Constraint-Based Verification of a Mobile App Game Designed for Nudging People to Attend Cancer Screening*

Arnaud Gotlieb and **Marine Louarn**
SIMULA, Norway

Mari Nygard and **Tomas Ruiz-Lopez**
Cancer Registry of Norway

Sagar Sen
SIMULA, Norway
Cancer Registry of Norway

Roberta Gori
University of Pisa, Italy

Abstract

In Norway, cervical cancer prevention involves the participation of as many eligible women aged 25-69 years as possible. However, reaching and inviting every eligible women to attend cervical cancer screening and HPV vaccination is difficult. Using social nudging and gamification in modern means of communication can encourage the participation of unscreened people. Simula Research Laboratory together with the Cancer Registry of Norway have developed *FightHPV*, a mobile app game intended to inform adolescent and eligible women about cervical cancer screening and HPV vaccination while they play and, to facilitate their further participation to prevention campaigns. However, game design and health information transfer can be hard to reconcile, as the design of each game episode is more guided by the release of information than gameplay and playing difficulty. In this paper, we propose a constraint-based model of *FightHPV* to evaluate the difficulty of each episode and to help the game designer in improving the player experience. This approach is relevant to facilitate social nudging of eligible women to participate to cervical cancer screening and HPV vaccination, as shown by the initial deployment of *FightHPV* and tests performed in focus groups. The design of this mobile app can thus be regarded as a new application case of Artificial Intelligence techniques such as gamification and constraint programming.

Introduction

Cervical cancer screening is of paramount importance to elevate the general population health and to decrease significantly the mortality rate due to this type of cancer. To be effective, screening campaigns have to maximize participation of relevant populations and to adopt appropriate communication about the risks and benefits of early detection. Knowing that cervical cancer was responsible in 2012 of about 528,000 new cases of invasive cancer and of about 266,000 deaths worldwide, intensive cervical cancer prevention programs grew up in many developed countries. For instance, as cervical cancer is almost always caused by Human papillomaviruses (HPV) infection, early HPV detection approaches have been developed in Norway, leading to the screening of 80% of women aged 25-69 years. Unfortunately, it turns out that reaching the latter 20% is difficult due to lack of health knowledge about cancer and ineffective communication to certain populations. For instance, in a study of 3,800 women living in Norway, less than half understood health information conveyed in a letter about a positive screening result (Burger et al. 2014). More generally, it is observed that sending personally targeted invitation letters has not been sufficient to motivate unscreened women to participate. More precisely, only 18% of unscreened women in last four years who received a reminder letter, went to perform a test in the next six months. In addition to screening, vaccination against HPV also reduces cervical cancer burden. In Norway, HPV vaccination has been increasing from 68% in 2009 to more than 80% in 2014 (Hansen et al. 2015), but again, the uptake of HPV vaccine is lower than other vaccines which indicates that there is room for improving HPV vaccination among the relevant populations.

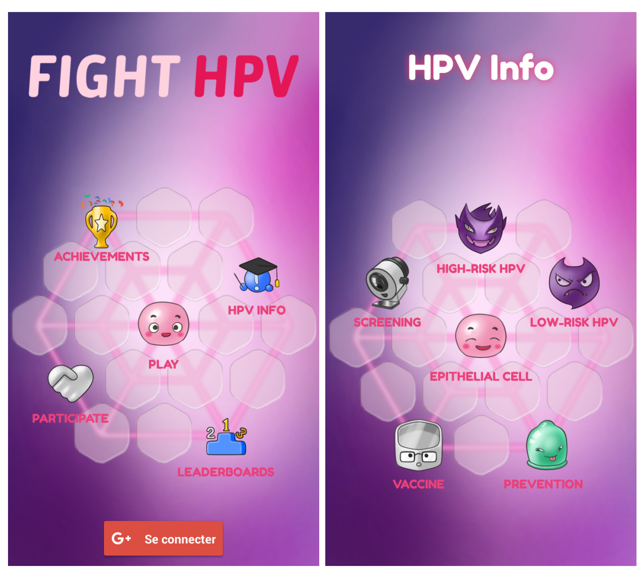

Figure 1: *FightHPV*: Starting page and characters.

Since 2014, Simula Research Laboratory and Cancer Registry of Norway have developed a research program to explore how social nudging with gamification and mobile

*This work is supported by the Research Council of Norway (RCN) through the research-based innovation center Certus, under the SFI programme

apps can help improving the communication of health information about cervical cancer screening and HPV vaccination. Social nudging is a well-known experimental psychology concept which aims at influencing certain behaviors by positive reinforcement and indirect suggestions (Thaler and Sunstein 2008). This research program led to the development of a mobile application game, called *FightHPV*, intended to nudge players to participate to cervical cancer prevention (Sen, Ruiz-Lopez, and Jacobsen 2015). Fig.1 shows the starting page of the game and briefly introduce some the characters of the game. By *characters*, we mean type of cells (including epithelial cell, cancerous cell, etc.), virus (cancerous HPV), and processes (screening, vaccination, etc.). *FightHPV* is a single-player game with a board which represents a body tissue where characters meet and recombine in order to change the state of the tissue. Beneath the game, there is a body of information related to HPV, cervical cancer, screening and vaccination. The game releases easy-to understand health information about cervical cancer screening and HPV vaccination and nudge people to participate to cervical cancer prevention. Fig.2 shows two distinct episodes of the game.

