## Amendment history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author (unit)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>29/01/2016</td>
<td>FRE</td>
<td>First draft</td>
</tr>
<tr>
<td>V2</td>
<td>09/02/2016</td>
<td>UOP</td>
<td>First review</td>
</tr>
<tr>
<td>V3</td>
<td>19/02/2016</td>
<td>FRE</td>
<td>Second draft</td>
</tr>
<tr>
<td>V4</td>
<td>04/03/2016</td>
<td>ISEP</td>
<td>Second Review</td>
</tr>
<tr>
<td>V5</td>
<td>04/03/2016</td>
<td>FRE</td>
<td>Third draft (PU)</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 649673. Disclaimer: The sole responsibility for the content of this material lies with the authors. It does not necessarily represent the views of the European Union, and neither EASME nor the European Commission are responsible for any use of this material.
Executive summary

Deliverable D3.1 presents the results of the task T3.1 (Serious game design) and T3.2 (software specifications and architecture). It first presents the serious game design in the form of the Game Design Document, a standard document used in the video games development industry presenting the guidelines to create the game envisioned. It describes precisely the content of the game in terms of game concepts, settings, interfaces, graphic, audio and sound design, game modes, assets, interactions, game mechanics, characters, missions, etc.

The deliverable then presents the artistic direction, describing the visual ideas thought for the EnerGAware serious game. The EnerGAware serious game artistic direction focuses on: the house concepts, as the house is at the centre of the gameplay; the evolution of the house, because it is a customisation game; and the integration to a real environment, because the game must relate to the everyday life of the players as previous works have concluded. The last part provides ideas and concept art around the main character of the game, the cat.

The last part of the D3.1 document is dedicated to the software specifications and architecture of the EnerGAware serious game. It provides all the relevant details to produce the EnerGAware from a software and server architecture point of view. The part relative to the game and game server is managed by FremenCorp and the part relative to the house server, by ISEP.
1 GAME DESIGN DOCUMENT ................................................................. 11
  1.1 THE MAIN IDEA .................................................................................... 11
    1.1.1 Story: the smart cat, king of the house ........................................... 11
    1.1.2 Game systems .................................................................................. 11
    1.1.3 The main game mechanics ............................................................... 12
    1.1.4 The main gameplay loop ................................................................. 13
    1.1.5 Balance between happiness and energy points ................................ 15
    1.1.6 Social dimension of the EnerGAware serious game ....................... 15
    1.1.7 Real consumption and game impact ................................................. 15
    1.1.8 Behaviour change objectives ........................................................... 16
  1.2 GENERIC GAME SETTINGS ................................................................. 16
    1.2.1 Interface ........................................................................................... 17
    1.2.2 Language .......................................................................................... 17
    1.2.3 Difficulty default settings ................................................................. 18
    1.2.4 Graphics default settings ................................................................. 18
    1.2.5 Controls default settings ................................................................. 18
    1.2.6 Music, Sound and FX settings .......................................................... 18
  1.3 IN-GAME MODE ...................................................................................... 19
    1.3.1 Camera: Field of view and Angle ..................................................... 19
    1.3.2 Controls: Finger(s) touch ................................................................. 20
    1.3.3 Messages and pop-up ...................................................................... 20
  1.4 EDITOR MODE ......................................................................................... 21
    1.4.1 Items’ information ............................................................................. 21
      1.4.1.1 Energy points consumption ....................................................... 21
      1.4.1.2 Happiness .................................................................................. 22
  1.5 RESOURCES .......................................................................................... 22
    1.5.1 Energy points .................................................................................... 22
    1.5.2 Unlock new items ............................................................................. 22
    1.5.3 Money .............................................................................................. 22
    1.5.4 Energy points, money, and happiness system .................................. 22
  1.6 CHARACTERS ......................................................................................... 23
1.6.1 Cat ............................................................................................................................................. 23
1.6.2 Humans ..................................................................................................................................... 23
1.6.3 Neighbours ............................................................................................................................... 23
1.7 SYSTEM INFORMATION ................................................................................................................................ 24
1.8 MISSIONS .................................................................................................................................................. 25
1.8.1 Global system .................................................................................................................................. 25
1.8.1.1 Triggers ................................................................................................................................................... 25
1.8.1.2 Interface ................................................................................................................................................ 25
1.9 TIME IN GAME ............................................................................................................................................ 26
1.9.1 Real calendar events .................................................................................................................. 26
1.9.2 Day and night cycle .................................................................................................................... 26
1.9.3 Seasons: in-game effects ............................................................................................................ 27
1.10 SPECIAL EVENTS ..................................................................................................................................... 27
1.11 ONLINE AND OFFLINE ............................................................................................................................ 27
2 ARTISTIC DIRECTION ........................................................................................................................... 28
2.1 HOUSE CONCEPT ....................................................................................................................................... 28
2.2 HOUSE DESIGN ........................................................................................................................................... 28
2.3 UPGRADES ................................................................................................................................................ 33
2.4 CHARACTER DESIGN .................................................................................................................................. 35
3 SOFTWARE SPECIFICATIONS AND ARCHITECTURE ............................................................................ 38
3.1 ARCHITECTURE AND TECHNICAL SPECIFICATIONS .................................................................................. 38
3.1.1 Game architecture .................................................................................................................................. 38
3.1.1.1 Global architecture ............................................................................................................................. 38
3.1.1.2 House server architecture ................................................................................................................... 38
3.1.1.3 Game server architecture .................................................................................................................. 40
3.1.1.4 Game client architecture ................................................................................................................... 41
3.1.2 Performance constraints on the game client .................................................................................... 42
3.1.2.1 Hudl2 is our target device ................................................................................................................... 42
3.1.2.2 Customisation game specifics ........................................................................................................... 42
3.1.2.3 Mobile game specifics ........................................................................................................................ 42
3.1.3 Build ........................................................................................................................................... 43
3.1.3.1 The .apk extension ............................................................................................................................... 43
3.1.3.2 Provide the build ................................................................................................................................. 43
3.2 GAME ENGINE, LANGUAGES AND TOOLS .....................................................................................................43

