Journée portes ouvertes 2024

UNIVERSITÉ DE BOURGOGNE
UFR LANGUES ET COMMUNICATION

Création d'un jeu vidéo :

exemple d'un travail réalisé sous forme de projet en Licence 3 Langues étrangères appliquées, parcours « Industries de la langue ».

Ce travail a servi à évaluer une partie du cours « Langue et industries du numérique » proposé au sein du parcours « Industries de la langue » en Licence 3 LEA. Le cours a pour objectif d'explorer les relations entre les métiers de la langue et les technologies numériques et le projet consiste en la création d'un jeu vidéo simple dans un petit groupe, sur un thème libre, en suivant un cahier des charges qui précise les différents types d'interaction à intégrer dans le jeu.

Le cours en question est dispensé en anglais mais les jeux doivent être développés en français. Ceux-ci doivent notamment éviter de produire des fautes de langue, même dans le cas où (par exemple) il faudrait insérer un nom d'objet pouvant être masculin ou féminin, singulier ou pluriel, au milieu d'une phrase contenant des adjectifs à accorder. Il s'agit quelque part d'un exercice de grammaire française ludique – mais parfois difficile – mais aussi et surtout d'une occasion pour réfléchir au rapport entre les langues informatiques et le langage humain.

Dans ce qui suit, nous pouvons remercier Louise Hupfer et Hugo Geneviève d'avoir accepté de partager leur travail.

Leur jeu « Ringborn » est disponible en accès libre sous le lien :
https://textadventures.co.uk/games/view/doyjcxfcf0o01fi5tueica/ringborn
N'hésitez pas à aller y jouer – il vaut mieux utiliser un ordinateur plutôt qu'un téléphone ou une tablette.

Sur les pages qui suivent, vous verrez les consignes pour le devoir et l'analyse du projet, rédigée en anglais, avec des commentaires de l'enseignant.

L3 LEA – Parcours « Industries de la langue », Language and digital cultures

Assessment Task 1: Development and linguistic analysis of a computer game

Grading: 50% of your final mark for the course
(the other 50% will be awarded for the second part of the class with L. Gautier).

Due: 8am, 13/11/23 (game to be shared online or submitted as a .zip file, analysis and any file submissions on Teams)

We will begin this activity during your class on 04/10 and will have time to work on your projects, ask questions and share problems during our remaining classes. You are welcome to use our Teams group to ask questions and share problems, screenshots, etc.

Your first assessment task for the semester may sound fun but may prove more of a challenge than you think. Working **in pairs or small groups**, you will be asked to design, develop and then analyse a text adventure game **in French** using the open-source game development platform Quest. You will have the choice of working in ActiveLit (a teaching version of Quest that allows us to share games within a space confined to our class) or using the desktop version of Quest and submitting your game as a .zip file.

Language is the main focus of this activity and one of the criteria for grading projects is your ability to produce a game in error-free French. You will also need to think carefully about how the text, characters and objects you create fit into the fictional world you create, and the storyline your player will act out inside it. The platform provides some support for images and media files, but multimodal content is not a compulsory part of your project.

What you will need to do:

Focus of class 4, 10/10:

- In class 4, we will walk through the basics of creating a game in Quest and make sure each group has a viable project.
- Take some time with your group to determine a location and storyline for your game. Games already published on Quest may provide some inspiration.
- Find a partner, look through the selection of Quest games provided on the ActiveLit platform and start working on a location and storyline for your game. Who is the "player" and who are the other characters, where are they, and what does the player need to do to play, win, lose, and otherwise interact with the game?
- Start designing your game world: how many 'rooms' or spaces will there be in your virtual world, how will you organise and describe them, and how will the player interact with each room? For example: are you in a forest? a castle? a house? a spaceship? a museum?
- Design characters and objects for your game world: how will the player interact with them? How will these objects help advance the gameplay?
- Think about how you will turn all this into text and make a list of useful words and phrases.

