Matching Game Mechanics and Human Computation Tasks in Games with a Purpose

Luca Galli
Politecnico di Milano, Italy
first.last@polimi.it

ABSTRACT

The massive amount of time that people spend in online gaming is being increasingly exploited by a particular kind of Serious Games, the Games with a Purpose (GWAP), used to solve complex problems as a byproduct of their gameplay. The design of the tasks and the choice of game mechanics able to solve them has been done so far without consolidated guidelines and with few considerations with respect to traditional game design princpiples.
Without proper best practices to follow, the design of a GWAP may incur in increased development time and costs or even failures.
This work attempts to solve these shortcomings for novel designers by providing: 1) a development process to follow when designing new GWAPs 2) the definition of the multimedia refinement tasks best suited to be solved with GWAPs and 3) the list of traditional game mechanics that best match these tasks.

Categories and Subject Descriptors

H.3 [**World Wide Web**]: Human Computation

General Terms

Design

Keywords

GWAP, Game Design, Games with a Purpose, Human Computation Tasks, Game Mechanics

1. INTRODUCTION

Game technologies, terminologies and practices are transcending the boundaries of pure entertainment. This phenomenon can be appreciated by the growth of a special subset of serious games, also known as Games with a Purpose (GWAP), as an industry and research field. A Game with a Purpose is a game in which players generate useful data as a by-product of play [18]. Despite the success of games like ESP [19] or FoldIt [5] and the considerable number of GWAP that has been developed, just few of them like Ontogalaxy[12] have considered common practices of traditional game design; most are just supporting the mere resolution of the hidden tasks even at the cost of sacrificing the overall gaming experience. Despite the presence of some GWAP templates that can be used to automatically validate the contribution of the players in reaching the object, no work has yet clarified the steps involved in the development of GWAPs from the idea formulation to the actual implementation. This has left researchers and developers with the only option of relying on their own past experiences, often not sufficient to avoid bad design choices that would nullify the considerable monetary and time effort spent in the creation of a GWAP.

This work tries to overcome these issues by defining the development process of GWAP, in order to guide a novel game designer in the choice of the most appropriate game mechanics for the human computation task to be solved. Section 3 describes the steps of the design process of a traditional videogame and the inferred design process for a GWAP.
Section 4 describes a human computation task, detailing in particular multimedia refinement tasks that suite GWAP Section 5 lists traditional game mechanics and matches each of them with the most suitable task, providing pointers to existing games that have already addressed the problem.

SeriousGames'14 November 07 2014, Orlando, FL, USA

http://dx.doi.org/10.1145/2656719.2656727.

2. RELATED WORK

Most works on GWAPs focus on embedding a specific problem solving task into an enjoyable user experience and on evaluating the quality and quantity of output produced by players. The classification of alternative game design patterns, based on different input-output templates, discussed in [20], is the first attempt to generalize GWAP design principles, while a comprehensive list of the games that have been developed so far is provided in [11].

Output agreement games induce humans to produce semantic annotations that describe as accurately as possible the input and to obtain a match for tasks like image tagging [19], image preference elicitation [8], ontology construction [17] and sentiment analysis [16]. In *Input Agreement* games, players must understand if they have been provided with the same content by describing it to the other players. Both input and output agreement games assign equivalent roles to the players, whereas the *Inversion-problem* template differentiates between them: at each round, one player assumes the role of the "describer"and the other one the role of the "guesser". The describer receives an input (e.g, an image, or a word) and based on it, sends suggestions to the guesser to help her identify some feature of the original input. Peekaboom [18] uses this approach with the aim of detecting objects in images. For tasks too expensive for being addressed with state-of-the-art computer algorithms, crowd wisdom has been used to quickly search and reduce the space of possible solutions, as it happens with the FoldIt game [4], in which crowds of online users compete and cooperate to predict biologically relevant low-energy protein conformations in the form of a 3D puzzle.