3.2.1 House server framework, languages and tools ..................................................................................43

3.2.1.1 FIWARE framework ...............................................................................................................................43

3.2.1.2 Runtime and languages .....................................................................................................................44

3.2.1.3 Data ontology ......................................................................................................................................45

3.2.2 Game client engine, languages and tools ..........................................................................................45

3.2.2.1 Unity game engine ..............................................................................................................................45

3.2.2.2 C# language ........................................................................................................................................45

3.2.2.3 MonoDevelop .......................................................................................................................................46

3.2.2.4 Android SDK ..........................................................................................................................................46

3.3 SECURITY ...................................................................................................................................................46
Figures

Figure 1: Simplified version of the main gameplay loop ................................................................. 14
Figure 2: Detailed version of the main gameplay loop ................................................................. 14
Figure 3: Left. Mockup for the global game settings interface. Right “Game settings” icon .......... 17
Figure 4: Mockup for the validation and saving menu ................................................................. 17
Figure 5: “Disabled FX sound” icon ............................................................................................. 19
Figure 6: Camera angle .................................................................................................................... 19
Figure 7: Finger(s) touch controls ............................................................................................... 20
Figure 8: Concept for messages and pop-ups .............................................................................. 21
Figure 9: 5 Stars’ gauge ................................................................................................................. 22
Figure 10: Mockup for the loading system interface ................................................................. 24
Figure 11: Concept for loading screens ....................................................................................... 25
Figure 12: Concept for the mission system menu ........................................................................ 26
Figure 13: Concept House ............................................................................................................. 29
Figure 14: Potential evolution of the house ................................................................................ 30
Figure 15: Concept art adding a user interface ........................................................................... 30
Figure 16: Concept houses in realistic environments .............................................................. 31
Figure 17: Typical Plymouth’s houses adapted for the game ...................................................... 32
Figure 18: Appliances with indicators supporting gameplay .................................................... 33
Figure 19: Databox design and implementation in the house environment ............................... 34
Figure 20: Leading role and players’ avatar of the EnerGAware serious game: The cat ........... 35
Figure 21: Robotic version of the cat ......................................................................................... 36
Figure 22: Concept arts for the cat ............................................................................................. 36
Figure 23: Concept art for the cat ............................................................................................. 37
Figure 24: Global architecture ..................................................................................................... 38
Figure 25: House server architecture .......................................................................................... 39
Figure 26: Architecture of the database and the Django server .................................................. 40
Figure 27: Game client architecture.
Glossary and abbreviations

**API:** Application Programming Interface. The API is a set of functions that aim to be an intermediary between two programs. The first program offers services to the second program through the API. In our case, the game client makes requests to the server to get and post data. The API represents this set of requests, it is the entry point of the client to get data from the server. The purpose of this architecture is the client does not need to know how the server works to get the information.

**Build:** The build is the package that contains the final game compiled and playable on the computer. For example, on a standalone PC game, the build is the .exe file you double-click on to open the game. A build for Android consists in a “.apk” package that can be installed on Android devices. In this document, we opposed the “built version” to the “editor version”, which corresponds to the whole unity project developers are working on.

**HUD:** Head-up display. The HUD is the group of information of the interface displayed at the screen borders during the game (its position is different from one game to another). The HUD often contains the main information about the player status. In EnerGAware, the HUD shows the player money, energy, happiness points, etc.

**Mesh:** A mesh is just a 3D model, regardless of any texture, material or script. It is made with a 3D modeller tool (Maya, 3Ds Max) and composed of a number of faces. The most faces a mesh has the most graphic instructions requires to display it. This is why a mobile game must be “low polygon”. This means all meshes contain the minimum faces and all the small details (for example, buttons on a microwave) will be painted directly on the texture that will be applied on the object.

**Shader:** Each object of the game has a material. Materials are definitions of how a surface should be rendered, including references to textures used, colour tints and more. The available options for a material depend on which shader the material is using. Shaders are small scripts that contain the mathematical calculations and algorithms needed to calculate the colour of each pixel.
rendered. If a shader is too complicated and contains a lot of calculations, it could be impossible for the hardware to display the game with a good framerate.

**Texture Atlases:** Each 3D model has a “texture”. A texture is an image (jpg, png, tga...) where the object faces are painted. A texture atlas is a texture where several objects faces are painted. These objects share the same texture, so if they are displayed in a view, only one texture is actually displayed, and it can help reduce performance demand if few conditions are respected.

**UI:** User Interface. Also called GUI (Graphical User Interface), is a type of interface that allows users to interact with a device through graphical icons and visual indicators. In 3D video game development, the UI is opposed to the 3D game scene. The UI represents all the 2D elements of the screen the player interacts with (dialog boxes, pop-ups and HUD buttons, contextual menus, etc.).
1 Game Design Document

1.1 The main idea

The following sections introduce the global game concept proposed at the time of this deliverable. The player plays and controls a cat who enjoys being at home and dreams about living in a smart, comfortable and energy efficient house of the future, such as the one it can see in a magazine. With that dream in mind it has a plan: rule its kingdom to make it evolve and ultimately reach its goal, its smart and energy efficient dream house. To accomplish its secret mission, the cat will need to manage the humans living in his house, helping them to behave more energy efficiently.

1.1.1 Story: the smart cat, king of the house

The cat is currently perceived as a cute, smart and popular animal on the internet. It represents a way to connect with a various range of players while providing a light and fantasy note on a game that is about the serious subject of reducing energy consumption.