- *Make sure you keep notes on your design and development work, as this will be part of the dossier you will need to submit with your game!*

Focus of class 5 and 6, 25/10 and 08/11:

- Put your ideas into practice as you continue developing your game, using your designs and other information you have collected. Since you have two weeks between classes, you should also make some time for group work. You will need to test the game as you go, checking whether the gameplay works as expected, and whether you are happy with how your descriptions, commands and other interactive elements fit together.
- In class 5 and 6, we will look at some more advanced features of Quest, including the use of scripts and at how the platform generates sentences by combining text with objects. The more organised you are before each class, the more I will be able to assist you.

What you will need to submit:

1. Your group's completed and tested game needs to be published on ActiveLit (or sent as a .zip file if you used the desktop version of Quest) by 8pm, Monday 13/11.

Your game should take the form of a **text adventure** (not an interactive fiction), with:

- at least five spaces or 'rooms' your character needs to navigate through, with descriptions of each room
- at least ten objects that your character can interact with
- at least one speaking character other than the player
- at least one **script** (we'll look at these in class 5)
- (*adding images, sounds and other multimodal elements is possible, but optional*).

The idea is to develop your game in French, and part of the exercise is to see how the game interface handles French grammatical and syntactical structures, including verbs, nouns (both inanimate objects and animate characters), and pronouns.

Your game needs to be thoroughly tested both for gameplay and for language. You will need to analyse the problems you encountered and solutions you found, and make sure that all language errors are identified and eliminated.

2. An analysis of your game for the same deadline, using the link available on Plubel.

Your analysis should be between 1-2 typed pages (roughly 1000 words) **per student**: if there are 3 students, I'll expect a single document of ~3000 words. It should include:

- A brief discussion of your game concept: what is the purpose of your game, what world is it set in and how do you expect the player to interact with it? You can submit diagrams, maps, and other documents you used to create the game.
- A discussion of the design process: how did you go about designing the game and its narrative content, what problems did you meet, and what solutions did you adopt?
- A list of the **interactive language elements** present in your game: which characters can interact with which objects using which verbs? How does the player know what to do? How did you prevent your game from generating language errors?
- A discussion of any technical or linguistic issues you met and how you solved them.

Analysis of our game : Ringborn

Concept of the game and its purpose:

We have decided to create a fantasy game where the player is a knight that needs to escape his cell first and then save his kingdom.

Global game ambiance has been inspired by FromSoftware Games, japanese game publisher which has made games such as Dark Souls, Bloodborne, Sekiro or Elden Ring.

The game takes place in a bewitched castle, full of enemies who were once human but have been transformed into zombies, inspired by the games mentioned above. The goal of the game is to find a way to defeat the enemy behind this chaos: a group of evil wizards to save his parents and people living in the kingdom. Players are expected to do their best to succeed in the game, inspecting each of the locations to which they have access and reading the dialogues carefully. In this way, they will avoid falling into the wrong paths or wasting time moving around the castle in search of an object or information crucial to their mission.
Some objects may have a dual purpose, but this is not necessary to complete the story. Some players may also see references to FromSoftware Games, such as certain pictures or the way certain dialogues are written.

Discussion about the game design process:

The very first thing to do was to get grips with the Quest software developed by ActiveLit, as we couldn't get access (we had your access but then a problem on the site made it unusable, loading infinitely then telling “error”). As we only had one Windows computer at our disposal, the entire game was developed on the same machine. Our first session was devoted solely to creating rooms and random objects of no interest, to see the potential limits of the software and what could and couldn't be done on it. We also watched the tutorials on offer, as well as some others on YouTube, but most of the content on the platform is made for an advanced level, so it was hard for us to follow what people were doing and understand it. We have directly noticed some such as the fact that we cannot use a hyphen for the names of parts or objects, this is same thing accents at the beginning of the names (e.g.: sword -> épée in French). An object also cannot contain too many spaces or words in their name.

