Adaptive Gameplay for Mobile Gaming

Yannick Francillette
Université du Quebec à Chicoutimi
LIARA
555, boulevard de l'université
Chicoutimi (Qc), Canada G7H 2B1
Email: yannick.francillette1@uqac.ca

Abdelkader Gouaich and Lylia Abrouk
Université de Montpellier
LIRMM
161, Rue Ada
Montpellier, France 34095
Email: (gouaich, abrouk)@lirmm.fr

***Abstract*—In this article we propose a game design approach to build context adaptive games. This approach is based on a model of the game structure and a generic adaptation model. Our method consists of designing different game scenarios involving different gameplay for the game then, game engine selects and proposes the appropriate one according to the context. We have conducted a pilot experiment in order to evaluate the effect of this adaptation on game and player satisfaction. We stated as hypotheses that context adaptation does not have effects on the game and player satisfaction. We have developed two versions of a mobile game and a virtual city to test these hypotheses. The results show that context adaptation increase the number of game objectives which are succeeded and decreases the number which are failed. However, player satisfaction is not always superior for the version that includes context adaptation.**

I. Introduction

Nowadays, mobile games are widely present in video games landscape. This kind of game has gained in popularity very much during the last decade. The success of mobile devices such as smartphone, tablet and application stores can explain this growing interest for mobile games. Indeed, these devices represent a suitable video gaming platform for most people. They can take them away and play everywhere and anytime.

In fact, mobile and connected devices are the starting point of ubiquitous computing. This expression defines an era where computing can be symbolized by a set of small devices that are everywhere in the environment and which interact with several users [1]. Computers are not set into one room and used by few people at a specific time. Computers are always connected and can be used anywhere and whenever.

In this technological context, game developers have access to new elements that make the creation of new gaming experiences possible. We can cite location-based games as an example of these new experiences. In these games, players must change their locations or do some actions in the real world in order to progress in the game. These game mechanics are not really new. However, technological context makes it easier to implement. Consequently, it helps in designing mechanics that are more rich, complex and deep. These new technologies provide game industry and researchers with new perspectives. In this document, we use the expression "mobile gaming" to refer to this kind of game.

Mobile gaming provides game developers with new design challenges. In this article, we are interested in issues related to maintain Flow state of players. Indeed, a mobile gaming is a subcategory of games. Consequently, they must meet the requirements of the Flow theory in order to offer an optimal experience. The theory states different requirements to create and maintain Flow state. In this article, we focus on elements related to game goals. The Flow theory states that the objectives must be (i) explicit and (ii) achievable with (iii) an acceptable level of challenge [2]. If the player is mobile, his context is evolving continuously and some objectives that were achievable in some contexts become inadequate in others.

For instance, let us consider a game using mechanisms that require players move into the real world to avoid virtual enemies. This game can be played well if players are in open and uncrowded places. Indeed, players have space and are able to move without constraint. However, if the game takes place in a crowded place or public transportation, players cannot move freely and avoid enemies properly. In this context, game objectives are not achievable. The Flow requirements are not met, that can cause frustration and create a bad experience.

One solution to prevent this game from creating a bad experience is to stop it whenever current context is not consistent with game objectives. One less radical alternative is to change the gameplay in response to current context. Basically, games must preserve a balance between challenge and players skill. In the case of mobile gaming, events from the real world can affect players skill. Because games cannot act on these skills, the only solution is to act on game structure and provide players with activities they can perform in current context. In transportation, our game should propose activities that do not require displacements. For instance, he can play a classical shooter game or carry out an activity that requires the touch screen only.

In this article, we present an approach to build games that adapt their gameplay to context. The main idea is to design different game scenarios. Each scenario proposes a specific gameplay, the game engine selects and executes one according to current context. Our approach is based on a gameplay formal model and a generic adaptation model. In this solution, game designers are responsible for defining these scenarios and their appropriate context. The game engine checks context changes and executes a selection algorithm in response.

This article is organized as follows: next Section presents the backgrounds for this work; Section III presents our ap-

proach; Section IV its experimental evaluation and the results to be gleaned from this work; Section V reviews current technologies and adaptation models for games. Finally, Section VI discusses about limitation and enhancement and concludes the article.

II. Background

A. Game definition

According to Schell, we can classify elements that structure a games in four categories [3]: (i) aesthetic; (ii) story; (iii) game mechanics; (iv) technology. Aesthetic defines all elements related to the visual and sound aspect of the game. Story refers to narrative elements, and the tale told to players. Game mechanics refer to all elements about interaction with the system. Basically, game mechanics can be considered as the gameplay of the game and define its heart. Finally, technology refers to technological elements used to implement other aspects. Because we focus on the adaptation of gameplay we need to identify main elements that compose the gameplay.

