Computational Thinking in a Game Design Course

Amber Settle
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604
(312) 362-5324
asettle@cdm.depaul.edu

ABSTRACT
As a part of an NSF-funded project to enhance computational thinking in undergraduate general education courses, activities and assessments were developed for a game design course taught at DePaul University. The focus of the course is on game analysis and design, but the course textbook uses an approach that is heavily grounded in computational thinking principles. We describe the course activities and assignments and discuss an initial assessment of those materials. Our results show that there is a gap in difficulty between several of the activities and indicate that the materials developed help students to better learn the computational thinking concepts in the course.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education

General Terms
Measurement, Design.

Keywords
Computational thinking, game design, evaluation.

1. INTRODUCTION
Since the beginning of the most recent decline in interest in information technology and computer science began in 2001, there has been a focus on finding ways to revitalize the way that computing is taught. One of the approaches taken during the past decade has been to use games as a medium for teaching core computing skills and ideas. Games have been used in many undergraduate classes, particularly in the introductory programming sequence [2, 5, 6, 10]. Games have also shown potential for helping K-12 teachers and students learn computing [1, 4, 17], and, under the right circumstances, for attracting underrepresented populations such as women [18].

In parallel with the development of games as a way to teach computing concepts, a number of information technology educators have approached the revitalization of the computing curriculum using an idea known as computational thinking. The term was originally coined in a 2006 CACM article written by Jeannette Wing [21] and has since been defined as “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” [22]. Spurred by the creation of the NSF CPATH (CISE Pathways to Revitalized Undergraduate Computing Education) Program, researchers have considered how to revamp undergraduate courses and entire curricula to place a greater emphasis on computational thinking concepts. A comprehensive list of the projects funded under the CPATH program can be found on the CPATH Community Website [7]. A number of CPATH projects have focused on finding ways to enhance computational thinking in courses outside of the standard information technology curriculum [8, 9, 11, 13, 14, 19, 20]. Our project in particular involved 18 faculty from across DePaul University, including the College of Computing and Digital Media (CDM) and the College of Liberal Arts and Sciences. Faculty from 8 different departments or schools developed computational thinking materials for 19 courses in the general education program at DePaul. The courses represented five different categories of the Liberal Studies Program: Scientific Inquiry, Arts and Literature, Understanding the Past, the First Year Program, and the Honors Program. Our previous work has discussed the project as a whole [14], and here we consider one of the courses in the project in more detail.

One of the first courses for which computational thinking materials were developed was GAM 224: Introduction to Game Design. The course is approved for credit in the Arts and Literature domain of the DePaul Liberal Studies Program, and we provide more details about the class in the next section. The course is a combination of the two approaches to revitalizing computing education discussed in this section. The focus is on game analysis and design, so that the course does not require students to implement computer games. However, the course textbook [15] uses an approach to game design that is heavily grounded in computational thinking principles, making it an excellent choice for our project. Some researchers have suggested that finding ways to show students that computing is more than “just programming” would be of benefit to the discipline and have argued that students should experience the foundations of computational thinking before their first programming experience [12]. Introduction to Game Design is precisely such a course, allowing students without any background in information technology to see computing concepts in a context that is of great interest to them. In the remainder of the paper we provide more information about the game design course, describe the computational thinking activities and assignments developed for the course, and discuss an initial assessment of those materials.

2. THE COURSE MATERIALS
As a course that combines game design and a strong computational thinking approach to game analysis, Introduction
to Game Design was a natural first course for our project. The materials produced for the game design course were created as a proof of concept for the project. It should be noted that the course materials described are expansions of ideas and examples presented in the course textbook [15].

2.1 The Game Design Course

Introduction to Game Design was first taught at CDM in the Spring 2004. It was designed to be one of the first courses taken by undergraduates in the then-new Computer Game Development program. It takes a writing-intensive approach to the analysis of games and was approved for the Arts and Literature Domain of the Liberal Studies Program in 2005. The Liberal Studies Program at DePaul is the common undergraduate curriculum, and the program is designed to develop students’ writing ability, computational and technological proficiencies, and critical and creative thinking. Introduction to Game Design requires students to study the principles of game design and use them to both analyze existing games and to develop their own original game ideas. Students do not implement computer games, focusing instead on creating board or card games or designs for computer games. Course topics, readings, and a week-by-week schedule for the class are provided in a previous article [16]. Teaching students to recognize computational thinking patterns through games is an approach that others have employed [3]. The difference here is that the course does not involve the implementation of games, instead analyzing games as media.

