touch the screen** for a certain amount of time, the Mii **gets bored**, start **singing or whistle** and a special animation is shown.
- Some **sounds effects** will be played in **certain area**, like a cat, dog, car, people ...

2.18 GuideMii : ITB – Sound Design.

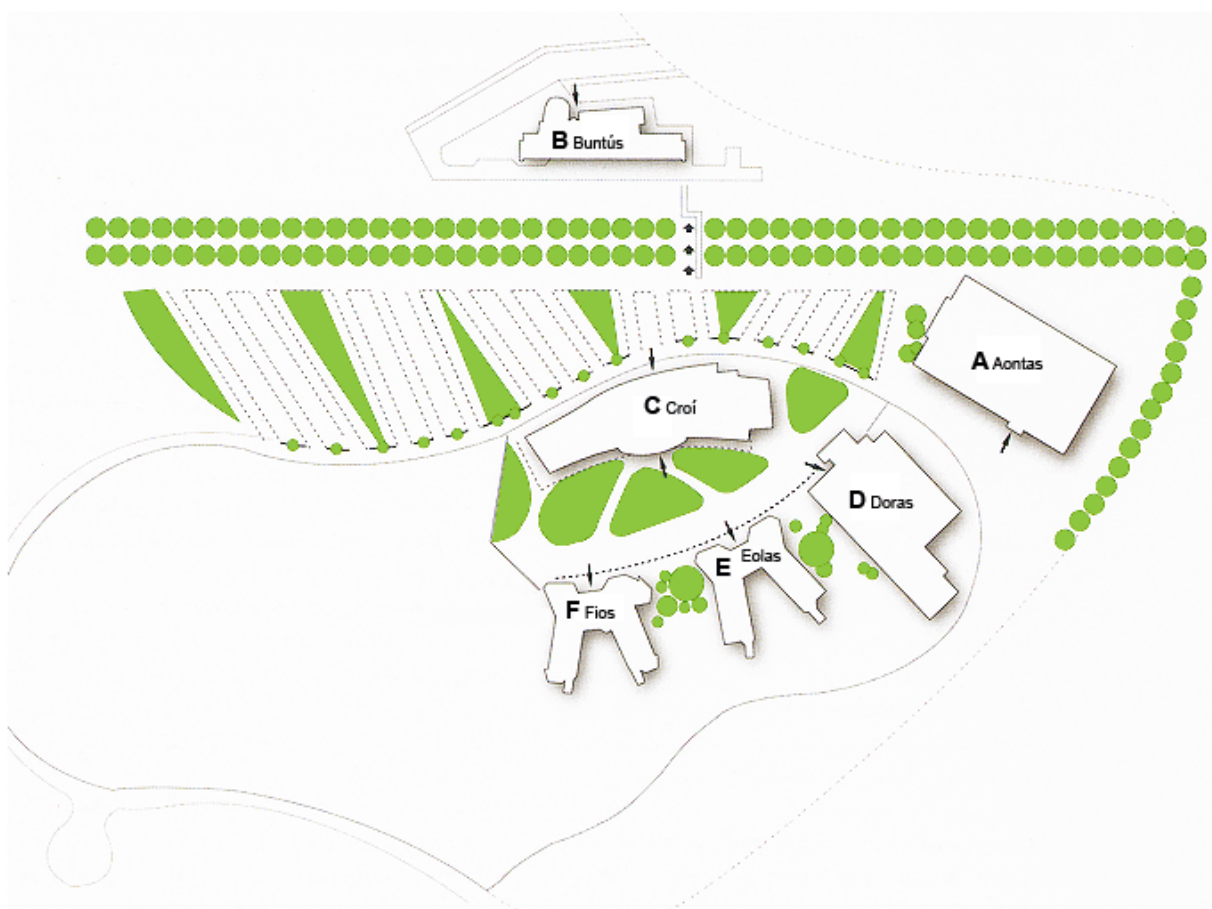
The music must be **relaxing, comforting** by default. But an option in the menu will allow the user to **choose between several themes** available in order to **satisfy different tastes**. The ambient sounds must remain **discreet**.

2.19 GuideMii : ITB – Level Design.

The environment must have everything instantly available to the user. He is really **free to go wherever he wants**.