A formal game is defined by the following main elements [4]: actions, states, rules, goals, players, challenge. All of these concepts are strongly related and, for didactic reasons, we propose to introduce them in a linear way.

The action concept is an interface between game and player. Indeed, player acts on game and modifies game state through actions. For example, in the chess game player modifies chessboard when he displaces a pawn or another chess piece. In this game moving a chess piece is an action and chessboard represents the game state.

The game state is the second main concept. A game necessarily has an explicit state which is represented by an adequacy formalism. Games have a set of states and player browses into this set through his actions. And more, some states in this ensemble are particularly interesting because they are starting points or ending points of the game. However, player's movements into this collection are not free, they are constrained by game's rules.

The rules define which game states can be reached from the current game state. Furthermore, rules are constitutive, they constitute the practice that is regulated by them, allowing to identify and distinguish the game from others. These formal and normative character of rules provide games with a fundamental property, reproducibility. Through the rules, games can exist independently of playing sessions.

However, the matches cannot exist without players. Indeed, this is the activities of players that make game lives. These are players activities which bring the game to life. Players are like agents adopting teleological behaviors, it means their behaviors are guided by game goals.

Goal is another main concept of the game structure. It defines particular states of game where players will be rewarded (success state) or punished (failure state). So, players browse into game states in order to reach success states. However, browsing is constrained by rules and another main concept of game structure, the challenge.

Challenge refers to all factors which prevent players from reaching success state and contribute to lead them to failure states.

B. Mobile games and mobile gaming

In the setting of this article, we distinguish between mobile games and mobile gaming. Mobile games are games designed as 'standard' video games: their gameplay and design process do not differ from a PC or console-based video games. However, they are targeted to run on lightweight hand-held devices: player's locations, or more generally the mobile context, are not used as elements of the game mechanics and rules. These games are present in the game industry since the 80's with handheld game consoles such as Nintendo GameBoys, Atari Lynx and Sega Game Gear. On the other hand, mobile gaming is about games that include player's mobile context as an element of the gameplay. Such games are illustrated by Geocaching[1], Tourality[2] or Parallel Kingdom[3].

III. Method

We propose a new, simple and generic game design approach to develop games that adapt their gameplay according to context. This approach uses two main components:

1) A generic game model to structure game activities.
2) A simple method to define contexts where each activity can be performed without problems.

These two elements allow us to define a game engine that can select and propose game scenarios according to current context. Here, the expression "game scenario" refers to a set of game activities that are structured according to a plan. This plan is a part of level design. Figure 1 shows an overview of game engine core elements. These elements act the following way:

1) At the beginning, game developers define a set of game scenarios (levels).
2) During the play session, a process (called "*Context observer*") collects pieces of information about player context. It sends these elements to another process (called "*Context sensitive game engine*")
3) "*Context sensitive game engine*" selects a game scenario from its database according to current context. Once the selection is done, game engine uses it to init a game scene.

A. Game formal model

We use "Gameplay Component" (GPC) Framework to model the game structure [5]. This framework proposes a formal model for OCR game loops. OCR game loops decompose a game as a set of activity units. An activity unit represents a sequence of three steps: Objective, Challenge, Reward [6]. Reward refers to elements that are given to the player when the goal is either achieved or missed. OCR game loops have

[1]http://www.geocaching.com
[2]http://www.tourality.com
[3]http://www.parallelkingdom.com

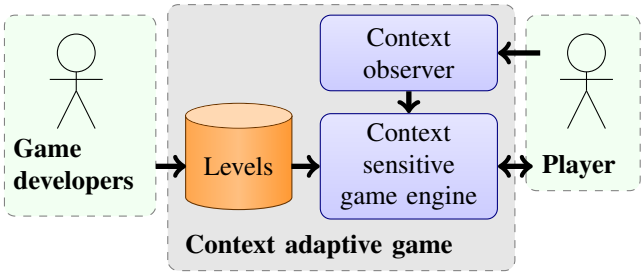

Fig. 1. Scheme giving an overview of the system

several sizes are nested and follow a hierarchical structure. Smallest loops represent a few seconds of play. Largest loop encompasses all OCR game loops and represents the overall game. When players interact with the game, basically they try to complete the main OCR game loop by reaching the smaller one.

