# Three Tiers of Gamification in a College Course on Problem Solving for Global Challenges

Steven L. Tanimoto

Paul G. Allen School of Computer Science and Engineering
University of Washington
Seattle, WA 98195
USA
tanimoto@uw.edu

**Abstract.** Three different forms of gamification in learning are described in the context of an intensive four-week course for incoming freshmen at the University of Washington. The course covers topics from several disciplines including computer science, game theory, and the learning sciences. The three tiers of gamification are (A) students learn while playing games, (B) students learn when they formulate complex global problems as games, and (C) students learn as they take on agile software-development team roles as they create digital games. Each of these tiers has motivational justifications, and in addition, these tiers offer complementary benefits. For example, the gamification of "wicked" problems in tier B requires and stimulates meta-cognitive thinking. When the students themselves are charged with formulating the problems as games, they end up better understanding the factors that go into successful problem solving, including "thinking outside of the box" and reformulating problems to make them more tractable to solution. Presented here are the design rationale for the course, observations about student learning and challenges, and how the course's pedagogy compares with methods described in the literature.

**Keywords:** gamification, learning, game design, wicked problem, global challenge, computer programming, Python, problem formulation, classical theory of problem solving, agile programming, scrum, collaborative design, tier, serious games, student designs.

## 1 Introduction

Game elements can be brought into courses in a variety of ways to foster motivation, engagement, and to deepen the learning that students do. This paper shares with readers the experiences and insights of the author on teaching a course that uses gamification at three different tiers, so that students learn interdisciplinary content related to global challenge problems and so that they learn how to formulate problems following the classical theory of problem solving.

### 1.1 Purposes of the Paper

The main goal of the paper is to share with other educators the reasons and the means for a possible way to exploit multiple aspects of gamification in a single course. As part of this sharing, a particular example course is offered that could be a starting point for others.

A secondary goal is to argue (to educators and researchers) for some of the benefits to students of using game design with the theory of problem solving as a vehicle for engaging with very challenging problems, such as climate change or nuclear proliferation.

While the conduct of the course was not intended to be a research project, part of this paper considers the gamification within the course from the perspective of the research literature in the gamification of learning. In retrospect, a variety of research questions can now be asked that warrant work in the future. Thus, a third purpose of the paper is to ask these questions from the context of the course retrospective.

### 1.2 Overview

First, we briefly describe the design-based research methodology assumed in this paper. Next, we describe the course to provide a context. (The course is an intensive four-week course with a focus on problem solving and formulation.) We then explain how students work with gamified material at tier A during week 1, then engage with tier B gamification starting in week 2, and finally start tier C in week 3. In weeks 3 and 4, the students are engaging with all three tiers of gamification: playing games, formulating games, and role playing in their agile development teams.

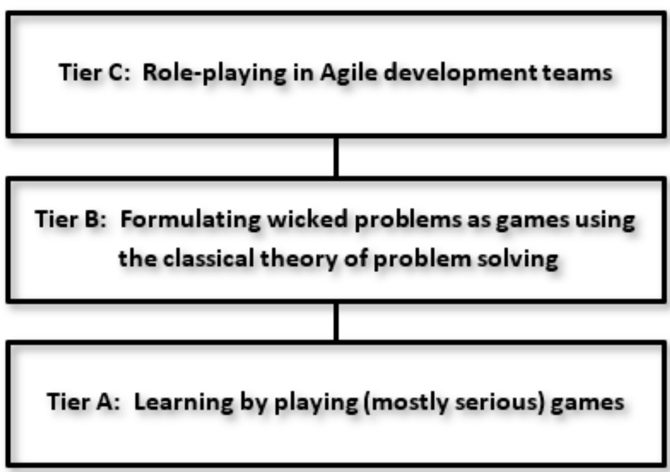


**Fig. 1**. The three tiers of gamification in the course.

Next, we give a brief introduction to the classical theory of problem solving which serves as a rigorous basis for gamification of problems. This is followed by a discussion of so-called “wicked” problems. Wicked problems play a central role in the course as each student team gamifies one wicked problem.

Section 5 describes the agile software-team development roles that the students play in the third tier of gamification in the course. Some modifications to traditional industry scrum structures have been made in order make the agile approach work in a classroom context.

The final sections of the paper describe how game evaluation is performed by students, how the gamification in the course relates to gamification in the literature and a discussion with the questions and suggestions for future research.

### 1.3 Design-Based Research

Research in educational practices is subject to challenges that tend to be less pronounced in other scientific research. For example, experiments with students are typically constrained not only by equity concerns, but by school and curriculum regulations, as well as difficult-to-control variables such as teacher personality, and numbers of student human subjects available within a given educational context. Design-based research is a methodology that takes these limitations into account yet permits obtaining valuable insights from conducting novel practical interventions [1, 2]. The case study described in this paper is an example of design-based research.

## 2 About the Course

### 2.1 Motivation for the Course

The primary knowledge students gain in the new course is about problem formulation and solving. Key elements of this knowledge come from cognitive psychology and artificial intelligence. The course aims to foster a “culture of problem solving.” Along the way, the students also learn specific techniques in game design, computer programming, the Python language, game theory, software development and about various global challenges. They also learn and practice iterative design.

Other goals of the course include helping new university students get oriented towards college-level academics and the university community, helping students make social connections, helping students develop metacognitive skills to become better learners, and to help students in class-level collaboration habits and skills that are likely to be helpful to them as they continue through university and afterwards. The course design is constrained by the scheduling requirements of the University of Washington’s Early Fall Start 4-week period, and its policy of not allowing pre-requisite courses for participation. Thus, the course description advises that students have prior programming experience, but it does not require that.

## 2.2 Administrative Aspects of the Course

After students have been admitted to the University of Washington as freshmen, they normally arrive in the autumn to begin their studies. However, they are offered the possibility of arriving about one month early to the Seattle campus and registering for an intensive 4-week, full-credit, graded course at an extra cost.

The course assumes that students will know or quickly learn basic computer programming. It also assumes that students will be ready to commit 4 weeks of full-time study to the course, meaning at least 40 hours/week.

| Week number | Main contents |
|---|---|
| 1 | Introduction + Python + Serious games at Tier A |
| 2 | Problem solving + formulation methods (Tier B) |
| 3 | Initial game designs, eval's in Agile teams (Tiers B, C) |
| 4 | Iterative design to final game and eval's (Tiers, A, B, C) |

(a)

| | | | |
|---|---|---|---|
| Tuesday, Aug. 23: Course Introduction; Begin P0: Playing and reviewing serious games | Wednesday, Aug. 24 Python Introduction; Python's basic data objects; working with strings in Python | Thursday, Aug. 25: Working with numbers and lists in Python; P0 (game reviews) due | Friday, Aug. 26: Python's control structures; defining functions in Python; Classical Theory of Problem Solving; GD Game design intro; control structures |
| Tuesday, Aug. 30: Python class definitions; Classic Puzzles | Wednesday, Aug. 31: Formulating Problems; A game with graphics; intro. to Tkinter; P1 (Python exercises) due | Thursday, Sept. 1: Wicked problems; Running SOLUZION from IDLE; Demo: Formulating a maze prob. | Friday, Sept. 2: Game structures; software dev. with agile teams; Project ideas and planning; P2 (prob. formulation exercises) due |
| Tuesday, Sept. 6: [Quiz 1 covering game structures] Continuing on software development with Scrum; P3 Milestone A (wicked prob. pre-formulations) due | Wednesday, Sept. 7: P3A presentations and stakeholder feedback; The Prisoner's Dilemma and related obstacles to prob. solving | Thursday, Sept. 8: Groups respond to P3A feedback and present the latest versions of their game ideas. | Friday, Sept. 9: [Quiz 2 covering the prisoner's dilemma]; Learning curves; Scrum practice. P3 Milestone B (storyline and first working code) due; |
| Tuesday, Sept. 13: Guest speaker on academic majors; [Quiz 3 on learning curves, theory of prob. solving, wicked prob.]; progress reports; P3 Milestone C (preliminary game) due | Wednesday, Sept. 14: Wicked Problem Case Study: Homelessness; Scrum practice, and further development. | Thursday, Sept. 15: Play-testing and stakeholder feedback; P3 Milestone D (semi-final game) due | Friday, Sept. 16: Final presentations, demos, and game evaluations; P3 Milestone E (final game, report, and responses to class feedback) due |

(b)

**Fig. 2**. (a) Main course activities by week, and (b) the detailed calendar.

### 2.3 Schedule and Content

The main topics of the course are scheduled as shown in Fig. 2. In the beginning is an introduction to the course.