We must also **avoid long loading time**.

We will use a **map of the campus** similar to that below as a base map for the game.



Ideally, the user is **asked to enter a room** when he is near a block or a room on the map.

If he wants to **get in**, he confirm by pressing **the A button**. If he doesn't want to get in, he doesn't press anything and continue to walk around.

There are 6 main blocks [6]:

- **Block A:** Aontas - Teaching Administration and N.L.N (national Learning Network)
- **Block B:** Buntús - Learning and Innovation centre (LINC)
- **Block C:** Croí - Multi-Purpose Building including Sports Hall and restaurant
- **Block D:** Doras - Apprentice Training
- **Block E:** Eolas - Teaching Building
- **Block F:** Fios - Reception, Library and Administration Building

Each block have at **least one sub map** corresponding to their internal representation, rooms and areas (some will have **two subs map** because of their arrangement, **two floor levels**).

There is also a **parking area** and a **green space** on the campus (**one map each**).

2.19.1 Sub Map:

Here is an example of a sub map, showing the outside and as you enter the building, show the inside:



Map, Outside.



Sub Map, Inside.

3. Game Design of “GuideMii : ITB”

“Easy graphical method to learn more about the campus of the Institute of Technology Blanchardstown”

3.1 Menu

3.1.1 How menus works.

At first launch of the game, the introduction runs (the user can skip it if he presses a button) and then the user gets directly on the main menu.

The first thing one sees is the intro page. Game title, cloudy animated background.

No FMV (Full motion video) or CG (Computer Graphic) video. The intro is made with the game engine. Simple logo with some random Mii walking around and a cool music. Introduction screen has to be funny, the first contact is important.

Note: The menu appears in transparency above the game screen, in a pop up style.



Screenshot of the introduction.

3.1.2 Menu Descriptions.

There is a special icon on the bottom game screen to go into the menu, called the menu! Icon.

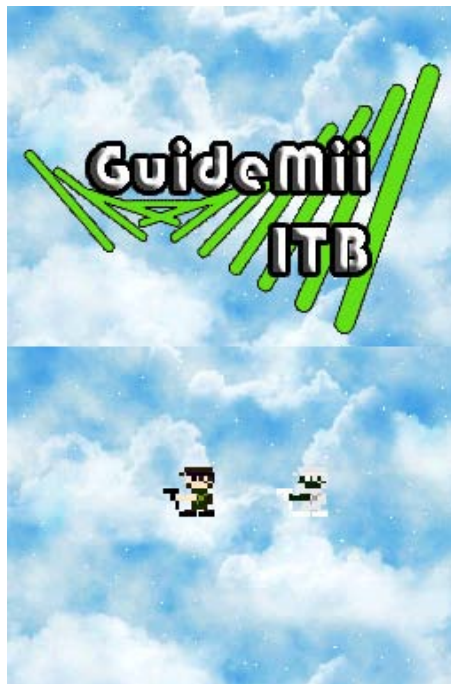
Main menu 1

- Start Game
- Options (Get back to the “Options!” menu)
- Credits (Display the Credits of the game)



Screenshot of the main menu

“Start Game” Just start the game and display a list of avatar to choose and play with. To select a Mii, just touch it with the stylus. Then the game starts and the player is free to walk on the map.



Screenshot of the avatar selection.



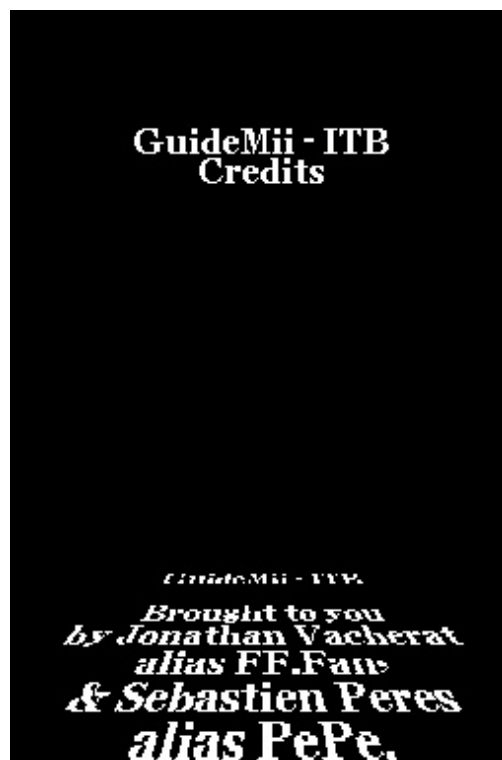
In game screenshot, displaying help.