In GPC framework, atomic GPCs represent the smallest OCR game loop and the elementary gameplay. Atomic GPC is an entity that provides players with an objective, set all elements of challenge and control the state of the objective by using an evaluation function. The return of the function is one of the following values:

- "*Succeeded*": means the objective has been reached by the player.
- "*Failed*": means the objective has not been reached and it will not be possible to reach the objective in the future states.
- "*Unknow*": means the objective is not in one of the previous state.

GPC framework defines a set of operators with operational semantics. Operators are used to define more complex situations. Basically, they allow to structure the game into a tree that reproduces the hierarchical structure of OCR game loops. Moreover, thanks to their operational semantics each operator defines how the game tree evolve according to the state of GPC.

B. Adaptation Model

In a non-context sensitive version, game engine selects trees according to predefined sequence. In mobile gaming, as game trees can be incompatible in some context, game engine requires some improvements in order to detect incompatible game trees.

In order to implement this feature, we use a method drawn from "*Responsive Web Design*" [7] and "*Adaptive Web Design*" [8]. In web design domain, designers use these approaches to build web pages that changes their presentations according to devices that display them. These solutions allow web designers to deal with the heterogeneous set of devices that are used to browse the Web. Our approach is based on three elements: "*Context queries*", a "*Context model*" and a "*Selection process*".

1) Context Queries: These web design methods use "*Media Query*" concept as a main element. A media query is a predicate that links a page layout instructions to device characteristics. Basically, a set of instruction is executed if evaluation of predicate that is linked to it returns "true" value. This allows to show a banner if only the screen resolution is higher to a defined size for example.

Media queries target originaly device properties. We extend media queries to target context properties. We call this extended version "*Context queries*". A context query is a predicate that links context requirements to a gameplay component. Consequently, a gameplay component (atomic or composite) is considered compatible if evaluation of its context query returns "true" value. Context queries need a model of context to make writing of these predicates possible.

2) Context Model: We use an Object-Oriented approach to model the context. This approach provides us with two main advantages. In fact, our main goal is not to provide a model that contains an exhaustive list of all pieces of information about context. On the one hand, we need a model easy to understand. On the other hand, we want be able to add new elements easily to extend the model. The Object-Oriented approach provides the inheritance concept. This one allows to add new elements easily. Moreover, we can exploit polymorphism concept during defining of context queries.

Actually, we started from pieces of information which are used by the state of the art to define core elements of our model. Thus, we transform the four categories we have identified into the four following objects:

- "*Device*" refers to information about devices which are involved, such as the kind of device, battery level, signal strength or screen size.
- "*Place*" refers to information about the place, such as topography, objects which are near.
- "*Player*" refers to information about player condition, such as tiredness level, if he is pedestrian or not.
- "*Time*" refers to temporal information such as maximum time for play session.

From this base and with inheritance concept, we are able to extend the model to fit with needs. For example, we can extend "*Player*" object and define the two following subclass:

- "*Static*" refers to a player who is not moving.
- "*Mobile*" refers to a player who is moving. It contains information such as speed or direction, for example.

Furthermore, the "*Mobile*" can also be extended into two subclasses:

- "*Pedestrian*" refers to a player who is pedestrian.
- "*Transport*" refers to players who are in a vehicle.

The Figure 2 shows the class diagram of our context model. Additionally, all the objects of our model inherit from an abstract object called "*ContextInfo*". This object represents a generic context information.

With this model, we are able to define global "*Context queries*" such as: "Player is Mobile". We can also be more

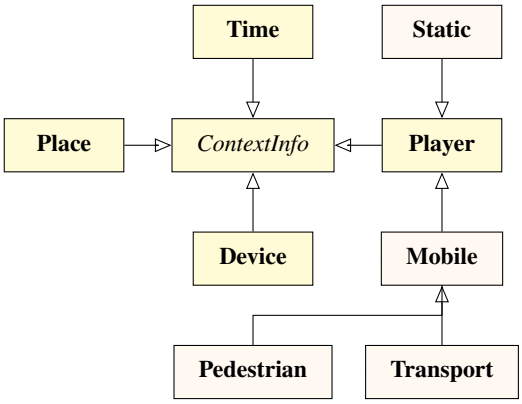

Fig. 2. Our class diagram for the context. In this diagram, yellow classes represent the core of our modeling. Light classes show how we can extend these objects to add new elements.

specific and target only pedestrian players for example by defining the following query: “Player is Pedestrian”.