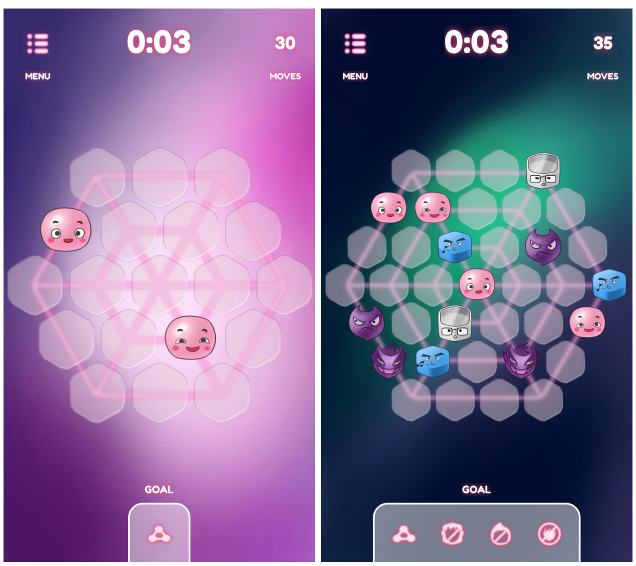

Figure 2: *FightHPV* game: screenshots of episodes 1 and 50.

Game design and information transfer are sometimes hard to reconcile. When the design of a new episode of the game is guided by the necessity of releasing appropriate health information in an incremental way, then the gameplay becomes less important for the designer. But, at the same time, ignoring the player satisfaction leads to the lack of user pleasure and eventually the abandonment of the game. Facing this challenge, we propose in this paper a constraint-based verification model of *FightHPV* which can be used to evaluate the playing difficulty of each episode and to help the game designer to improve the player satisfaction. In this application, using Constraint Programming allows us to perform an automatic and exhaustive exploration of all paths leading to winning states of the game in a reasonable amount of time. Then, measuring some timing aspects, it becomes possible to classify the episodes according to their level of difficulty and thus facilitate the design of the gameplay. This paper describes our constraint-based verification model of *FightHPV* and its usage in the gameplay design process. The paper contains the results of an experimental analysis to evaluate the computational difficulty of game episodes. Regarding the game deployment process, *FightHPV* has been tested in focus groups and its public release to the general population is planned for early 2017, making of this mobile app a new application case of Artificial Intelligence techniques such as gamification and constraint programming.

Background

Game modeling and algorithmics have a long history in AI. We focus our presentation on alignment board games that have been checked with Constraint Programming (CP) and we elaborate on the pecularities of our approach for *FightHPV*.

One of the most famous board game is the *N-queens* problem over a chessboard (Bell and Stevens 2009). Many CP-based approaches have been tried out until the million-queens threshold was beaten in a few seconds by using constraint-based local search (Sosic and Gu 1994). The *FightHPV* game shares with N-queens the difficulty to characterize hardness by level. Indeed, the absence of factors for explaining how the difficulty grows with each level is problematic. N-queens can be extremely difficult to solve for a level i, while it is much easier for level $i + 1$ and so on. According to (Bell and Stevens 2009), *"there is no closed form expression for the total number of solutions for the standard or modular board of arbitrary size"*. *FightHPV* also suffers from the absence of general theory as boards of different size do not lead necessarily to an increase of difficulty in the game. Another famous chessboard problem is the *Knight's tour* which is related to the Hamiltonian circuit problem in graph theory, but, unlike the general Hamiltonian circuit problem, knight's tour can be solved in linear time. Similarly to some constraint-based models of N-queens and knight's tour which both utilize global constraints to ease the encoding and solving, our model of *FightHPV* also exploits a global constraint, namely the TABLE global constraint (Régin 2004).

Sudoku is another example of single-player game, which has been completely solved using CP (Simonis 2005). Interestingly, the difficulty of a sudoku is characterized by requirements on value-guessing. In fact, a sudoku is hard to solve when many guesses have to be made on the possible values of its free cells. Note that even if a general framework for modelling games with constraint programming is now available (Nguyen and Lallouet 2014), there is no general principle to guide the design of levels of increased difficulty. Despite some attempts to find formal ways to maximize user satisfaction in video games (Andrade et al. 2006), practitioners still have to empirically evaluate the playing experience to maximize user satisfaction.

Recently, following the idea that game design elements can facilitate knowledge acquisition (Deterding et al. 2011), *gamification* was used in the healthcare domain (Lister et al.

2014; Tong et al. 2015). Serious games were also developed on mobile devices for helping adolescents in cancer treatment (Stinson et al. 2013). *FightHPV* can also be regarded as a serious game based on gamification of cervical cancer information. A screenshot example is given in Fig.3 to show how cervical cancer related notions have been used in *FightHPV* as gamification elements.