Recently the mechanics of GWAPs have begun to be modelled formally [3], with the intent of standardizing the design of games with a purpose, deriving some interesting properties from the interaction patterns and the users. All the presented works are lacking detailed description on how the game design phase has been performed and no work provide the detailed description of suitable tasks that could be solved with the use of GWAP and which traditional game mechanics could be feasible to be applied in their resolution.

3. DEVELOPMENT PROCESS

Before trying to describe how tasks and game mechanics can be matched together, it is necessary to define where the design of tasks and mechanics are introduced in the development process of a GWAP. A reference model of the game design workflow of a typical Videogame is needed to lay the basis of GWAP development, however such a model is not readily available from the literature. For this reason, in Section 3.1 we "reverse engineer" a possible process model from the game development best practices detailed by Crawford [6] and Fullerton [7], while in Section 3.2 the process of developing this special genre of Serious Games is provided.

3.1 Traditional Game Development Process

The literature on game design does not prescribe a structured development process, yet companies and designers have distilled their experience into best practices and guidelines useful for organizing game production. One justification advocated for such an informal approach is that "*game design is primarily an artistic process and reliance on formal procedures is inimical to creativity*"[6]. However, with the transformation of games into a consolidated industry with time and budget constraints comparable to those of business software products, some authors [7][15] have claimed that iterative and rapid software methods, most notably agile methodologies such as "Scrum", may be adequately applied also to the development of games.

Fig. 1 shows a possible representation of the game development process, obtained by modeling the guidelines and practices suggested by widely recognized designers, such as Chris Crawford and Stacy Fullerton [6][7].

In the following, each phase of the process model is briefly described.

- *Player Experience Definition* pinpoints the goals of the game, the players' interactions, and the emotions induced. The output is a narrative document defining the game concept at a high level.
- *Game Mechanics Design* defines the actions, challenges, and rules of the game. The output is a document outlining the translation of the game concept into the actual game dynamics.
- *Physical Prototype Development* involves the creation of a simplistic model that can be used to play and refine the game mechanics. Output of this phase is a playable physical game and a preliminary document, *The Game Design Document*, that defines the rules, the challenges and the components of a game, and the workflows of activities and actions that can be performed, better known as *Gameplay.*
- *Aesthetics Design* creates the visual and aural characteristics of the game, including the general look&feel (e.g., cartoon, futuristic, or historical), graphical resources (color palettes, indoor and outdoor scenarios, graphical resources, etc), the graphic models of characters and objects, and the sound themes and effects. The output comprises graphical and audio resources.
- *Interface Design* represents the user's viewpoint of the game, i.e., the display of the game status and of the controls that allow the user to play.

The iterative refinement of the Aesthetics, Interface and Game Mechanics design phases produce as output a final version of the Game Design Document, detailing the gaming experience as a whole.

- *Digital Prototype Development and Testing* involves the implementation of the design specifications into a digital product for testing purposes. This phase is conducted in subsequent iterations until the testing with the target audience achieves the intended goals. Iterations may affect also the prior steps, as shown in Fig. 1.
- *Publication and Maintenance* happens when the game has reached a consistency that allows it to be published and distributed to the whole audience; after publication a game may require maintenance in terms of bug fixes and possible refinements in the game mechanics. Novel functionalities can be added by following the entire process from the beginning, which normally results in a new edition of the game.

3.2 GWAP Development Process

The main goal of a Game with a Purpose is not the engagement of its user but to involve humans in the computational process of complex tasks; without a task to be solved, a GWAP would be just a traditional digital game, but a GWAP that does not offer an interesting experience for its players will fail to accomplish its goals, since there will not be enough performers to solve the defined problems. For this reason, the development process for a GWAP involves the definition of activities that have to be delegated to human performers and their integration within a game (existing or novel); a possible model is shown in Fig. 2, derived from the experience of the author and the design guidelines defined in [20]. In particular the *Task Design* and *Task Matching* phase, the main contributions of this work, will be covered in sections 4 and 5.