Moreover, set of solutions for ”*Context queries*” can be extended or reduced by using ”*and*,” ”*or*” and ”*negation*” operators form boolean logic. For example, we can define the following ”*Context queries*”:

- ”Player is Pedestrian *or* Player’s speed is under 10 km/h”. It means the player can be pedestrian or passager. But if the player is passager his speed must be under 10 km/h.
- ””Player is Pedestrian *and* Player’s speed is under 10 km/h”” It means the player must be pedestrian with a speed under 10 km/h.

3) Game tree selection process: Our approach takes account of operational semantic of GC operators during game tree filtering. Basically, our filtering process aims to detect if it is exist at least one way to complete game tree in current context. It starts from the root and evaluate context queries. If an evaluation return false, it checks if the GC using the context queries is mandatory to complete the game tree. Figure 3 shows the flowchart of our method.

4) Game trees transition: We want to emphasise the fact that in our model, game designers are involved with the management of game tree transitions. In fact, when game engine needs to change a game tree, it sends a signal to all nodes. Game designers must define for each node what to do when this signal is received.

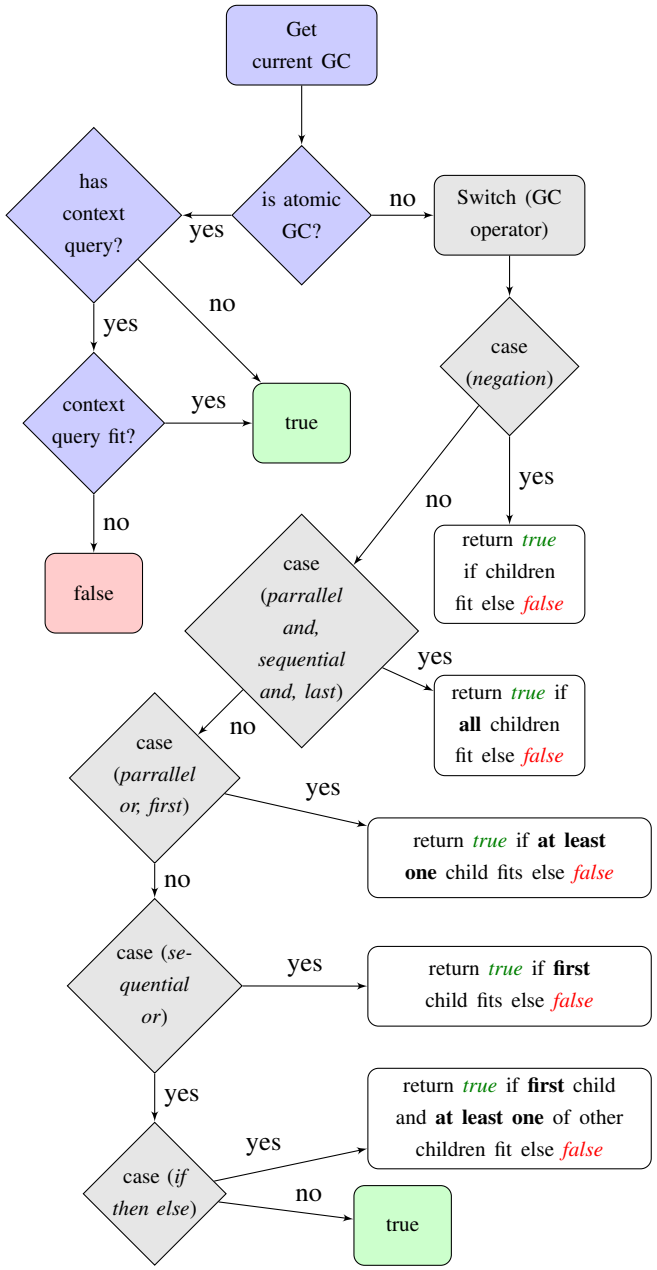

Fig. 3. Flowchart of our algorithm that checks if a gameplay tree is compatible with a context. True means the current component is compatible with current context. False means the current component is not compatible.

IV. Results

Our pilot experiment aims to study the impact of our adaptation approach on player’s experience. The study consists of observing some player metrics during different play session. We describe the experimental design and protocol we adopted in the following sub-section.

A. Experimental Design

We have created and developed a mobile game to conduct the experiment. We have defined the three following atomic gameplay components:

- “*Collect*“: player has to collect an object.
- “*Avoid*“: player has to avoid an entity that tries to catch him.
- “*Eliminate*“: player has to eliminate an entity by sending a projectile toward it.