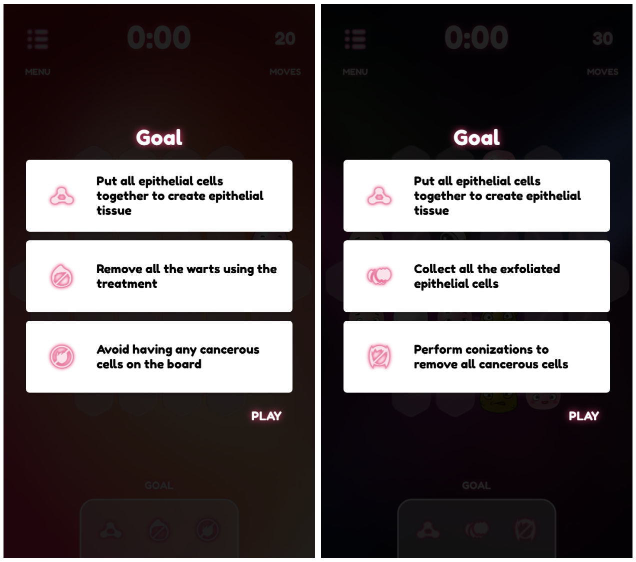

Figure 3: Screenshots of gamification elements.

Constraint-Based Verification of *FightHPV*

Constraint-based verification (Delzanno and Podelski 2001) is a formal paradigm which entails the design of a constraint model of a system in order to verify it. Interestingly, each game episode of *FightHPV* can be seen as a user-interacting system which needs to be precisely analysed to extract its level of difficulty. It requires formal definitions for the game board, user actions, character interations and winning conditions, as well as a presentation of the constraint-based search process.

Board Description

The *FightHPV* board is composed of concentric hexagons as shown in Fig.4.

In this view, Tri denotes an integer associated to one of six possible triangles of a given hexagon (Tri in $0..5$). By convention, numbering follows the anti-clockwise direction starting from the top of the mobile device. Hex denotes the level of a selected hexagon Hex in $0..HexMax$, where $HexMax$ is the maximum level of the board. Note that not all levels have the same number of hexagons. $HexMax_i$ denotes the maximum number of hexagons for each level i inside the same triangle. It can be observed that given a level Hex, the maximum number of hexagons inside the same triangle $HexMax_i$ is equal to Hex. Pos denotes the position of a cell on one side of the hexagon (Pos in $0..HexMax_i - 1$). The number of possible values for Pos is exactly equal to the value of the hexagon. For instance, there are 3 possible values for Pos on one side of hexagon numbered 3. Finally, Typ denotes the type of character on the board (Typ in $0..13$). Typ can take one of the following possible values, *Epithelial cell* (value 0), *Non-cancerous HPV* (1), *Cancerous HPV* (2), *HPV Vaccine* (3), *Immune System* (4), *HPV Antibody* (5), *Wart* (6), *Cancerous Cell* (7), *Conization* (8), *Imiquimod* (9), *Intercourse* (10), *Prevention method* (11), *Screening* (12), *Exfoliated Cell* (13).

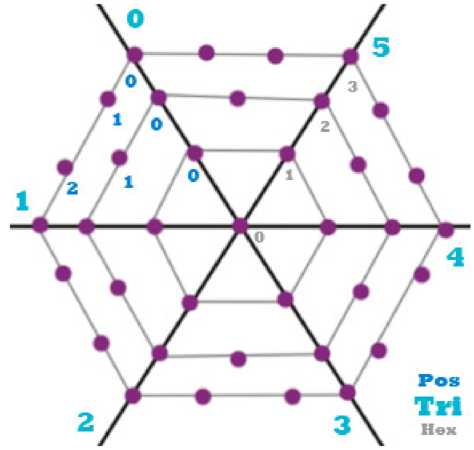

Figure 4: Schematic view of the board of *FightHPV*

Character Definition

A *Character* in *FightHPV* is uniquely represented by a quadruplet (Hex, Tri, Pos, Typ) where $Pos < Hex$ in addition to the following domain constraints Hex in $0..HexMax_i$, Tri in $0..5$, Pos in $0..HexMax_i - 1$, Typ in $0..13$. We have selected this encoding after several rounds of formalization as not only the allowed movements on the board had to be considered but also the interactions between characters and the winning conditions. In order to ease a cost-efficient encoding of these constraints, we have used a global constraint called TABLE (Régin 2004). This constraint encodes the various possible combinations of values for a tuple of variables.

An initial state is composed of two or more characters placed on the board. For instance, the state given in Fig.5 contains two characters, namely an epithelial cell $P1 = (2, 0, 1, 0)$ and a cancerous HPV $P2 = (3, 3, 2, 2)$. The central point of the board has coordinates $(0, 0, 0, \_)$ where $\_$ stands for any value (from $0..13$ in this particular case).

Game Actions

The *FightHPV* gamer has two possible actions over the board: rotation and translation.

Rotation.