During the first week, most of the class time is spent reviewing or (for some students, learning for the first time) elementary computer programming. However, as an introduction to serious games aspect of the course, students must select two games from a provided list of serious games, play the games, and write a formal review of each one covering each part of a given game-evaluation rubric. As they play these two serious games, they are participating in gamified learning at the first level: Tier A. They learn about the problems addressed by the games, such as global climate change or the tragedy of poverty in the world. They also learn about game mechanics and affective aspects of a game's design. This paper gives less attention to Tier A of gamification than the other two, as that subject has been given much attention by others (see, for example, Tobias and Fletcher [3]; Connelly et al [4]; a recent meta-analysis show that interest in learning through game playing remains an important realm for innovation in learning [5].)

During the second week, students learn the basics of problem-solving theory, and a structured method for formulating problems. They learn key aspects of wicked, global-challenge problems, and how to evaluate the wickedness of such a problem. Weeks 3 and 4 are dedicated to the student projects, which involve selecting and formulating a wicked problem, and gamifying it (learning at Tier B of gamification). During this second half, students are working in teams of 4 to 5 students each, and they are role playing as members of agile development teams (and as they play these roles they are learning at Tier C of gamification).

## 3 Classical Theory of Problem Solving

### 3.1 Rationale

In terms of student learning in this course, Tier B is the most important level of gamification. The gamification of globally challenging problems requires that students not only learn something about the problems they are gamifying but that they learn how to formulate any problem using the classical theory. The classical theory itself provides a rigorous basis for the gamification, and it allows evaluation of the game in terms of a variety of game mechanics and effective modeling of phenomena relevant to the problem. In order to explain this tier of gamification accurately, this section describes the theory itself.

Credit for the classical theory of problem solving goes to many researchers who developed the theory from about 1955 to 1985. A good summary of the more basic concepts and applications of the theory is the review by Newell and Simon (1971 [6]), which was further expanded in their book *Human Problem Solving* (1972 [7]). Unlike earlier studies of problem solving such as Polya's guide to solving mathematical problems (1945 [8]) the classical theory can be broadly applied to both human and computer solving of problems. Consequently, later work such as Pearl's book on heuristics (1984 [9]) carried the classical theory to an advanced level.

Herbert Simon argued that the theory could be applied not only to traditional mathematical and engineering problems but also to design problems such as in architecture, urban planning, and public policy (Simon 1969 [10]). Some planners, such as Rittel and Weber (1971 [11]) pushed back on this, arguing that some problems are too "wicked" to be amenable to the classical theory. More recent thought allows for the theory to apply broadly, but with elaborations to the theory. Scott Klemmer [12] has set out a spectrum of types of problems that spans a continuum of what Jonassen would call a well-structured problem (Jonassen 2007 [13]) to artistic production that many would not consider problems at all. Figure 3 illustrates the spectrum, with the well-structured Towers of Hanoi puzzle on the left end, the writing of poetry on the right end, and design (e.g., architectural) in the middle. The text in the figure explains that the expected degree of agreement, by the community, of what a valid solution is to the problem serves a proxy for how well-structured the problem is.

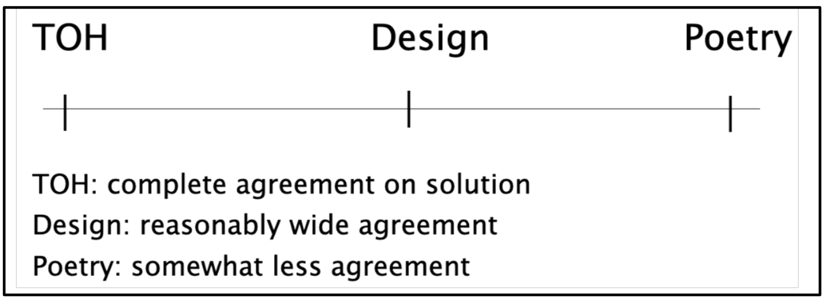


**Fig. 3**. Continuum of problem types, with the Towers of Hanoi at one end, and art/poetry at the other. Students learn formulation first at the left end but later learn to apply it in the middle. Design (e.g., architectural) is in the middle (after Klemmer [12]).

### 3.2 States and Operators

The theory begins with what a problem is. Informally, it is a need, or at least a possible need, to be satisfied in some way. A problem typically is presented with a starting situation, such as, to take a simple example, a person being at some location inside of a maze, and a statement about what it would mean to solve the problem, such escaping from the maze by finding a path to an exit. The starting situation we will call the "initial state" of the problem. Each time the solver (i.e., the person trying to solve the problem) takes an action (such as a step from the current location to a new location in the maze) we call that making a move or taking an action. In a maze, a possible move might be from a location (x, y) = (13, 7) to an adjacent location (14, 7); this results from walking in the east direction, and this changes the solver's situation to a new state (the state of being at position 14, 7 in the maze). Another move, such as from (3, 5) to (4, 5) is also due to walking in the east direction. Here we can call "East" an operator; it's like a general kind of action that might be applicable at any location in the maze, provided there is no wall blocking the person from going east from that location. Using the cartesian coordinates again, we can say that the East operator can change the current state (x,y) to a new state (x+1, y). At any location in the maze, there might be four possible choices of operator to try: North, East, West, and South. The solver tries to find a solution by trying sequences of operators, starting from the initial state, that lead to a "goal state" (e.g., location of an exit from the maze).

In the classical theory, we must represent the problem by clearly specifying (a) its initial state, (b) its set of operators, and (c) what its goal states are. Once this has been accomplished, a rich world of potential solving strategies typically arises. It works whether the problem is as simple as getting out of a maze or designing a modern university building. There may be additional sources of information about the problem that go beyond these three components and are helpful (see, e.g., Pearl [9]), but these are the basics, analogous to the atomic theory of matter in chemistry and physics.

The three problem components mentioned above are (a) initial state, (b) set of operators, and (b) goal states. Each of these needs to be specified unambiguously when a problem has been properly *formulated*. Formulation can be a simple technical exercise for puzzles and games such as maze solving. However, it is completely non-trivial for problems further to the right in the spectrum of Fig. 3. The steps of problem formulation will be described later. However, the process of formulation is greatly facilitated by understanding effective ways to represent the three problem components. A first step in that understanding involves the use of a symbolic notation that is mathematical in style, although it does not involve advanced mathematics. Through the use of the notation, students start to learn to distinguish unambiguous from ambiguous representations of a problem. This is essential to the kind of gamification they will do. The following lines show the notation.

i. A problem formulation is a triple: $(\sigma_0, \Phi, \Gamma)$ where $\sigma_0$ is the initial state of the problem, $\Phi$ is a set of operators, and $\Gamma$ is a set of goal states.

ii. Each operator $\varphi_i \in \Phi$ has a precondition and a state-transformation function.

iii. These implicitly define $\Sigma$ , the set of all states reachable from $\sigma_0$ by applying members of $\Phi$ zero or more times.