In “*Collect*“ and ”*Avoid*” components player has to move in the real world. Moreover, in “*Avoid*“ player must be in large space such as green parks in order to be able to move without severe constraints. Thus, we associate these gameplay components with the following context queries:

- "*Collect*": player is pedestrian.
- "*Avoid*": player is pedestrian and is in a rural place.
- "*Eliminate*": nothing.

"*Rural place*" extends the "*place*" class in our context model. "*Rural place*" involves places such as green parks and rural zone. It is the opposite of "*Urban place*". This latter involves places with several buildings and streets.

We have built ten game trees from these atomic gameplay components. Figures 4, 5 and 6 show screenshots of our game.

Fig. 4. Screenshot of the game. (1) character controlled by the player (2) an egg to collect.

Fig. 5. Screenshot of the game during an avoiding phase. (1) character controlled by the player (2) an enemy to avoid.

We have not implemented a process to detect if player is pedestrian or not. In order to get this piece of information, player has to use the button on the left top of screen. It switches between pedestrian and not pedestrian.

Subjects were not outside in a town but indoor during the experiment. In order to simulate players displacement in a real world, we have implemented a virtual town. In our simulator, player can use tramways. They can go into some green zone where they are able to run in order to move faster in the game. They can also walk to places where they are unable to run.

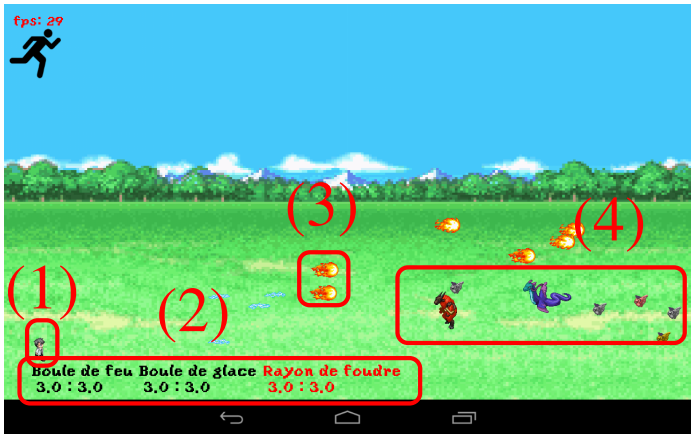

Fig. 6. Screenshot of the game during an elimination phase. (1) character which throws projectiles (2) menu to select the kind of projectile (3) two kinds of projectiles thrown (4) a group of enemies who walk toward the character.

Some buildings and walls constraint player's displacements these elements are represented by black zone in the simulator.

B. Design and Hypotheses

We used a repeated measures design where subjects have to play two versions of our game. One version uses our context adaptive approach. The game engine selects game tree according to current context. The other version provides players with a game tree according to a defined sequence.

Furthermore, in order to counterbalance order effects that may occur, we split subjects into two groups: (1) First group plays adaptive version first; (2) second group plays version without adaptation first. Players neither know their groups nor the version they play during the experiment.

Main steps of the experiment are the following:

1) Subject is introduced with the goal of experiments;
2) he plays a training session to get familiar with the game;
3) he plays the first version of the game then the second version;
4) we conduct an interview with the subject to collect data about his player's experience.

Our objective is to compare the effects of both versions on player's experience. So, we set the following null hypotheses:

1) *H.A: context adaptation has no effect on the number of objectives reached.*
2) *H.B: context adaptation has no effect on the number of objectives failed.*
3) *H.C: context adaptation has no effect on the player's satisfaction.*

We record the following data:

1) *Number of objectives reached*: numbers of GC that ended in succeeded state.
2) *Number of objectives failed*: numbers of GC that ended in failed state.
3) *Player's satisfaction* players have to report their level of satisfaction through a 1 to 5 scale (1: Not satisfied

at all; 2: Not satisfied; 3: Moderately; 4: Satisfied; 5: Very satisfied) during playing sessions. This allow us to estimate the player's level of satisfaction for each second of play.

C. Participants

Subjects were between 23 and 37 years old. This group contains different player profiles. The Table I shows main information about our subjects.

D. Material

Subjects use two materials for the experiment: (1) a tablet computer executing the game; (2) a laptop executing the town simulator. The Figure 7 shows a photo of a subject during the experiment. In this picture, we can see the two devices.

Fig. 7. Photo of a candidate during the experiment.

E. Result

The Table II shows data gathered from the subjects.

We compute satisfaction means of each subject by using the following formula:

$$\frac{\sum_{i=1}^{n} v_i \times s_i}{duration}$$

where

v_i : represents the level of satisfaction,
s_i : represents seconds passed with level of satisfaction v_i,
$duration$: represents session duration.