Performing this action means rotating one of the hexagon, including all its characters, while keeping all the other points of the board unchanged. Note that the formulas depend of the direction of the rotation (clockwise or anti-clockwise). If **HexRot** is the selected hexagon of the rotation, and β is

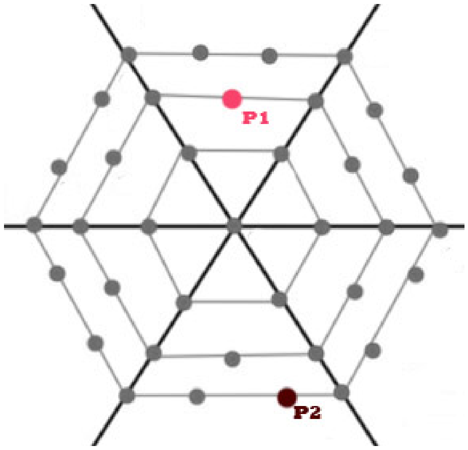

Figure 5: An initial state of the *FightHPV* board

the number of points which are rotated, we get the following coordinates for a rotation in the anti-clockwise direction, where a quo b denotes the quotient of a by b, while a mod b denotes the rest:

$$rotation^{+}_{HexRot,\beta}(Hex, Tri, Pos, Typ) =$$

$$\begin{cases} Hex' \leftarrow HexRot \text{ if } Hex = HexRot \\ Tri' \leftarrow (Tri + (Pos + \beta) \text{ quo } HexRot) \text{mod } 6 \\ Pos' \leftarrow (Pos + \beta) \text{ mod } HexRot \\ Typ' \leftarrow Typ \end{cases}$$

For the clockwise direction, the formula is:

$$rotation^{-}_{HexRot,\beta}(Hex, Tri, Pos, Typ) =$$

$$\begin{cases} Hex' \leftarrow HexRot \text{ if } Hex = HexRot \\ Tri' \leftarrow (Tri + (Pos + \alpha) \text{ quo } HexRot) \text{mod } 6 \\ Pos' \leftarrow (Pos + \alpha) \text{ mod } HexRot \\ \quad \text{where } \alpha = 6 \times HexRot - \beta \\ Typ' \leftarrow Typ \end{cases}$$

Translation.

With translations, the gamer can grab a single point which stays on the diagonals and move all diagonal of the *FightHPV* board in straight line of β positions. Note that the diagonals are circular, which means that any character which goes off the board reappears on the other side of the diagonal.

Our formalization consider two distinct cases: the point to be moved has $Hex \neq 0$ or it has $Hex = 0$ (i.e., it is the central point). For the case $Hex \neq 0$ and the movement of the point is toward the centre, if $\alpha = (\beta \text{ mod } 2HexMax + 1) \neq 0$ we have

$$move(Hex_1, Tri_1, 0, Typ_1, \alpha):$$

if $|Hex_1 - \alpha| \leq HexMax$ then

$$Pos' = 0 \wedge Hex'_1 = |Hex_1 - \alpha| \wedge$$

$$Tri'_1 = \begin{cases} 0 & \text{if } Hex_1 = \alpha; \\ Tri_1 & \text{if } Hex_1 - \alpha > 0; \\ (Tri_1 + 3) \text{ mod } 6 & \text{if } Hex_1 - \alpha < 0. \end{cases}$$

else

$$move(HexMax, (Tri_1 + 3) \text{ mod } 6, 0, Typ_1, \alpha')$$

where $\alpha' = \alpha - Hex_1 - HexMax - 1$.

Character Interactions

Character interactions take place only when the characters meet on the board. Conditions for this to happen also have to be formalized. For any character placed outside a diagonal (i.e., when $Pos \neq 0$), there are either 4 or 6 adjacents points, as shown in Fig.6. Let

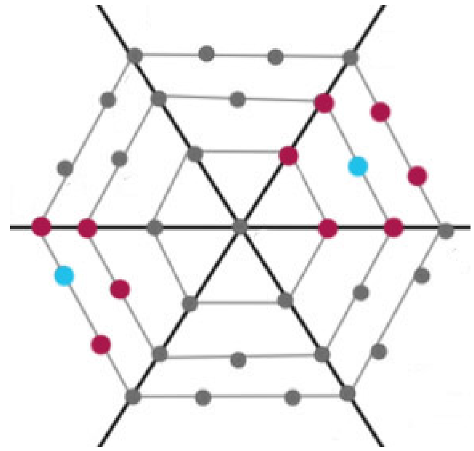

Figure 6: Character interactions and adjacence relation on the board when $Pos \neq 0$.

$P_a = (Hex_a, Tri_a, Pos_a, Typ_a)$ be one of the blue characters in Fig.6, and $P_b = (Hex_b, Tri_b, Pos_b, Typ_b)$ any of the adjacent characters placed in the hexagon $Hex_b = Hex_a - 1$, we have:

$adj(P_a, P_b)$ holds if

$$\begin{cases} ((Pos_b = Pos_a - 1 \wedge Tri_b = Tri_a) \vee \\ (Pos_b = Pos_a \text{ mod } (Hex_a - 1) \wedge \\ (Tri_b = (Tri_a + (Pos_a \text{ quo } (Hex_a - 1))) \text{ mod } 6))) \end{cases}$$

For the hexagon $Hex_b = Hex_a + 1$ with $Hex_a + 1 \leq HexMax$, we get:

$adj(P_a, P_b)$ holds if

$$\begin{cases} ((Pos_b = Pos_a \wedge Tri_b = Tri_a) \vee \\ (Pos_b = (Pos_a + 1) \wedge Tri_b = Tri_a)) \end{cases}$$

Finally, for the characters placed on the same hexagon $Hex_b = Hex_a$:

$adj(P_a, P_b)$ holds if

$$\begin{cases} ((Pos_b = Pos_a - 1 \wedge Tri_b = Tri_a) \vee \\ (Pos_b = (Pos_a + 1) \text{ mod } (Hex_a) \wedge \\ (Tri_b = (Tri_a + ((Pos_a + 1) \text{ quo } (Hex_a))) \text{ mod } 6))) \end{cases}$$

Similar formulas can be devised for characters with $Pos = 0$, an example of which is shown on Fig.7. In this case, some characters may have 6 or 3 adjacents points.