The curriculum at CDM has been significantly modified since Introduction to Game Design was first created. The course is no longer required of Computer Game Development students, having been replaced by a course for majors only. However, the class continues to count as an elective in the Animation program and can be taken as a Liberal Studies course by any CDM student other than gaming majors. At its peak there were 5 sections of the course offered every quarter, but the class continues to be popular with a total of 6 sections offered during the 2010-2011 academic year.

There were two sets of activities and assessments created for the game design course. The first focused on various types of game rules and the relationship between them, and the second required students to understand and classify feedback loops in games.

2.2 Representation of Game Rules

Game rules can be categorized into three types: constitutive, operational, and implicit. Operational rules are the guidelines players require in order to play, such as the rules printed on the box of a board game. Constitutive rules are the underlying logical and mathematical structures in the game. Implicit rules are the unwritten rules of the game, such as rules about decorum. Important here are the first two types of rules: constitutive and operational. Two games are considered to be the same if there is a one-to-one mapping between the constitutive rules of the two games, so that if you can find a winning strategy in one game you can use the mapping to find a winning strategy in the other game. At the same time, the operational rules for two structurally identical games can vary significantly. While the operational rules are what make a game enjoyable to play, the constitutive rules are the ones that more experienced players use when they find winning strategies. Abstracting game rules in different ways is an example of computational thinking; it allows students to see the relationship between different abstractions of rules, the modeling of game behavior, and the underlying structure of a game. The computational thinking learning goal for the rules activity can be described as follows: Students will be able to abstract the operational rules of a simple board game to find the underlying constitutive rules for the game and use the constitutive rules to comment on strategies that may exist for the game.

In the first game-rules activity, students consider the well-known rules of Tic-Tac-Toe as well as the rules of 3-to-15 described as follows:

1. Two players alternate turns.
2. On your turn, pick a number from 1 to 9. You may not pick a number that has already been picked by either player.
3. The first person to obtain a set of exactly 3 numbers that sum to 15 wins the game.
4. If all numbers between 1 and 9 have been chosen and no player has a subset that sums to 15, the game ends in a draw.

In the class session, the students are first asked to play 5 minutes of Tic-Tac-Toe. The 3-to-15 game is then described and students play the game for 5 minutes. After the time elapses a class discussion focused on the following questions is conducted:

1. What strategies exist for Tic-Tac-Toe? Include both strategies for winning and for preventing the other player from winning.
2. What strategies exist for 3-to-15? Include both strategies for winning and for preventing the other player from winning.
3. Are the two games the same? Why or why not?

Students are then shown the following grid which provides a mapping from Tic-Tac-Toe to 3-to-15 by indicating what number needs to be chosen for a particular position on the grid:

| Table 1: A mapping of Tic-Tac-Toe and 3-to-15 moves |
|---|---|---|
| 2 | 9 | 4 |
| 7 | 5 | 3 |
| 6 | 1 | 8 |

The equivalence between the two games is usually a surprise to students. The exercise is concluded by discussing the following additional questions:

4. Translate a strategy for Tic-Tac-Toe into a strategy for 3-to-15. For example, describe a blocking strategy for 3-to-15 derived from a blocking strategy for Tic-Tac-Toe.
5. Which game is easier to play using its operational rules? Why?

The goal of the second activity is to find the set of constitutive rules for the board game Chutes and Ladders. The students are reminded of the rules, and most importantly, the board for Chutes and Ladders. They are then asked a series of questions including how to represent the game without using a board, what rules are necessary to implement the spinner, movement on the board, chutes, ladders, and the winning condition for the game, and the implications for understanding strategies using the new representation. The full list of questions can be found in our earlier work [14]. After completing the two class activities, the students are given an assessment either as a homework assignment or on a quiz that asks them to develop a set of
constitutive rules for the board game Candyland. The details can be found in the previous article [14].