Line (i) above gives a symbol for each of the three problem components: initial state, set of operators, and set of goal states. The lower-case sigma, $\sigma$, is used to represent a state, and the subscript 0 makes it specifically the initial state. The symbol $\Phi$ (capital Phi) represents the set of operators, which could also be written, more verbosely, as
$\{ \varphi_0 , \varphi_1 , \varphi_2 , \ldots, \varphi_{m-1} \}$.

The symbol $\Gamma$ (upper-case Gamma) represents the problem's goal states. These could also be written as $\{ \gamma_0 , \gamma_1 , \gamma_2 , \ldots, \gamma_{g-1} \}$.

Line (ii) indicates that each operator has two important parts. The precondition provides a way of telling, given a particular state, whether that operator can legally be used in that state. For example, the maze operator East has a precondition that there not be a wall on the east side of the solver that blocks the way. The operator also has a "state-transformation function." This is the function that takes the current state and determines the new state, such as the function that inputs (x, y) and outputs (x+1, y) which represents the state after moving to the east one step. In a game, the state-transformation function is essentially a game mechanic that implements the "rule" for what happens when the player chooses a particular action.

Line (iii) introduces the symbol Σ (capital Sigma) which is the set of all the possible states of the problem (or game, when formulated this way). This set we call the *state space* for the problem. We can write this as follows:

$\Sigma = \{ \sigma_0 , \sigma_1 , \sigma_2 , \ldots, \sigma_{n-1} \}$

Here *n* is the number of possible states for the problem. It could be finite or infinite, depending on the problem. It represents the size of the state space, and it is an interesting characteristic of the problem in its own right. If *n* is small, then the problem is likely to be simple to solve (unless it is impossible to solve – for example a small maze with no way out from the middle would have a small *n* but be impossible to solve.) Similarly, a student-designed game with a small *n* might not be fun to play if it is perceived as being too easy to win. Here, we are considering an extremely simple and somewhat shallow means of evaluating a game, but it comes immediately from the use of the theory and is perhaps analogous to counting the number of protons in an atom in the subject of chemistry or the number of possible crystal structures for a given molecule.

### 3.3 Example Problem Formulation for a Toy Problem

In order to illustrate the theory here, as we also do for students, we present the essential elements of a formulation of the Towers of Hanoi problem. This puzzle was published by Edouard Lucas in 1895 [14], and since then has been widely used as a motivation for functional recursion in beginning computer science courses in colleges. Here we are not interested in recursion but in how the problem itself is formulated unambiguously.

The Towers of Hanoi problem is based on a legend that describes a temple in Asia where a group of monks have a pile of 64 disks (with holes in their centers) of increasing sizes that they must transfer from one of three poles to another of the poles, by moving one disk per day from the top of one pile to the top of another pile, such that a disk is never placed on top of another disk having smaller diameter. Although the monks should make correct moves each day, the world will end when they reach the goal state. Fortunately for us all, the size of the state space is $n = 3^{64}$, and the minimum number of moves required to reach the goal is $2^{64}-1$ which represents so many days that it is orders of magnitude longer than the current age of the universe.

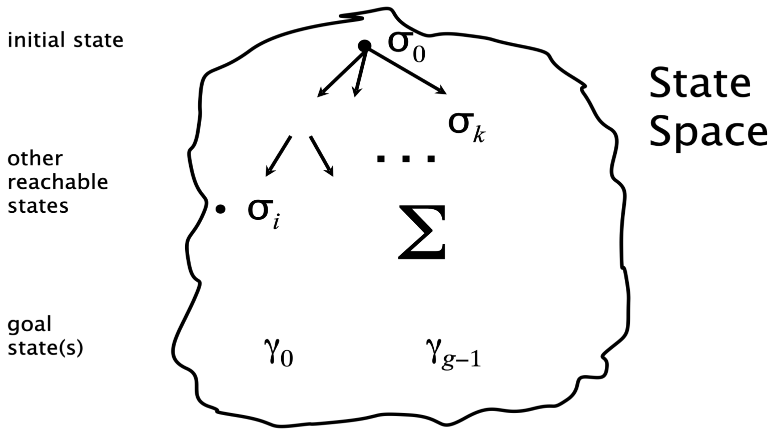


**Fig. 4**. A general state space, the basis for gamifying many problems. If we include not only the states, but also the moves from one state to another, we get the problem-space graph by considering the states as the nodes of the graph and the arrows as the edges of the graph.

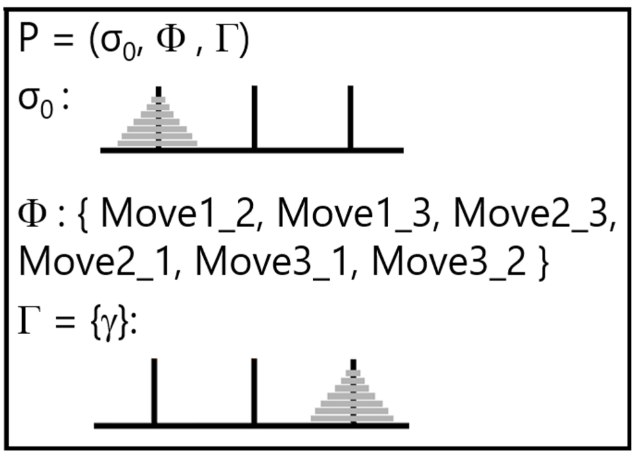


**Fig. 5.** Formulation for the Towers of Hanoi puzzle using the classical theory.

A formulation for the Towers of Hanoi problem, following the classical theory of problem solving, is shown in Fig. 5. The operators here have been given descriptive names rather than their $\varphi_i$ symbols. The first operator "Move1_2" represents moving the top disk on peg 1 to peg 2. The precondition for this operator is that there must be at least one disk on peg 1, and if there are any disks on peg 2, they must all be larger in diameter than the topmost disk on peg 1. When in a particular state of the puzzle, such as the initial state shown in Fig. 5, this precondition is true, then the move is allowable. Note that this problem has only one goal state $\gamma$ and so no subscript is needed to distinguish it from any other goal state.

Using this formulation, and setting a particular number of disks for the puzzle, say 4, we can proceed to enumerate the states and the moves that lead from one state to another. With some special attention to laying out an arrangement of the states as vertices for the problem-space graph, we can get the visualized graph shown in Fig. 6. In this drawing, the initial state corresponds to the node in the lower left corner, and the goal state corresponds to the node in the lower-right initial state and moving in a somewhat irregular way towards the right. This is not a solution path since it does not connect the initial state all the way to the goal state.