Based on this adjacence relation, interactions between characters can happen resulting in changes of character in most of the cases. For example, if a epithelial cell ($Typ_a = 0$) is adjacent to a non-cancerous cell ($Typ_b = 1$), then the epithelial cell will change to a so-called wart ($Typ_a = 6$). If a cancerous HPV ($Typ_a = 2$) or a cancerous cell ($Typ_a = 7$) is next to an epithelial cell ($Typ_b = 0$), then this cell will turn into a cancerous cell ($Typ_b = 7$).

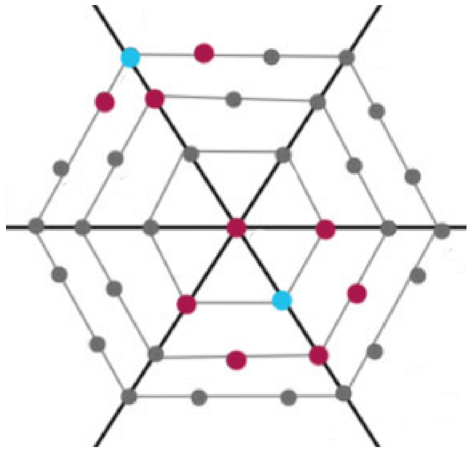

Figure 7: Adjacence relation on the board when characters with $Pos = 0$ are involved.

Winning Conditions

Depending on the game level, there are distinct winning conditions, i.e., conditions over the characters on the board which make the player win the game level. Formalizing and implementing them was not easy as the board is formed of concentric hexagons. Actually, no less than 9 distinct winning conditions were identified and formalized. The most tricky one required that the characters associated to epithelial cells to be pairwise adjacent over the board, so that they can form an epithelial tissue. We implemented this winning condition by follwing all paths made of adjacent characters and backtracking when it was necessary. Let suppose that $P_1, \ldots, P_n$ are characters on the board and $adj(P_i, P_j)$ means that P_i and P_j are adjacent on the board, then this winning condition is as follows: a set of epithelial cells $\{P_1, \ldots, P_n\}$ is an epithelial tissue iff $\forall i, j$ in $1..n, i \neq j$, either $adj(P_i, P_j)$ holds or there exists a finite sequence of epithelial cells $P^1, \ldots, P^l$, with $P^i \in \{P_1, \ldots, P_n\}$, such that $adj(P_i, P^1) \wedge adj(P^1, P^2) \wedge \ldots \wedge adj(P^{l-1}, P^l) \wedge adj(P^l, P_j)$ holds.

Constraint-Based Search

There are basically two ways of exploring the space of game actions. *Forward exploration* starts with an initial state (some characters on the board) of the board and then tries out all the possible actions until the process falls into a winning state of the game or until a boundary is reached on the number of elementary steps. *Backward exploration* starts from a winning state and goes back (applying reverse actions) until it founds an initial state or some boundaries are reached. In our approach, we selected forward exploration as our goal was to mimick the actions of an actual player. Note also that our objective was to find all the winning paths from an initial state which is thus more easily reachable with forward analysis. In any case, limiting the number of actions for the exploration guarantees termination and controls the search to an acceptable effort for the player.

Implementing constraint-based search with forward exploration requires to create a new state each time an action is operated (translation, rotation). Using constraint logic programming over finite domains helps a lot here as creating new variables and new constraints for handling a new state is easy with this framework. In fact, using recursion our constraint logic model can generate a new constraint system as an extension of the previous one at each step of the forward exploration process. Even if the number of allowed steps must be bounded (to ensure termination), this number does not need to be known at compile-time. This facilitates the implementation of the search process.

Experimental Analysis and Deployment

All the experiments were run a 2,66GHz Intel core i7 Macbook pro with 4GB RAM.

Implementation

We implemented the constraint model described above in SICStus Prolog with the `clpfd` library (Carlsson, Ottosson, and Carlson 1997) which provides a constraint solver over finite domains, a parametrized labelling procedure and global constraints implementation. Running the model, which accounts for 0.6KLOC, involves an exhaustive exploration of all winning paths, that are sequences of intermediate states which lead to a winning state of the game. These paths are composed of an ordered succession of player actions. When a maximum number of actions is specified, the constraint-based search presented above can be used to find all the winning paths. Fig.8 shows an output example of a run with two winning paths.

Gameplay Design

In *FightHPV*, each level of the game is called an *Episode* (Ep.) and there is a timer on each Ep. session. The game now incorporates more than 60 Ep. In principle, any gameplay should respect the basic principle that *"the higher is the episode number, the harder is the gameplay"* to nurture the pleasure of the players. However, with the initial release version of *FightHPV*, it was observed for example that any player finds Ep. 5 much harder to win than Ep. 8. Such observations motivated the work of this paper and conducted to totally revise the gameplay of *FightHPV*.

