# Designing Programming Games for Diversity to Teach CS1

Jecton Tocho Anyango[1][0000−0002−7295−2137] and Hussein Suleman[2][]

[1] University of Cape Town, Private Bag X3, Rondebosch 7701, Cape Town, South Africa
[2] University of Cape Town, Private Bag X3, Rondebosch 7701, Cape Town, South Africa

**Abstract.** Diverse learners from different backgrounds present both significant instructional design challenges and opportunities. Particularly in programming, most serious games that have been created to aid lecturers lack support for diversity. As a result, domain experts who may wish to adopt a Game Based Learning (GBL) approach lack diverse games that are relevant to their local contexts. This paper reports on the design considerations necessary to create diverse programming games for teaching recursion to different novice students. Interviews were conducted with 17 introductory programming (CS1) lecturers from Kenya and South Africa. This was followed by qualitative thematic content analysis. Findings were reviewed by a game expert to validate them from a games design perspective. Results suggest that student background, gender, and culture as well as other factors such as local context, game attributes, pedagogy and practical teaching aspects are core to creating diverse programming games targeting different learners in this generation.

**Keywords:** Diversity · Games · Recursion.

## 1 Introduction

The debate on diversity in Computer Science Education (CSE) is not new. The early work by Camp [11] brought to the attention of the community the declining number of female graduates in Computer Science (CS), pointing out the urgent need for gender inclusivity. Further, the community was called upon to come up with innovative ways to attract and retain students, particularly female, in CS to address the shrinking pipeline. Reiterating low enrollment rates and the lack of diversity in Computer Science education, Bruckman et al. called for changes across CS pipeline [10]. In South Africa, some works have recognized the diverse nature of most classrooms as demonstrated by efforts aimed at preparing lecturers to cope with and handle students from culturally and economically divergent backgrounds [5, 41].

In this paper, diversity refers to differences among students due to race/ gender as well as diversity in (i) motivations, learning needs, educational level and (ii) academic/ skills as well as cultural and economic backgrounds [33, 47]. Lecturers of introductory programming (CS1) encounter diverse multitudes of students [33, 47], bringing with it many challenges [33]. Guzdial and Soloway reiterate that most of these students - the Nintendo or MTV generation - are not easy to motivate using invisible and abstract traditional teaching approaches[19]. Consequently, using old teaching techniques may not be beneficial when engaging with them in the classroom. Probably, new approaches such as digital GBL could be worthwhile. To this extent, some works have suggested the use of games to motivate this generation of students who apparently like engaging with multimedia and animations [19] as well as games [30].

For instance, in South Africa, given the Apartheid system and the diversity of this generation of students entering universities, a pedagogical design for teaching novice programmers CS1 within the local context has been proposed [13]. One of the design principles involves students using construction tools such as puzzles and games while interacting in small groups and speaking in their local languages - mainly Zulu. The results seem promising [13].

Although the motivational and engagement promise of games in learning has been reported in the past [6, 28, 36, 37], designing and developing playable games remains difficult, time consuming and expensive [18, 48]. Moreover, most of the material describing games for teaching computing in higher education does not describe how these games have been developed [8]. Additionally, there seems to be no consideration for diversity in design. This was echoed in a report by a special working group on game development for Computer Science Education (CSE) that noted the significant design challenges as well as opportunities presented by different learners given varied gaming backgrounds [24]. If the motivational need of the 21st century learner is to be met through Game Based Learning (GBL) approach then design need to cater to diverse students, especially in CS1 courses.

Given these findings, coupled with the well known difficulty of the recursion topic among students [4, 29, 32, 31, 42], the purpose of this study is to identify needs of particular groups of learners with an attempt to understand how programming games that appeal to diverse learners could be designed. It answers the following three questions:

1. What teaching practices are used by programming lecturers in Kenya and South Africa when teaching undergraduate novice students the recursion topic?
2. What teaching examples/ analogies are suitable for designing games to teach the recursion topic to diverse novice students?
3. What design considerations are useful when developing a game authoring tool to create games that can teach diverse novice students the recursion topic?

By answering these questions, the novelty and hence contributions of this paper are as follows:

- It is a unique study on teaching CS in developing countries that could be extended to others.
- It highlights factors affecting teaching the recursion topic via games and suggests practical ways to address them as well as suitable teaching analogies.
- It proposes design principles to guide the creation of programming games to teach diverse learners.

The rest of the paper is organised as follows: Section two is a summary of related work. Section three describes the methodology adopted. Results and discussion are presented in section four, followed by limitations in section five. Section six is conclusions and future work.

## 2 Related Work

In a recent study, when White, Asian, Hispanic/ Latin, and African American/ Black students were asked whether they had an interest in video games, their selections for “A lot” were: 52, 48, 27, and 38% respectively [22]. Consequently, a recommendation was made that examples relevant to different groups, including females, and under represented minorities be included in the introductory CS curriculum [22]. In a contrary study, the Lightbot game was used to introduce functions and recursion to students in two post-secondary classrooms from different countries [16]. These were Peru and United States of America (USA) that had different cultural contexts and curricular. In USA, the study was conducted in Berea College (BC), one of the most racially diverse private arts colleges. Results suggested that the game could offer an effective and engaging introduction to the programming concepts regardless of cultural context or course contexts [16]. Given the mixed results from such early works, it would therefore appear that still little is known about how best programming games for diversity could be designed. The work reported in this paper builds on current literature on the topic, but from a different context. It leans on unique experiences drawn from universities from two African countries (Kenya-low-income and South Africa-middle-income).

