Mathematical Modeling of Social Games

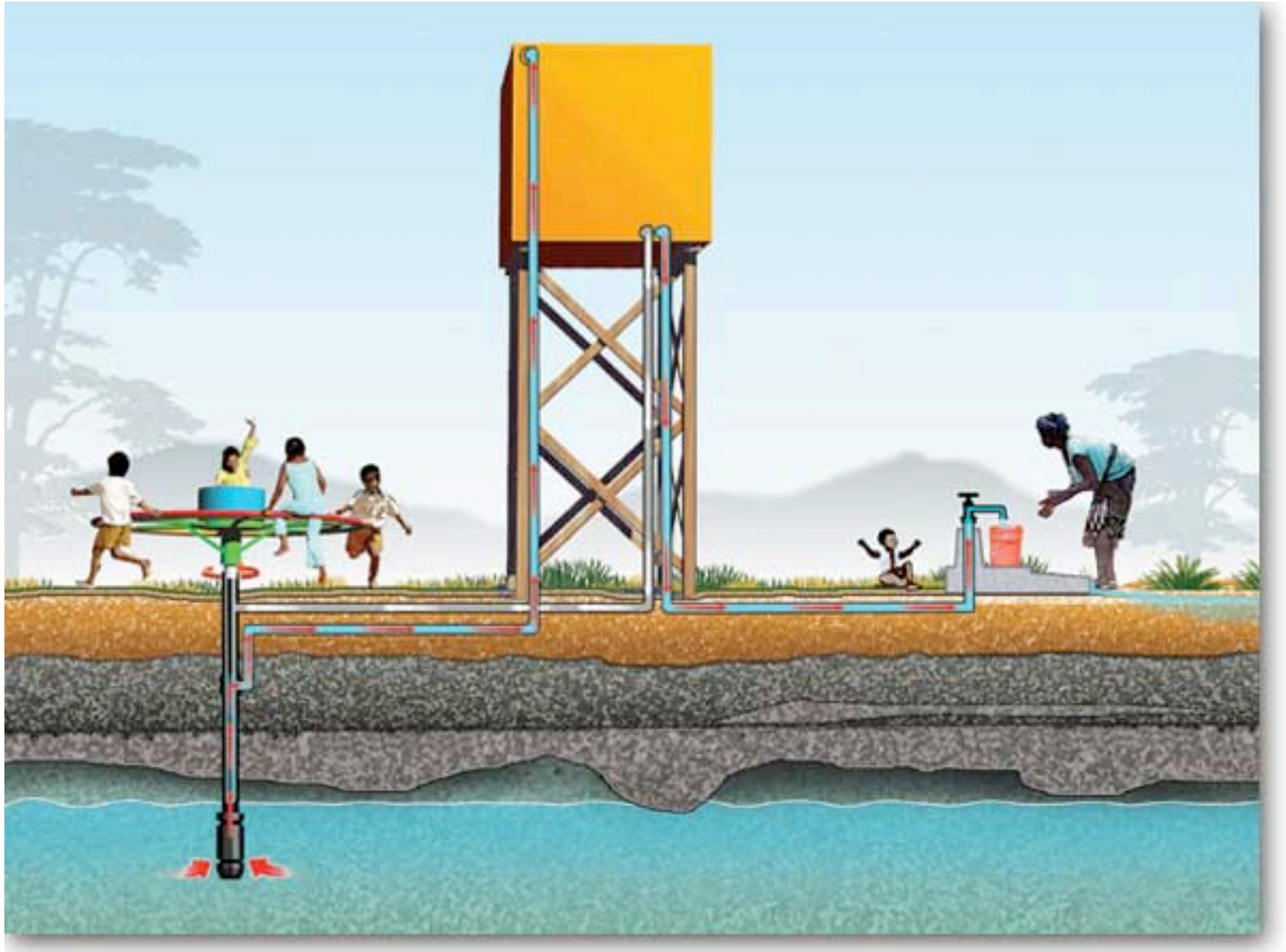
Kam Tong Chan, Irwin King, and Man-Ching Yuen

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Hong Kong

http://www.cse.cuhk.edu.hk/~king

Playing/Having Fun ⟷ ? Work/Computation

Idea of Human Computation



- Take advantage of people's desire to be entertained and perform useful tasks as a side effect

Motivations

- Recently the online games become more popular, Human Computation Systems (HCS) are proposed to collect accurate information from players as a side effect of their playing.
- Motivations
  - Many popular HCS were based on ad hoc design
  - Not easy to design a game for solving a specific problem without formal modeling
- Contributions
  - To formulate a model for HCS
  - To analyze the characteristics of problems and properties of games to solve these problems
  - To describe how to design a social game for solving a specific problem using our model

Outline

- Background and Motivation
- Related Work
- Social Game Modeling
- Social Game Properties
- Design Guidelines
- Further Improvement

Background

- Human Computation Systems (HCS) aim to solve Artificial Intelligence (AI) problems through the human human interactions.
- There are problems that computers are poor at solving but human can solve them easily, e.g., vision, cognitive, etc.
- In order to ensure the collected information to be useful, we have to:

1. guarantee the quality of collected information
2. attract more people to contribute information