Statistical analysis was performed with R software in 3.1.2 version [4]. We have used paired Student's t-test to contest our hypotheses.

The hypothesis *H.A* seems to be rejected according to Student's t-test results. The difference between the two versions for success mean rate is statically significant with our data. With df=7, the average of differences M=-0.56445, t=-9.3744 and p-value=3.27e-05, the version with adaptation has a bigger success rate than the version without adaptation.

The hypothesis *H.B* seems to be rejected according to Student's t-test results. The difference between the two versions for fail mean rate is statically significant with our data. With df=7, the average of differences M=0.20478, t=6.3075 and p-value=0.0004, the version with adaptation has a bigger success rate than the version without adaptation.

The hypothesis *H.C* cannot be rejected according to Student's t-test results. The difference between the two versions for satisfaction mean is not statically significant with our data. With df=7, the average of differences M=-0.39475 and t=-1.4439 we have a p-value=0.192.

F. Discussion

We can use observations obtained from the interviews and play sessions to explain the difference between the success rate and failure rate for the two game versions. One reason is related to our game tree proposition strategy in the version without adaptation. Because it proposes game tree according to a predefined sequence, several times, players have received a tree involving "avoid" components, while they were not able to run. Consequently, they failed these objectives.

Moreover, players have emphasized the fact that it is difficult to focus on several things. In the case of our game they refer to think about a strategy and watch the two screens. They reported that they failed several times because they focused on tablet screen and collided with buildings or walls in virtual town. Inversely, they collided with enemies because they focused on virtual town. Buildings and walls are less frequent in green areas of our virtual town. Players said that is hard to escape from enemies if they are not in these space.

Because of these two reasons, players said that the version without adaptation is harder than the version with adaptation. Indeed, we can understand this opinion because the version with adaptation cannot propose a game tree with this component if player are not in good place.

In order to explain the result of the hypothesis *H.C*, we can analyze the average satisfaction of each player. Table III shows these data.

It may be observed that four subjects (half of the group) show a level of satisfaction higher in the version without adaptation. We can use observations from interviews and play sessions to understand this phenomenon.

We want to highlight the fact that three of these candidates have played the version without adaptation first. They played several game trees involving "avoid" components and they have shown a high interest in this kind of game trees. They reported that difficulty of these game trees is a source of motivation for them. One of them considered that his failures were because of his mistakes such as a lack of attention or a bad choice. Consequently, when they played the second version (with adaptation), because they played less game trees involving this component, they shown a lower level of satisfaction. However, game engine has not proposed this kind of game tree because players were not in a right place (as

[4]R official website: http://www.r-project.org/. Last visited :04/04/2017

Candidate	1	2	3	4	5	6	7	8
Age	26	23	30	26	37	24	25	29
Play frequency	$>$ 1/week	$>$ 1/week	$<$ 1/week	$\approx$ 1/week	$<$ 1/week	$\approx$ 1/week	$\approx$ 1/week	everyday
Mobile game player?	No	Yes (only)	Yes	Yes	Yes	Yes	No	No

TABLE I
SUBJECTS DATA

Version	Without adaptation	With adaptation
Success means rate (value on 1)	0.17 ($\pm$ 0.14)	0.73 ($\pm$ 0.19)
Fail mean rate(value on 1)	0.28 ($\pm$ 0.05)	0.07 ($\pm$ 0.06)
Satisfaction mean	2.71 ($\pm$ 0.57)	3.26 ($\pm$ 0.27)

TABLE II
DATA CAUGHT FROM THE EXPERIMENT.

Id	1	2	3	4	5	6	7	8
Co	2.335	2.619	2.964	1.764	2.847	3.48	3.336	2.358
Ad	3.018	3.426	2.936	3.72	3.368	3.188	3.168	2.037

TABLE III
AVERAGE SATISFACTION OF PLAYERS FOR EACH VERSION OF THE GAME. 'CO' MEANS GROUP CONTROL. 'AD' MEANS GROUP WITH ADAPTATION.

defined by our context queries). They have been few times in green areas. However, as these subjects considered they were able to play this kind of objective, they were less satisfied.

The fourth subject have played version with adaptation first. He have shown a good level of satisfaction on average for this version. When he played version without adaptation, by chance, he received game trees involving "avoid" component when he was in green areas. Because he was in the right place, he was able to complete the game tree without severe problem. Consequently, he did not show frustration. On the contrary, he shown a better satisfaction because he has liked this kind of game tree.