“Options!” menu : Choose different theme



Credits

A screen displaying the credits of the game (name of the author, message ...)



Screenshot of the Credits.

3.2 The Mii – Interactive Guide Avatar

GuideMii : ITB is roughly based on the concept of virtual interactive guide.

The aim of this game is to teach the user about the ITB campus, in a funny and pleasant way.

He can learn about the location of blocks, rooms, information about them, and extra tips about some places.

To get there, we offer the user some actions he can do like: Walk, Ask Info, and Enter into a sub map. He can also drive a car in order to go faster, and also teleport the guide where he wants on the map.

The guide is very important because he has to be friendly and funny. The user can also identify himself as a guide and learn naturally.

3.2.1 Actions:

Actions cause specific animations of the Mii. Some can't be interrupted.

The User can:

- **Walk:** The user can move the Mii with the **traditional Directional Pad**. The walk animation of the Mii is displayed as he is moving up/down/left or right.
- **View the help:** Touching the **help! Icon** will bring up a menu with some help topics.
- **View the menu:** Touching the **menu! Icon** will bring get back to the main menu interface.
- **Ask for Info:** Touching the **moreInfo! Icon** will pop up a text with information in relation to the current zone. This icon will be **disabled in gray** if the user is **not in a particular area**. When there is **more info about a zone**, the icon will be **displayed properly and animated**, in order **to attract the user attention**.
- **Confirm an entrance into a block or a room:** Pressing the **button A** when near a block or a room (pop up).
- **Drive the car:** Bringing the **avatar on top of the car** and **press the A Button to get in**. **Left/Right to steer the car**, **Up/Down to go forward/backward**. Press **B** to get out.
- **Teleport:** **Double tap with the stylus** the location in the map will **teleport the avatar** to that location.

3.2.2 Movements:

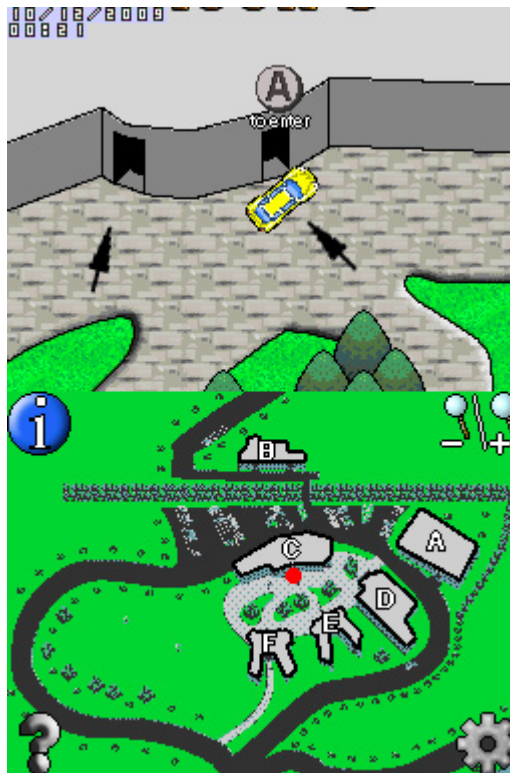
The movements of the Mii are simple and intuitive: the user can move the Mii with the **traditional Directional Pad**.

If the user stops pressing a direction on the D-Pad, the Mii stops where he is.

There will be **collision detection** on the campus map, the user can **only walk on the road path**. He can't walk on the grass. When the user is in **the car**, **he can go wherever he wants**, even on the places he couldn't reach walking (even on the water and on top of the buildings!).

The way of the avatar and the car move are different.

- Avatar: Playing a direction will move the avatar in that direction.
- Car: Pushing left/right will rotate the car, like a pivot, and pressing up/down, will move the car in that direction (forward/backward).