Related Work

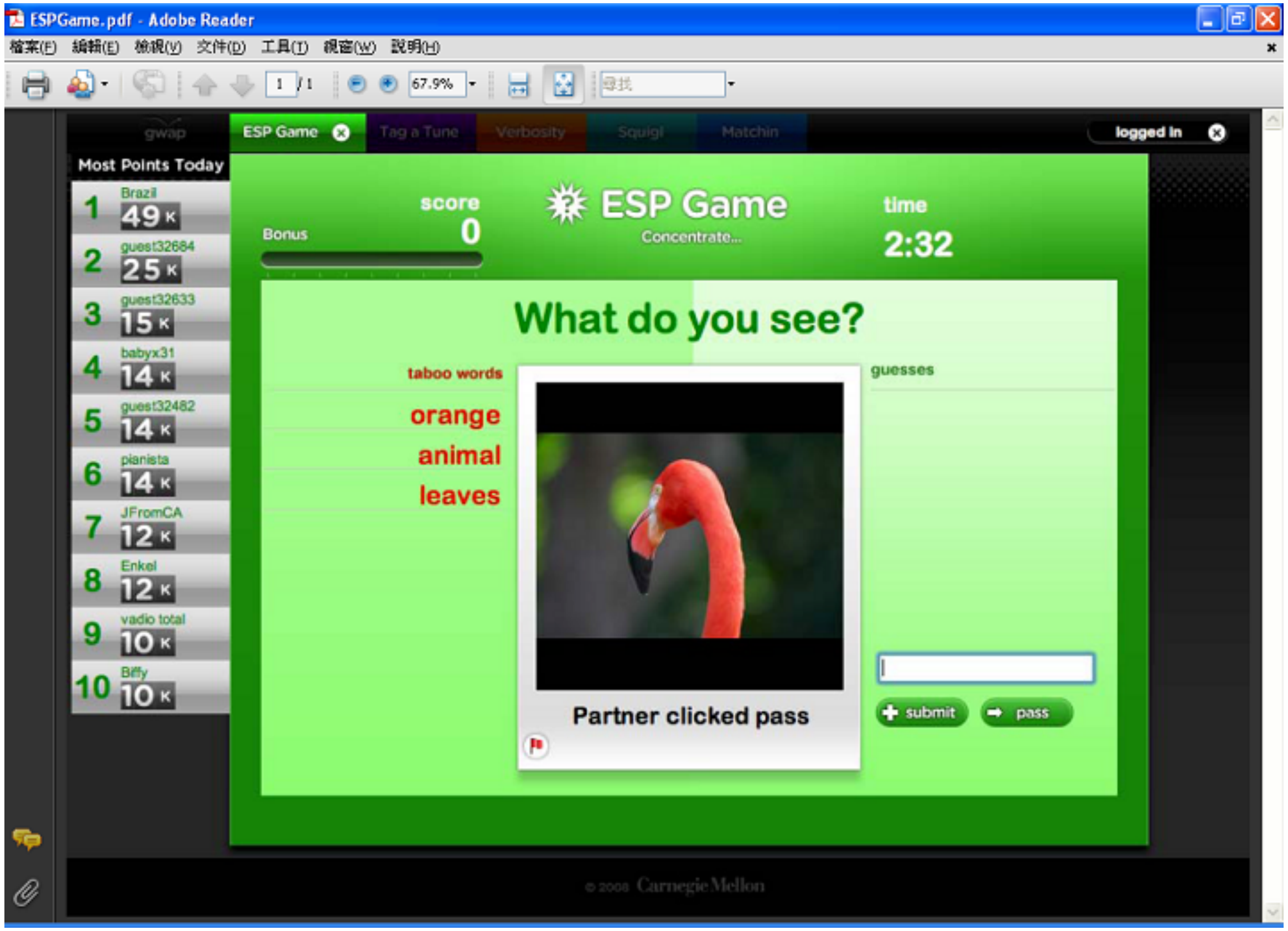
- Most of the games at early stage aimed to collect commonsense knowledge.
- Examples: Cyc, Open Mind, Mindpixel.
- Disadvantages:
  - Rely on online volunteers or paid engineers to enter information explicitly
  - Unable to scale up the system due to high cost
  - No validation mechanism to guarantee that the information collected is accurate

Social Games

- Social games were proposed to collect information from the players as a side effect of their playing.
- Advantage:
  - It encouraged more Internet users to contribute information to solve the AI problems because of the increasingly popularity of online game.
- TWO important factors for collecting information effectively from players through a social game:
  - Guarantee the quality of collected information.
  - Maintain the enjoyment of players in the game.