Once we understood the basics, we immediately started working on our game, trying to include our ideas. The first thought we made is that it is very easy to get lost and linger on details while wanting to add more and more things (images then sound effects then font, color, background without end) so we focused on the main skeleton of the game first so that once it was finished we could add the more or less important details that could contribute to the player's immersion.

Commented [WN1]: This is a very good effort overall. Your game is engaging and atmospheric, mostly with very good attention to detail and only a few minor language issues (sous-sol, zombies, morts-vivants, use of masc_pl to address the player character) to spoil an otherwise professional effect. One or two aspects of the gameplay deserve another look (for example, you could add a script to the dialogue sequence with the counsellor to avoid it repeating) and I encountered a couple of minor bugs (needing to pick up the ring already in my inventory to teleport to the throne room, wizard sequence repeating after they are defeated) that could be solved, but this is already a more than satisfactory effort.
Likewise, your analysis is clear and coherent, one minor critique is that it could go into a little more detail on technical issues, but we can't have everything.
16/20

According to our first idea, we wanted to create an RPG with weapons, attacks, and health points but we quickly realized how complex this option would be. As we really wanted to stick to the RPG principle, we decided to adapt classic FromSoftware game rules such as life bars etc. to focus on the story and ease of play. That's why there's no combat system as such in our game. From the very start of the design, this created a problem for our game and especially its ending: how to defeat the wizards without fighting directly with a combat system.

We were afraid that this would take too much time so we came up with an alternative solution: by recovering an object considered mythical and very powerful, and by bringing it to the right place and using it, we could defeat the enemy indirectly and finish the game. There was also the idea of inserting the object as the centerpiece of a mechanism that would be the cause of the wizards' defeat and return monsters and zombies to their original form, which is the people of the kingdom.

But coming back strictly to the game, with the player starting in a locked dungeon, we had to learn how to create a working key on the door leading to the exit, and a way to hide the key so it wasn't too obvious to find. We chose to place it behind a stone in the wall, defining the latter as a container in the settings. As for other verbs that we will see later, as we could not change the names of the predefined verbs, we had to use another one, namely “open”, and replace it with “move”, attributing to the latter the function of opening the dungeon door.
Once out of the latter, the player can go back to the entrance of the castle via other rooms (stairs, kitchen…). Having created a certain number of rooms, we thought that using images to help the player visualize them and find their way in space could be useful, as well as the descriptions of the rooms. We then added several objects in the following rooms as well as ambient sounds, so that the player can soak up the atmosphere of the place and begin to feed the plot. From the hall of the castle, several paths are now available, all of them are useful for the story but it gives the player a little more freedom in their choices. Later on, when the player arrives at the dungeon, the first interaction with another human character takes place : a former advisor to the kingdom who was stuck escaping the fate suffered by the other inhabitants. This is when we started using "typewriter" which allows us to select how quickly the text is displayed on the screen, this allows us to transcribe the emotions in the words of the characters (faster to show the impatience or surprise and fear, slower for hesitation or to show that the character is reassured for example). We also used a new feature that allows another character to hand us an item directly. The object in question here is an amulet. This amulet is an object which allowed us, once used, to unlock a new path in a room where the player had previously passed. It also indicates the direction of this place which makes it easier to find for the player.

During the creation of these rooms and objects, we encountered several minor problems such as display bugs on certain words, or certain actions which continued to appear even though they had previously been deleted. All I had to do was save and relaunch quest to correct this. We did, however, we have an issue in-game where some text appeared twice, and we were unable to fix this. This is not particularly disturbing for the player, but it is important to note.

Interactive language elements:

Below is a list of all the interactive language elements that are present in the game. The verbs in bold indicate the ones we created, while the other ones are the automatic ones.