To design the gameplay of an Ep., a designer starts by creating an initial board state and specifies final board winning conditions. Then, this person attempts to solve the Ep. by himself to get an initial idea of the hardness of the Ep. Even if the process is subjective, it allows the designer to discard trivial or intractable Ep. immediately. The constraint-based model presented in this paper is helpful here to provide more objective, fine-grained and automated feedback on the computational difficulty of each Ep. to the designer. The computational difficulty is based on two factors, namely, a) the number of winning paths b) the average time required to find a winning path. Tab.1 enlists the possible factors for an Ep. and provides simple guidelines to evaluate the difficulty. For the sake of simplicity, the number of actions in different winning paths can vary from 'Low' to 'High', as well as the average time required to find a winning path. The actual values of 'Low' and 'High' are directly linked to the

```
| ?- movement([[2,2,0,0], [1,4,0,0]], 2, 2).
Winning path
Initial state: [[2,2,0,0],[1,4,0,0]]
Max number of Hexagons: 2
Max number of Moves   : 2
Path: [[[2,2,0,0],[1,2,0,0]],[[2,2,0,0],[1,0,0,0]],[[2,2,0,0],[1,5,0,0]],[[2,2,0,0],[1,4,0,0]]]
Number of steps: 3

Winning path
Initial state: [[2,2,0,0],[1,4,0,0]]
Max number of Hexagons: 2
Max number of Moves   : 2
Path: [[[2,5,1,0],[1,0,0,0]],[[2,2,0,0],[1,0,0,0]],[[2,2,0,0],[1,5,0,0]],[[2,2,0,0],[1,4,0,0]]]
Number of steps: 3
```

Figure 8: *FightHPV*-Ep.1: Output example of our constraint model.

amount of playing time allocated to each Ep. According to Tab.1, 'Zero' winning paths simply means that the Ep. cannot be solved and hence is intractable. In the other cases, the guidelines estimate the difficulty based on how hard it was for a computational approach to solve an Ep. This information can then be used to guide the designer in his choices to order the various Ep.

This approach however does not guarantee *playability* or *pleasure*. Both these factors should ideally be objectively evaluated through user testing, but gameplay design is still a craftman's activity and large-scale experiments are difficult and costly to set up. Using the number of winning paths and the average time required to find a winning state with an automated procedure, to evaluate the difficulty of an episode is discussable. On the one hand, human reasoning is not comparable and reductible to these numbers, but on the other hand, adopting this methodology while the game is not yet deployed at the larger scale (i.e., worldwide) accelerates its development and analysis. For *FightHPV*, using the proposed constraint model, it is possible to evaluate objectively the computational difficulty and thus to use the proposed guidlines to improve the gaming experience in further releases of the game.

Nb. of winning paths	Av. Time to find a win. path	Ep. Difficulty
Zero	N/A	Intractable
Low	Low	Easy/Medium
Low	High	Hard
High	Low	Easy
High	High	Medium/Hard

Table 1: Guideline for determining the difficulty of an Ep.

Our experimental analysis aimed at evaluating the level of difficulty of each Ep. and compare them.

Evaluating Each Episode

For each episode, the constraint model was exhaustively explored up to a certain boundary to evaluate its difficulty. The results for Ep.3 are given in Tab.2. Given a maximum number of actions (1^{st} col.), i.e., rotation and translations, we computed the number of paths leading to one winning condition of the game (2^{nd} col.). We also measured the total runtime required to compute all these paths (3^{rd} col.) and computed the average runtime to find one such path (4^{th} col.). We computed these tables for each episode up to Ep.10. In

Max. num. of act.	Num. of win. pat.	Runtime	Average Runtime
1	0	0min 0sec 02	–
2	5	0min 0sec 01	2.00ms
3	27	0min 0sec 04	1.48ms
4	265	0min 0sec 29	1.09ms
5	1654	0min 1sec 92	1.16ms
6	12810	0min 14sec 66	1.14ms
7	84805	1min 44sec 98	1.24ms
8	615719	12min 4sec 96	1.18ms

Table 2: *FightHPV*-Ep.3: Number of winning paths, runtime to find all solutions, average runtime to find one solution.

fact, computing these tables for Ep. higher than 7 already required more than 4 hours.

Comparing the Difficulty of Episodes

We compared the tables computed for each episode when the maximum number of actions was set up to 7. This arbitrary number was selected because it corresponds to a reasonable effort of the player. By reasonable effort, we meant an appropriate level of effort to find one solution path.

Tab.3 shows the comparison for all episodes of the game up to Ep.10.

Level	Number of win. paths	Runtime	Average Runtime to find a win. path (ms)
1	123276	2min 55sec	1.02ms
2	96510	52min 56sec	32.91ms
3	84805	1min 44sec	1.24ms
4	61602	28min 59sec	28.24ms
5	828	245min 49sec	17,813.97ms
6	68	46min 4sec	40,945.44ms
7	1	75min 28sec	452,8440.00ms
8	0	157min 37sec	–
9	48	26min 33sec	33,191.25ms
10	0	158min 19sec	–

Table 3: Number of winning paths and runtime to find solutions for all levels ($MaxMove = 7$).