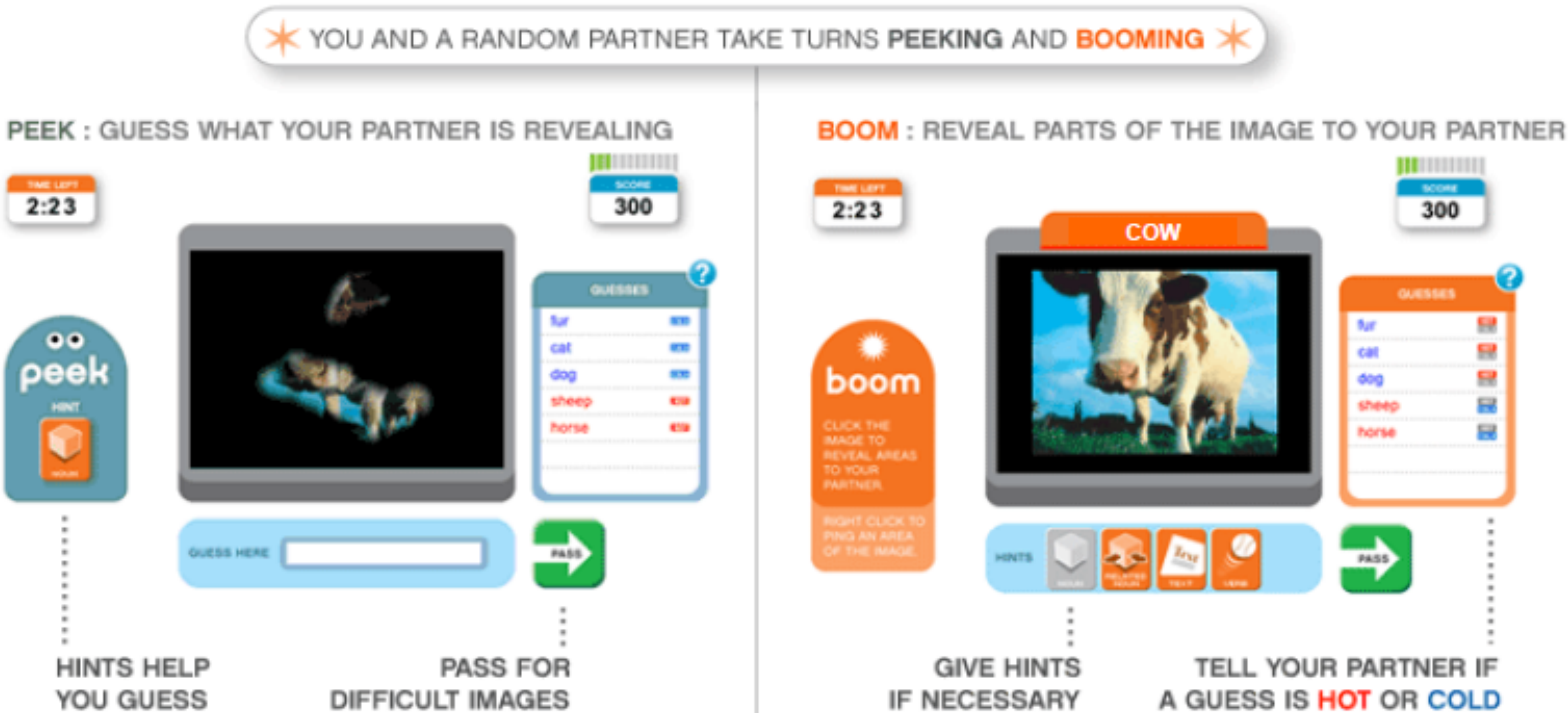
Example of Social Games (1)

- To collect text information from images
  - Examples (1): ESP game



Example of Social Games (2)

- To collect text information for images:
    - Examples (2): Peekaboom



Example of Social Games (3)