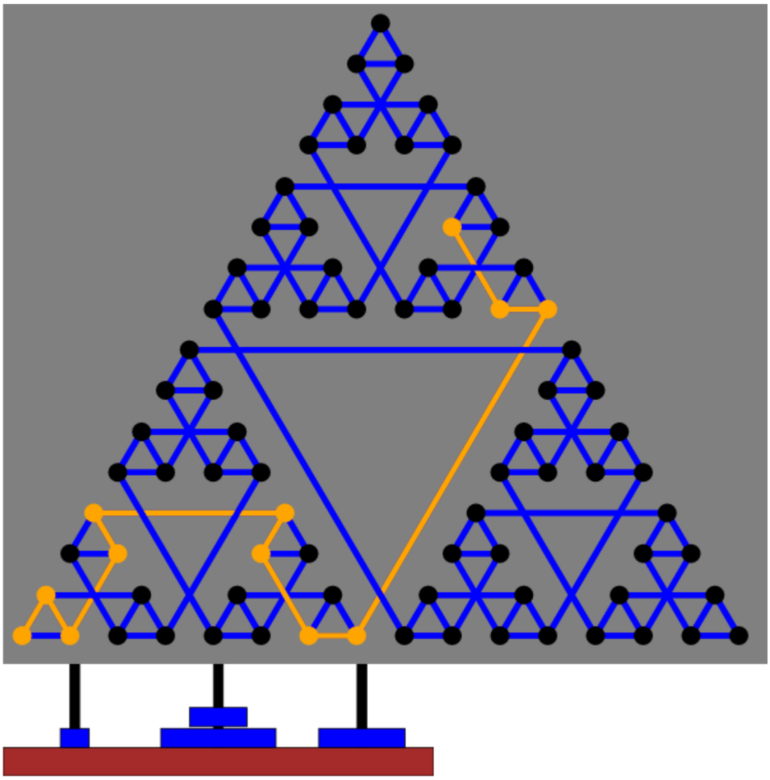

**Fig. 6.** Visualization of solving a Towers of Hanoi problem with 4 disks as path-finding in a maze, using a problem-space graph. The last (uppermost) node on the yellow path represents the game state shown below the graph, with the blue disks scattered among the three pegs.

### 3.4 Basic Methodology for Problem Formulation

In our course, the key to gamification at Tier B is the formulation of a problem. The students in each group choose a global-challenge problem from a list that they themselves, as a class, come up with, and they have to create a serious game out of it.

The formulation process is an important part of the content of the course. Our method of formulation borrows from the work of educational psychologists.

In his introduction to his edited book on learning through solving complex scientific problems, David Jonassen made a distinction between ill-structured problems and well-structured problems, and he outlined a means to transform the former into the latter [13]. We have adapted his methodology to better fit the context of student projects as well as reaching formulations that work with computational interfaces. The steps in our methodology are the following, organized into three phases.

Phase I: Pre-formulation

1. Describing a need
2. Identifying resources

Phase II: Posing

3. Restriction and simplification
4. Designing a state representation
5. Designing a set of operators
6. Listing constraints and desiderata

Phase III: Coding

7. Specifying in code the state representation, operators, constraints, evaluation criteria, and goal criterion.
8. Specifying in code a state visualization method.
9. If appropriate, providing for multiple roles within teams of solvers.

In the pre-formulation phase, students come up with short descriptions of their problems, in terms of needs to be satisfied, and they perform library or internet research to find resources that are relevant to the problem. Typical types of resources are Wikipedia articles, other web pages, TED talks, YouTube videos, journal articles, and books. In a group of 4 or 5 students, with each student responsible for finding and analyzing two resources, each group begins with 8 or 10 resources that provide reference material for their problem formulation.

Posing is the phase in which the vague and complex problem that the student group starts with gets refined into something much more definite and limited. The students determine what variables related to the problem should contribute to each state in the problem's state space and they propose a representational structure for the states, giving particulars for the initial state. They come up with a set of operators, representing actions that players will be able to perform during the turns of the game. These operators will typically modify the values of state variables, and if there is a board-like layout, operators might move tokens for resources or avatars to new locations on the board. Constraints should be identified that will limit what legal moves are; for example, if an operator involves spending game money, a constraint could be that players are not permitted to spend more money than they have. Desiderata typically apply to the goal state; for example, in a climate-change game, the average surface temperature of the earth should be as low as possible.

This posing phase is the most intellectually challenging for most students. A key question for them to answer is "How do we want our players to be thinking about the wicked problem as they play our game?" In answering this, they themselves engage in metacognitive thinking. As they add or remove operators and constraints, they are shaping the problem space of their formulation (see Jackman et al [15]). The problem space can be thought of as the interior of a problem space "box" and then by refining the formulation, obtaining a different box requires thinking outside of the first box. Formulations are the box-like artifacts to be thoughtfully shaped rather than inert givens.

The coding phase involves expressing the initial state, operators, and goal-state criterion as computer source code. We scaffold this process by providing a code template file with commenting and boiler-plate code into which students can put their code snippets that express these elements of their formulation. We also provide the students with software tools to help them get their formulations to comply with the requirements of being well-formed. For example, each operator needs a name, a precondition function, and a state-transformation function.

The formulation should include a means for each game state to be visualized, typically in two ways: as a piece of text that describes the state, and as a graphic, which shows the state using graphical elements such as a chart, map, image, styled text, or combination of these.

Once the problem has been formulated, it can immediately be played as a game using what we call a "client" piece of software. We offer students several different clients, so that they can more easily develop their game first as a text-only game and then add simple graphics, and finally, more complex graphics.

### 3.5 Example Formulation for Climate Change.

One strategy for students to use in formulating a complex problem such as climate change is to break the design into a factual part and a fictional part, such that the factual part represents a scientific model of a key phenomenon underlying the problem, and the fictional part consists of actions that we might wish to be able to take, such as spending large amounts of money to replant forests after deforestation or build solar power stations.

For climate change, the earth's thermal equilibrium based on the black-body radiation formula can serve as the scientific model, and various imaginary government programs can represent the fictional actions that might be taken when working to solve the problem. Both the scientific model and the fictional actions contribute to the design of the operators in the formulation. The actions affect a simulation which then affect changes to variables in game's state representation. Here is the thermal equilibrium equation. While it's not important to understanding the essence of this paper, the fact it can be solved for T and used in a game to predict changes in the earth's temperature in reaction to small changes in the earth's albedo and emissivity as a result of actions such as reforestation or reducing greenhouse gases is significant.

$$(1 - a)\, S\, \pi\, r^2 = 4\, \pi\, r^2\, \varepsilon\, \sigma\, T^4$$

Here $a$ is the earth's albedo (an indicator of reflectiveness), S is the solar constant, r is the earth's radius, $\varepsilon$ is the emissivity of the earth, $\sigma$ is the Stefan-Bolzmann constant, and T is the temperature of the earth in degrees Kelvin.

For purposes of creating a serious game that helps players learn the relationships among human actions (at a large scale) and changes to the average surface temperature of the earth, many plausible but fictional operators can be proposed. Making their effects realistic will require additional research into, say, how changing concentrations of greenhouse gases in the upper atmosphere affect the earth's emissivity, etc. Students who become sufficiently interested in gamification of climate change can deepen their engagement and understanding by playing commercially produced games such as Fate of The World (Red Redemption 2011 [16]).

## 4 Taming Wicked Problems

The second half of our course focuses on the student projects, done in groups of 4 or 5. Each group selects a problem and formulates it as a game. The problem must pass a test of "wickedness" so that the game will be a serious game, and the students will have an ample opportunity to apply the formulation methodology and meet classroom-community obligations to be described in the context of Tier C gamification. The term "wicked problem" is said by Churchman (1967 [17]) to have originated with Rittel, but the definitive explanation of what a wicked problem is came later in the paper by Rittel and Webber [11]. (See also Skaburskis [18].) Their ten characteristics of wicked problems are listed in the next section. Our students are exposed to these but defend the wickedness of their problem using our own modified criteria, which comport better with the classical theory and the aim of serious gamification. By labeling and formalizing wicked problems, we clarify for students what sort of serious games they will be creating, and we encourage and empower them to apply our methodology to them.

### 4.1 Rittel and Webber Criteria

Here are the ten original criteria. (For detailed explanations, see Rittel and Webber, 1973 [11].) (1) There is no definitive formulation of a wicked problem. (2) Wicked problems have no stopping rule. (3) Solutions to wicked problems are not true-or-false, but good-or-bad. (4) There is no immediate and no ultimate test of a solution to a wicked problem. (5) Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial-and-error, every attempt counts significantly. (6) Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan. (7) Every wicked problem is essentially unique. (8). Every wicked problem can be considered to be a symptom of another problem. (9) The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution. (10) The planner has no right to be wrong.