Results Interpretation

When one looks at the number of winning paths while the episode number increases, a clear correlation appears as shown in Fig.9. Put in simpler words, the number of solutions decreases with the episode number. Note that some episodes are without any solution when the maximum number of actions is set up to 7. It means that these episodes are not easily solvable by an average player. More surprisingly,

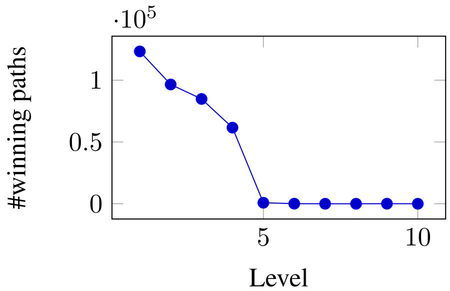

Figure 9: Number of Winning Paths per level ($MaxMove = 7$).

when looking at the runtime required to find all solutions in Fig.10, we observe a large variation from one episode to another. To understand this surprising result, we looked at the minimum number of actions required to find a first solution. Fig.11 shows that, from ep. 7, there is a huge variability in

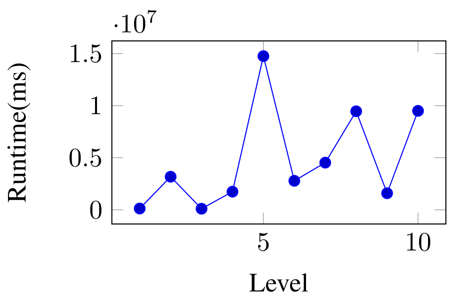

Figure 10: Total runtime per level ($MaxMove = 7$).

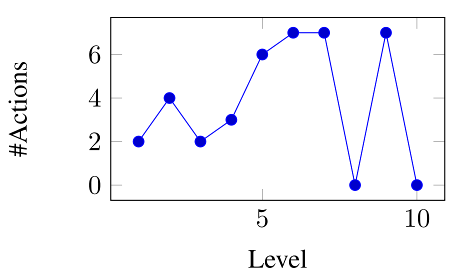

Figure 11: Minimum number of actions required to find one solution per episode.

the minimum number of actions required to find one solution. This means that the episode difficulty varies a lot from Ep. 5 to Ep. 10, which contributes to explain the decrease of interest of the players from episode 5.

Threats-to-Validity

Our experimental analysis has several threats-to-validity. Firstly, we were unfortunately not able to process all the levels (until 60) and we have used $MaxMove = 7$ as the maximum number of actions that could be played. Even if we do not expect large deviations with other episodes or an increased number of actions, it would be interesting to evaluate the difficulty of all episodes. Secondly, by adopting the runtime required to compute all the winning paths (or the average runtime for each solution) as measurement of the episode difficulty, we compute only the computational difficulty which can be a poor approximation of the actual difficulty. Indeed, human players may adopt much more subtle strategies than the ones implemented in our constraint model to find solutions. In order to tame this risk, we also looked at the minimum number of actions required to find one solution (in Fig.11) but it could have been interesting to consider other measurements such as the average number of backtracks in the constraint model or the activity of each variable of the model. Thirdly, there is room for optimizing the constraint model by looking at symmetrical solutions. Symmetry detection and reduction could be used to dramatically reduce the number of winning paths and thus improve the time required to find solutions. This could also impact the qualitative analysis of the results since many actual players probably detect symmetry and exploit it to boost the finding of a winning state.

Deployment of *FightHPV*

The deployment of *FightHPV* has started since 2015 with a 3-step process. The first step, which is now completed, aimed at validating the game itself by asking known and selected users to play with game. The goal of this initial small-scale deployment was to validate the technology involved (gamification, gameplay, health information release). We started the development of our constraint-based model after having observed that the gameplay and the difficulty of levels were key-aspects of the adoption of the game. The second step, also completed, aimed at deploying *FightHPV* on three focus groups and beta-testing groups. The goal of this step was to evaluate precisely the capability of the game to instruct distinct categories of people to cervical cancer, screening and vaccination. The three groups and the findings are described below. Finally, the third step aims at releasing *FightHPV* to the general public with no restriction and this step is scheduled for early 2017.

We considered three focus groups with distinct evaluation criteria. The goal was to evaluate the capability of the game to transfer health information to high-risk adolescents group, general adult public and specialized nurses.

1. **Zambian focus group**: Teenagers in Lusaka, Zambia were invited from both urban and rural schools to participate to a playing experience with *FightHPV*. The focus group comprised of 8 boys and 8 girls from each school. The children were give a total time of two hours to play

the game. Teenagers finished the game until different levels, one of them finished all 60 levels of the game in the short time. Then, a focus group interview was conducted by us with the children to understand their perception of the HPV and cervical cancer. This experience led us to revise considerably the presentation of health information in the game ;

2. **CRN focus group**: This was a beta-testing group established in the Cancer registry of Norway comprising of 40 employees who were invited to play. Data about their game play was collected using Google Analytics and Google Play. The data consisted of information regarding number of wins, retries, and fails per level and clicks on information sections. The main observation was that some levels were much harder than others, creating not a very smooth gameplay ;

3. **Sannitetskvinner nurses focus group**: A focus five nurses in the age group 50-60 were asked to play the game and give qualitative feedback through extensive interviews. The interviews resulted in improvements suggested to finer details of the game. One of the key changes that was implemented following this experience is the direct access to advanced episodes without having to play basic episodes.