- To collect commonsense knowledge:
  - Examples (3): Verbosity

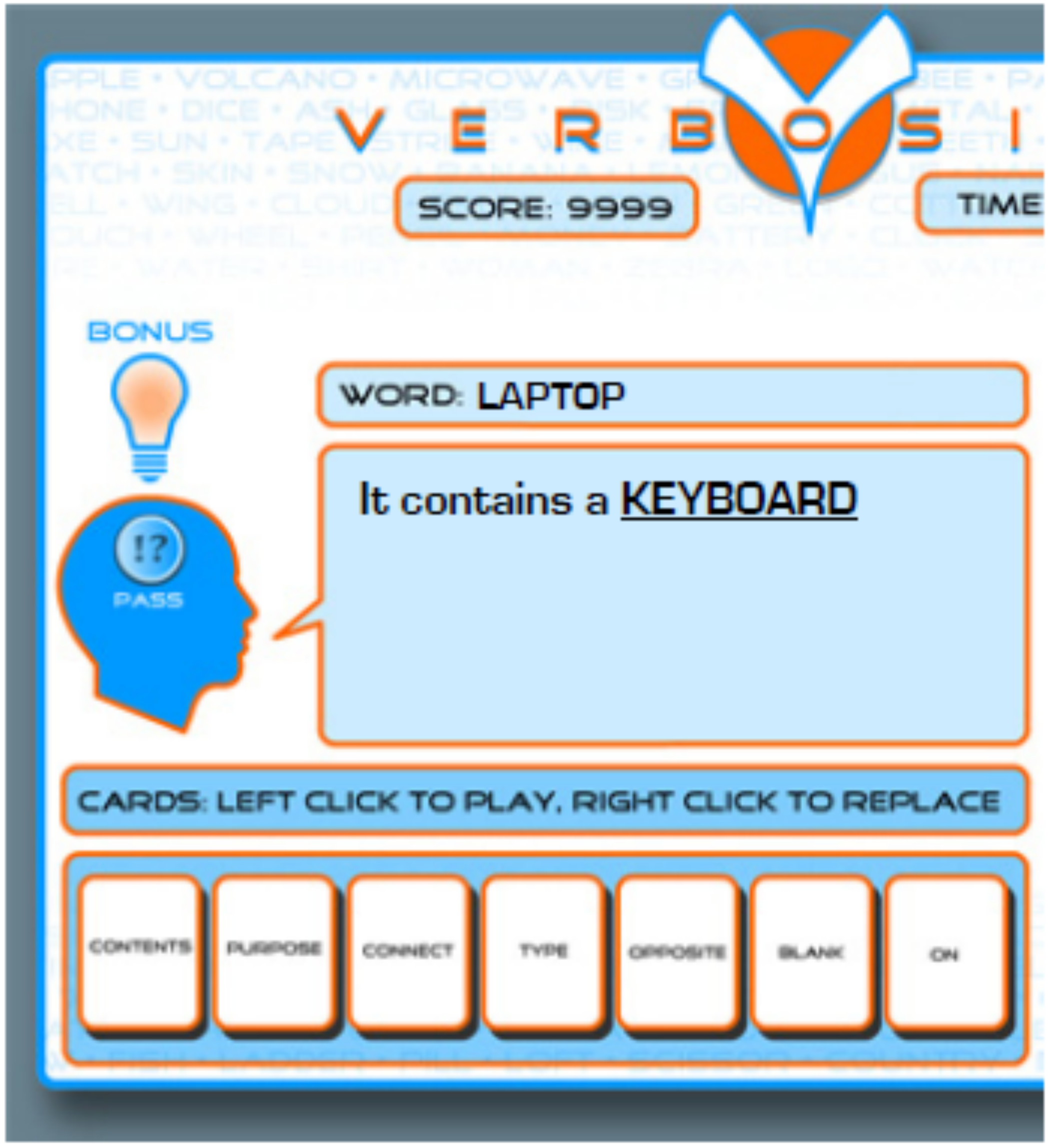
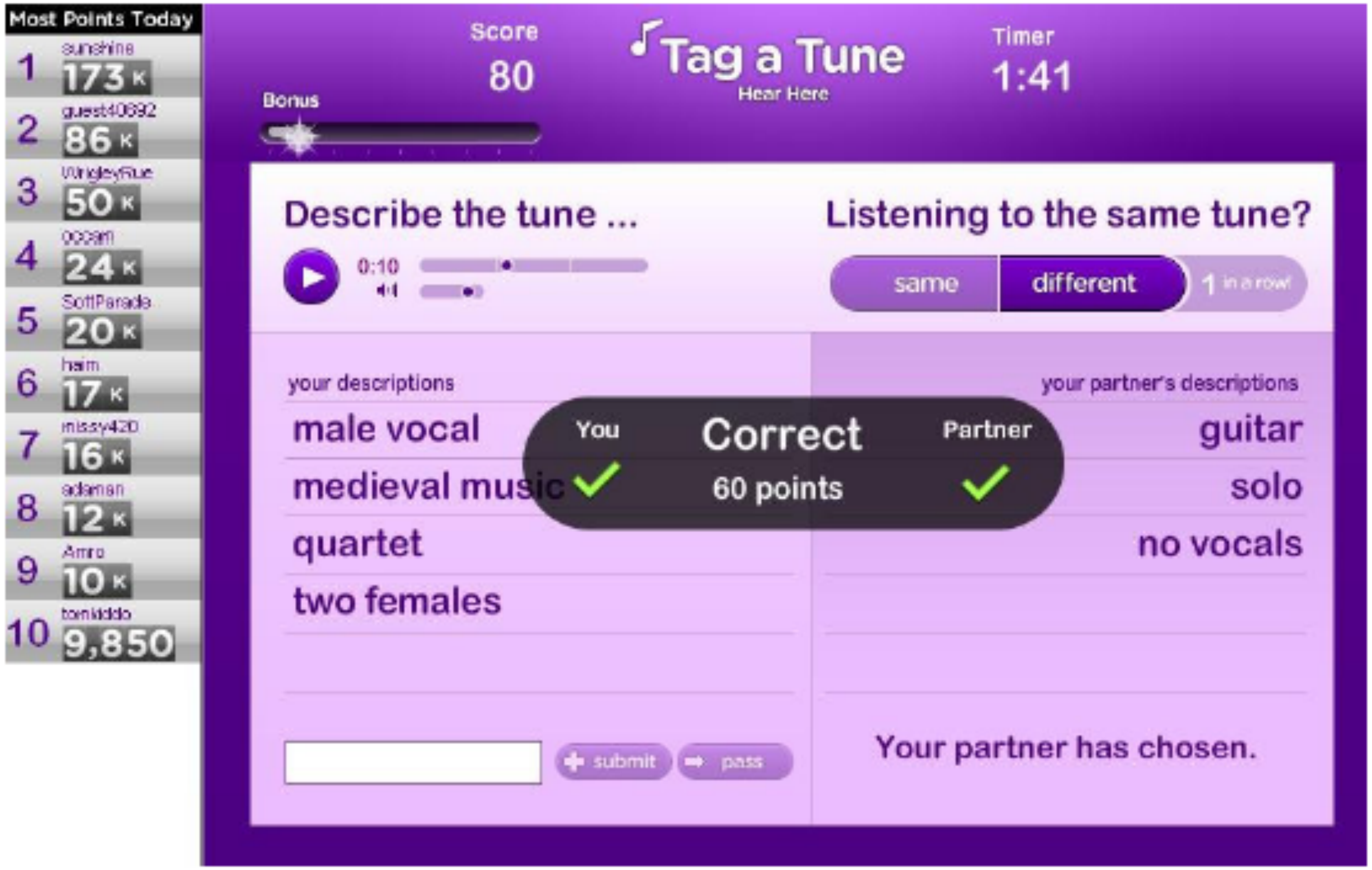


Figure 1. Part of the Narrator's screen.