2.3 Classifying Feedback Loops

While discrete models can be used to understand simple games like Tic-Tac-Toe, complex games such as computer games are better modeled as cybernetic systems. In a cybernetic system, the encouragement of movement in a certain direction is called a positive feedback loop. A positive feedback loop has the effect of creating significant and increasing growth or decline in some aspect of the system. On the other hand, the discouragement of movement in a certain direction is called a negative feedback loop, and its effect is to stabilize a system or maintain equilibrium. To create a game that is balanced, students must understand how the alteration of the game state through positive and negative feedback loops affects the players’ perception of the fairness and competitiveness of the game. It requires students to model and understand the internal state of a game system, the mechanisms detecting the game state, and the mechanisms altering the game state. The computational thinking learning goal for the feedback-loops activity can be described as follows: Students will be able to describe what a negative or positive feedback loop is, how such a feedback loop can be incorporated into an existing game, and how that feedback loop impacts the experience of both winning and losing players in the game.

2.3.1 Classifying and Describing Feedback Loops

A cybernetic system is one in which the behavior of the system is controlled by a negative feedback loop, a positive feedback loop, or in some cases, several of each. A negative feedback loop acts to move the state of the system in the direction of its previous state. A positive feedback loop acts to move the state of the system in the direction it is currently moving [15]. Students will consider game rules found in common board games and be asked the following questions for each rule:

1. Classify each rule as a positive feedback loop, negative feedback loop, or neither.
2. If positive, indicate what direction the game state is moving in and how the rule aids the movement of the state of the system.
3. If negative, indicate what state the rule is trying to preserve and how the rule keeps the game in that state.
4. If neither, indicate why.
5. If negative or positive, describe the impact that the rule has on a player who is winning and the impact that the rule has on a player who is losing. How would each type of player perceive the rule?

Game rules that are considered in this exercise are:

1. You must reach the final/winning square (square 100) in Chutes and Ladders with an exact roll. For example, if you are on square 96 and you roll a 6, you do not advance but instead lose your turn.
2. In Checkers, if a piece reaches the far end of the board, then it becomes a “king”. A king is allowed to move and jump diagonally backward and forward, unlike ordinary pieces which may only move and jump diagonally forward.
3. In Candyland, penalty spaces on the board cause players to remain stuck in a certain position until they draw a specific card that frees them from the space.
4. Players landing on properties owned by another player in Monopoly must pay rent. A player who owns an entire color group of properties may charge double the rent for any unimproved property in that color group.

After the initial activity, students are led through an in-class activity that asks them to describe feedback loops. They are asked to pick one example of a positive feedback loop and one example of a negative feedback loop in a computer game they have played. For each of the positive and negative feedback loop examples they describe, they are required to answer the following questions:

1. Describe the direction that the game state is moving when the positive feedback loop begins, and how the positive feedback loop enhances that game state.
2. Describe the state that the situation or rule is trying to preserve, and how the negative feedback loop causes the game state to be stabilized.
3. Describe how the positive feedback loop enhances the game play. Be specific about how the positive feedback loop modifies game play, paying special attention to its impact on both the winning and losing players.
4. Describe how the negative feedback loop enhances the game play. Be specific about how the negative feedback loop modifies game play, paying special attention to its impact on both the winning and losing players.

2.3.2 Assessment

After the two in-class activities, the students complete an assessment as a homework assignment or quiz question. In the assessment the students consider an existing board, card, or computer game. They are asked to modify the game to include a positive or negative feedback loop not already present in the game. In doing so, they are asked the following questions:

- Precisely where in the game (under what circumstances and when) are you introducing the negative or positive feedback loop?
- How does the negative or positive feedback loop work, and what game state it is either enhancing (in the case of a positive feedback loop) or stabilizing (in the case of a negative feedback loop)?
- How does your change to the game make it more fun? What balance issues is the change to the game addressing? How will the change to the game affect winning and losing players? Do you anticipate any potential problems that your change could introduce?