Advocacy for consideration of diversity, equity, and inclusion in CS curriculum by having content with examples relevant to diverse groups such as female and under represented communities has been on the rise [22, 34]. The work by Collain and Trytten [15] investigated diversity by understanding home and pre-college computer experiences of students and found out that (i) fathers preferentially taught their sons computing and (ii) there are different pathways students use to join CS or Computer Engineering (CE). Some authors have also argued that well-designed culturally responsive computing can make classrooms more equitable and inclusive to a diverse student population [40]. In the case of South Africa, there is need to design teaching and learning strategies that benefit diverse learners in a regular classroom [50]. Engelbrecht goes on to add that the recognition of diversity in learners is one of the core pillars of inclusivity in education in South Africa [17].

Findings from another work on game design for CSE showed that while gender identity could affect player enjoyment, genre preferences had impact on both experience and outcome performance - independent from gender, hence the need to have a better understanding of target audience [21]. In addition, in an evaluation of a game to enable students to learn the Spanish language, users criticized it for lacking diversity such as diverse game characters, aspects of Spanish culture in game tasks and the inability to customize the game play experience [39].

A claim has also been made that it is best to design games with the background culture of the intended audience in mind [27]. The work by Malliarakis et al. [30] showed that existing educational games designed with a focus on computer programming do not enable lecturers to configure the game environment according to the pedagogical goals related to the respective unit of learning. Moreover, a comparative analysis of five Serious games specifically designed to teach the recursion topic in CS1 (see Table 1) reveals that none are customizable, use simple textbook examples, consider culture or student background in their design. Further, only two are designed with constructivism learning theory [2, 20] in mind. In this theory, learning is a constructive process that involves problem solving and understanding as learners build internal models of knowledge [2].

This paper approaches programming game design through the diversity lens. It considers the missing aspects identified in the reviewed literature such as genre preferences, relevant curriculum examples, pedagogy, gender, culture, learning context, game customisation as well as student computer background.

## 3 Method

### 3.1 Sampling

The questions raised in this paper are addressed by interviewing 17 CS1 lecturers from higher learning institutions based in Kenya and South Africa. Four participants were female and 13 male. Of the females, 3 came from South Africa and 1 from Kenya. Among the males, 5 were from Kenya and 8 from South Africa. Twelve participants (70%) were very experienced. Overall, the most experienced had 30 years and the least 3 years of experience. One of the participants was a professor and an expert in game development who was recruited on purpose to give insights from game design and development perspective. Six respondents had used games in teaching. Four (66%)

**Table 1.** Comparative Analysis of Serious Games Designed to Teach Recursion (PYR = Program your Robot)

| Feature | Elemental [12] | Cargo Bot [46] | Light Bot [1] | PYR [26] | Saving Cera [7] |
|---|---|---|---|---|---|
| Supports writing code | ✓ | ✗ | ✗ | ✗ | ✓ |
| Stack visualisation | ✓ | ✓ | ✗ | ✗ | ✗ |
| Debugging feature | ✓ | ✗ | ✗ | ✓ | ✗ |
| Drag and drop actions | ✗ | ✓ | ✓ | ✓ | ✗ |
| Learning goals clearly aligned | ✗ | ✓ | ✗ | ✓ | ✗ |
| Supports learning by construction | ✓ | ✗ | ✗ | ✗ | ✗ |
| Offers scaffolding | ✓ | ✗ | ✗ | ✗ | ✓ |
| Freedom to explore level | ✗ | ✓ | ✗ | ✗ | ✗ |
| Allows customisation | ✗ | ✗ | ✗ | ✗ | ✗ |
| Considers culture and diversity | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses simple textbook examples | ✗ | ✗ | ✗ | ✗ | ✗ |

had personally designed the games used. Consequently, the data gathered is deemed relevant from both pedagogy and game design perspectives [48]. All the participants from South Africa were recruited from the Southern African Computer Lecturers'Association (SACLA) CS1 lecturers' list created in 2018. On the other hand, it was convenient to recruit respondents from Kenya because of the already established contacts. A summary of the participants is shown in Table 2.

### 3.2 Procedure

Ethical clearance was first sought from the Faculty of Science Research Ethics Committee of the University of Cape Town and relevant institutions. Invitation emails were then sent out explaining the purpose of the study to the potential participants. Thereafter, interviews were scheduled with the 17 willing participants. This was followed by sending written informed consent forms for signing before the interviews. Each interview lasted an average of 45 minutes. The interviews were structured to have focused topics aimed at eliciting particular themes relevant to the focus of the research. The interviews contained questions about the teaching experiences, game adoption barriers, teaching practices (e.g. teaching time and class examples) as well as the recommendations on suitable teaching analogies for designing games to teach the recursion topic to diverse students. Additionally, there was a question regarding what the lecturers thought could guide the design and development of a game authoring tool capable of creating different games for diverse students. As the interviews progressed, we began to realise the recurrence of similar data. By the final participant, data collection reached a point where no new insights were generated i.e. data saturation was reached [44]. At this point, no new participants were recruited. Skype interview sessions were recorded using FlashBack Express video screen recorder. For the respondents interviewed face to face, recording was done using a mobile phone audio recorder. A game expert validated the findings.