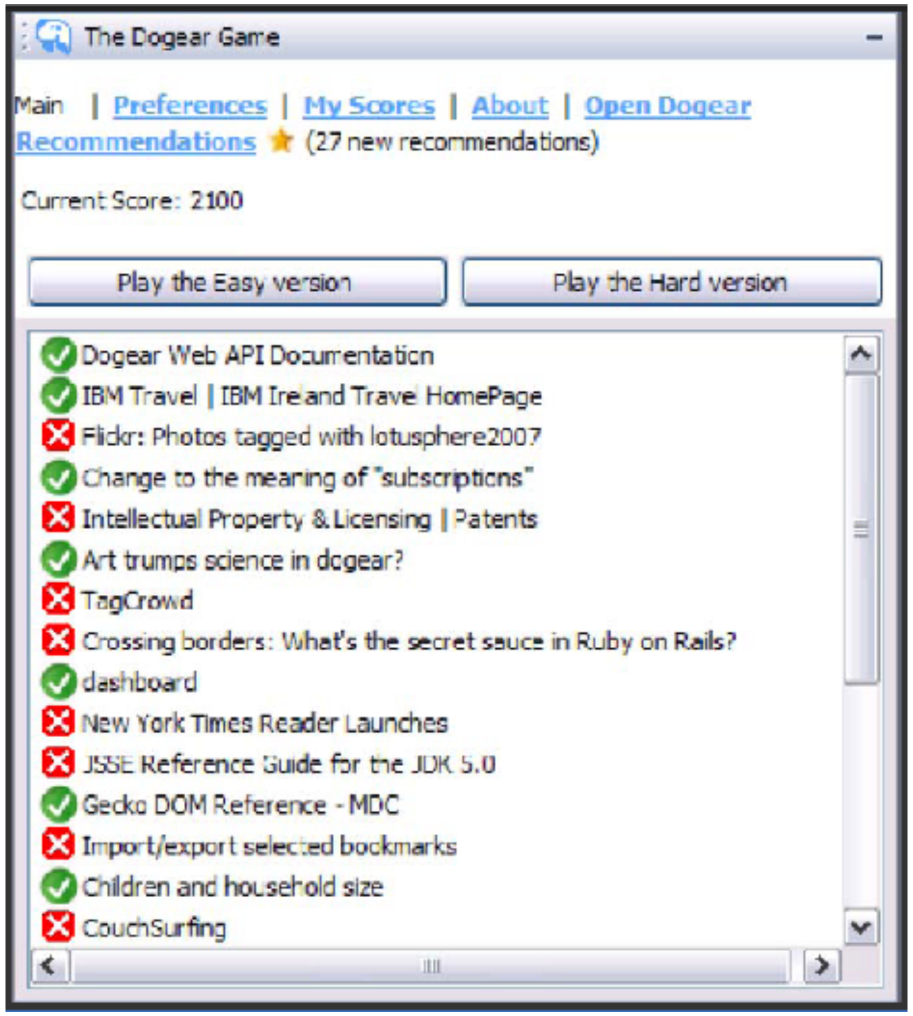
Example of Social Games (4)

- To collect subjective descriptions of sounds and music:
  - Example (4): Tagatune



Example of Social Games (5)

- To learn colleagues' bookmarks in an organizational goal:
  - Example (5): Dogear Game



Example of Social Games (6)

- To tag locations in the real world through gameplay in mobile social games:
  - Example (6): Gopher guessing game