Screenshot showing the car drive, the pop up press A button and the info icon color changed.

3.2.3 Specifications:

- The user move faster when he is in the car than when he walks.

3.2.4 Non-Player Characters (NPC):

There will not be any NPC on this game for the moment.

3.2.5 Artificial Intelligence (AI):

There will not be any AI on this game for the moment.

3.3 Map – Levels

The world must have everything available "immediately" to the user. He is really **free to go wherever he wants**.

We must also **avoid long loading time**.

This **map of the campus** below is used for the map of the campus in the game.



The user is **asked to enter a block/room** when he is near a block or a room on the map.

- If he wants to **get in**, he confirm by pressing **the A button**. A fade out occurs as the Mii leaves the map, then a fade in as he enter a sub map.

- If he doesn't want to get in, he doesn't press anything and continue to walk around.

There are 6 main blocks:

- **Block A:** Aontas - Teaching Administration and N.L.N (national Learning Network)
- **Block B:** Buntús - Learning and Innovation centre (LINC)
- **Block C:** Croí - Multi-Purpose Building including Sports Hall and restaurant
- **Block D:** Doras - Apprentice Training
- **Block E:** Eolas - Teaching Building
- **Block F:** Fios - Reception, Library and Administration Building

One block, the C one, have **one sub map** corresponding to its internal representation, rooms and areas.

- **Block C:** Sub map 1



- **Block C:** Sub map 2



Note:

- Sound music change every time you enter the building. It's random.

- The maps are scrollable except the C building one. The dialog pop up will appears on top of the maps on the top screen.

3.4 Interface

3.4.1 Main Interface.



Screenshot of the interface, in game, icons displayed in the corners of the bottom screen, and the time & date on the top screen.

As shown on the screenshot, there are 3 icons on the bottom screen:

- **View the help:** Touching the **help! Icon** will bring up a menu with some help topics.



- **View the menu:** Touching the **menu! Icon** will bring get back to the main menu interface.



- **Ask for Info:** Touching the **moreInfo! Icon** will pop up a text with information in relation to the current zone. This icon will be **disabled in gray** if the user is **not in a particular area**. When the there is **more info about a zone**, the icon will be **displayed properly and animated**, in order to **attract the user attention**.



There is also a mini map of the entire campus map, and a red point telling the user in real time where he is on the campus!

On the top screen there is a date & time display, may be useful for students.

3.4.2 Stylus

The stylus is used to navigate in the main menu and choose the avatar at the beginning of the game.

But it is also used in game to touch the different icon on the bottom screen and to teleport the avatar on the map by double tapping it.

3.5 Sound

The music must be **relaxing, comforting** by default. But an option in the menu will allow **choosing between several themes** available in order to **satisfy a lot of people** and **not getting bored** with the same music. The ambient sounds must remain **discreet**.

There are **3 different songs, playing randomly**. When one is **finished, another starts**.

Each time you enter the C block, another music is playing.

4. Technical Design of “GuideMii : ITB”

4.1 Tools

4.1.1 DevkitPro

DevKitPro_[8] is a free, unofficial SDK for video game console. It allows developers to code homebrew_[9] games and applications.

DevkitPro is a toolchains of choice for homebrew game development, currently available for GameBoy Advance, GP32, Playstation Portable and GameCube.

The one we use is a version of GCC for ARM processor (the Nintendo DS has an ARM7 and ARM9) and it's called devkitARM.

4.1.2 PALib

PALib_[10] (Programmer's Arsenal Library) is a free Nintendo DS library used to create homebrew games. It is designed to be used easily by anyone in C or C++.

It allows an easy access to all the NDS hardware, the buttons, stylus, sound, double screen...

PALib come with a lot of simple example to demonstrate how each function works. The documentation is also well done.

It provides a specific tool called PAGfx, used to convert image in a format comprehensive for the NDS.

4.1.3 PC NDS Emulator: no\$gba

NO\$GBA_[11] pronounced as No Cash GBA is a Nintendo DS / DS Lite and GameBoy Advance emulator for Windows XP, Windows VISTA and MS-DOS. NO\$GBA f options include emulating multiple save types as well as and multiple cartridge reading. NoGBA supports multiplayer and is able to load multiple NDS ROM files for linking games. Multiplayer for GBA Games is also supported.