### 3.3 Analysis and Presentation

Findings were manually transcribed and summarised into textual data using Microsoft Word. This was followed by Thematic Content Analysis (TCA) [3, 43, 51]. TCA is a descriptive presentation of

**Table 2.** Respondents Demographic Information

| Lecturer | Experience(Yrs) | Gender | Country of Origin |
|---|---|---|---|
| L1 | 30 | Female | South Africa |
| L2 | 13 | Male | Kenya |
| L3 | 18 | Male | South Africa |
| L4 | 16 | Male | South Africa |
| L5 | 8 | Female | South Africa |
| L6 | 10 | Male | Kenya |
| L7 | 3 | Male | South Africa |
| L8 | 6 | Male | South Africa |
| L9 | 18 | Female | Kenya |
| L10 | 5 | Male | Kenya |
| L11 | 10 | Male | South Africa |
| L12 | 10 | Female | South Africa |
| L13 | 5 | Male | South Africa |
| L14 | 25 | Male | South Africa |
| L15 | 10 | Male | Kenya |
| L16 | 4 | Male | Kenya |
| L17 | 14 | Male | South Africa |

qualitative data such as interview transcripts gathered from respondents or other identified texts on the topic of study [3]. Common concepts/ ideas in the textual data were identified/ highlighted and grouped into common categories/ themes. Codes (concepts) were either included or excluded guided by the interview questions. Coding was iteratively done until no new themes emerged. Overall, three iterations were done. The two authors who conducted the analysis met regularly to discuss and agree on the coding/ analysis of data. Where appropriate, tables were used to guide discussions.

## 4 Results and Discussion

**Game Adoption Barriers** After asking respondents about their demographic information, an inquiry was done to find out potential reasons why games have not been widely adopted in teaching. Fourteen (82%) reported 'lack of time in the CS1 curriculum and the effort required'; another ten (59%) said 'lecturer background and lack of awareness'; nine (53%) mentioned 'difficulty to get relevant - well designed games and tools'; while three (18%) were not sure if game use would enable lecturers to effectively deliver content. Some remarks follow:

- *"It is hard to infuse games into the curriculum considering time"* [L5].

- *" In my programming education up bringing I have never seen games used in teaching ... so the way I was introduced to programming is what I give back"*[L10].

- *"when I was taught programming in my university I never learned games"* [L9].

- *"I do not know how effective it would be compared to the traditional learning and so I fear trying to use it because I am not sure of the impact"* [L2].

- *"It takes time and again there are no right game resources"[L8].*

In a broader sense, the findings on adoption barriers could extend the debate on the diversity topic by stressing the need to expose students to introductory programming [15], and use of alternative learning media such as games in an attempt to address the shrinking pipeline problem in Science Technology Engineering and Mathematics (STEM).

## 4.1 Teaching Practice

Responses to the question on teaching practices and approaches adopted by lecturers when covering the recursion topic can be summarised under two themes: (i) teaching time and number of students and (ii) class examples.

**Teaching time** When asked to comment on time taken to teach the recursion topic and how it was organised, most lecturers said they spent between two and three lectures in total on the topic. However, we noted some variation on the number of hours spent on the topic and how it was spread. As an illustration, a summary of time spent by nine respondents is shown in Table3.

**Table 3.** Recursion Teaching Time

| Lecturer | No of Lessons | Minutes Per Lesson |
|---|---|---|
| L6, L15 | 3 to 4 | 80 |
| L2, L10 | 2 to 3 | 60 |
| L3, L4, L13 | 3 | 45 |
| L9 | 3 to 4 | 60 |
| L5 | 5 to 7 | 45 |

While acknowledging the importance of recursion in computer science, some respondents alluded that the CS1 curriculum did not allocate much time for the topic. Consequently, they were forced to find extra time to reinforce the concept. Comments from such respondents included mentions like:

- *"... in our current curriculum we spend only 2 or 3 lectures if ... may be only 2 but not much"* [L2]

- *"I think it is always sort of the pillar of all these programming courses ... you find it is the one always you repeat after doing ... students will always want you to come back and back and back and back from the first time you introduce it"* [L8],

- *"I usually find time and come back to it after teaching binary search and sorting just to try and reinforce it"* [L5].

Given the pressure of time, two models of practical game adoption were proposed: (i) short games for quick just in time class illustration and (ii) long games for after class practice. Lecturer L1 proposed a design approach that could give students games that allow them to practice for longer hours after class while at the same time give a lecturer the ability to track their progress. She made particular reference to universities that enroll large numbers of students like hers (700 to 1000) for CS1 with different backgrounds. Her proposal was supported by lecturer L9 who added: *"there was a time we wanted to change one of our introductory course ... only to realise that over 70% of the students were government sponsored with no computer background"*. Regarding the number of students, respondents were unanimous that CS1 classes attract large numbers, mostly drawn from different departments.

The finding on time is relevant and in agreement with that of Battistella and Wangenheim [8] on games for teaching computing in higher education that showed that most games are designed to fit into the duration of a class or to be played rapidly (about 10 minutes). Additionally, other than the practical aspect of time, these results further supplement current literature on programming game design by introducing the practically large number and diverse CS1 students aspect.