As answer to the question "what is your favorite version ?", four candidates choose the version with adaptation because it has created less frustration. Two candidates choose the version without adaptation. They said this version is more challenging and it is a source of motivation. The last two candidates said they did not be able to chose. The candidate of the previous paragraph is a member of this group. One of the three candidates that have played the version without adaptation first and who has shown a higher level of satisfaction in this version is the second member. In fact, they consider the two versions can be proposed. The version without adaptation should be proposed to players who agree to get invested at 100% into the game. For example, people who agree to get out from a tramway in order to reach the game objective. The version with adaptation is a good choice for people who want to play while they do something else at the same time.

This opinion raise an interesting point about our experiment. Indeed, we used a virtual town and we have not provides candidates with scenario. They were able to move with no goal, no cost or time constraint. They move according to their will and they used the city to experience different aspect of the game. If the game is run in a real town during a candidate's daily life, we can think that they will not act in the same way. For example, public transportation will mostly be a tool for their daily tasks and not a tool for the game.

V. RELATED WORKS

Some authors have proposed game with adaptive gameplay. In [9] and [10], authors present two games that modify game rules according to context. In these games, size, speed and a number of resources are affected by context elements such as battery level, player's speed in the real world signal strength. These elements are part of challenge of the game. The adaptation model uses a rule-based system to set size and velocity of elements according to context information. They adapt rules in order to propose an appropriate levels of difficulty. However, the approach limit the reusability of elements. Indeed, the rules are defined for a particular game and cannot be reused for another one.

The video game *7's wild Ride* use another approach to adapt game difficulty according to these two elements player's performance and playing time left [11]. This is a game of physical skill where main goal is to lead a character which is balancing on a snow ball to the goal line. The topology of terrain acts on the snow ball's speed and the character's balance. Some obstacles are added to the terrain in order to make the character fall. The game generates terrain on the fly and adapts the topology according to player's performance and playing time left. So, player gives his desired time of play at the start of applications and the game engine records player's performance. In order to adapt the game's ground, authors split it in several chunks. Each chunk is linked with metadata describing his level of difficulty and which elements can be linked with them. Obstacles are also linked with these metadata. So, at each generating cycle, the engine creates a request in order to get all the elements which respond to the required level of challenge. Then, it sequences chunks in order to create a coherent path.

Our method is similar to this one. Indeed, we filter game elements according to context's information. One main difference between the two approaches is the nature of these elements. In [11], the gameplay is set and the authors work directly on entities used in the game. Consequently, the adaptation engine needs to be defined again for another game. With GPC, we work at a higher level, we target game activities. So, our adaptation process is generic and common to all kinds of games.

In [12] and [13], two exergames who adapt their structure according to player's tiredness are proposed. In order to adapt games, they propose a state machine-based approach. A state is a subversion of game its own rules and goals. Each subversion is linked with a set of conditions which allow to go to another subversion, these conditions are based on player

state. Adaptation engine switches the current working model according to the current state of player. This approach is a simple way to design self-adaptive games to context; it is very intuitive and it offers a graphical representation. However, this solution can require to develop a high number of game modes and transition. This number grows according to the number of context's information. Furthermore, this solution seems better if context changes in a linear way. That is the case for tiredness where player pass through several states well defined, for instance.

The exergame *ZombieSC* uses another approach to adapt position and type of enemies according to activity of player. The game controls if player is moving and if he is interacting with him. Values of these variables are given to a generator which decides where to put enemies and their types.

In [14], authors propose another approach based on filtering. Their model is able to generate a sequence of objectives according to elements that are around to the player. In this model, the system contains a database that allows him to detect objects that are near to the player. Adaptation module gets player's real position in order to deduct the object. Authors link each object with metadata describing its characteristics. In parallel, activity patterns are defined. These patterns must be completed by a set of objects in order to be used. Patterns which cannot be completed from the list of available objects are filtered out. Then, the game creates a sequence of activity according to authors' policy. With this model, metadata are linked to game objects and the context informations are used to generate filters for these game objects. However, this solution can require to define the filter generator for each game according to metadata set and context's information.

VI. Conclusion and future work

We have tackled the problem mobile games endure with changing context during play. We have proposed a solution to build context adaptive game in order to maintain the "*Flow*" balance. We have introduced the main elements of games structure (actions, states, rules, goals, players and challenge) in Section II and different approaches exist in order to build an adaptive engine which adapts this structure to player context. Unlike existing works, our solution is built with a game structure model, an extensible context model and an adaptive strategy.