We are using it for test purpose and emulate the NDS software.

4.1.4 Important Functions

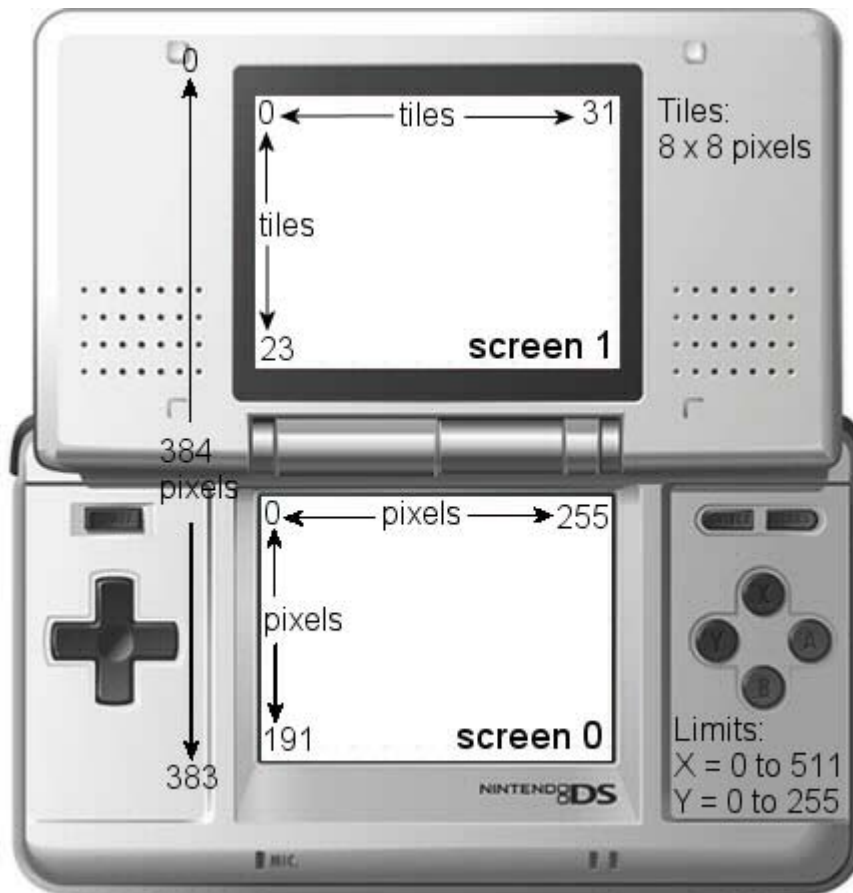
4.1.4.1 Basic code, game loop.

```
////////////////////////////////////  
// PALib project template //  
////////////////////////////////////  
  
// Lines starting with two slashes are ignored by the compiler  
// Basically you can use them to comment what are you doing  
// In fact, this kind of lines are called comments :P  
  
// Include PALib so that you can use it  
#include <PA9.h>  
  
int main(){  
    // Initialize PALib  
    PA_Init();  
    PA_InitVBL();  
  
    // Put your initialization code here  
  
    // Infinite loop to keep the program running  
    while(true){  
        // Put your game logic here  
  
        // Wait until the next frame.  
        // The DS runs at 60 frames per second.  
        PA_WaitForVBL();  
    }  
}
```

Very basic loop code for compiling using PALib.

4.1.4.2 Display a background.

The NDS work with layers. There are four for each screen and as the following dimension.



The NDS VRAM (Video RAM) doesn't have an unlimited memory, so we have to manage the memory more than once. Very likely, we will not be able to have more than 3 backgrounds at once on each screen, so we'll have to do some fancy tricks in order to avoid problems like backgrounds not appearing right or being all corrupt.