- *Requirement Specification* involves the collection of information necessary for the definition of a task, a unit of work performed by human worker in the process of solving computational problems (e.g. cropping the silhouette of the models in a picture).
- *Task Design* involves the design of the algorithms able to solve the defined problem for which well known methods, based on the decomposition of the problem in operations and controls, are known [10]. Two proper-

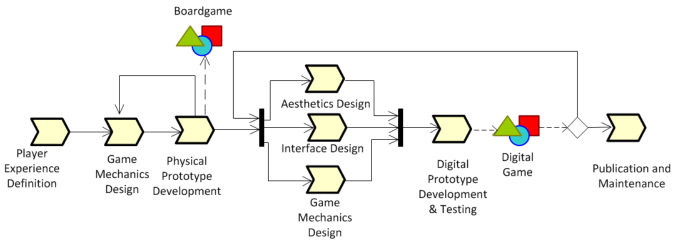

Figure 1: Software Process Engineering Metamodel (SPEM) for game development

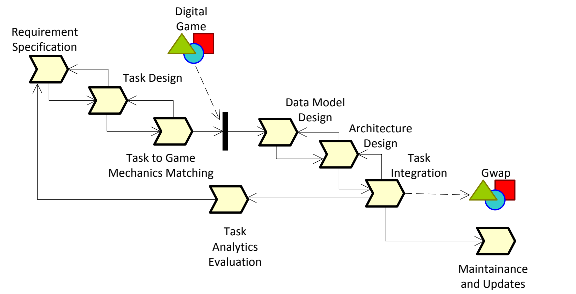

Figure 2: Software Process Engineering Metamodel (SPEM) for GWAP Development

ties are fundamental due the necessity of having human contributors: finiteness and effectiveness. *Finiteness* requires the answer to be provided and verifiable in a known and finite number of steps. *Effectiveness* requires each operation of the algorithm to be executed by a human performer that has no prior knowledge on how to solve that particular problem. The noisy nature of the results requires the definition of an *aggregation strategy*, borrowed for instance from the field of *computational social choice* with techniques such as Plurality Vote, Borda Count, Kemeny[9] or Maximin[14]. If a voting mechanism does not suffice, due to the complexity of the unstructured data generated, a tailored aggregation strategy has to be defined.

- *Task Matching* involves the analysis of the operations that have to be performed to solve the task and the identification of known game mechanics used in the gameplay of an already existing game that involve similar actions; it is described in Section 5. If a known game mechanic cannot be found, then a novel game has to be designed to propose the task to be solved as a conflict within the gameplay. Once the game mechanics have been defined, a working prototype of the game must be created following the steps defined in 3.1, before committing to the integration phase or, if the existing game allows modifications or the source code is present, the integration will be performed on the existing project.
- *Data Model Design* involves the definition of the schema of the inputs/outputs of the task to be performed, along with the ones used to represent the state of the game and its players.
- *Architecture Design* involves the definition of the hardware and software components that has to be used to create the game and the backend used to process the data necessary to accomplish the tasks.
- *Task Integration* In this phase, the set of operations defined to solve a particular task are implemented within the game: the retrieval and visualization of input data within the game are coded as the initial condition of a challenge to be solved during the gameplay, the admissible operations for the task are implemented as gameplay actions that a player could perform and the validation techniques on the provided output are coupled with the algorithms of the game to provide immediate feedback to the users. Output of this phase is a first prototype of the *GWAP*, to be refined if it fails to solve the planned task or poorly designed game mechanics.
- *Task Results Evaluation* verifies if the output of the game maps properly to the particular inputs that were fed into it and usually requires the presence of a ground truth or human judgement to make the comparison.