3. ASSESSMENT

In the Fall 2009 we conducted an initial evaluation of the materials developed for Introduction to Game Design. There were four sections of the class taught in the Fall 2009, and two of those instructors agreed to participate in the assessment.

3.1 The Fully-implemented Assessment

The section in which the assessment described in Section 2.3 was fully implemented was taught once a week during the evening for 3 hours a session. The instructor for the course was unable to teach one week and the author taught the class in place of the instructor. The class session consisted of a lecture about cybernetics in games, a warm-up activity and discussion of the solution to the activity, then a class exercise and discussion of some solutions to the class exercise. Finally the students were
given an assignment to be handed in the following week that built on the lecture and in-class activities.

The warm-up activity is the first one described in Section 2.3.1. The activity was completed as a pair exercise, and 25 students completed the activity. Twenty two students agreed to have their submissions used for the study, resulting in eleven submissions to be analyzed. The submissions were analyzed in three areas: 1. Did the students provide the correct classification of the game element? 2. Of what quality was the explanation the students provided? 3. If applicable (i.e. the game element was either a positive or negative feedback loop), what characteristics did the answer the students provided for the impact of the game element have? The table below summarizes the scores earned on the warm-up activity:

The game state and enhancement but one is partially incorrect, 4 = includes both the game state and enhancement and is nearly correct, 5 = includes both the game state and enhancement and is completely correct. The rubric used for the game play enhancement was the following: 1 = none, 2 = incorrect or irrelevant to the particular game chosen, 3 = relevant to only one player in the chosen game, 4 = relevant to both players in the chosen game, 5 = highly relevant to both players in the chosen game and discusses player perception. The following table gives the summary statistics for the class exercise:

Table 3: Scores earned on the individual exercise

<table>
<thead>
<tr>
<th>Q</th>
<th>Correct</th>
<th>Explanation</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Avg</td>
<td>Median</td>
</tr>
<tr>
<td>1</td>
<td>100%</td>
<td>2.63</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>2.82</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>90.90%</td>
<td>3.54</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>100%</td>
<td>2.73</td>
<td>2</td>
</tr>
</tbody>
</table>

It is a positive sign that nearly all of the students were able to correctly classify the type of feedback loop each game element represented. This seems to indicate that the lecture enabled students to recognize feedback loops effectively. It is slightly less encouraging that the explanations provided by students were so poor. This may indicate that after a single activity the students did not have a strong enough grasp on the concept of feedback loops to be able to effectively explain their classification. On the other hand, many students were able to describe the impact that feedback loops would have on at least one player and in one case from the perspectives of both players.

Immediately after the warm-up activity, the class exercise described in Section 2.3.1 was conducted. The games the students considered were: Bridge, Checkers, Chess, Monopoly, Poker, and Risk. The exercise was completed as an individual in-class activity, and 25 students completed the activity. Twenty one students agreed to have their submissions used for the study. The students were allowed to choose from the following games and did so as indicated in parentheses, i.e. (number for positive feedback, number for negative feedback): Checkers (3, 2), Chess (9, 3), Monopoly (2, 7), Poker (5, 4), Risk(1, 2), Other (1, 1), and None (0, 2). The students’ submissions were analyzed in three areas: 1. Did the student correctly identify a feedback loop of the appropriate type? 2. Did the description the student provided for the feedback loop accurately describe both the game state and the enhancement? 3. Did the description of how game play is enhanced include elements relevant to both players? The rubric used for the description was the following: 1 = none, 2 = incorrect or missing the game state and/or enhancement, 3 = includes both the game state and enhancement but one is partially incorrect, 4 = includes both the game state and enhancement and is nearly correct, 5 = includes both the game state and enhancement and is completely correct. The rubric used for the feedback loop was the following: 1 = none, 2 = incorrect or irrelevant to the particular game chosen, 3 = relevant to only one player in the chosen game, 4 = relevant to both players in the chosen game, 5 = highly relevant to both players in the chosen game and discusses player perception.