All you need to do for loading a simple background is to convert it with PAGfx and do:

```
// Load a simple background, very easy and simple...

// Includes
#include <PA9.h>          // Include for PA_Lib

// Converted using PAGfx
#include "gfx/all_gfx.c"
#include "gfx/all_gfx.h"

// Function: main()
int main(int argc, char ** argv)
{
    PA_Init();           // Initializes PA_Lib
    PA_InitVBL();        // Initializes a standard VBL

    // Load Backgrounds with their palettes !
    PA_EasyBgLoad(0, // screen
                  3, // background number (0-3)
                  bg0); // Background name, used by PAGfx...
    PA_EasyBgLoad(1, 0, bg0); // Same thing, but on the top screen...

    // Infinite loop to keep the program running
    while (1)
    {
        PA_WaitForVBL();
    }

    return 0;
} // End of main()
```

4.1.4.3 Display text.

All you need to do is add the following lines to the template's file just after PA_InitVBL();

PA_InitText(1, 2); // Tell it to put text on screen 1, background number 2

PA_OutputSimpleText(1, 1, 1, "Hello World!"); // Print on screen 1, with coordinate 1, 1

4.1.4.4 Display a sprite.

The NDS can display 128 sprites max in each screen, so 256 on both.

All you need to do is to convert your sprite with PAGfx and do:

```
#include <PA9.h>

// PAGfxConverter Include
#include "gfx/all_gfx.h"
#include "gfx/all_gfx.c"

int main(void){

    PA_Init(); //PALib inits
    PA_InitVBL();

    PA_LoadSpritePal(0, // Screen
                    0, // Palette number
                    (void*)sprite0_Pal); // Palette name

    PA_CreateSprite(0, // Screen
                   0, // Sprite number
                   (void*)vaisseau_Sprite, // Sprite name
                   OBJ_SIZE_32X32, // Sprite size
                   1, // 256 color mode
                   0, // Sprite palette number
                   50, 50); // X and Y position on the screen

    while(1) // Infinite loops
    {
        PA_WaitForVBL();
    }
    return 0;
}
```

4.1.4.4 Scrolling and Parallax.

Here is a sample of simple background scrolling:

```
PA_InitText(1, 0); // Init text on the top screen, background 0...

// Load Backgrounds and their palettes...
PA_EasyBgLoad(0, 3, BG3);
PA_EasyBgLoad(1, 3, BG3);

s32 scrollx = 0; // No X scroll by default...
s32 scrolly = 0; // No Y scroll by default...

// Infinite loop to keep the program running
while (1)
{
    // We'll modify scrollx and scrolly according to the keys pressed
    scrollx += (Pad.Held.Left - Pad.Held.Right) * 4; // Move 4 pixels per press
    scrolly += (Pad.Held.Up - Pad.Held.Down) * 4; // Move 4 pixels per press

    // Scroll the background to scrollx, scrolly...
    PA_EasyBgScrollXY(0, // Screen
                      3, // Background number
                      scrollx, // X scroll
                      scrolly); // Y scroll

    // Display the X and Y scrolls :
    PA_OutputText(1, 0, 0, "x : %d \ny : %d ", scrollx, scrolly);

    PA_WaitForVBL();
}
```

Parallax Scrolling is a great technique that does a different scrolling speed on all background, giving an impression of depth and a sort of 3d effect... It looks good and is so easy to put in place. It's used in many games, from shoot'em up (scrolling the close stars at one speed, and the further ones at another) to platform games (mario has the blocks at one speed, the mountains at another speed, and the clouds at still another one, and it looks perfect !), etc... This can be used in tons of situations.

```
// Load the 4 Backgrounds on the bottom screen...
PA_EasyBgLoad(0, 1, BG1); //screen, background number, background name
PA_EasyBgLoad(0, 2, BG2);
PA_EasyBgLoad(0, 3, BG3);

// Initialise parallax vertically (Y axis) for both backgrounds
// 256 is normal speed, 128 half speed, 512 twice as fast...
PA_InitParallaxY(0, //screen
                 0, //Parallax speed for Background 0. 0 is no parallax (will scroll independently with PA_EasyBgScrollXY)
                 256, // Normal speed for Bg1
                 192, // 3/4 speed
                 128); // Half speed
PA_InitParallaxY(1, 0, 256, 192, 128); // Same thing, but for Screen 1...

s32 scroll = 0;

// Infinite loop to keep the program running
while (1)
{
    scroll += 1; // Scroll by one pixel...
    // Backgrounds with a parallax speed of 256 will scroll 1 pixel, 192 will scroll 0.75, and 128 0.5
    // We could also have put a negative parallax speed to have some backgrounds scroll in different directions

    PA_ParallaxScrollY(0, -scroll); // Scroll the screen 0 backgrounds.

    PA_WaitForVBL();
}
```