Within the game, the cat will be presented as the true ruler of the home: humans feed it when it demands, they open the door as soon as it wishes, it sits where it wants, etc. The cat will be also introduced as an intelligent animal, who will be able to give advice to humans. It will have his own personality, ask itself questions, and even comment and joke about humans’ inappropriate energy efficient habits.

1.1.2 Game systems

The EnerGAware serious game is composed of two game systems (also called game “modes”) that interact to create a unique game experience: A house customisation and a mission mode. The house customisation mode is based on the game “The sims” in which the player is able to create his/her dream house without any restrictions. The implementation of this system in the EnerGAware serious game follows the same general principles by providing an “Editor” function with which the player will buy new appliances, insulation and decoration in a realistic environment. The focus will however be placed on energy consumption and energy efficiency of the installation the player created. As a cat, the player will be a spectator of the events and also benefit from the counsel of the cat, who will guide her/him and the humans through the game. For instance, the cat may provide feedback about thermal comfort and energy consumption issues.

The house customisation mode will be at the centre of the game world. A mission mode will provide stories and background that will instill at the same time knowledge about energy efficiency and educate the player about right energy management behaviours. Missions will be given directly in
the player’s house, but missions about specific energy efficiency issues will be provided in additional locations, the neighbours’ houses. The neighbours’ houses will provide scenes tailored for issues that cannot be shown in the player’s house (e.g. a house with infants, external wall insulation issues), but also provide ideas about how the player’s house could evolve. Furthermore, the neighbours’ house will have a fixed geometry that will allow providing precise energy consumption simulations obtained by UOP’s energy simulation engine from T3.4 (contrary to the player’s house that will be unique to each player, and change frequently).

Simulations will also be provided for the player’s house, but due to the highly variable nature of this part of the game, UOP cannot provide precise simulations for the near infinite number of possibilities. Simulations have been run for 369 combinations containing variations on the size of the house, extensions on the four cardinal directions and upward with a floor, and each time with three different qualities of wall and roof insulation, as well as windows and boilers.

Every day the player may complete missions that will reward him/her with extra money, special items, or a happiness boost to motivate the humans and make them bring more daily money or perform extra actions. Special missions with middle and long term goals could motivate the player to save energy points. It would reward them by unlocking new items or sets of items (e.g. more efficient, lower consumption...).

1.1.3 The main game mechanics

The main strategy of the game focuses on resource management and house customisation to make the player’s “kingdom” evolve. The two main resources are money and points representing the impact of the actions of the player “energy-wise”. Such points could be called “Energy points”, “Biodiversity points”, “Future points” or “anti-waste points”. Focus groups held during the production of the EnerGAware serious game with UOP will help identify the best label. The term “Energy points” will be used in this document for the sake of simplification.

Energy points is the most valuable resource of the game. The player directly has an influence on the number of energy points he/she will save. Saving energy points allows unlocking game content (such as new appliances and house furniture) but could also be converted into money to reach secondary goals. Energy points can be saved by upgrading electric appliances as well as boilers for more energy-efficient models; improving the building’s thermal performance of external walls, roof and windows; change the behaviour of the humans living in the house towards energy efficient actions (examples of actions that should be corrected: let the windows opened whilst the house is being heated; use the shower for a long time; leave the light on when the room is unoccupied).
Money on the other hand is not directly saved by the player but by the humans living in the player’s house. Money can then be used to: buy items that have been previously unlocked using energy points; upgrade current appliances or invest in energy efficient upgrades for the house (e.g. improve wall insulation, improve the performance of the boiler, etc.). Every day in the morning, players will get the money earned by the human. This will encourage them coming every day to get their daily reward. Rewards depend on the last actions the player did, but can also be hidden in the house. By regularly providing rewards hidden in the house, it will motivate players to play every day, to look into every room, to look for bad behaviours. Special hidden rewards may also be provided during events like Christmas, Halloween, etc.

Energy points and money interact with a third parameter which we will call happiness, but could be named differently after feedbacks from focus groups (e.g. comfort, well-being). The happiness level of the humans of the house depends on the number of happiness points all the appliances and furniture of the house provide the humans. At the beginning of the game, it will be difficult to have completely happy humans because most appliances are not energy efficient and they provide little happiness compared to the number of energy points they consume. The more the humans are happy, the more the earn money. The game mechanics therefore revolve around using enough energy points to have “happy” humans that earn enough money to buy new content, while saving enough energy points to unlock new game content.

### 1.1.4 The main gameplay loop

The main gameplay loop describes how game mechanics interact with each other to create the whole game. It must be emphasised that presented in a single schematic (see Figure 1 and Figure 2), the main gameplay loop may seem complicated, but the player will experience it and learn it progressively without realizing it.

In its simplified version (Figure 1), the main gameplay loop starts with a daily pool of energy points. The player has a running house with a global energy consumption and she/he will have to save energy points to complete energy efficiency objectives. It will allow him/her to unlock game content, mainly new items and upgrades for already possessed items (e.g. appliances, insulation). New items will consume less energy points and impact favourably global energy consumption. Upgrades and new items will increase global happiness which will in turn increase daily money income. Money is then used to buy upgrades and new items.
Figure 1: Simplified version of the main gameplay loop

The detailed version of the main gameplay loop (Figure 2) adds parameters such as “extra consumption” of energy points that decrease daily money income, missions’ rewards that add extra money earned and that can also unlock new items and upgrades.

Figure 2: Detailed version of the main gameplay loop
1.1.5 Balance between happiness and energy points

The EnerGAware serious game will be deployed to social housing communities in which health, energy consumption and spending money are complex problems. For instance, people in difficult financial situations might want to cut all energy spending, including heating in winter which will cause illnesses and more financial problems later. The concept of happiness allows evoking the subject without dramatic displays, humans being only “unhappy” when living conditions are not good enough. Furthermore, the cat being at the centre of the gameplay, information might be given in a humorous and indirect way. For instance, the cat might sneeze, but not the humans so that the consequence of having a house insufficiently heated is only reflected on the cat.