Table 3: Scores earned on the individual exercise

<table>
<thead>
<tr>
<th>Correct</th>
<th>Positive loop</th>
<th>Negative loop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg</td>
<td>Median</td>
</tr>
<tr>
<td>Describe</td>
<td>2.95</td>
<td>3</td>
</tr>
<tr>
<td>Enhance</td>
<td>2.38</td>
<td>2</td>
</tr>
</tbody>
</table>

The individual activity showed a significant decrease in performance from the pair activity. Less than 60% of the students were able to identify a positive feedback loop in their chosen game and only 22% of students correctly identified a negative feedback loop in their chosen game. A vast majority of descriptions provided were either partially or completely incorrect. The decrease in scores remains if individual student performance on the pair activity and class exercise are compared. The average decrease in scores by student was 56.58%, although interestingly the average decrease in scores for the explanation was only 6.32%. These results seem to indicate that the students’ weak explanations on the pair activity were due to a lack of understanding of the nuances of feedback loops, particularly for negative feedback loops. Identifying feedback loops rather than simply classifying them is a harder problem, and the students did not have sufficient knowledge to do so consistently after the lecture and pair activity. It appears from these results that the gap in difficulty between the first activity and the second exercise may too large.

The assignment due the following week is the one described in Section 2.3.2. One additional question was added to the ones given in that section: “Describe what game you have chosen in sufficient detail so that someone unfamiliar with the game understands the premise and basic game play.” There were 24 students who submitted the assignment, and 16 of the students agreed to have their submissions included in the study. Of the 16, 3 had not participated in either in-class activity the previous week. The students were allowed to choose from a list of games, and the student selections were as follows: Advance Wars: 1, Canabalt: 3, Frantic: 3, Katamari Damacy: 2, Left 4 Dead: 7. Seven students chose to add negative feedback loops and the remaining 9 added positive feedback loops.

Three aspects of the submissions were considered: 1. How well did the student describe how the proposed feedback loop works? 2. Was the student’s description of the alteration of the game state clear and detailed? 3. Was the effect of the change on the players described clearly? The rubric used for the feedback loop was: 1 = none, 2 = missing crucial details, 3 = adequate description, 4 = good description, 5 = completely clear description. The rubric for the alteration of the game state was: 1 = none, 2 = incorrect or
missing crucial elements, 3 = adequate but missing some details, 4 = good but missing at least one element, 5 = excellent. And the rubric for the effect on players was: 1 = none, 2 = incorrect or irrelevant, 3 = limited relevance, 4 = relevant to all players, 5 = highly relevant to all players. The table below provides summary statistics for the assignment:

Table 4: Scores earned on the assignment

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Avg</th>
<th>Median</th>
<th>St dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>How it works</td>
<td>3.68</td>
<td>4</td>
<td>1.19</td>
</tr>
<tr>
<td>Game state</td>
<td>3.31</td>
<td>3</td>
<td>1.08</td>
</tr>
<tr>
<td>Effects on players</td>
<td>3.44</td>
<td>3.5</td>
<td>0.63</td>
</tr>
</tbody>
</table>

These results show a distinct improvement over the similar in-class exercise. Considering only those students who participated in the class session, there was an average 3.78% improvement in the description of the game state and effect on the players over the warm-up activity and an average 18.77% improvement over the individual in-class exercise. There are several possible explanations for this improvement. It may be that the students were more familiar with the self-selected games, improving their ability to identify feedback loops. The additional time to work on the assignment may have also helped them. Finally, it may be that the discussion following the in-class activities improved their knowledge of feedback loops. There is some evidence supporting the latter assertion. The table below gives assignment statistics for the game state and impact on the players for two different student populations:

Table 5: Game state and player impact scores by population

<table>
<thead>
<tr>
<th>Population</th>
<th>Avg</th>
<th>Median</th>
<th>St dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>In class</td>
<td>70.77%</td>
<td>70%</td>
<td>14.41</td>
</tr>
<tr>
<td>Not in class</td>
<td>53.33%</td>
<td>50%</td>
<td>15.28</td>
</tr>
</tbody>
</table>

Students who were not present in class had access to the same textbook materials, lecture notes, and activity descriptions as the rest, but were not present for the discussion of the activities. It appears that the discussion of the individual activity improved student understanding of feedback loops.