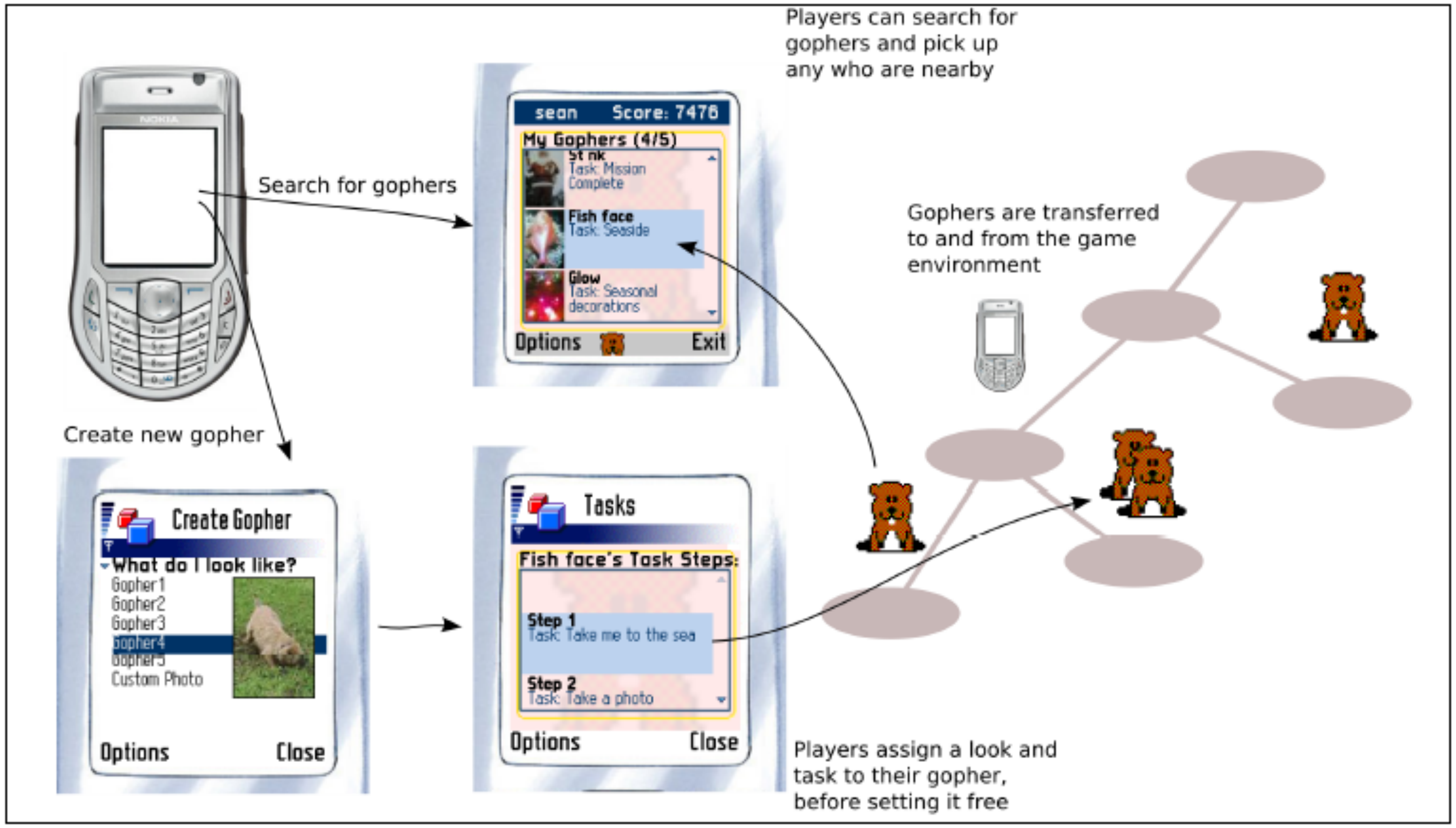


Figure 1. Real world experience, acquiring gophers

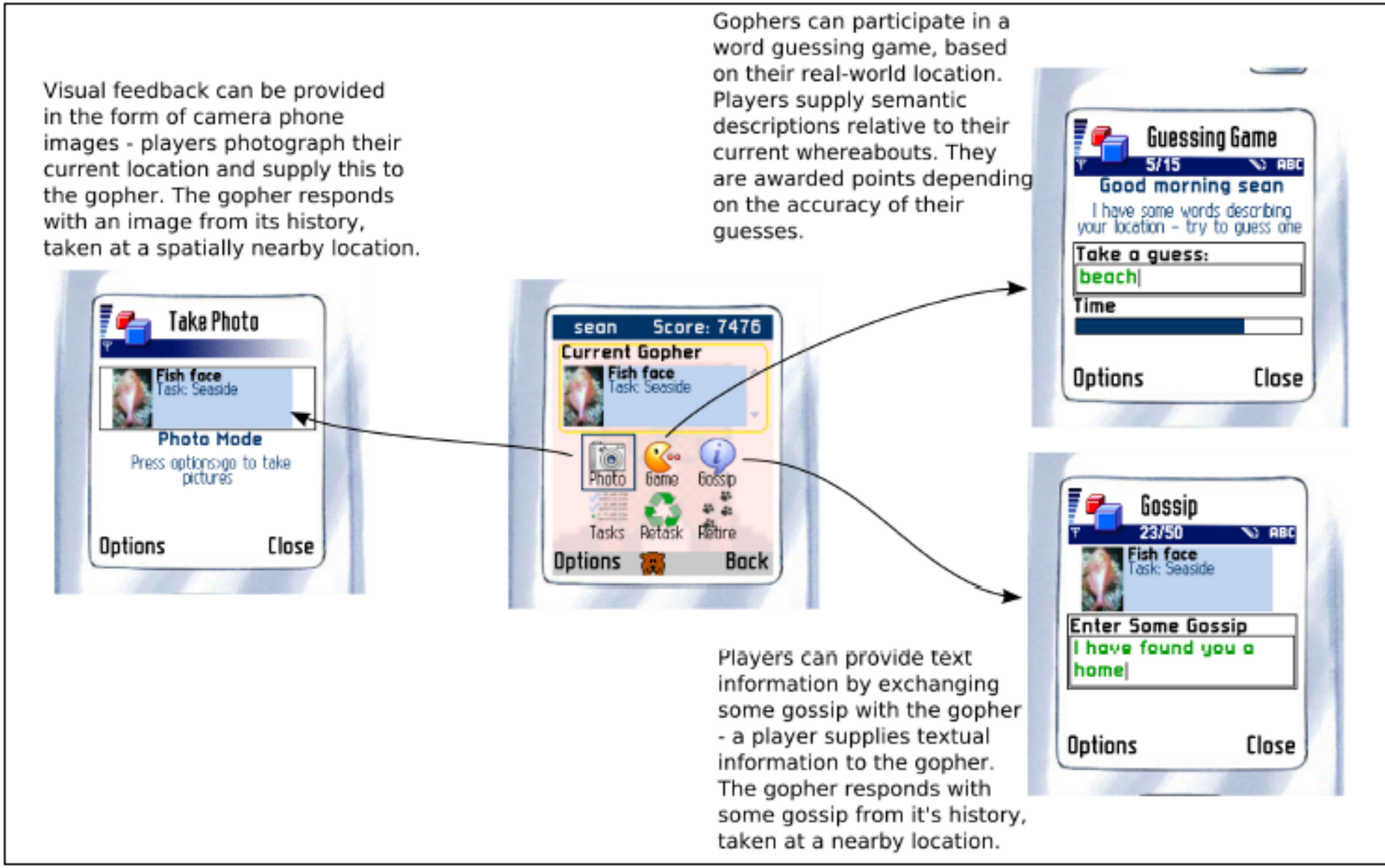


Figure 2. Real world experience, interacting with gophers

Example of Social Games

- Existing social games are casual games.
- Casual games are designed to have simple game play, and are intended for use by a wide player demographic.
- They are developed on an ad-hoc basis without a systematic approach, a formal framework does not exist for designing a social game in general.
- In the previous work of von Ahn et al., they listed out the design principles of current social games. Their study is description-based, but not in a formal framework.

Social Game Model

- Definition of data of a general data type
- Definition of a social game

Definition of Data

Definition 1 *A data $\mathcal{D}$ is an object with a data type $\mathcal{T}$ and a set of attributes denotes as $\mathcal{A}$:*

$$\mathcal{T} \in \{text, image, video, sound, URL\}$$

$$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_{\mathcal{X}})$$

where the date type $\mathcal{T}$ is the media type presented by $\mathcal{D}$; and each attribute $\mathcal{A}_{\mathcal{X}}$ has a relationship $Rel(\mathcal{A}_{\mathcal{X}})$ and a set of value $\mathcal{V}(\mathcal{A}_{\mathcal{X}}) = \{\mathcal{V}_1(\mathcal{A}_{\mathcal{X}})$, $\mathcal{V}_2(\mathcal{A}_{\mathcal{X}})$, ..., $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})\}$; and each value $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})$ is an object with its own data type and contains its set of attributes. $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})$ is also called metadata of $\mathcal{D}$.