In our course, we place particular importance on criteria 1 and 6, and we add an additional criterion, which we find underlies most of the global-challenge problems in the world today. That is what we call the "opposing-stakeholders" criterion.

### 4.2 Additional Opposing-Stakeholders Criterion

The following additional criterion is one we have found to be not only a characteristic of global-challenge problems, but readily recognizable and helpful in discovering game-theoretic features of the problem: "There are multiple stakeholders in the problem situation, at least two of which have inconsistent objective functions." For example, oil companies have a financial stake in the use of fossil fuels, while many other citizens of the world, including environmental leaders have opposing interests. The existence of opposing stakeholders in a problem explains and underlies the implicit social disagreement in Rittel and Webber criteria 1 and 9 and impacts possible definitions of goal state (related to criteria 2 and 4), value judgments (criterion 3), and permissible operators (criterion 6). Once students have identified opposing stakeholders in their problem, they can consider game-theoretic modeling such as the Prisoners' Dilemma and Nash Equilibria (Kimbrough 2012 [19]).

### 4.3 Typical Development Milestones for Students

The main Tier B activities in the course begin with suggestions of problems through a collaborative class activity. Each student makes a list of problems they think might be wicked. In small groups (but not their project groups) they refine these lists, eliminating duplicates. Then we go through all the groups' lists and put the suggestions on a whiteboard (when in-person, and on a virtual whiteboard when remote teaching). We group the suggestions into categories such as economic issues, world peace issues, social inequality, biodiversity, etc., and we identify a leading problem from each category. Then we obtain student preference information using a computer-based group-formation tool, and we form the project groups. Students who are unhappy with our algorithm's suggested groupings are given an opportunity to adjust their preferences and we iterate on the groupings until we get the best groupings we can. (The details of the grouping algorithm can be found in [20].) The student groups begin their formulation by performing steps 1 and 2 (pre-formulation) and reporting their results to the class. As they begin the posing phase of formulation, they also develop an argument for the wickedness of their problem. They iteratively refine their formulation, overlapping the steps of posing and coding. On the second-to-last day of class, the games are play-tested by the other members of the class, formal evaluations are made, and the teams have one final day to make last-minute changes, and write up their responses to feedback. Additional aspects of this process are described in the next section, dealing with the Tier C gamification.

## 5 Agile Software Development and Tier C

This section explains the Tier C gamification in the course. At this tier, students are role-playing in a gamified version of agile development teams. They not only learn about the agile methodology popular in corporate software development, but they learn to collaborate at a classroom-community level that goes beyond the teams to which they belong. At this level, the students are all supporting the game development and learning success of all their classmates and not only their own groupmates.

### 5.1 Collaboration and Role-Playing on an Agile Development Team

During the first half of the course, students do their assignments either individually or in partnerships. They are getting to know each other as well as gaining background knowledge and skills. Near the end of the first half, we begin to consider what it will mean to collaborate in larger groups. The students do an exercise in which they enumerate the characteristics of ideal teammates --- things like reliability, readiness to share and explain ideas, willingness to listen, etc. This prompts them to consider what they might expect from their teammates and what will be expected of them. After the project groups are formed, they get a lesson on software development methodologies with a critique of the waterfall model and a detailed introduction to agile development with the scrum approach (Sims and Johnson 2011 [21]). Each student team must elect (through any process they like) one person to be the “scrum master” (a standard role in industrial agile development teams) and another to be the “scrum ambassador.” This latter role is not found in industry but serves a very important purpose in our gamified version of agile development teams.

### 5.2 The Scrum Ambassadors

While the scrum master on each team has the responsibility to keep the group aware of and following scrum protocols, helping to arrange group meetings outside of class, and supervising the maintenance of the scrum “backlog” of tasks to complete, the scrum ambassador has primary responsibility for communications between the group and the rest of the class. For example, when the group gives a presentation to the rest of the class about their design progress, the ambassador will introduce the other team members, who will each have a short part of the presentation to give. The ambassador also has the responsibility to collect, transmit, receive, and distribute peer evaluations to and from other groups via those groups’ ambassadors. (This aspect of the ambassador role is depicted in Fig. 7.) In actuality, the ambassadors have often turned out to be de-facto leaders of their groups, even though the scrum masters are charged with managing the group’s adherence to the scrum process.

Thus, as we have adopted the agile methodology, we have adapted the standard industrial structure to match our educational context. The result is a set of rules that offer an understandable and reasonably fair playing context in which the students can readily give their best efforts to their teams and the whole class.

Unlike some gamified learning contexts, we have adjusted the overarching goal of the third-tier game away from competition and towards cooperation. This is consistent with the notion that wicked, global challenge problems are owned by everyone on earth, and it will take global cooperation to actually solve any of them. This is described further in the next subsection.

### 5.3 Who Owns the Games?

In agile development, there is typically a "project owner" that does not include any of the team members. That owner is an external client who has commissioned the team to design and implement a product. While that is an appropriate model for many project-oriented learning courses, especially when real-world external "customers" such as industrial groups, community or government organizations, or other units on campus bring real problems to the class, that is not the model used in the present course which deals with wicked problems rather than problems belonging to specific external customers.

Yet, we have gamified at our third tier in such a manner as to indeed have a "project owner" entity. Our answer to the need for an owner to critique designs is to make the entire class the owner of each of the teams' games and designs. For any team $T_i$ in the class, the entire class, including the members of $T_i$, are the owners of $T_i$'s game. With ownership comes responsibility, and thus each student in the class has responsibility not only for their own team's success, but for the success of all the other teams' designs. Naturally, there is increased importance on a student's own team's game for that student, but they do indeed have explicit obligations to the entire class, in terms of thoughtful peer reviewing and responses to the peer reviews of others. The whole class is the project owner for the game developed by each team.

### 5.4 Iterative Design: Game Ideas, Prototyping, Evaluating, and Repeating

Although the four-week course period is short, the students learn and practice an iterative design methodology (see, e.g., Aslan 2016 [22]). While the duration of a scrum sprint in industry is typically two weeks, in our course it is about 3 days long. The students find time to meet either in person or online between classes to coordinate work on their projects. Over a period of two weeks, they meet five milestones for which they have to submit work; they receive oral feedback from the class after each milestone and written peer feedback after two of these. The last milestone involves a final presentation, a written report, and final code.

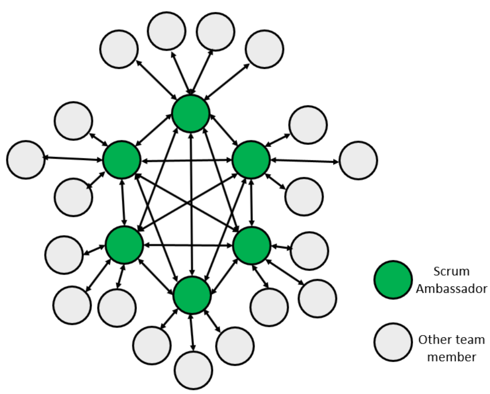


**Fig. 7.** Tier C flow of communication during the formal peer review cycle. Each of the six teams has either 4 or 5 members, one of whom is the Scrum Ambassador charged with collecting and sharing peer review papers or messages.

# 6 Formal Evaluations of Serious Games.

Student's prior conceptions of game design sometimes are tied up with the game mechanics of commercial, action-oriented epics situated in three dimensions. The students are typically not attuned to the criteria specific to serious games. For this reason, the course uses forms for evaluation that prompt students to make judgments about the way that a game has its problem formulated, what the purpose of the game is, and how well it achieves its purpose. These criteria are in line with those proposed by Mitgutsch and Alvarado (2012 [23]). Here we mention four points at which written evaluations of games take place in the course.