Conclusions

This paper introduced a constraint-based verification framework of *FightHPV*, a mobile app game designed to nudge people to attend cervical cancer screening and vaccination. Modelling the game with constraint programming (with global constraints) led us to analyze the difficulty of each episode of the game and thus to improve the gameplay. Gamification was also used to ease the knowledge transfer about cervical cancer screening and HPV vaccination to distinct categories of people. *FightHPV* has been deployed to 3 focus groups before its public release in early 2017.

In terms of perspectives, gamification for social nudging combined with constraint-based verification appear to be a strong combination of Artificial Intelligence techniques. We can mention two perspectives to this work. Firstly, an interesting development could be to automatically generate new episodes for *FightHPV*. By using our constraint-based verification model, originally designed for gameplay evaluation, it could be possible to derive new initial states composed of distinct characters and new rules of play for the characters. Secondly, observing that human players tend to explore moves based on trial and error until they see recognisable spatial-patterns while the automatic processing of our constraint-based model is more systematic, an approach based on reinforcement learning that attempts to develop spatial reasoning in a computer could be interesting to develop in future. In addition, the development of serious games for nudging people to participate to other cancer prevention campaigns (breast, colorectal, etc.) is a general trend where Artificial Intelligence techniques will have a strong role to play.

References

Andrade, G.; Ramalho, G.; Gomes, A. S.; and Corruble, V. 2006. Dynamic Game Balancing: An Evaluation of User Satisfaction. In *AAAI conf. on Art. Int. and Interactive Digital Entertainement*, 3–8.

Bell, J., and Stevens, B. 2009. A survey of known results and research areas for nqueens. *Discrete Math.* 309(1):1–31.

Burger, E.; Nygard, M.; GyrdHansen, D.; Moger, T.; and Kristiansen, I. S. 2014. Does the primary screening test influence women's anxiety and intention to screen for cervical cancer? *BMC Public Health* 14(360).

Carlsson, M.; Ottosson, G.; and Carlson, B. 1997. An open–ended finite domain constraint solver. In *Proc. of Programming Languages: Implementations, Logics, and Programs*.

Delzanno, G., and Podelski, A. 2001. Constraint-based deductive model checking. *STTT* 3(3):250–270.

Deterding, S.; Sicart, M.; Nacke, L.; O'Hara, K.; and Dixon, D. 2011. *Extended Abstracts on Human Factors in Computing Systems*. ACM. chapter Gamification: Using Game-Design Elements in Non-Gaming Contexts, 2425–2428.

Hansen, B.; Campbell, S.; Burger, E.; and Nygard, M. 2015. Correlates of hpv vaccine uptake in schoolbased routine vaccination of preadolescent girls in norway: A registerbased study of 90,000 girls and their parents. *Preventive Medicine* 77:4–10.

Lister, C.; West, H.; Cannon, B.; Sax, T.; and Brodegard, D. 2014. Just a fad? gamification in health and fitness apps. *JMIR Serious Games* 2(2).

Nguyen, T., and Lallouet, A. 2014. A complete solver for constraint games. In *Principles and Practice of Constraint Prog. CP'14, Lyon, France, 2014*, 58–74.

Régin, J.-C. 2004. *Global Constraints and Filtering Algorithms*. Boston, MA: Springer US. 89–135.

Sen, S.; Ruiz-Lopez, T.; and Jacobsen, E. 2015. *FightHPV: Et spill som skal øke bevisstheten rundt HPV, og dulte folk til a forebygge livmorhalskreft*. Oslo, Kreftregisteret. 137–143.

Simonis, H. 2005. Sudoku as a constraint problem. In *In Workshop on Modeling and Reformulating Constraint Satisfaction Problems (MODREF'05)*, 13–27.

Sosic, R., and Gu, J. 1994. Efficient local search with conflict minimization: a case study of the n-queens problem. *IEEE Trans. on Knowledge and Data Eng.* 6(5):661–668.

Stinson, N.; Jibb, A.; Nguyen, C.; Nathan, C.; Maloney, M.; Dupuis, L.; Gerstle, T.; Alman, B.; Hopyan, S.; Strahlendorf, C.; Portwine, C.; Johnston, L.; and Orr, M. 2013. Development and testing of a multidimensional iphone pain assessment application for adolescents with cancer. *J. Med. Internet Res.* 15(3).

Thaler, R., and Sunstein, C. 2008. *Nudge: Improving Decisions about Health, Wealth and Happiness*. Yale Univ. Press.

Tong, T.; Yeung, J.; Sandrakumar, J.; Chignell, M.; Tierney, M.; and Lee, J. 2015. Improving the ergonomics of cognitive assessment with serious games. *Int. Symp. on Human Factors and Ergonomics in Health Care* 4(1):1–5.