**Class examples** Next, participants were asked to comment on the examples they use in class to teach recursion. Findings indicate that a majority (77%) use textbook examples. Popularly mentioned were: Factorial, Fibonacci series, GCD, binary search, power of a number, palindrome, strings, towers of Hanoi, and recursive patterns. Lecturer L11 recalls clearly his experience in teaching recursion by saying: *"I have used patterns of very simple problems to teach recursion"*. It is surprising to note that only two (12%) lecturers said that they use real-life scenarios in class.

## 4.2 Teaching Analogies and Design Ideas

After questions about recursion teaching practices, participants were asked to recommend suitable teaching analogies (examples) they felt could be designed when creating different programming games to teach novices recursion. Results indicate that fifteen (88%) participants believe that such games should use simple algorithms. In their view, consideration of the fact that the examples used do not introduce extra cognitive load to students is crucial. As exemplars, some of the respondents express their opinion using quotes like:

- *"...give them simple tasks"* [L1]
- *"if not made very complicated can be of value"* [L2]
- *"let them be simple ... not very complicated algorithms ... I tried using a normal fan like pattern in my CS1 class but from experience students found it difficult to understand recursion"* [L3]
- *"simple and basic'* [L4]
- *"simple and neat with simple rewards, levels and scores ... 2D not 3D"* [L5].

A mixed reaction was noticed on whether to use 2D or 3D games. While lecturer L5 who also teaches a design course and lecturer L14 - a professor in game design proposed a 2D game with simple and neat rewards arguing that it will keep it simple and reduce distraction, lecturer L13 with an interest in graphics recommended 3D for their visual learning appeal.

Though textbook examples were suggested by most participants, some respondents brought up a useful debate on the extent to include them. Lecturer L3 felt that it was mandatory to fully include them by arguing: *"classical recursive examples like factorial, binary search, Fibonacci must be fully there in the design knowing very well the importance of maths in computer science"*. On the contrary, L8 saw it differently, commenting: *"I think their use is a disadvantage to students with poor mathematical background"*. This view was supported by lecturer L13 who also cautioned against using only mathematical examples, particularly, those that novice students had not encountered in their high school. He said: *"for me the expectation that every student entering university understands factorial is reasonable ... expectation that they will understand Fibonacci am not sure, I don 't think that much thought has gone into the possibility of using real-life scenarios"*.

Findings on class examples and proposed teaching analogies, potentially, could equally add to the literature on the topic in CSE. It seems that inclusivity in programming games design could be advanced by using a variety of teaching examples that would cater for different learning needs. This is in line with the recommendation by Ibe et al. [22]. It could therefore be argued that a design approach that adopts both real life and textbook examples would increase the potential of diverse CS1 educators and learners to embrace a Game Based Learning (GBL) approach.

## 4.3 Creating Diverse Games

Views of respondents were also sought regarding what could guide the design and development of a game authoring tool capable of creating different instances of games for diverse novice students. To elicit this information, three aspects were investigated: (i) unique design factors; (ii) game attributes and (iii) pedagogy.

**Unique design factors** From the text analysis of the responses received about unique factors that lecturers thought could guide the design of programming games for diverse students, four key themes emerged, namely: (a) context; (b) gender; (c) religion; and (d) childhood game experience.

One lecturer (relatively young in age) could not hide her frustration with the fact that most digital games played in Africa borrow a lot from the West. She lamented: *"why not contextualize to local games such as ... 'Banu'and local characters"* [L9]. Banu, also known as 'Banta'in Kenya, is a marble game commonly played by two people during childhood. Lecturer L15 who has taught a multimedia course using games also suggested a localised game scenario where a Masai Moran (warrior) fights lions to protect a Manyatta (hut) with the warrior as the main character. Masai is a well known community in Kenya for maintaining their culture.

A similar remark came from another lecturer whose opinion was: *"if you are to design a game for the youth in Kenya then probably, characters based on local games, football stars, celebrities in music industry, and surprisingly politicians will make them interested"* [L2]. A respondent from South Africa echoed the same sentiments by saying: *"background and culture will affect the icons ... shapes ... symbols that you use ... people from different regions are sensitive to different symbols ... a certain icon in Cape Town will mean a totally different thing in Kenya"*[L8].

Three (18%) of the participants considered gender an important factor when designing effective programming games. For instance, after the interview, respondent L12 further sent an email suggesting two different design scenarios for male and female students. Adding his voice to and in support of the gender debate, L14 - a game expert, however suggested gender neutrality in design: "while considering gender ... why not just diverse backgrounds in a neutral way".

Some feedback from four (24%) respondents illustrated that it is equally important to design games that would interest learners based on how they grew up. One of them described his experience when teaching an introductory programming course in the Northern hemisphere. According to him, classes started in October where students did an objects first introductory course from October to December. This was followed by a project in December during the last week of the semester. Students were allowed to pick a project to work on based on scenarios they were interested in. He said: *"one year, two of my students collaborated in a battle Bot game ... a game of Robots ... and they really enjoyed the fact that it was something they could relate with remembering playing it when growing up"* [L11].

Another practical demonstration on how childhood game experience could influence game design for different students in Computer Science Education (CSE) was a revelation by a lecturer from Kenya. She recalls vividly one particular teaching experience saying: *"...I remember there was a time when we had some students actually using AI techniques to develop a local game ... we picked this game so that students could relate with things they have grown up with..."* [L9]. In one unique case, one respondent suggested that lecturers should be careful when using games that include gambling since they may offend some religious beliefs [L11].