The delicate balance between energy consumption and occupants’ happiness will therefore be an important part of the gameplay. Reducing the happiness level too much to save energy points will decrease humans’ productivity and as a consequence they will earn less money. The goal is to make the players understand the need to invest in better equipment, smart connected devices and insulation to reduce their need in energy points without decreasing the happiness level of their humans.

1.1.6 Social dimension of the EnerGAware serious game

The EnerGAware serious game has a social impact objective. Energy efficient behaviours should be shared and disseminated in order to induce large scale and widespread results. Cooperative missions are specially designed to generate motivation through challenge by involving part or all of the community. For instance, the game will contain special events relevant to the current season. In winter, a cooperative mission to light the neighbourhood Christmas tree will need each player to save energy points and donate them to complete a stretch goal. At each stretch goal completed, every contributor will win a reward. Each consecutive stretch goal will require more energy points to be completed and will unlock special rewards.

The EnerGAware serious game will also contain features to share screenshots, achievements and tips directly from the game to social networks such as Facebook. It will allow improving the dissemination of the game while building competition and cooperation with a community revolving around energy efficiency and proper energy consumption behaviours.

1.1.7 Real consumption and game impact

The EnerGAware serious game is connected to the real energy consumption of the house. One year prior to the release of the beta version of the game, smart metres should have been installed in the test houses to record a full year baseline of energy consumption. When the beta version will
be released, the energy consumption will still be measured and could have an impact in the game. For instance, each week the energy consumption could be compared to the energy consumption of the same week the year before. If a significant reduction is observed, the player could get an extra money bonus. This bonus could be proportional to the reduction measured to motivate the player. It could also be cumulated should the player not log in every week.

1.1.8 Behaviour change objectives

The EnerGAware serious game serves several purposes: provide educational content about energy efficiency, change energy consumption behaviour and teach through gameplay ways of improving the house fabric (e.g. insulation) as well as selecting better appliances. The whole game has therefore been designed to support this content with gameplay features and interfaces that allow the player to learn through the game.

Several missions will be designed and a story written during the production of the Serious Game. In this document we present a few examples of game features that serve educational purposes:

- During a winter mission, the player will be asked to close a window that was left opened by the humans, adjust the thermostat for optimal heating/happiness ratio, etc.

- During a mission, a human will use the shower for a long period of time, and the cat will have to try to stop him/her in order to reduce hot water heating and therefore reduce energy consumption.

- Interfaces will provide feedback on the real energy consumption of the player’s household, and give rewards when the player decreases his/her consumption relative to the previous year. If the energy consumption did not decrease the player will be provided encouragements and tips to help her/him.

- Interfaces will provide detailed information about energy consumption of various appliances and insulation efficiency as well as hints and tips (“to know more”) to decrease energy consumption and links to relevant websites.

1.2 Generic game settings

Game settings represent the general parameters the player will be able to modify in his/her game. These parameters include the language selection, the graphic quality of the game, the game controls and the sound level of the game.
1.2.1 Interface

The Figure 3 below is a mockup for the global game settings interface (Figure 3, left). It can be accessed via a “game settings” icon on the main game screen (Figure 3, right). When the player taps on this icon, the “game settings” panel appears.

![Figure 3: Left. Mockup for the global game settings interface. Right “Game settings” icon](image)

If the game settings are changed, a pop-up will appear to ask the user if he/she wants to save the modifications (Figure 4). If the player quitting without changing the settings, no pop-up will appear.

![Figure 4: Mockup for the validation and saving menu](image)

1.2.2 Language

The default language is English but French will also be proposed. Nonetheless, the choice for more languages is planned and will be taken into account in the software architecture: all texts of the game will be stored in specific files to ease the translation without having to make multiple modifications spread in the code. In order to have an international impact, more languages may be added by the end of the project should finances allow it.
1.2.3 Difficulty default settings

There is no level of difficulty in the EnerGAware serious game. The missions and objectives will define the difficulty according to a learning curve. Challenges will be accessible in order to motivate the player to get rewards. The player cannot lose completely and go to a “game over”. As in most social games, the player can face problems to solve (i.e. not enough happiness, not enough money, etc.) slowing the progress, but cannot lose everything which would force them to start again from the beginning.

1.2.4 Graphics default settings

For the beta version, the game will be optimised for the Hudl 2 tablet given to the pilot homes. For the final version, it will be possible to adjust settings according to the performances of the tablet, so the game can be more beautiful on powerful tablets (better lighting, better textures definition).

1.2.5 Controls default settings

“Finger touch” is the unique control system of the game on the Hudl2 tablet. There are no parameters to modify in the setting. However, during the development, some versions of the game will be made available on PC with mouse and keyboard controls, mainly to demonstrate the game during events or presentations. But these controls won’t be maintained for the end-user version.

1.2.6 Music, Sound and FX settings

No music will be included in the game, as most of the social games today are played with no music.

Effects (FX) represents all the jingles, feedbacks sounds, etc. that will help the player or bring some immersion feeling. Sounds are often used in games when: unlocking items, completing missions, failing missions, earning money, performing contextual actions and touching a button of the interface.

A 100 % graduate gauge will allow the user to change the game sound level (the default sound level is set at 50%). A cursor on the gauge may slide with a finger movement to increase or decrease the sound level. The global gauge is divided in 10 segments, each one represents 10% of the sound level. The current level is clearly indicated above the cursor.