4. TASK DESIGN

A Human Computation Task (HCT) is a "unit of work" assigned to a user of a Human Computation system; removing duplicates or inappropriate content, cropping the silhouette of the models in a picture or recognizing and identifying the people contained in a set of images are examples of what a task involving a GWAP's player may look like and its goals may greatly vary based on the business objectives that have to be met. A generic task can be designed by specifying the different components shown in figure 3. A task is defined

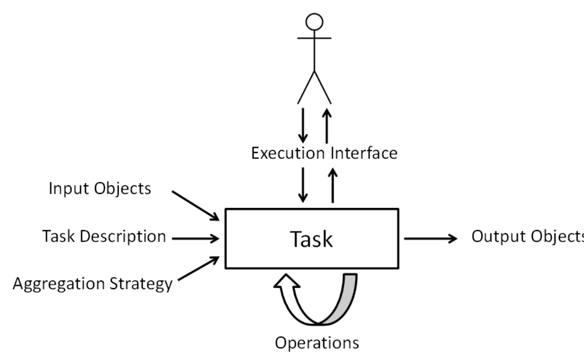

Figure 3: Components of a Task

by a *description*, generally a textual one, of the goal that has to be reached and a set of admissible operations that represent the mean with which this particular task can be accomplished by the user. Usually a HCT is created with the purpose of creating or modifying multimedia content or its annotations. For this reason, a task has to be defined not only based on the operations that can be performed but also on the data that will be manipulated and produced, the input and output objects, that may be represented as structured data or multimedia content. Depending on the specific nature of the task, it may also be useful to have a certain number of users of the platform to perform the same task several times, to achieve the redundancy needed to overcome inaccurate responses or personal biases. In such a case, the output objects may require further processing to be able to retrieve a meaningful result and thus it may

Task Type	Category	Task Description	Human Performer Operations
Object Recognition Object Identification Object Detection	Decision Generative	Recognize one or several pre-specified or learned objects together with their 2D positions in the image or 3D poses in the scene. Recognize an individual instance of an object. Recognize specific condition or anomalies	Given a specific object, identify it in the image or environment with an annotation which selects a subset of samples with a particular meaning. Their representation depends on the context and the dimension of the space that is being considered. (2D,3D, 4D...)
Clustering	Decision	Task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).	Define a (subjective) similarity measure to compare the input data with and group objects into clusters based on it.
Ordering	Decision	Arranging items of the same kind, class, nature, etc. in some ordered sequence, based on a particular criteria.	Define a (subjective) evaluation criteria to compare the input data and order the objects based on the chosen criteria.
Natural Language Processing (NLP)	Decision Generative	Performing various operations related to natural language understanding and manipulation, such as Summarizing, Question answering, Sentiment Analysis, Speech recognition...	Performing the requested tasks by exploiting humans' ability to understand natural language
State Exploration	Decision	Problems for which the solution can be measured and evaluated but exploring the whole solution space is intractable. Exploring the set of all possible points of an optimization problem that satisfy the problem's constraints, potentially including inequalities, equalities, and integer constraints, to obtain the best solution.	Intuitively recognize optimization patterns that may lead to the best solution for the problem at hand
Content Generation Content Submission	Generative	Generating novel content for the problem at hand, respecting the constraints or providing content based on particular requests	Use one's own ability to generate the requested content or choosing the best content to be provided based on personal judgment
User Preferences Opinion Elicitation	Decision Generative	Gathering synthesis of opinions of authorities of a subject where there is uncertainty	Submit an opinion or a preference related to a particular topic

Table 1: Most meaningful multimedia refinement tasks for GWAPs

be necessary to define an *aggregation strategy* associated to the task, like ranking, clustering or majority voting[13]. The *operations* are the activity that the user is requested to perform in order to accomplish a specific objective. By analyzing existing works in literature [11] and typical AI hard problems[1], the most common operations that could be performed effectively by human players within a GWAP have been collected, taking into consideration the context of multimedia meta-data refinement, and have been reported in Table 1. They may fall into two broad categories:

- *Generative Tasks* include tasks which aim at generating new artifacts as the solution of the problem at hand, e.g. Labeling, Segmenting, Ontology linking...
- *Decision Tasks* are related to decisions that the users has to perform over already existing data, e.g. Preferences elicitation, Ordering...