As in the National Science Foundation [35] report, this study's findings on gender, context, religion and childhood game experiences contribute to the diversity topic by considering target learners, but from Africa. Particularly, in the context of South Africa, universities predominantly enroll CS1 students from diverse racial, cultural and computer knowledge backgrounds.

**Customisable game attributes** Participants were further asked to propose game attributes or features a lecturer could change/ customise to create different instances of games with unique game play experiences using a game authoring tool. Pointing out the lack of such tools, L15 said: *"I have not seen any educational game development software"*. Supporting this, two other participants observed that game generation would help lecturers save on time spent searching for appropriate teaching games and also present them with variety. One of them said: *"a number of years if you keep using the same examples over and over the students don't see the value ... there is need for refreshing examples used by lecturers"* [L11]. A similar view was held by another who remarked: *"do not constrain them to a particular example"* [L7]. The two views illustrate the need for variety.

Game complexity and challenge levels was also suggested by sixteen (94%) respondents. Some comments included: *"design for challenge at different game levels"* [L1]; *"different levels of difficulty/ complexity-for different types of students"* [L6]; *"change game complexity"* [L8]; and *"vary the levels of complexity such that they start by coding simple ones and also to take care of slow learners"* [L10].

Lecturers L2, L6, L9, L10, L12, L15, L16 and L17 felt that characters and their unique parameters like clothes and colour could also provide an opportunity to create unique game experiences. While backing up the idea of game generation, one respondent who considered himself a gamer said: *"I think a nice feature would be a game that has a lot of customization or configuration options for instance I remember in the car race game you could change it to your specification ... change the color, tyres"* [L6]. On the contrary, while stressing the role of visual learning, lecturer L13 observed that the design should steer way from using characters since such could bring in personal issues. Instead, he advised that neutral objects could be used in their place. Other customisable game features suggested were: scene, background, theme, texture, light intensity, interaction style, and character. Lecturer L14 added: "I think that it is very important to consider game genre as another possible attribute". A summary of some customisable game features is presented in Table 4.

**Table 4.** Customisable Game Design Attributes

| Game Genre | Game scenario | Back ground | Character | Interaction style | Complexity |
|---|---|---|---|---|---|
| Adventure | Maze type game | Color | Cloth | Drag and drop | Simple |
| Role Playing | Factorial of a number | Scene | Hair | Write full code | Intermediate |
| Simulation | Simple pattern | Texture | Body color | Complete snippet | Complex |
| Action | Tree traversal game | Sound | Eye color | | |

**Pedagogical considerations** Finally, participants were asked to reflect on their teaching experiences and share pedagogical/ instructional design considerations they deemed suitable for games to teach recursion to different students. It emerged that there is need to balance game play and learning given diversity in a class. Two lecturers were quick to caution that from their experience one of the reasons why lecturers do not use games in class and some students do not take them seriously is the fact that they do not directly see their educational value. One lecturer advised: *"design to engage them in such a way that they see relevance ... value in it"* [L11]. This was supported by another who added: *"students may feel they are just playing if the feel of playing to learn is not there, particularly the bright ones"* [L8].

Twelve (70%) participants pointed out that individual differences among learners in a class ought to be considered. One remarked: *"I tried using a game in class and the challenge was to get tasks to take care of the weak students ... pitch a game at the very weak students and their capabilities ... design for different types of learners"* [L4]. Another thirteen respondents (76%) made remarks suggesting that game goal and learning goal should be clearly aligned [38], and that a deliberate effort should be made to balance design to cater for all types of learners. In CS, the Cargo Bot serious game by Tessler et al. [46] and Program Your Robot by Kazimoglu et al. [26] are illustrations of design by aligning learning and game goals.

Two lecturers felt that programming game design should allow students to actively construct their programming knowledge through active coding. For instance, lecturer L2 said: *" my experience is that it is important to let students be in control of their learning by writing code"* [L2]. His position was supported by another respondent who commented: *"..design to take them through step by step and to practice by forcing them to write and compile code..."* [L7]. An example of a game in CSE that enables students to learn programming by practicing coding is Elemental the Recurrence [12]. These findings suggest consideration for constructivism learning theory [2, 20] in design. Particularly for diverse learners in CS1 [47].

Careful scaffolding and student support, especially for weak students, was yet another admirable instructional design feature according to lecturer L8. He said : *" design for user guided inquiry learning to allow them try different things"*. Scaffolding is demonstrated in Saving Cera serious game by Barnes et al. [7] and Elemental the Recurrence by Chaffin et al. [12] in CSE. Additionally, embedding motivation, reward and challenge in the design was seen as important by respondents L5, L8 and L16. Participant L5 remarked: *"I have taught second year course with games ... students were very interested ... make it fun and amazing ... amusing ...engaging...challenging...motivating"*. Singling out the use of leader boards and points, lecturer L16 who had used games in teaching a third year computer course additionally stressed the need for competition and rewards in design. Agreeing with this position, lecturer L14 (a professor and game expert) added his voice by saying: " reward system is very important from game design point of view ... furthermore the core aim is pedagogy ... to design educational games, particularly to teach the recursion topic".