FX sound on/off: The player may toggle on / toggle off the sound FX.
Graphic Feedback: A disabled sound and FX icon is visible on the in-game and editor interface (Figure 5).

![Disabled FX sound icon](image)

Figure 5: “Disabled FX sound” icon

Sound Feedback: Each segment reached will play an FX sound while the sound level increases or decreases in order to display the sound level of this segment.

1.3 In-game mode

This mode is the “action” game. Player may interact contextually with the items and the cat’s avatar. This mode offers missions to the player, which she/he will need to complete to gain rewards (money, items, bonuses, etc.). Interactivity, feedbacks, rewards and messages represent the biggest part of this mode.

1.3.1 Camera: Field of view and Angle

The Field of View (FOV) should be between 35° and 50°. This value should be enough to offer a good game experience and to let the player have an efficient perception of the rooms. Note: Tests will be needed in order to set the final FOV value. Movements of the camera and feedbacks may influence the settings. The angle of the camera can be seen in Figure 6.

![Camera angle diagram](image)

Figure 6: Camera angle
1.3.2 Controls: Finger(s) touch

The EnerGAware serious game is released on tablets with a touch interface. The Figure 7 shows the five types of touches the player will be able to perform in the game.

![Figure 7: Finger(s) touch controls](image)

1.3.3 Messages and pop-up

In order for the player to interact efficiently with the game, she/he needs to stay informed about what is happening. Messages will provide such information and can be given in the form of pop-ups that appear at various places on the screen depending on their importance (e.g. in the middle if the message is important, or on a corner if it is a minor notification).
A message in a pop-up might appear when the player: opens a detailed mission, unlocks a new feature, finishes a mission, earned money, double taped on an item to see its statistics or if there is a special event. They might contain: pictures, icons, text, action buttons (e.g. Ok, Confirm, Cancel, Claim, Close)

1.4 Editor mode

The editor mode allows the player to customise the house. While the game is in Editor mode, all the characters stop their current actions. The player can buy, move, add or remove any item he/she may have on the scene.

1.4.1 Items' information

1.4.1.1 Energy points consumption

Each item has a consumption rate which value (to be defined with UOP) could depend on the ranking consumption of the item. The value may be 0 to 10. “0” being possible for any item that does not consume energy (e.g. furniture).
1.4.1.2 Happiness

The happiness level could be a 5 stars’ gauge with each item having a happiness rank from 1 to 5 stars (1 being the lower happiness level and 5 the best – Figure 9).

![5 Stars' gauge](image)

Figure 9: 5 Stars’ gauge

1.5 Resources

1.5.1 Energy points

As stated in 1.1.3, the impact of the player regarding energy matters will be represented in the form of points. We use the term “Energy Point” or E.P. although it will most likely change after focus groups are held and UOP defines the best label to use. Energy points is the main resource of the game. It represents any kind of source of energy that can be used in the house to feed all the devices and household appliances.

1.5.2 Unlock new items

Several items will be locked at the start of the game. Statistics (such as energy consumed by the appliance) and design will be visible in order to motivate the player to unlock them. Each unlocking step is linked to a bundle of items and need a certain amount of saved E.P. to be reached: as soon as the player reaches a step to unlock, items will be available and buyable (note: These figures will need to be tuned to give a good progression feeling).

1.5.3 Money

Money is one of the resources of the game the player will have to manage to buy more energy efficient appliances, or upgrade the house thermal performance (e.g. higher levels of wall insulation).

1.5.4 Energy points, money, and happiness system

Energy points, money and happiness are linked because happiness increases the money earned by the humans, but happiness also depend on items that cost energy points. A system of chain reaction could be used in order to balance further those resources. When the player does not have enough E.P., the amount of money could be impacted by decreasing progressively as an extra consumption fee. Should the player not have money anymore, the items would automatically be
tuned off. In order to avoid frustration or misunderstanding, a pop-up “Cat’s advice” could appear to offer some help to the player. (Money could be offered for a single time).

1.6 Characters

1.6.1 Cat

The cat is the main character of the game and the only one that can be controlled by the player. In the game mode the player can control the cat. The cat can move to the position pointed by the player, with a single tap. The camera does not follow the cat. The player always controls the camera to allow him/her to do other actions while the cat is doing something (move, contextual actions…). Contextual actions are possible: the player needs to tap an item which has contextual actions (e.g. turn on or off action). The cat will move directly to the targeted item in the scene and then will play the animation of the action.

1.6.2 Humans

Humans are Non Player Characters (NPC) and are controlled by the computer. They are the ones who bring daily money and have weird behaviours according to the cat.

Humans NPCs can do actions according to their behaviour. NPC’s behaviour is scripted and follows the Artificial Intelligence (AI) program. NPC may move to any object in order to “simulate” the life of real humans. The AI program selects actions to perform from a list of action relevant to the objects present in the scene.

Human NPCs may have an awareness behaviour. They could detect the cat when the player tries to make it do an action that would be suspicious according to a human (ex: turn off the light). In such event, the NPC would play an animation while the cat’s action would be cancelled. The human could stay a moment in the same room or could follow the cat (during few seconds) to try to spy the animal. During that time, the player would not be able to make the cat do any action.

1.6.3 Neighbours

Neighbours are NPCs that may request the player’s help in several kind of situation (e.g. advices to choose an energy provider or actions such as turning off all the lights).

They will be used to show to the player, different kind of house settings with rare unlocked items to motivate her/him to reach the right amount of saved EP to get them too. Neighbours will allow the player to observe how a better or worst insulated house may influence the energy consumption;
the player will be able to check the statistics, real time energy consumption rate, etc. of their items and insulation.

1.7 System information

1.7.1 Loading system

A game is composed of a large amount of data that needs to be retrieved in the computer when the game requires it. Because a whole game cannot usually be loaded in the Random Access Memory (RAM), parts needed at a particular point are loaded. This process takes time, and the player must be informed that during this period of time, the game cannot be played. An interface shows the loading process as a bar that fills progressively.