4.1.4.5 Collision detection.

4.1.4.5.1 Color collision.

We used a function available in PALib, called:

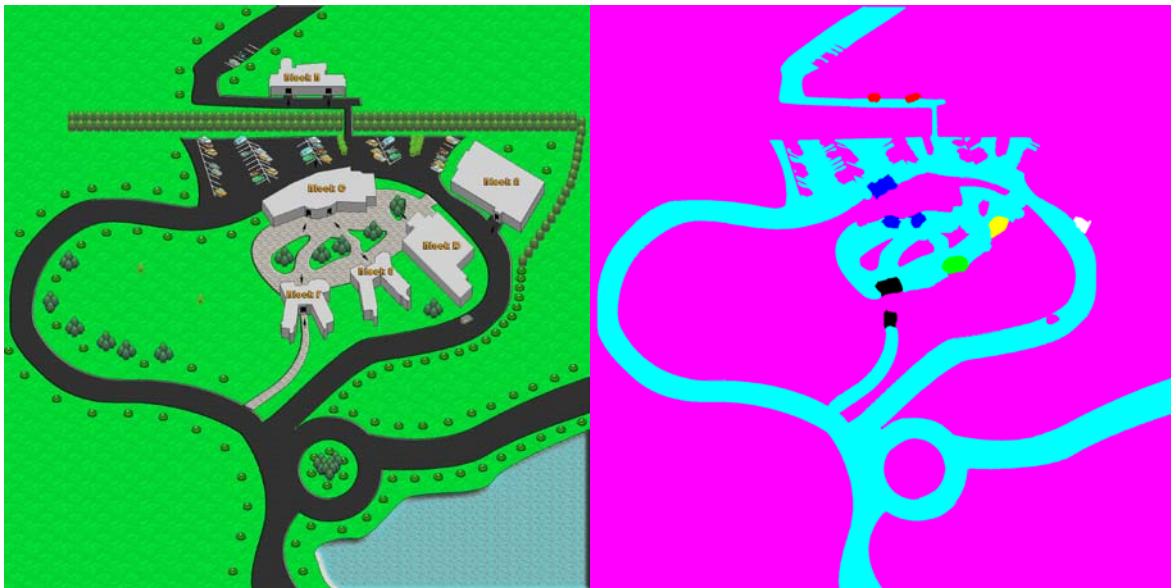
```
inline u8 PA_EasyBgGetPixel (    u8    screen,
                                u8    bg_number,
                                s32    x,
                                s32    y
                                )
```

Returns the color (number in the palette) of the screen pixel...

Parameters:

screen Chose de screen (0 or 1)
bg_number Background number (0-3)
x X screen pixel position
y Y screen pixel position

But what is the relation with collision? Well it's pretty simple. Here is two images for the map.



Left – Displayed map on the console screen. Right – Map in memory, underneath the displayed one.

Now it's easy to understand the use of the function described above.

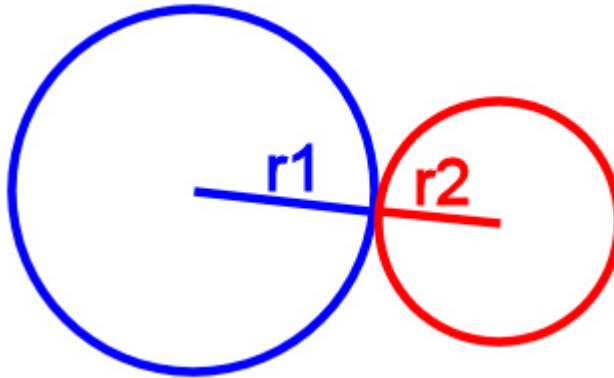
If the color is light-blue, the user is able to walk, otherwise, not.

The other colors are used in the same way to detect the building entrance.