## 4.4 Summary

Evidence presented in this paper suggests that game adoption rate still remains low despite the potential motivational and retention impact on learners. Although most lecturers use textbook examples in class, it appears this could also potentially disadvantage some students and further propagate the lack of diversity among CS students. An educational game design approach that strictly adopts only such examples might in itself cause a learning barrier to some learners. Consequently, when designing CS1 educational games for diverse learners, it would be advisable to use textbook examples learners are familiar with, simple algorithms and real life analogies. For example, Depth First Search (DFS) algorithm used in Chaffin et al. [12] would be considered complex for

CS1. In addition, ability to have unique characters is equally important. However, caution should be taken and where necessary, neutrality could be considered in design. Some scholars have reiterated the lack of design guidelines/ principles to create serious games [14]. This paper attempts to address this gap by proposing the following eight design principles that could guide requirements engineering and creation of diverse games to teach different students programming:

1. Use diverse simple mathematical examples learners are already familiar with to reduce cognitive load, while factoring in practically available time in the curriculum.
2. Use a variety of fun and enjoyable real life analogies that students can easily relate with.
3. Consider particular gender needs.
4. Carefully factor in student's background in terms of computer knowledge, culture and local contexts.
5. Carefully choose game genres bearing in mind religion.
6. Consider students'childhood game experiences.
7. Pitch the design for different learning needs of students in a class, particularly the very weak.
8. Design for game customization to produce a variety of game experiences.

The proposed design principles are supported by findings of other related works. For instance, while calling for a more inclusive design approach, participants in the study by Rankin [39] criticised the game design for lack of (i) diverse game characters that consider gender (principle 3); (ii) cultural aspects (principle 4); and (iii) ability to customise the game play experience (principle 8). Similarly, findings from the works by Khalifa et al. [27] and Rankin [39] confirm that culture (principle 4), customization and game attribute configuration (principle 8) are crucial design considerations when creating diverse games targeting different audiences. Moreover, two key findings from another study that sort the views of 41 game designers and practitioners agree with principles 2 and 7 [23]. In this study, fun is identified as a key design feature for serious games for diversity in line with principle 2. The second key finding suggest that design should gradually release information and allow practice and mastery that can be built upon to give a sense of progression (principle 7). This supports individual progression of students in a CS1 class given different learning abilities [33]. On the other hand, the work by Katz [25] presents seven broad Universal Design (UD) principles for an inclusive education that considers diversity. The first principle, equitable use, maps to design principles 1, 2 and 3 in this paper. The second, flexibility in use, aligns to the proposed principles 4, 5, 6 and 7. Lastly, simple and intuitive use, the third principle aligns to the proposed principle 1.

## 5 Limitations

In total, 17 CS1 lecturers (domain experts) participated in the qualitative study which was less than expected, but interviews were conducted until data saturation was reached. In addition, most respondents (70%) were very experienced. Another limitation could be the small number of participants who had used games in teaching. Nonetheless, it should be noted that some studies have equally shown that lack of educational game authoring tools [45] and other potential problems/ barriers [9, 49] have resulted in low GBL adoption in mainstream teaching particularly in developing countries. Therefore, the small number is a clear reflection of this population. Furthermore, a game expert (professor) assisted in reviewing and validating the findings from game design perspective. Subsequently, the small number does not significantly affect the result. Consequently, we argue that findings of the study are valid, broadly representative and could be generalised to teaching CS to other underrepresented minorities and learning situations that represent diversity regardless of country.

## 6 Conclusions and Future Work

In this paper, we presented findings from a unique study of teaching CS in developing countries. We conducted interviews with 17 CS1 lecturers from higher education and explored how diversity could drive the design and development of programming games. We showed that practical aspects (time, number of students in class and teaching examples) may affect teaching the recursion topic with games. We found that simple mathematical and real life analogies could be designed while considering CS1 syllabus time constraints and diversity among learners. Further, results suggests that domain experts (lecturers and instructors) would be more productive if they were to be given variety of simple games adapted to gender, culture, religion and other environmental factors to aid them in their teaching. Guided by these findings, our next step is to develop a prototype of a game authoring tool (an innovative tool) to be used by programming lecturers to create diverse instances of games to teach the difficult topic of recursion.

## 7 Acknowledgements

This research was partially funded by the Hasso Plattner Institute, the National Research Foundation (NRF) of South Africa (Grant numbers: 85470 and 88209) and University of Cape Town. The authors acknowledge that opinions, findings and conclusions or recommendations expressed in this publication are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