The loading screen appears when the game data are loading after the splash screen (i.e. the first image of the game when it is launching) and when the user selects a profile. Wise or smart messages, advices or energy information may be showed during the loading screen. Several messages could be randomly selected from a list (Figure 10).

![Mockup for the loading system interface](image)

Figure 10: Mockup for the loading system interface

While the game is loading, random advices and funny messages from the cat could be picked from a pool and appear on the screen (see Figure 11 for concepts).
1.8 Missions

1.8.1 Global system

1.8.1.1 Triggers

A trigger causes something to happen in a game. In the case of missions, several actions and situations may trigger a new mission. It may be that the previous mission has been completed, that a specific mission has been completed (using the ID of the mission), that the date corresponds to a specific event (for instance in case of special events like Halloween and Christmas), that a specific item (potentially required for the mission) has been unlocked, that a specific amount of resource has been attained (E.P., money, happiness), that the player reached a specific location, that a new neighbour has been unlocked and because of the player’s activity or scripted events. A mission may have one or several triggers to be activated (e.g. Getting 3 000 E.P. and completing a specific mission).

1.8.1.2 Interface

The mission system is the second largest part of the EnerGAware serious game. A specific interface would help the player know the missions available at any point in time, the current progress on active missions, the intermediary and final objectives of the missions, as well as the rewards they could earn (see Figure 12).
1.9 Time in game

The player’s might want to play for a long time, which would consume a large part of the content rapidly. In order to be played for the whole year duration of testing, a daily session will be tailored to last between 15 and 20 minutes.

1.9.1 Real calendar events

The game will follow our everyday real calendar (e.g.: Christmas will be the 25th of December). There might be a synchronisation with the server to check the dates when the game will be released, in order to be sure the dates match.

1.9.2 Day and night cycle

Day and night influence humans’ behaviour and activities. Some missions may depend on the moment of the day they were activated or completed. The system simulates the cycle but in a shorter time represented in 2 turns: “day turn” and “night turn”. Each turn duration will be around 10 to 20 minutes in order to let the humans do actions and players complete some missions. Day and
night influence the energy consumption. The player can wait the end of the “day turn” to let the system switch automatically or may switch manually by using a button.

1.9.3 Seasons: in-game effects

With seasons changing, insulation will play an important role in the energy consumption. Global consumption will be affected by a modifier according to the season. This parameter may be tuned for the whole season or for a special event day (ex: exceptional blizzard event). The modifiers will be tuned according to the energy simulation engine provided by UOP (T3.4 – M18).

Ground texture modification and FX sounds will increase the immersion of the player. Snow, fog, butterflies, leaves, rain, flowers, rainbow, ..., may be part of the whole graphic impact. FX sounds may be added in the background to recreate season atmospheres like: bugs, birds, barking, rain, wind... Season could also unlock special items to decorate the outside: barbeque, snowman, ...

1.10 Special events

Special events can be a celebration (birthday, Christmas, Saint Patrick, Halloween, etc.), or a special scripted mission for an educational purpose. Events unlock rare special items according to the theme. ex: Christmas tree, pumpkins, ...

1.11 Online and Offline

Playing online: While the user plays the game with an internet connexion, data may be exchanged between the server and the application. Statistics may be saved on the server and the server version can check and inform the player if a new version may be updated on the tablet.

Playing offline: The player can continue to play the game but all the saves will be done temporarily on the tablet. As soon as an internet connection is accessible, the game will update automatically the data and if needed: download new parameters and/or new versions, and upload the local save of the game.
2 Artistic Direction

Artistic direction handles the artistic definition of a content (here a serious game) from the proposition of ideas in the form of concept arts to the final design that will be released. It is therefore important to note that the artworks presented below are all concept arts that will never be implemented in the EnerGAware serious game. They are used as propositions, and as such, most of them will be discarded because they do not fit in the global design ultimately chosen. Furthermore, the ones that are chosen merely guide the artistic conception of the game, acting as a broad objective to reach rather than a specific content to produce.

As a consequence, some aspects of the final result might be close to the initial art concepts, others will show improvements and others weaknesses. Technical constraints and gameplay will greatly impact the final artistic design. The EnerGAware serious game early prototype (of which many screenshots are provided in 1 Game Design Document) followed the guidelines of the artistic direction, but is also a temporary version of the game as well as of the implementation of the artistic design.

2.1 House concept

The House represents the central element of the EnerGAware concept: “Its heart must beat”. Preliminary research has been performed to determine the design of “the house of the future” without imposing constraints. It allowed providing numerous ideas from which the best were selected.

Buildings from contemporary architects, especially those reflecting eco-design have been the primary inspiration for the artistic direction. Supplementary research has been performed on insulation methods and natural as well as renewable sources of energy.

2.2 House design

The mood board has been composed mainly with ecological elements such as grass, water and wood. A cubic approach was chosen as the game is a Sims-like with basic 3D graphics. More modern materials similar to those found in shipbuilding, aerospace manufacture and the car industry have been added in order to bring futuristic features to the house.
Breaking the right angles brings more roundness and aesthetics to the global design of the game. The overall aim is to create a singular and recognisable style that allows identifying the houses of the EnerGAware serious game in any context, thanks to a shape-centered artistic direction.

Figure 13: Concept House

A symbolic significance of the natural evolution of the house appears step by step. The number of ways to optimise the house increases as it evolves.
Figure 14: Potential evolution of the house

The concept arts are then adapted to a game environment, adding a User Interface (UI) on a slightly isometric perspective.

Figure 15: Concept art adding a user interface
The outdoor design of the concept houses has been transposed in a realistic environment from Plymouth (in which the game will first be tested) that projects artistic ideas for the game in a credible existing style.