We have conducted a pilot experimentation in order to evaluate the effects of adaptation in game and player's experience. The results show that adaptation has an impact on average success and fail rate of game objectives. Players achieve more objective and fail less when the game is context adaptive. However, we notice that average satisfaction of players is not necessarily higher. In fact, adaptation engines can generate frustration. This occurs when a player does not receive the game scenario that he is waiting for because his current context does not fit scenario prerequisites.

Our pilot experiment highlights several points which can be developed in order to enhance the adaptive engine. The first point consists of considering the player profile during game three selection process. Indeed, some authors such as Bartle [15] or Stewart [16] have identified different psychologies of players. According to his psychology, player tries to satisfy an intrinsic goal while he plays. For example, some people use games to create social interactions whereas other people use it to evaluate their skills. These goals can be used to explain players' behavior and also to improve players' satisfaction. In fact, the adaptation engine needs to be enhanced in order to select game trees according to player profiles and context. For example, it must give a high priority to trees which require high levels of skill if a player likes to play to overcome tasks with high level of challenge.

Finally, our pilot experiment has to be extended and include more subjects in order to be getting results that can be considered as statically more significant. We also need to conduct the experiment in a real city in order to observe results in real conditions.

References

[1] M. Weiser and J. S. Brown, "The coming age of calm technology," in *Beyond calculation*. Springer, 1997, pp. 75–85.

[2] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. Harper & Row, 1990. [Online]. Available: http://books.google.fr/books?id=V9KrQgAACAAJ

[3] J. Schell, *The art of game design: A deck of lenses*. Schell Games, 2008.

[4] S. Chauvier, *Qu'est-ce qu'un jeu?* Vrin, 2007.

[5] Y. Francillette, A. Gouaich, N. Hocine, and J. Pons, "A gameplay loops formal language," in *Computer Games (CGAMES), 2012 17th International Conference on*. IEEE, juillet 2012, pp. 94–101.

[6] M. Albinet, *Concevoir un jeu vidéo: Les méthodes et les outils des professionnels expliqués à tous !*, ser. Entreprendre: Développement professionnel. Fyp éditions, 2011. [Online]. Available: http://books.google.fr/books?id=iwOFZwEACAAJ

[7] E. Marcotte, "Responsive web design," http://alistapart.com/article/responsive-web-design, 2010.

[8] S. Viljami, "Adaptive vs. responsive, what's the difference," http://viljamis.com/blog/2012/adaptive-vs-responsive-whats-the-difference.php, 2012, [Last visited : 06/01/2017].

[9] A. Sahami Shirazi, S. Stünkel, and A. Schmidt, "Exploring context-aware mobile games," *In adjacent Proceeding of Pervasive 2010*, 2010.

[10] P. Holleis, A. Sahami Shirazi, and A. Schmidt, "Providing generic context for mobile games on phones," in *Proceedings of the 7th international and interdisciplinary conference on Modeling and using context*, ser. CONTEXT'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 5–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=2045502.2045504

[11] R. Lopes, K. Hilf, L. Jayapalan, and R. Bidarra, "Mobile adaptive procedural content generation," in *Proceedings of the fourth workshop on Procedural Content Generation in Games (PCG 2013), Chania, Crete, Greece*, 2013.

[12] F. Buttussi, L. Chittaro, R. Ranon, and A. Verona, "Adaptation of graphics and gameplay in fitness games by exploiting motion and physiological sensors," in *Smart Graphics*, ser. Lecture Notes in Computer Science, A. Butz, B. Fisher, A. Krüger, P. Olivier, and S. Owada, Eds., vol. 4569. Springer Berlin Heidelberg, 2007, pp. 85–96. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73214-3_8

[13] F. Buttussi and L. Chittaro, "Smarter phones for healthier lifestyles: An adaptive fitness game," *Pervasive Computing, IEEE*, vol. 9, no. 4, pp. 51–57, 2010.

[14] C. Lu, M. Chang, D. Kinshuk, E. Huang, and C.-W. Chen, "Usability of context-aware mobile educational game," *Knowledge Management & E-Learning: An International Journal (KM&EL)*, vol. 3, no. 3, pp. 448–477, 2011.

[15] R. Bartle, "Hearts, clubs, diamonds, spades : Players who suit muds," *The Journal of Virtual Environments*, 1996.

[16] B. Stewart, "Personality and play styles: A unified model," http://www.gamasutra.com/view/feature/6474/personality, 2011.