3.2 The Isolated Assessment

The section in which an isolated assessment was conducted was taught twice a week during the day for 1 ½ hours a session. The course did not directly involve the project researchers. However, the instructor for the course provided his materials when the author was first assigned to teach the course, so that his approach to the course is compatible with the computational thinking materials. In this section of the course the instructor led a discussion of feedback loops in the game Crazy Eights as a regular part of the curriculum.

In a class session after the discussion of feedback loops the students were given the assessment described in Section 2.3.2 on a quiz. One additional question was added to the ones given in that section: “Describe the game you have chosen in sufficient detail so that someone unfamiliar with the game understands the premise and basic gameplay.” The last question was also slightly modified to state: “What are the likely gameplay impacts of your change? Would it make the game more fun? Would it change the balance of the game? Would it effect (sic) winning and losing players differently? Do you anticipate any potential problems that your change could introduce?” No restrictions were given on the games the students could consider when answering the question. Twenty students who completed the quiz agreed to participate in the study. The students chose 20 different games, including card games like Go Fish, board games like Chess and Risk, and computer games like Resident Evil and World of Warcraft.

The same three aspects of the submissions were considered as for the assignment given in the evening section of the course, and the rubric used for evaluation was the same. The table below provides summary statistics for the quiz question:

Table 6: Scores earned on the quiz question

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Avg</th>
<th>Median</th>
<th>St dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>How it works</td>
<td>3.36</td>
<td>3</td>
<td>1.43</td>
</tr>
<tr>
<td>Game state</td>
<td>2.54</td>
<td>2</td>
<td>1.01</td>
</tr>
<tr>
<td>Effects on players</td>
<td>3.09</td>
<td>3</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Despite the fact that both the quiz question in the day section and the assignment in the evening section were identical in content, one needs to be cautious in comparing the two assessments. Students were given only 20 minutes to complete the quiz question whereas the assignment could be completed at home under no significant time constraints. On the other hand students were given a restricted list of games to choose from on the assignment, but students taking the quiz were allowed to choose the game for the question arbitrarily.

What is clear from the statistics in Tables 4 and 6 that the students did better on the assignment than on the quiz question. Particularly striking is the difference between the scores on the game state. While the difference between the description of the feedback loops are not notably different between sections, the difference become more striking when one considers the number of students in each section who were unable to correctly identify a positive or negative feedback loop. There were no students who incorrectly identified a loop on the assignment; however, there were 7 out of 22 students taking the quiz who were unable to correctly identify either a positive or a negative feedback loop. It would appear that the sequencing of the computational thinking activities is more effective at teaching students to identify feedback loops and to describe the game state that the feedback loops are altering.

4. CONCLUSION AND FUTURE WORK

Here we have described several of the computational thinking approaches taken in a writing-intensive game design course. The game design class was one of 19 general education courses modified in an NSF-funded project. We described several course activities and assessments and provided an overview of an initial assessment of the computational thinking materials developed for the class. The assessment highlighted the gap in difficulty between the first and second activities for feedback loops. It also gave some indication that the feedback-loop materials help students to better classify those game structures and understand how the loops modify the game state.

One limitation of the assessment of the course materials is the narrow focus of the evaluation. An interesting question that some researchers have considered is how the computational thinking...
ideas that students learn in a given context, for example when developing games, can be applied when the students encounter another situation where similar patterns can be found, such as in scientific simulations [3]. An extension of this work is to develop materials that ask students who have seen mappings between sets of game rules or have classified feedback loops in games to apply those concepts to either another area of game design or to another discipline. Understanding how well computational thinking concepts can be transferred to a different context is an important direction for future work.

5. ACKNOWLEDGMENTS
This work was supported by the National Science Foundation under Grant No. 0829671. I am deeply grateful to Ljubomir Perković for his crucial role in the project. Thanks also to Robin Burke and Jose Zagal for the opportunity to assess the materials developed for the gaming course and for their feedback. Finally, thanks to André Berthiaume for his comments on an earlier draft of this paper.

6. REFERENCES