Figure 16: Concept houses in realistic environments
Figure 17: Typical Plymouth’s houses adapted for the game.
2.3 Upgrades

The design of some appliances supports the gameplay, helping the player optimising the house by indicating energy use.

![Figure 18: Appliances with indicators supporting gameplay](image)

The “Databox” is a unique recognisable object the player can refer to in order to get energy consumption information as well as controlling some appliances.
Figure 19: Databox design and implementation in the house environment
2.4 Character design

As the most important feature of the EnerGAware serious game, the house is a vessel and its residents are the passengers. The living mind of the house is embodied by a cat, a pet that is always present and knows everything that happens. The player will interact with the game through its omnipotent presence. The cat, mascot of the EnerGAware serious game, will bring a cozy and friendly side to the game.

Figure 20: Leading role and players’ avatar of the EnerGAware serious game: The cat

The human characters are nevertheless still important. As supporting roles, they will represent a drive to become better, even when their presence is only suggested. The storyline will contain successive missions and tasks showing how human’s behaviour can be improved to decrease energy consumption. The player, through the cat, will be there to ensure the optimisation of the energy consumption and that everything is functioning well in the house.
The cat’s design has not been fixed yet. However, ideas are focused on a stylistic design that must be linked to the story. Furthermore, the cat could incarnate our ever evolving lives, as a robot, or part robot that resonates with modern smart houses in which automatic vacuum cleaners move around as if they were pets.

Figure 22: Concept arts for the cat
The cat, like a human being, has his own personality traits such as laziness, nonchalance, trickery, etc. These traits will be used, diverted and exaggerated to make it a mascot appreciated by the gamers' community.

Figure 23: Concept art for the cat
3 Software specifications and architecture

3.1 Architecture and technical specifications

3.1.1 Game architecture

3.1.1.1 Global architecture

The EnerGAware serious game (also referred to as “Client”) relies on two servers, the “House” server and the “Game” server that interact with each other through an Application Programming Interface (API). The game server interacts with the Client with an API. ISEP will manage the “House” server and FremenCorp the “Game” server and the “Client” (Figure 24).

Figure 24: Global architecture

3.1.1.2 House server architecture

The House Server is a repository that aggregates data from diverse sources that will permit to quantitatively assess the potential energy savings during the time period of the project (Figure 25).
The House Server is based on the FIWARE framework (https://www.fiware.org/), a middleware platform for development of Future Internet applications, which is sponsored by the European Commission. At its core, the Orion Context Broker is a publish/subscribe context broker that provides two interfaces (NGSI9 and NGSI10) through a REpresentational State Transfer (REST) service\(^1\). The Orion Context Broker communicates with specific context updaters, encapsulating data with the JavaScript Object Notation (JSON).

The Orion Context Broker uses an underlying MongoDB database that, unlike relational databases, also uses the JSON format on a document style table to store data.

Specific EnerGAware Context Updaters modules, executing in a Node.js runtime environment, are used to interoperate with external systems. One module fetches the energy metering readings from

\(^1\)https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/NGSI-9/NGSI-10_information_model
the Advantic server, through a Simple Object Access Protocol (SOAP) web service; another is responsible to fetch weather data of Plymouth². Other two connect with the Game Server to collect game progress tracking data and to inform about weekly energy savings.

3.1.1.3 Game server architecture

The “Game” server manages the data generated by the “Client” and redirects them on a Database which is a PostgreSQL server that stores anonymous data from the “House” server but also game statuses and tracking data from the “Client”. The relation between the “Game” server and the Database is a CRUD which stands for “Create, Read, Update and Delete” (Figure 26).

Figure 26: Architecture of the database and the Django server.

² Due to the unavailability of online services from the Plymouth weather station, the context updater extracts information from monthly log files.
3.1.1.4 Game client architecture

The game “Client” architecture is separated in three levels: Data, Gameplay and User Interface (UI) and Scene. The data part is composed of dynamic data that represent all information that can change throughout the player’s game (e.g. player’s money, state of the virtual house), and of static data that will not change (e.g. graphic assets, game rules).

The gameplay part contains the different modes of the game: the object mode (manipulating objects) and the house mode (manipulating walls, etc.) that are the two part of the “editor” mode, the game mode in which the player manipulated the cat. They are all linked through the mission system.

The User Interface and scene part represents the part the player is going to interact directly with: the contextual view (to interact with objects), the scene content that adds the user interface composed of the Head Up Display (HUD) and the missions’ UI.

Figure 27: Game client architecture.
3.1.2 Performance constraints on the game client

3.1.2.1 Hudl2 is our target device

The guideline of all performance constraints is that the game needs to work fluently on the chosen Android tablet Hudl2. From this point, we evaluated the necessary graphic constraints to reach a correct performance level.

3.1.2.2 Customisation game specifics

The main feature of a customisation game is that nothing is ever fixed. The player can add, delete, move and rotate the objects, as well as changing the size and number of rooms. The consequences are the following:

- The scene needs to be handled as a grid with separated tiles on which objects can be placed. The objects take space in number of tiles (minimum size is one tile).
- All 3D objects in the scene are distinct meshes that can be moved independently of each other.
- It is impossible to create a lightmap for the whole scene, as it can change completely from one frame to another. Lightmaps work only for static objects.
- Even if we use texture atlases, the player can create a room with many different textures. Indeed, if a view contains two objects that share the same texture (by using an atlas), it reduces performance cost. But if the player puts in a same view only very different objects that are not on the same texture, the result is the same than if there were no atlases at all.

3.1.2.3 Mobile game specifics

Mobile devices are characterized by low GPU (Graphics Processing Unit) and CPU (Central Processing Unit) performances compared to a PC. When targeting a mobile platform, optimization must be a considered from the beginning of the project.