## References

1. Aedo Lopez, M., Vidal Duarte, E., Castro Gutierrez, E., Paz Valderrama, A.: Teaching abstraction, function and reuse in the first class of cs1: A lightbot experience. In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. pp. 256–257. ITiCSE '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2899415.2925505, http://doi.acm.org/10.1145/2899415.2925505
2. Amineh, R.J., Asl, H.D.: Review of constructivism and social constructivism. Journal of Social Sciences, Literature and Languages **1**(1), 9–16 (2015)
3. Anderson, R.: Thematic content analysis (tca). Descriptive presentation of qualitative data (2007)
4. Anyango, J.T., Suleman, H.: Teaching programming in kenya and south africa: What is difficult and is it universal? In: Proceedings of the 18th Koli Calling International Conference on Computing Education Research. pp. 24:1–24:2. Koli Calling '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3279720.3279744, http://doi.acm.org/10.1145/3279720.3279744
5. Ball, A.F.: Preparing teachers for diversity: Lessons learned from the us and south africa. Teaching and Teacher Education **16**(4), 491–509 (2000)
6. Barnes, T., Powell, E., Chaffin, A., Lipford, H.: Game2learn: Improving the motivation of cs1 students. In: Proceedings of the 3rd International Conference on Game Development in Computer Science Education. pp. 1–5. GDCSE '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1463673.1463674, http://doi.acm.org/10.1145/1463673.1463674
7. Barnes, T., Richter, H., Powell, E., Chaffin, A., Godwin, A.: Game2learn: Building cs1 learning games for retention. In: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education. pp. 121–125. ITiCSE '07, ACM, New York, NY, USA (2007). https://doi.org/10.1145/1268784.1268821, http://doi.acm.org/10.1145/1268784.1268821
8. Battistella, P., von Wangenheim, C.G.: Games for teaching computing in higher education–a systematic review. IEEE Technology and Engineering Education Journal **9**(1), 8–30 (2016)

9. Bayliss, J.D.: Using games in introductory courses: Tips from the trenches. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education. pp. 337–341. SIGCSE '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1508865.1508989, http://doi.acm.org/10.1145/1508865.1508989
10. Bruckman, A., Biggers, M., Ericson, B., McKlin, T., Dimond, J., DiSalvo, B., Hewner, M., Ni, L., Yardi, S.: " georgia computes!" improving the computing education pipeline. ACM SIGCSE Bulletin **41**(1), 86–90 (2009)
11. Camp, T.: The incredible shrinking pipeline. Communications of the ACM **40**(10), 103–110 (1997)
12. Chaffin, A., Doran, K., Hicks, D., Barnes, T.: Experimental evaluation of teaching recursion in a video game. In: Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games. pp. 79–86. Sandbox '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1581073.1581086, http://doi.acm.org/10.1145/1581073.1581086
13. Chetty, J., van der Westhuizen, D.: Towards a pedagogical design for teaching novice programmers: Design-based research as an empirical determinant for success. In: Proceedings of the 15th Koli Calling Conference on Computing Education Research. pp. 5–12. Koli Calling '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2828959.2828976, http://doi.acm.org/10.1145/2828959.2828976
14. Chorianopoulos, K., Giannakos, M.N., Chrisochoides, N.: Design principles for serious games in mathematics. In: Proceedings of the 18th Panhellenic Conference on Informatics. p. 1–5. PCI '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2645791.2645843, https://doi.org/10.1145/2645791.2645843
15. Collain, M., Trytten, D.: "you don't have to be a white male that was learning how to program since he was five". In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. pp. 968–974. SIGCSE '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3287324.3287383, http://doi.acm.org/10.1145/3287324.3287383
16. Duarte, E.V., Pearce, J.L.: A cross-cultural review of lightbot for introducing functions and code reuse. J. Comput. Sci. Coll. **33**(2), 100–105 (Dec 2017), http://dl.acm.org/citation.cfm?id=3144645.3144660
17. Engelbrecht, P.: The implementation of inclusive education in south africa after ten years of democracy. European journal of psychology of education **21**(3), 253 (2006)
18. Green, M.C., Khalifa, A., Barros, G.A., Nealen, A., Togelius, J.: Generating levels that teach mechanics. In: Proceedings of the 13th International Conference on the Foundations of Digital Games. p. 55. ACM (2018)
19. Guzdial, M., Soloway, E.: Teaching the nintendo generation to program. Communications of the ACM **45**(4), 17–21 (2002)
20. Hadjerrouit, S.: A constructivist framework for integrating the java paradigm into the undergraduate curriculum. In: Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education. pp. 105–107. ITiCSE '98, ACM, New York, NY, USA (1998). https://doi.org/10.1145/282991.283079, http://doi.acm.org/10.1145/282991.283079
21. Horn, B., Clark, C., Strom, O., Chao, H., Stahl, A.J., Harteveld, C., Smith, G.: Design insights into the creation and evaluation of a computer science educational game. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. pp. 576–581. SIGCSE '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2839509.2844656, http://doi.acm.org/10.1145/2839509.2844656
22. Ibe, N.A., Howsmon, R., Penney, L., Granor, N., DeLyser, L.A., Wang, K.: Reflections of a diversity, equity, and inclusion working group based on data from a national cs education program. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. pp. 711–716. SIGCSE '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3159450.3159594, http://doi.acm.org/10.1145/3159450.3159594
23. Isbister, K., Flanagan, M., Hash, C.: Designing games for learning: Insights from conversations with designers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 2041–2044. CHI '10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1753326.1753637, https://doi.org/10.1145/1753326.1753637