Room : dungeon

- Journal : look at ; **read** (the player can read the journal to find out more about the story); take
- Wobbly stone : look at ; **move** (opens the stone which is set as container to reveal the key)
- Key : look at ; take ; **use** (opens the dungeon room)

Room : underground

- Blood stain : **touch** (describes the blood stain)

Room : kitchen

- Butcher knife : look at ; take

Room : courtyard

- Zombies : look at ; **rob** (takes the sword from one of them and adds to inventory)

Room : dungeon tower

- Sword : take ; **hit** (hits the door of the dungeon library to crack it open)

Room : dungeon library

- Counselor (character but will be listed as an object) : look at ; **talk to** (generates a script for a dialog that ends with the player being given the charm)
- Shiny potion : look at ; take
- Work table : **use** (requires the shiny potion, makes a magic sword appear and the previous one disappear)
- Magic sword : look at ; take
- Charm : **use** (needs to be used once in the courtyard, makes an entrance that leads underground appear)

Room : deep undergrounds

- Mystic monster : look at ; attack (runs a script that narrates the defeat of the monster, makes it disappear and make a ring appear on the ground)
- Skeleton of an ancient king on his throne : look at (runs a script that indicate the dagger next to him)
- Dagger : look at ; take
- Ring : look at ; **retrieve** (runs a script that works on a condition : if the player is carrying the dagger, it plays a bell sound and teleports him into the throne room, and if not a message saying “make sure to look everywhere in the room” appears enI)

Commented [WN2]: I encountered a problem after picking up the ring, then the dagger: how do I get into the throne room?

Room : throne room

- Wizards : **talk** (runs a dialog script and plays a whisper sound) ; attack (runs a script that work on different conditions : if the player is carrying the ring, the player has trouble battling the wizards and decides to run to the throne to put the ring on the king, which has just appeared along with the queen after the script began. The player can type in the verb “run” to get to the throne. If he is not carrying the ring, the script of the next level, when the king has become human again, begins, describing the battle of the player, the king and the queen against the wizards).
- King : put ring on (runs a script with the scene of the king and the queen’s awakening).
- Dagger : give (the player gives the dagger to the king that uses it and defeat the wizard).

Most of the objects present in the game were created for the purpose of being useful to the story and helping the player get to the end of the game. We created in total 10 objects that are used throughout the game. One of these, the charm, is given to the main player by the former advisor. Some of the others are taken from other characters such as monsters (the sword, the ring, the dagger). A challenge was trying figuring out how to create an action that requires at least 2 objects : using the sword to break the door of the library or putting the ring on the finger of the king. Trying to do that was actually difficult for the shiny potion found in the dungeon library. We at first wanted to be able to use the potion and directly apply it to the sword, using a verb called “enduire l’épée” and to then create a script that would replace the current sword with a new object, a magic sword. But that wouldn’t work, so we chose to install a work table that “contained” the magic sword, so that when the player uses the table it makes the basic sword disappear and the magic one appear in the room. For an unknown reason the “add to inventory” function wouldn’t work, so we just made it appear.

Another challenge was figuring out the underground monster fight level. There were 2 objects that were needed to be taken from the room, and at first we didn’t put any conditions to these objects. But we realized that the player couldn’t know that these objects were necessary, and as the level ends with a teleportation into the throne room, which is locked otherwise, they would be stuck inside the room without getting the chance to finish the game, making it frustrating. So to make sure that would not happen, we created a script for the verb “récupérer” to take the

ring, so depending on whether the player is carrying the dagger or not, different scenarios would take place. If the player is carrying the dagger, he is teleported to the throne room. If he isn't, a message telling him to make sure he found everything in the previous rooms, preventing him from being teleported without the dagger that will be needed later on.

Making exits appear and disappear was useful, as for example the one leading to the undergrounds from the courtyard. The exit remains invisible until the player is carrying the charm, that indicates the exit by projecting a light in its direction.

Making the game intuitive was challenging, because as creators, we immediately know what to do when arriving in a room or seeing an object. But as we tried to put ourselves in the shoes of the player, we realized that something as simple as interacting with the objects in a different order than planned totally interfered with the story.

To avoid language errors, we had to sometimes create new verbs to fix the syntax, and modify the description automatic phrase of a room (for example the courtyard), using "Vous voyez au loin une procession de....", as the first object described is the zombies. We also had to change the suffixes and articles of objects.