Definition of a Social Game (1)

Definition 1 *A **social game** is a 4-tuple* $(\mathcal{SGPD}, \mathcal{GR}, \mathcal{GF}, \mathcal{ANS})$*, where sets:*

1. $\mathcal{SGPD} = (\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{C})$ *is the social game problem domain.*

   (a) $\mathcal{E} = \{e_i | i = 1, ..., x\}$ *is a set of problems that we want to solve where the problem* e_i *is to collect metadata of an input data* $\mathcal{D}$.

   (b) $\mathcal{F} = \{f_i | i = 1, ..., y\}$ *is the answer domain. Solutions to any* $e_i \in \mathcal{E}$*, which* f_i *is a value of an attribute of* $\mathcal{D}$ *that we want to collect, can only exist in* $\mathcal{F}$.

   (c) $\mathcal{G} : \mathcal{E} \times \mathcal{F} \rightarrow \Re \in [0..1]$ *is a function that determine whether an answer is correct to a problem.*

   (d) $\mathcal{C}$ *is a set of constraints in the game that*

      i. *indicating the attribute(s) we want to collect such that* $\mathcal{A}_{\mathcal{X}} \in \mathcal{A}$;

      ii. *indicating the set of values that we want to collect within* $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.

Definition of a Social Game (2)

Definition 1 *A **social game** is a 4-tuple ($\mathcal{SGPD}$, $\mathcal{GR}$, $\mathcal{GF}$, $\mathcal{ANS}$), where sets:*

1. $\mathcal{GR} = (\mathcal{D}, \mathcal{M}, \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{I}, \mathcal{O}, \mathcal{G}, \mathcal{W})$ *represents rules of a social game.*

   (a) $\mathcal{D}$ *is input data that we want to collect its metadata.*

   (b) $\mathcal{M} = \{m_i | i = 1, ..., x\}$ *is a set of metadata which are the values of attributes of* $\mathcal{D}$ *that we want to collect.*

   (c) $\mathcal{C}$ *is a set of constraints in the game that*

      i. *indicating the attribute(s) we want to collect such that* $\mathcal{A}_{\mathcal{X}} \in \mathcal{A}$;

      ii. *indicating the set of values that we want to collect within* $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.

   (d) $\mathcal{R} = \{r_k | k = 1, ..., nR\}$ *is the set of roles that players could have during a game.*

   (e) $\mathcal{P}(r_k) = \{p_j^k | j = 1, ..., n\mathcal{P}(r_k)\}$ *is the set of players that are assigned to the role* r_k *during a game.*

   (f) $\mathcal{I}(p_j^k) = \{i_m^{k,j} | m = 1, ..., n\mathcal{I}(p_j^k)\}$ *is the set of input given to the player* p_j^k *for solving the problem of input* $\mathcal{D}$ *during a game.*

   (g) $\mathcal{O}(p_j^k) = \{o_m^{k,j} | m = 1, ..., n\mathcal{O}(p_j^k)\}$ *is the set of output provided by the player* p_j^k *for solving the problem of input* $\mathcal{D}$ *during a game.*

   (h) $\mathcal{G}()$ *is a procedure that determines whether players have produced outputs that meet specific requirements within a game segment. If so, return a possible answer* $f \in \mathcal{F}$.

   (i) $\mathcal{W}(p_j^k)$ *is the reward that the player can receive for solving the problem of input* $\mathcal{D}$ *during a game where* $\mathcal{W}(p_j^k) \in \{w_i | i = 1, ..., y\}$. *Players will receive a reward when achieving the winning condition of the game.*

Definition of a Social Game (3)

Definition 1 *A **social game** is a 4-tuple* $(\mathcal{SGPD}, \mathcal{GR}, \mathcal{GF}, \mathcal{ANS})$*, where sets:*

1. $\mathcal{GF} = \{pSel, eSel, tMax, pNum, \mathcal{GM}, \mathcal{UI}\}$ *represents the flow of a social game.*
   (a) *pSel() is a procedure that selects players to play a game and assigns roles to them.*
   (b) *eSel() is a procedure that picks a problem from the problem set.*
   (c) *tMax is the maximum duration of a game.*
   (d) *pNum is the number of players of a game. It may be a single-player game, two-player game or multi-player game.*
   (e) $\mathcal{GM} \in$ *{collaborative, competitive, hybrid} is the mechanism of a game.*
   (f) $\mathcal{UI} = \{ui_j | j = 1, 2, ..., x\}$ *is the set of design characteristics of user interface.*

Definition of a Social Game (4)

Definition 1 *A **social game** is a 4-tuple* $(\mathcal{SGPD}, \mathcal{GR}, \mathcal{GF}, \mathcal{ANS})$*, where sets:*

1. $\mathcal{ANS} = (\xi, \tau)$ *represents answer extraction. It defines how answers are generated for each problem based on all the games played.*
   (a) ξ *is a data structure that supports the following operations:*
      i. ***add()*** *takes* $e \in \mathcal{E}$ *as input and updates its internal counters.*
      ii. ***count()*** *returns the internal count for a particular* $f \in \mathcal{F}$
   (b) τ *is a frequency threshold for accepting an answer.*

Answer Extraction Procedure

- It generates answers to each problem based on all the games played in the system.
- Answers with a frequency lower than the threshold will be pruned away.