5. GAME MECHANICS & OPERATION MATCHING

Game mechanics represent the artificial conflicts and interaction means that are introduced in a GWAP or in a traditional game to drive the behaviors of players. One of the greatest issues that a GWAP designer could face is the difficulty in finding the right mechanics that have to be applied in a specific context. In [20] a list of structured templates for the design of GWAPS, namely input-agreement, output-agreement and inversion problem is defined. These templates alone, even though they are fundamental for what concerns the validation of the submitted results, are not sufficient for creating a gaming experience.

Defining a list of possible mechanics that have been used in traditional games could hint a novel designer on the available choices that she could exploit; for these reason based on the best practices described in [2][7], a list of possible game mechanics suitable to be applied to a GWAP is provided, along with examples. These mechanics can be combined together to produce a variety of experiences that form the structure of what can be recognized as a game; the list is not exhaustive, since game design is fundamentally a creative process, but can be seen as a starting point able to cover most of the experiences provided by the GWAPs that has been developed so far. The results are shown in Table 2, in which the mechanics are paired with the most suited tasks.

Game Mechanic	Task Type	Significative Examples
Agreement	Object recognition, Clustering, Ordering, NLP	ESP Game, TagATune
Tile(Resource)-Placement	Clustering, Ordering	Phylo
Line Drawing	Object Recognition	Sketchness, Squigl
Memory	Clustering	FlipIt
Betting/Wagering	User Preferences/Opinion Elicitation	N/A
Pattern Building	State Exploration	FoldIt, Eyewire
Bluffing	Ordering, Object Identification	Disguise, SearchWar
Trivia	Natural Language Processing	Verbosity, WebPardy
Area Enclosure	Object Recognition	PeekABoom, Ask'nSeek

Table 2: Game Mechanics to Task Type matching

- ***Agreement*** Players are requested to reach an agreement over a question or a topic based on some hints provided by the game. Agreement is one of the most widely used mechanics in GWAP, being the foundation on which templates like input-agreement and output-agreement rely on to be able to automatically validate the contributions of different players. The ESP game for instance requires two players to agree on the same submitted tag by using as the only hint a common image.
- ***Tile(Resource)-Placement*** Tile Placement games feature placing a piece to score points, with the amount often based on adjacent pieces or pieces in the same group/cluster, and keying off non-spatial properties like color, feature completion, cluster size etc. The visual nature of the mechanic is particularly suited for exploiting the capabilities of humans to visually identify patterns through abstraction and intuition. Placing multimedia assets that share some commonalities spatially near each other allows for easy human clustering tasks. In Phylo, players solve pattern-matching puzzles that represent nucleotide sequences of different phylogenetic taxonomies to optimize alignments over a computer algorithm.
- ***Line Drawing*** Games that make use of this mechanics involve drawing lines in one way or another. Line Drawing is a mechanic that allows to identify regions of interest in images and thus to solve object recognition problems. Squigl was a GWAP in which two users where asked to draw the contour of the same object and were judged based on how close their outlines were. Sketchness is a Draw-And-Guess game similar to Pictionary in which one player is given an image and an object to segment by drawing the contour, while the other players, without being able to see the image, have to guess the underlying object based just on the drawn contour.
- ***Memory*** Games that use the Memory mechanic require players to recall previous game events or information in order to reach an objective. By using just their memory, which is likely not to be able to recall all the details of a multimedia asset but just the salient features, the player can be asked to cluster assets by creating implicit mental relationship between object. FliptIt used this mechanic to cluster images that were portraying the same subjects. The players were presented with tiles hiding images and were requested to clear the board by pairing two tiles, picked up sequentially and removed just if the content of the image was the same or similar.
- ***Betting/Wagering*** Involves games that encourage or require players to bet resources or commodities on certain outcomes within the game. Often the values of the commodities are continually changing throughout the game, and the players buy and sell the commodities to make money off of their investment. No known GWAP make use of this mechanic so far, nonetheless it could help develop games involving preference elicitation or human judgment.
- ***Pattern Building*** Players place game components in specific patterns in order to gain specific or variable game results. The objective of FoldIt is to fold the structure of selected proteins as well as possible, using various tools provided within the game. The highest scoring solutions are analyzed by researchers, who determine whether or not there is a native structural configuration (or native state) that can be applied to the relevant proteins.
- ***Bluffing*** In games with the bluffing mechanic, players need to hide their true intent or actions by using bluff, lies or deceiving. In Disguise, a GWAP used to evaluate the capabilities of different color blending algorithms, some of the enemies in the game are semi-transparent in order to disguise themselves among useful resources; it is duty of the player to exploit her perceptions to identify the intruders.
- ***Trivia*** Games that make players answer questions based on their knowledge. In Verbosity, one player is giving textual clues related to a particular word or subject to be guessed by the other player, in order to obtain meaningful semantic annotations.
- ***Area Enclosure*** Players try to surround or reveal an area to score points or to gain other advantages. Similarly to Line Drawing, this mechanics allows to identify regions of interest in images. PeekABoom used this techniques by allowing one player to unveil part of an hidden image that contained salient information regarding the object within the image that another