## 6.1 Pre-Project Game Reviews

The course's first assignment is to play and perform detailed evaluations of two serious games. One game is pre-chosen for all students to evaluate, and the other gets chosen by each student from a list of options. The reviewing criteria are given to the students on a two-page web-based form that expands as they fill in their responses to particular prompts. This rubric is shown in Appendix A. This exercise helps familiarize students with serious games and their evaluation criteria as well as a variety of game mechanics.

In the earlier offerings of the course, the pre-chosen game was BBC Climate Challenge [24]. Due to the Flash plug-in no longer being supported by browsers, we have changed the pre-chosen game to Climate Quest [25]. Then students evaluate another game, choosing from a list of four other games.

## 6.2 First Written Peer Feedback

The first written peer feedback occurs in the early stages of problem formulation, after pre-formulation has been completed and students are working on posing their problem (class day 10 on the calendar). Evaluations here represent "project owner feedback," and must contain constructive suggestions in addition to any criticism of the formulation so far. Each team is required to respond to this feedback.

## 6.3 Peer Evaluation of Near-Final Games

Although there are opportunities for oral feedback before this, on the second-to-last day of the course, students see presentations of the games and in most cases play-test each game. Then they fill out a detailed evaluation form. Again, this represents "project owner" feedback. Each team must respond to each criticism or suggestion made, and these responses must be mentioned in the final reports, whether or not they resulted in changes to the game code or assets.

### 6.4 Post-Presentation Celebratory Evaluations

Near the end of the last class meeting, after the final group presentations have been made, students get to vote for "Best game for X" in ten categories. The results are shared immediately. (In the past we have used Survey Monkey to manage the voting.) Although this sounds competitive, it does not contribute towards grades and serves only as a way to celebrate the end of the course and show peer appreciation for the efforts of all teams.

## 7 Student Outcomes

### 7.1 A Retrospective

The course has been given once a year for the last six years (starting in 2017). It has consistently obtained high ratings from the students. So far, no student team has ever failed to turn in a final project game. Aside from the features described in this paper to increase student engagement, the small-class format is no-doubt a positive factor in their experience. However, it would be nice to scale up the course to larger classes, in order to serve more students, and that's just one challenge for the future.

This section of the paper discusses what students can learn and what they do learn in the course.

### 7.2 Tier A Results

By playing and reviewing two pre-approved serious games, the students have learned (a) what the characteristics are of some serious games, and (b) specific lessons about subject matter, such as a conceptualization of climate-change moderation in terms of government actions (in the case of BBC Climate Challenge). The review form uses the rubric shown in Appendix A. The following is one student's answer to the review-form prompt "Did you learn anything as a result of playing this game? If so, what?"

> "Through this game, I learned about leaders' difficulty in managing to keep both society happy, and caring for the good of the planet."

There are two main reasons for having students play these games: (a) to familiarize students with the types of subject matter related to global challenges that have been gamified in the past and that they will themselves be gamifying, and (2) to acquaint students with the sort of formulations that must be done to gamify such complex problems. Although the student answer above does not directly show they have learned these lessons, it demonstrates that the student was engaged in the game, and experienced it in roughly the way intended.

### 7.3 Tier B Results

At the end of the course, each group not only presents the final version of their game to the class but turns in their game code and a written report. One required section of the report is a learning retrospective in which the students are asked to mention particular lessons they believe they learned during the process of creating their game. Table 1 shows what topics students mention in these retrospectives and how frequently each has occurred over the 6 offerings of the course. The total number of mentions is 368. The number of students covered here is 147.

**Table 1.** What students say they have learned in the process of creating their games.

| *Topic or Lesson Learned* | *# mentions* |
|---|---|
| Collaboration (importance of, or how to do it) | 62 |
| Software eng., incl. debugging | 48 |
| Python (either as a new programmer, or one just gaining fluency) | 45 |
| A specific wicked problem (e.g., world poverty, info. Security) | 45 |
| What wicked problems are (i.e., Rittel-Weber criteria) | 29 |
| Formulation of problems (methodology) | 28 |
| Graphical User Interface development (in general or with Tkinter) | 26 |
| Game structure and game design (esp. applying the classical theory) | 19 |
| Communication (need for and practice of) | 15 |
| Scrum methodology | 13 |
| GIT usage for code sharing and version control | 10 |
| SOLUZION game-software framework provided by the instructor | 11 |
| Social & cultural lessons (esp. when working in international groups) | 9 |
| Problem solving theory | 8 |

The most commonly mentioned item is the need for and/or practice of collaboration, including coordination of design and programming efforts. Other lessons mentioned include specific skills or technical topics, and references to learning about particular global-challenge problems are often included. Also mentioned are some social lessons, such as how to interact with teammates, or remarks about how much fun it was to work together into the night. The "software engineering" category here covered debugging, (often mentioned), and learning about putting software components together, or how to use object-oriented class definitions.

The games students created were all based on global-challenge problems such as climate change, world poverty, biodiversity loss, unemployment due to automation, war, refugee crises, hunger, homelessness, and others. An example screenshot from one group's game about global and local access to clean water is shown below. It illustrates the map-based approach to game-state visualization. During the game, the colors of individual land areas change to reflect the player's progress in fixing the issue of poor access to clean water.

While the above-mentioned topics learned by students can generally be considered positive outcomes, one group in the first offering wrote that they had experienced frustration at a point in their game creation when development was slow. Some students who are new to programming find it difficult to pick it up within the first half of the course. However, they still find ways to contribute to both their team's game and, through their peer reviewing, to the games of the other teams in the class.

### 7.4 Tier C Results

Many students have mentioned the learning they experienced in terms of collaboration, communication, and/or the scrum development methodology itself in their retrospectives. The relationship between our classroom version of scrum and industrial scrum has been close enough that students have never expressed any reservations to the instructor about our use of it, but often cited its benefits. At the very least, it offers students a means for organizing their development processes. They adopt the use of "backlogs" and "burn-down charts" in their progress reports.

## 8 Related Work

Prior work by others has already been cited in the previous narrative, and Tier A gamification for learning – student learning through the playing of games -- has been so widely studied that existing surveys such as Connelly et al (2021 [4]) cover it quite well. In the following two sections, some additional work is mentioned that relate to gamification in learning at Tiers B and C.

Before those, however, let's admit that our separation of Tier A and Tier B (learning from playing games vs learning from creating games) will not cleanly classify all gamification approaches. For example, the design game created by Kloeckner et al [26] gamifies a general design process whose design products are educational materials (such as an architectural design portfolio structure) but which they say could also be games. In this case, depending on whether the output is a game or not, we could say that their method spans both Tier A and Tier B, or stays in Tier A.

### 8.1 Work Related to Tier B

A good perspective on game-making for learning literature up to 2015 by Earp [27] documents the increasing interest in learning by making games. However, a systematic model for this kind of education has been given by Weitz. In her IxD&A paper and Ph.D. thesis (Weitze [28, 29]), she described a research project in which two populations of students (adults and $7^{th}$ graders) were engaged in creating games for learning. Her "Smiley" model specifies a structure for student groups and peer review that has much in common with the structure described in this paper. Her "small digital games" correspond with our student project wicked-problem gamifications. Her "big game learning design" is analogous to our Tier C, but without our agile-development roles structure. These two references of hers give a thorough description of the theoretical justifications for the learning design. By contrast, our work is focused on the gamification of a course specifically on problem solving and how the tiers of gamification contribute in this context.