TABLE I
THE ANSWER EXTRACTION PROCEDURE

for each $e \in \mathcal{E}$ **do**
    for each game segment GS working on problem e **do**
        if $\mathcal{G}() = TRUE$ **then**
            $\xi.add(f)$
        end if
    end for
    for each $f \in \mathcal{F}$ **do**
        if $\xi.count(f) \geq \tau$ **then**
            f is regarded as an answer for e
        end if
    end for
end for

Properties of Social Games

1. Type of information to be collected
2. Game Structure
    1. Output-agreement Game
    2. Input-agreement Game
    3. Inversion-problem Game
    4. Output-optimization Game
3. Verification Method
    1. Symmetric
    2. Asymmetric
4. Game Mechanism
    1. Collaborative
    2. Competitive
    3. Hybrid
5. Player Requirement

Categorization of Social Games

TABLE I
CATEGORIZATION OF SOCIAL GAMES

Game Structure	Verification Method	Game Mechanism
Output-agreement	Symmetric	Collaborative or Hybrid
Input-agreement	Symmetric	Collaborative or Hybrid
Inversion-problem	Asymmetric	Collaborative or Competitive or Hybrid
Output-optimization	Symmetric or Asymmetric	Collaborative or Competitive or Hybrid

Subjective vs. Objective Information

- For subjective information, the information presented for the same subject is affected by users because of different choices of vocabularies for the same subject.
  - lower probability on players' correct outputs being the same
- For objective information, the information presented for the same subject is NOT affected by users because of same choices of vocabularies for the same subject.
  - higher probability on players' correct outputs being the same

Game Structure (1)

- Game structure defines the key elements of a game including players' input, players' output, the relationship among the input and output of players, and the winning condition
- Four types of game structure

1. Output-agreement Game
2. Input-agreement Game
3. Inversion-problem Game
4. Output-optimization Game

Game Structure (2)

- Output-agreement Games: All players are given the same input and must produce outputs based on the common input
  - An output-agreement game should be used to collect objective information
- Input-agreement Games: All players are given inputs that are known by the game (but not by the players) to be the same or different. The players are instructed to produce outputs describing their input, so their partners are able to assess whether their inputs are the same or different. Players see only each other's outputs
  - An input-agreement game should be used to collect subjective information

Game Structure (3)

- Inversion-problem Games: The first player has access to the whole problem and gives hints to the second player to make a guess. If the second player is able to guess the secret, we assume that the hints given by the first player are correct.
- Output-optimization Games: All players are given the same input and their outputs are the hints of other players' outputs.
  - An output-optimization game should be used to collect subjective information, because the output pattern of players reflects outputs of players are strongly affected by others' outputs. It is subjective.

Verification Methods

- Verification method of a game defines the method to check the output accuracy of players by asking players to do the same task or different tasks
- Symmetric Verification Games: Either an output-agreement game or an input-agreement game is symmetric verification
- Asymmetric Verification Games: Players are assigned to one of the roles to do different tasks

Game Mechanism

- Game mechanism defines the relationship of all players in the game in order to achieve the winning condition
- Collaborative Games determine the winning condition of all players. The accuracy of output is guaranteed by collaboration of all players.
- Competitive Games determine the winning condition of a player. Output accuracy is guaranteed by information stored in a database. Players' enjoyment in the game can be increased in competition.
- Hybrid Game

Player Requirements (1)

- Player requirement defines the rules on accessing the game of all players.
- In Synchronous Games, players have to give real-time response to other players' action.
- In Asynchronous Games, players do not have to give real-time response to other players' action. The information collected from one player is stored in a database and will be used to determine the correctness of other players' output.

Player Requirements (2)

- Number of players define the following types:
- Single-player Games: It allows one player to play and the other's moves can be simulated from the prerecorded game. Only inversion-problem game can be a single-player game.
- Two-player Games: It allows two players to play together.
- Multi-player Games: It allows multiple players to play together. Only hybrid games can be a multi-player game.

Summary

TABLE II
CATEGORIZATION OF SOCIAL GAMES WITH EXAMPLES

Game Structure	Verification Method	Game Mechanism	Player Requirement		Examples
			Num of Player	Game Play	
Output-agreement	Symmetric	Collaborative	2	Synchronous	ESP, Matchi, Squigl, OntoGame
		Hybrid	Multi-players	Synchronous	Common Consensus, Social Heroes
		Hybrid	Multi-players	Asynchronous	Gopher Game
Input-agreement	Symmetric	Collaborative	2	Synchronous	TagATune
		Hybrid	N/A	N/A	N/A
Inversion-problem	Asymmetric	Collaborative	1 or 2	Synchronous	Peekaboom, Verbosity
		Competitive	2	Asynchronous	Dogear, CyPRESS, CARS
		Hybrid	1 or Multi-players	Synchronous	Phetch
Output-optimization	Symmetric	Collaborative	2	Synchronous	Restaurant Game
		Competitive	N/A	N/A	N/A
		Hybrid	Multi-players	Synchronous	Diplomacy

Design Guidelines

TABLE III
THE DESIGN GUIDELINES ON SOCIAL GAMES

```
if data.attr.value = objective then
    struct = (output-agreement or inversion-problem)
else if (data.attr.value = subjective and
data.attr.value.data-type = output-pattern) then
    struct = output-optimization
else if (data.attr.value = subjective and
data.attr.value.data-type ≠ output-pattern) then
    struct = (input-agreement or inversion-problem)
end if
if struct = (output-agreement or input-agreement) then
    if no-of-players > 2 then
        mechanism = hybrid
    else if no-of-players = 2 then
        mechanism = collaborative
    end if
end if
if struct = inversion-problem then
    if no-of-players > 2 then
        mechanism = hybrid
    else if no-of-players = 2 then
        if ans verification based on outputs of players then
            mechanism = collaborative
        else if ans verification based on info stored in DB then
            (mechanism = competitive and time = async)
        end if
    else if no-of-players = 1 then
        (mechanism = (collaborative or hybrid) and
        moves simulated from the prerecorded game)
    end if
end if
if struct = output-optimization then
    if no-of-players > 2 then
        mechanism = hybrid
    else if no-of-players = 2 then
        mechanism = (collaborative or competitive)
    end if
end if
```

Final Remarks

- Formal description of HCS
- Categorization of HCS

- To consider using the model to design HCS for solving a set of inter-related problems
- To consider using the model to handle different data types under different environmental context

Q & A