player was asked to identify. The least the area unveiled, the more were the points received by the first player. The traces submitted by the players are aggregated in order to build bounding boxes identifying the position of a particular object.

6. CONCLUSIONS

A better understanding of the multimedia refinement tasks that may benefit from human contribution and the game mechanics that can support the development of new Games with a Purpose is key to increase their adoption. In this paper, we present a development process for the design of new GWAP, detailing the core problems in which this kind of games have been applied and the mechanics that could be adopted to foster the active engagement of humans in their resolution.

7. ACKNOWLEDGMENTS

This work is supported by the EU FP7 CUbRIK[1] Integrating Project.

8. REFERENCES

[1] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford. Captcha: Using hard ai problems for security. EUROCRYPT'03, 2003.

[2] S. Bjork and J. Holopainen. *Patterns in Game Design*. Charles River Media game development series. Charles River Media, 2005.

[3] K. T. Chan, I. King, and M.-C. Yuen. Mathematical modeling of social games. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, 2009.

[4] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, and Z. Popovic. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[5] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, and F. Players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, Aug. 2010.

[6] C. Crawford. *The Art of Computer Game Design*. Washington State University Vancouver, 1982.

[7] T. Fullerton, C. Swain, and S. Hoffman. *Game Design Workshop: A playcentric approach to creating innovative games*. Morgan Kauffmann, 2008.

[8] S. Hacker and L. von Ahn. Matchin: eliciting user preferences with an online game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, 2009.

[9] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88, 1959.

[10] R. Kowalski. Algorithm = logic + control. *Commun. ACM*, 22, 1979.

[11] M. Krause and J. Smeddinck. Human computation games: A survey. 2011.

[12] M. Krause, A. Takhtamysheva, M. Wittstock, and R. Malaka. Frontiers of a paradigm: Exploring human computation with digital games. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 22–25, New York, NY, USA, 2010. ACM.

[13] E. Law and L. Von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning Series. Morgan & Claypool, 2011.

[14] J. Levin and B. Nalebuff. An introduction to vote-counting schemes. *Journal of Economic Perspectives*, 9(1):3–26, Winter 1995.

[15] A. Rollings and D. Morris. *Game architecture and design*. New Riders, 2004.

[16] N. Seemakurty, J. Chu, L. von Ahn, and A. Tomasic. Word sense disambiguation via human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2010.

[17] D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet. A game-based approach for collecting semantic annotations of music. In *In 8th International Conference on Music Information Retrieval (ISMIR*, 2007.

[18] L. von Ahn. Games with a purpose. *Computer*, 39:92–94, 2006.

[19] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *SIGCHI*, 2004.

[20] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51:58–67, 2008.

[1] http://www.cubrikproject.eu/