Many rules exist to increase performances on mobile. We present below a few examples:

- Limit the number of triangles and materials in the scene
- Use very simple shaders
- Use texture atlases and compressed textures
- Avoid expensive features like dynamic lightning, alpha blending, camera effects
Limit the number of separated objects in the scene

The last example is the most difficult in a customisation game. All objects must be independent to be moved in the scene. All other points are important, but the last one has an overall heavy cost for graphic rendering.

### 3.1.3 Build

#### 3.1.3.1 The .apk extension

The final project will be delivered as an apk file. The apk extension corresponds to the Android application package. Executing this file, the player will be able to install the game on his Android device.

#### 3.1.3.2 Provide the build

There are two ways to install an apk file on an Android device:

- From Google Store.
- From any other source than Google Store. In that case, the user needs to activate "authorize unknown sources" in the device parameters. It means the user agrees to installations of apps that are not directly provided by Google Store.

If there is no Google account created for the EnerGAware project, the build will be hosted somewhere else (to define). If that is the case, the web page that will allow downloading the app should be presented as a very simple tutorial that explains users the process to install the build. This is the solution chosen by Amazon for instance.

### 3.2 Game engine, languages and tools

#### 3.2.1 House server framework, languages and tools

The development of the House Server uses the following technologies.

#### 3.2.1.1 FIWARE framework

FIWARE (www.fiware.org) is a middleware platform that establishes a set of open, public, royalty-free Application Programming Interfaces (API), with the intention to facilitate the development and global deployment of services and applications for Future Internet.
FIWARE is supported by the European Commission. From this framework, the Home Server is using the following components: Orion Context Broker and MongoDB and will provide data information through Wirecloud.

**Orion Context Broker**

This is an implementation of the FIWARE Generic Enabler (GE) publish/subscribe context broker. This broker allows client applications to register, update and query context information. Additionally, the Orion Context Broker can notify client applications regularly at a selected frequency, or when selected events (modification of selected data) take place.


**MongoDB**

MongoDB is a document-oriented database in which data is stored in documents, following the JSON format.

Link: [https://www.mongodb.org/](https://www.mongodb.org/)

**Wirecloud**

Wirecloud is a web application mashup platform, where it is possible to add widgets related to data filtering and visualisation. This will be used to provide the visualization layer of EnergAware data.


### 3.2.1.2 Runtime and languages

**Node.js**

Node.js is an open-source cross-platform runtime environment for server-side Web applications. Applications are coded in JavaScript, but executed in this environment at the server-side.

Node.js is used to execute the Context Updaters that establish the communication between the Home Server and external systems.

Link: [https://nodejs.org/](https://nodejs.org/)

**Javascript language**
Javascript is an interpreted programming language with many similarities with the Java programming language. It was originally developed to execute in web-browsers, but, currently, Javascript is also used to code server-side applications that are executed in the Node.js environment.

Link: http://www.ecmascript.org/

3.2.1.3 Data ontology

Data is persistently stored in the House Server following the recent Smart Appliances REFerence ontology (SAREF) European standard. This standard has been developed by the European Commission in cooperation with industry and the European Telecommunications Standards Institute (ETSI), to establish a common language for energy-related information, with the goal to permit the development of products and applications to improve the efficient use of energy.

Link: https://sites.google.com/site/smartappliancesproject/home

3.2.2 Game client engine, languages and tools

3.2.2.1 Unity game engine

Unity is a game engine for creating 3D and 2D game. With Unity, it is possible to create complex scenes with a powerful editor and control the game through scripting. The basic version contains comprehensive set of tools that allows developers to script behaviours for characters, cameras and custom object. Using the editor, the lightning of your scene and the graphics settings can easily be managed taking into account the launching platform and the performance constraints.

Unity provides the following advantages:

- FremenCorp already has expertise in using this engine for gameplay development.
- Unity makes the Android development almost as easy as for a classic PC game.
- The documentation is clear and up to date, and there is a large community that can provide help on any matter.

Link: https://unity3d.com/unity

3.2.2.2 C# language

Unity scripts can be written in three different languages: C#, UnityScript (JavaScript for Unity) and Boo. EnerGAware scripts are written in C#, as we have an experience of this object-oriented
language for Unity. C# is also the preferred language of the community, so Unity's documentation provides the best support for C#.

### 3.2.2.3 MonoDevelop

MonoDevelop is the standard IDE provided with Unity. We use MonoDevelop as it offers the basic features we need to code.

### 3.2.2.4 Android SDK

Android SDK is the development kit a developer requires to code an Android application. The game engine, Unity, is able to launch an Android application by using the Android SDK setup on the computer.


### 3.3 Security

The development of the project components will adhere to standard security features, both to prevent illegal access to the game data (which can be used to cheat) as well as unauthorized access to the information being stored by the project.

Hacking a game is always possible for someone who has the skills, the means and the motivation. Therefore, the security subject encompasses the different means (shields) that can be coded in the game in order to discourage the hackers.

- For the online part, mandatory measures should be encrypted data communication and an authentication mechanism (which remains to be defined).
- For the local part (apk package and offline saves), three layers of protection could be used to protect the sources and an eventual offline save: checksum, encryption, obfuscation.

Concerning the information stored by the project, as specified in the Data Management Plan, no information is stored online that can correlate the information with actual houses. Nevertheless, it is important to prevent unauthorized access to detailed data (the correlation of social and family aggregate information can give hints on potential tenants), as well as guaranteeing data integrity. This will be done by:

- Encrypted data communication between all components of the data platform.
— Definition of access protection layers. Access to aggregate data might be made publicly available, but access to detailed data will make use of an extra authorization procedure.

— Data updates will only be allowed through very specific and limited functionality components.