**Fig. 8.** Game-state visualization from "World of Water," a game about the problem of access to clean water around the globe. (Image courtesy of T. Bhan, M. Seyer, J. Wang, and J. Zemek).

Mårell-Olsson [30] recently described a sort of hybrid of Tiers B and A in which university students created educational games that were play-tested by upper secondary students, with a university teacher supervising the university students and secondary teachers supervising the secondary school students. She found that the complexity of making this arrangement work could be attributed not only to the four different kinds of participants, but the lack of enough background knowledge about educational gamification, especially on the parts of the secondary school teachers. In the future, building these complex collaborations may become easier as it becomes more commonplace, and secondary teachers themselves become more familiar with the technological tools and practices associated with 21st century skills. Unlike her relatively complex participants structure, our class is relatively homogeneous at least in terms of student academic level, and only university faculty were involved. Also, we used one main form of gamification (via problem-solving theory) rather than the many diverse gamification techniques her population used based on the Octalysis framework.

A set of research recommendations were made in 2021 by Weitze based on a survey of the literature on learning through educational game design [31]. From an initial pool of 700 articles, 17 were identified as relevant to her research questions. The results offer suggestions for researching pedagogical approaches (including learning theories, and collaborative learning processes), learning design frameworks, ways of helping students with educational content formulation, and game design skills, measurement of learning outcomes, design of student and teacher support. The course described in this paper perhaps contributes best towards her item "4.2. Useful learning design frameworks and methods (RQ2)." At least in connection with the educational objective of helping students learn about ways to approach the formulation and solving of global-challenge problems, the course offers a unique approach. This consists of the three tiers of gamification, together with the use of the classical theory of problem solving, and original software that helps in the Tier B task of turning proper formulations into playable games.

### 8.2 Work Related to Tier C

As mentioned above, Weitze's Smiley model includes a "big game" that indicates a degree of gamification at the classroom organizational level, with students in groups. One can ask what sorts of roles and rules might work best at this tier. If there is simply a set of rules, students might follow them and gain benefits, but they may perceive these rules to be "the teacher's rules" or "school is a game" rather than using their imaginations to feel that they are playing "realistic" roles related to real-world situations. Role-playing itself can work to foster learning in fantasy worlds; for example, Barab et al found that students playing Quest Atlantis gained motivation through having a role with importance to a team [32]. Rodriguez et al found that playing fantasy roles while learning initially had a positive effect but the benefits wore off after a few weeks [33]. More realistic roles are helpful in developing leadership skills as found by Oropeza Hernández et al [34]. In our Tier C gamification, the agile development roles are real roles in two senses: there are no avatars as in Quest Atlantis, and students do the same kinds of tasks that industrial software development teams would do. Although the guidelines have been altered for the classroom, students see them more as industrial practice guidelines rather than "school rules" or "teacher rules." This is an important aspect of our Tier C gamification.

## 9 Discussion and Future Work

### 9.1 Prerequisites and Leveling the Playing Field

A pedagogical issue in this course, as in many group project situations involving computer programming, is accommodating differing levels of programming skill. Programming is an important course component, and in the public course description, it's recommended, but not required, that students have prior programming experience. Not requiring it is a result of the institutional program's policy on avoiding prerequisites for students in this program. (If the course is adopted elsewhere, it might be best to require the prior programming experience.) To help students who are beginning programmers not to be intimidated by the advanced skills of a few of the other students, all teams, including those with hot-shot programmers, are required to code to a particular API which both scaffolds and limits what the students can do in such a way that they focus most of their time on the formulation (gamification) of their wicked problem (with modeling and analysis), rather than building fancy-looking user-interface features or fast-action game elements. However, to partially allow fancier programming by students who really want to add cool things to their games, we permit students to create game-state visualizations that involve 2D graphics without animations.

### 9.2 Future Work

As mentioned, one challenge for the methodology is in scaling up the course to more students. Currently, with 25 students divided into 6 groups of 4 to 5 students, the class as project-owner convention in Tier C gamification makes sense. How well would the methodology work if students were not stakeholders in every other team's game, but perhaps in only a subset of them? How could this work in a class of 100 students or an online class of 500? These are some configurations that could be tried.

Another issue, identified by Weitze [31] as a concern of teachers, is assessing student learning during game play. The games that students build in our course are intended to be serious, educational games for their players. Although students are encouraged to include indicators of progress, such as scoring mechanisms or game-play milestones, we do not expect the games to include educational assessment components. The inclusion of built-in, unobtrusive assessment such as that described by Shute [35] is something that could be tried in the future.

In this paper, the evidence of student learning from the gamification comes from the student retrospectives. Future work could be to modify the course and create alternative versions use pre-tests and post-tests as well as alternative gamification conditions (e.g., with and without Tier C) in an attempt to determine, quantitatively, what benefits each form of gamification brings to learning outcomes.

Some of the students who take this course wish that they could use more sophisticated graphics in their games that the current guidelines allow. How to offer richer possibilities without changing the course's emphasis on serious formulations of wicked problems or raising the stress levels of students who are completely new to programming is another challenge in the further development of this course's infrastructure and shaping of the student experience. Then, there is an open question of what would be the best way to facilitate the adoption of this course or something like it by other instructors or institutions. Related to this is what the minimal training for an instructor or teaching team might be that would make it feasible to offer the course in a new setting. One additional aspect of the fancier-technology direction takes note of Earle and Leyva-de la Hiz' study on the potential for using virtual reality and augmented reality in teaching about wicked problems related to environmental sustainability [36]. The emerging research question here is how best could students be empowered at Tier B to create serious VR or AR games about sustainability.

### 9.3 Concluding Remarks

The three tiers of gamification presented here each involve a transformation of an existing pattern of teaching and learning. In Tier A, students play games to learn, instead of reading books, listening to lectures or doing pencil-and-paper exercises. In Tier B, they become the makers of the games (as opposed to, say, merely reading and writing about global problems). As they create games, they continue to play games as they evaluate their own and peers' games. As they design their games, they better master the game content as well as anticipate how others will react to their formulations. In Tier C, they play game-like roles as they pursue their Tier A and Tier B activities. Tier C gives them responsibilities to foster learning by the whole class, and it deepens their feelings of participation as they continue to perform their Tier A and B activities.

The author hopes that the ways that the three Tiers of gamification have been presented here may prove to be useful to other educators and researchers. While gamification of each form (Tier A, B, C) has been used by others in the past, the particular way the tiers are combined here works well in a class of 25 college freshman, and the use of the classical theory of problem-solving to scaffold their formulations of wicked problems as games is unique. It has turned out to be effective in engaging these students in studying and understanding aspects of global-challenge problems.

**Acknowledgments.** The author would like to thank the editors and organizers of the special issue of IxD&A for administering the issue and encouraging authors to submit their experiences and insights about gamification of learning. The author also thanks the 147 students who have participated in offerings over the past six years of the gamified course described here. Finally, the constructive comments of the anonymous reviewers are much appreciated.