4.1.4.5.1 Circular collision.

There is another collision detection used in our game too. This one is more common. It's based on a mathematic formula and called the Circular Collision. We use it to determine if the avatar is near the car.

If you consider sprites as round entities, it becomes really easy to make them collide correctly... Look at this small drawing, with 2 circles on it:



As you can see, these circles collide. At what distance are they ? Their centers are at exactly distance $r1 + r2$... So if you calculate the distance between the circles' centers, they collide if the distance is less than $r1 + r2$.

All we need is to call a function available in PALib called :

```
u32 PA_Distance (      s32    x1,
                    s32    y1,
                    s32    x2,
                    s32    y2
                  ) [inline]
```

Calculate the distance (squared) between 2 points

Parameters:

x1	X coordinate of the first point
y1	Y coordinate of the first point
x2	X coordinate of the second point
y2	Y coordinate of the second point

Further work

In order to improve our game, we consider the following points:

- Finish all the sub map for all the buildings
- Put more textual information about the different area
- Code the Customize Mii section described in the Game Concept
- Add mini games in order to entertain the user with simple games (snake, pong, tetris, airhockey ...) (sources codes are available on the internet for free)
- Add an encyclopedia listing all the visited area
- Add a special reward if the encyclopedia is fulfilled (special Mii character like a spiderman or other, secret map, secret theme song ...)
- Add extra animation and sound for the Mii (like touching him make it laugh, or scream ...)
- Add NPCs (non playable characters) & AI.
- Add Wi-Fi timetable retrieve for each student. (we have some code working that can retrieve an image file via Wifi !)
- Add ability to see every student connected on wifi and chat with.
- Use a recreated ITB campus in 3D instead of a single image map.
- Use a 3D Mii model.

Conclusion

In spite of lack of time, the results of this project are satisfying. In the game, we can walk, drive a car or teleport through the ITB map with collision detection and with sound playing in the background. We can also enter in a sub map of the C building.

This project was an opportunity to work for the first time on a game console and enter into the amateur video game community. We learned a lot during this project not only in code, but also on a game management project. We would like to use our knowledge to improve this game or create another one later on.

References

- [1] A. Hensman, 2009, ITB. Spoken words, first project meeting.
- [2] Nintendo. 2009-10-29. p. 9. "Consolidated Financial Highlights" (Portable Document Format).
URL: <http://www.nintendo.co.jp/ir/pdf/2009/091029e.pdf>
- [3] Animal Crossing, Nintendo, 2004.
URL: <http://www.animal-crossing.com/>
- [4] Harvest Moon, Ubisoft/Natsume, 2004.
URL: <http://www.mmv-i.co.jp/bokumono/index.html>
- [5] Mii Channel, Wii game console, Nintendo, 2006.
URL: http://www.nintendo.com/consumer/systems/wii/en_na/channelsMii.jsp
- [6] ITB.
URL: <http://www.itb.ie/>
- [7] R4, Linker for NDS, 2007.
URL: <http://www.r4ds.cn/index.html>
- [8] devkitPro, free toolchain for game dev.
URL: <http://www.devkitpro.org/>
- [9] Famous amateur homebrew download site.
URL: <http://www.pdroms.de/files/nintendods/>
- [10] PALib, free Nintendo DS library for game dev, Author: Mollusk
URL: <http://forum.palib.info/>
URL: <http://www.palib.info/wiki/doku.php?id=homepage>
URL: <http://palib.info/Doc/PALibDoc%20Eng/modules.html>
- [11] NO\$GBA, NDS emulator, 2007.
URL: <http://nocash.emubase.de/gba.htm>
- [12] Gambit2099, PODS & snake_48, Site du Zero, 2009. "Programmez sur votre Nintendo DS !"
URL: <http://www.siteduzero.com/tutoriel-3-101567-programmez-sur-votre-nintendo-ds.html>
- [13] Raphaël CALABRO, Développer en Amateur sur Nintendo DS, 2008.
URL: <http://www-igm.univ-mlv.fr/~dr/XPOSE2007/rcalabroDevNintendoDS/index.html#introduction>
- [14] French Homebrew community dev for NDS Forums, Player Advance.
URL: <http://playeradvance.org/forum/forumdisplay.php?f=122>