## References

1. Brown, A.L.: Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. Journal of the Learning Sciences, 2(2), pp.141--178. (1992) https://doi.org/10.1207/s15327809jls0202_2
2. Reeves, T., Herrington, J., Oliver, R.: Design Research: A socially responsible approach to instructional technology research in higher education. Journal of Computing in Higher Education, 16(2), pp.97--116. (2005) https://doi.org/10.1007/BF02961476
3. Tobias, S., Fletcher, J.: Computer games and instruction. Charlotte, NC: Information Age Publishing. (2011)
4. Connolly, T.M., Boyle, E.A., MacArthur, E., Hainey, T., Boyle, J.M.: A systematic literature review of empirical evidence on computer games and serious games. Computers & Education, 59(2), pp.661-686. (2021) https://doi.org/10.1016/j.compedu.2012.03.004
5. Sailer, M., Homner, L.: The Gamification of learning: a Meta-analysis. Educational Psychology Review 32, pp.77-112. (2020) https://doi.org/10.1007/s10648-019-09498-w
6. Simon, H. A., Newell, A.: Human problem solving: The state of the theory in 1970. American Psychologist, 26: 2, pp.145-159. (1971) https://doi.org/10.1037/h0030806
7. Newell, A., Simon, H.: Human Problem Solving. Englewood Cliffs, N.J.: Prentice-Hall. (1972)
8. Polya, G.: How to Solve It. Princeton University Press. (1945) https://doi.org/10.1515/9781400828678
9. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Reading, MA: Addisson-Wesley. (1984)
10. Simon, H.: The Sciences of the Artificial. Cambridge, MA: MIT Press. (1969)

11. Rittel, H. W. J, Webber, M. M.: Dilemmas in a general theory of planning. Policy Sciences, Vol. 4, pp.155-169 (1973) https://doi.org/10.1007/BF01405730
12. Klemmer, S.: Personal communication. (2017)
13. Jonassen, D. (ed): Learning to Solve Complex Scientific Problems. New York, NY: Routledge (2007)
14. Lucas, E.: L'Arithmétique Amusante. Paris: Gauthier-Villars et Fils (1895) https://doi.org/10.1038/053079c0
15. Jackman, J., Ryan, S., Olafsson, S., Dark, V. J.: Metaproblem spaces and problem structure. (Ch. 11 of Jonassen) (2007)
16. Red Redemption, Ltd.: Fate of The World. (game). Oxford, UK. (2011)
17. Churchman, C.: Free for All Guest Editorial. Management Science.Vol. 14, No. 4. (1967) https://doi.org/10.1287/mnsc.14.4.B141
18. Skaburskis, A.: The Origin of "Wicked Problems", Planning Theory & Practice, 9:2, pp.277-280. (2008). https://doi.org/10.1080/14649350802041654
19. Kimbrough, S. O.: Agents, Games, and Evolution: Strategies at Work and Play. Boca Raton, FL: CRC Press (2012)
20. Tanimoto, S. L.: The Squeaky Wheel algorithm: Automatic grouping of students for collaborative projects. Proc. Workshop on Personalization in e-Learning Environments at Individual and Group Level (PING), Corfu, Greece, pp.79--80. (2007)
21. Sims, C., Johnson, H. L.: The Elements of Scrum. Foster City, CA: Dymaxicon. (2011)
22. Aslan, S.: Digital Educational Games: Methodologies for Development and Software Quality. Ph.D. dissertation, Virginia Polytechnic University, Blackburg, VA. (2016)
23. Mitgutsch, K., Alvarado, N.: Purposeful by design? A serious game design assessment framework. Proceedings of the International Conference on the foundations of digital games. New York: ACM, pp.121-128. (2012) https://doi.org/10.1145/2282338.2282364
24. Red Redemption, Ltd.: BBC Climate Challenge. (2006). https://www.bbc.co.uk/sn/hottopics/climatechange/climate_challenge/
25. Earthgames.org: Climate Quest. (2021)
26. Kloeckner, A. P., Scherer, J. O., Ribeiro, J. L. D.: A game to teach and apply design thinking for innovation. International Journal of Innovation - IJI, São Paulo, 9(3), pp.557-587. (2021, Sept./Dec.). https://doi.org/10.5585/iji.v9i3.20286
27. Earp, J.: Game Making for learning: A systematic review of the research literature. Proc. 8th International Conference of Education, Research and Innovation, ICERI 2015, Seville, Spain, (2015)
28. Weitze, C.L.: Designing for learning and play: The Smiley model as a framework. IxD&A, No.29, pp. 52-75. (2016)
29. Weitze, C. L.: Innovative Pedagogical Processes Involving Educational Technology. Ph.D. Thesis. Aalborg Universitetsforlag (2016)
30. Mårell-Olsson, E.: Using gamification as an online teaching strategy to develop students' 21st century skills. IxD&A No. 47, pp.69-93. (2020-21) https://doi.org/10.55612/s-5002-047-004
31. Weitze, C.L.: Recommendations for learning through educational game design: A systematic literature review. Proc.15th European Conference on Games Based Learning (ECGBL). (2021). DOI:10.34190/GBL.21.035.
32. Barab, S.A., Gresalfi, M., Ingram-Goble, A.: Transformational play using games to position person, content, and context. Educational Researcher, 39(7), pp.525-536. (2010) https://doi.org/10.3102/0013189X10386593
33. Rodrigues, L., Pereira, F. D., Toda, A. M., Palomino, P. T., Pessoa, M., Galvao Carvalho, L. S., Fernandes, D., Oliveiro, E. H. T., Cristea, A. I., Isotani, S.: Gamification suffers from the novelty effect but benefits from the familiarization effect: Findings from a longitudinal study.

Int'l. J. Technol. in Higher Education. Vol. 19, No. 13. (2022). https://doi.org/10.1186/s41239-021-00314-6. https://doi.org/10.1186/s41239-021-00314-6
34. Oropeza Hernández, M. I., Muñoz Lezama, R., Madero Gómez, S.: Work-in-Progress: The Road to learning, using gamification. 2021 IEEE Global Engineering Education Conference (EDUCON), pp.1393-1397. (2021) https://doi.org/10.1109/EDUCON46332.2021.9453892
35. Shute, V.J.: Stealth assessment in computer-based games to support learning. Computer games and instruction, Vol. 55, No. 2, pp.503--524. (2011)
36. Earle, A., Leyva-de la Hiz, D. I.: The wicked problem of teaching about wicked problems: Design thinking and emerging technologies in sustainability education. Management Learning. December (2020). https://doi.org/10.1177/1350507620974857

## Appendix A. The Serious Game Reviewing Rubric for Tier A.

Reviewed by: (Your name goes here)

Name of the game: (The name of the game goes here)

Type of game: (Describe the type of game here. E.g., card game, board game, branching story, turn-based strategy game, computer simulation-based game)

Authors of the game: (Who developed the game -- authors and/or company?)

Release date and version: (When was the game released, and which version are you reviewing?)

Where available: (Where is the game available? If online, give its URL)

How many can play? (How many players -- call this n -- can play together in the game? Does it work better for certain values of n?)

Appropriate for ages... (For what age groups is the game appropriate? Pre-school? Primary school? Middle-school? High-school? College students? Adults in general? Is it intended for professionals?)

"Serious" game? (To what extent does the game deal with a challenging problem?)

Modeled aspect: (What aspect of the problem or the phenomenon is modeled?)

Player roles: (Does each player take on a role in the game? If so, what role? What other roles are available?)

Knowledge needed to play: (What knowledge is expected of a player?)

What a player can learn: (What can a player learn in the game?)

Main game mechanics: (What are the important game mechanics involved? e.g., turn-taking, use of cards -- real or simulated, navigating a maze, using money or other currency, health points, game points, levels, adversaries, randomness, etc.)

Variety of possible experiences: (What variety of experiences is available over the course of multiple sessions?)

Production values: (Comment on the production values or software complexity)

Your own try-out: (Describe your experience as you played this game)

Engagement: (How engaging did you find the game? How much time did it take you to play one game? Did you play this game multiple times? If so, roughly how many times?)

Learning from playing: (Did you learn anything as a result of playing this game? If so, what? Was it worth your time?)

Strengths and weaknesses: (In your opinion, what are the main strengths and weaknesses of the game?)

Any other references: (If you used any other references than the game itself in answering the review questions, cite those here, with each URL or author and publisher)