24. Johnson, C., McGill, M., Bouchard, D., Bradshaw, M.K., Bucheli, V.A., Merkle, L.D., Scott, M.J., Sweedyk, Z., Ángel, J., Xiao, Z., et al.: Game development for computer science education. In: Proceedings of the 2016 ITiCSE Working Group Reports. pp. 23–44. ACM (2016)
25. Katz, J.: Teaching to diversity: The three-block model of universal design for learning. Portage & Main Press (2012)
26. Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Procedia-Social and Behavioural Sciences **47**, 1991–1999 (2012)
27. Khalifa, A., Perez-Liebana, D., Lucas, S.M., Togelius, J.: General video game level generation. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 253–259. ACM (2016)
28. Kirriemuir, J., McFarlane, A.: Literature review in games and learning (2004)
29. Lahtinen, E., Ala-Mutka, K., Järvinen, H.M.: A study of the difficulties of novice programmers. SIGCSE Bull. **37**(3), 14–18 (Jun 2005). https://doi.org/10.1145/1151954.1067453, http://doi.acm.org/10.1145/1151954.1067453
30. Malliarakis, C., Satratzemi, M., Xinogalos, S.: Educational games for teaching computer programming. In: Research on e-Learning and ICT in Education, pp. 87–98. Springer (2014)
31. McCauley, R., Grissom, S., Fitzgerald, S., Murphy, L.: Teaching and learning recursive programming: a review of the research literature. Computer Science Education **25**(1), 37–66 (2015). https://doi.org/10.1080/08993408.2015.1033205, https://doi.org/10.1080/08993408.2015.1033205
32. Miller, C.S., Settle, A., Lalor, J.: Learning object-oriented programming in python: Towards an inventory of difficulties and testing pitfalls. In: Proceedings of the 16th Annual Conference on Information Technology Education. pp. 59–64. SIGITE '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2808006.2808017, http://doi.acm.org/10.1145/2808006.2808017
33. Mohamed, A.: Designing a cs1 programming course for a mixed-ability class. In: Proceedings of the Western Canadian Conference on Computing Education. pp. 8:1–8:6. WCCCE '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3314994.3325084, http://doi.acm.org/10.1145/3314994.3325084
34. Monge, A.E., Fadjo, C.L., Quinn, B.A., Barker, L.J.: Engagecsedu: Engaging and retaining cs1 and cs2 students. ACM Inroads **6**(1), 6–11 (Feb 2015). https://doi.org/10.1145/2714569, http://doi.acm.org/10.1145/2714569
35. National Science Foundation: Women, minorities, and persons with disabilities in science and engineering: 2013.technical report, national center for science and engineering statistics, directorate for social, behavioral and economics science, 2013. Tech. rep. (2013)
36. Papastergiou, M.: Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. Computers & Education **52**(1), 1–12 (2009)
37. Parberry, I., Kazemzadeh, M.B., Roden, T.: The art and science of game programming. In: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education. pp. 510–514. SIGCSE '06, ACM, New York, NY, USA (2006). https://doi.org/10.1145/1121341.1121500, http://doi.acm.org/10.1145/1121341.1121500
38. Prensky, M.: Digital game-based learning. Comput. Entertain. **1**(1), 21–21 (Oct 2003). https://doi.org/10.1145/950566.950596, http://doi.acm.org/10.1145/950566.950596
39. Rankin, Y.A.: Diversity by design: Female students' perception of a spanish language learning game. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 670–679. CHI EA '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2851581.2851610, http://doi.acm.org/10.1145/2851581.2851610
40. Richard, G.T.: Video games, gender, diversity, and learning as cultural practice: Implications for equitable learning and computing participation through games. Educational Technology pp. 36–43 (2017)
41. Robinson, M., Zinn, D.: Teacher preparation for diversity at three south african universities. Journal of Education **42**(1), 61–81 (2007)

42. Schulte, C., Bennedsen, J.: What do teachers teach in introductory programming? In: Proceedings of the Second International Workshop on Computing Education Research. pp. 17–28. ICER '06, ACM, New York, NY, USA (2006). https://doi.org/10.1145/1151588.1151593, http://doi.acm.org/10.1145/1151588.1151593
43. Spiel, K., Alharthi, S.A., Cen, A.J.l., Hammer, J., Nacke, L.E., Toups, Z.O., Tanenbaum, T.: "it started as a joke": On the design of idle games. In: Proceedings of the Annual Symposium on Computer-Human Interaction in Play. pp. 495–508. CHI PLAY '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3311350.3347180, http://doi.acm.org/10.1145/3311350.3347180
44. Strauss, A.: 8c corbin, j.(1998). basics of qualitative research: Techniques and procedures for developing grounded theory. Thousand Oaks, CA: Sage
45. Tang, S., Hanneghan, M.: A model-driven framework to support development of serious games for game-based learning. In: Developments in E-systems Engineering (DESE), 2010. pp. 95–100. IEEE (2010)
46. Tessler, J., Beth, B., Lin, C.: Using cargo-bot to provide contextualized learning of recursion. In: Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research. pp. 161–168. ICER '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2493394.2493411, http://doi.acm.org/10.1145/2493394.2493411
47. Thevathayan, C., Hamilton, M.: Supporting diverse novice programming cohorts through flexible and incremental visual constructivist pathways. In: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education. pp. 296–301. ITiCSE '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2729094.2742609, http://doi.acm.org/10.1145/2729094.2742609
48. Torrente, J., del Blanco, A., Cañizal, G., Moreno-Ger, P., Fernández-Manjón, B.: &lt;e-adventure3d&gt;: An open source authoring environment for 3d adventure games in education. In: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology. pp. 191–194. ACE '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1501750.1501795, http://doi.acm.org/10.1145/1501750.1501795
49. Torrente, J., Blanco, A., Marchiori, E., Moreno Ger, P., Fernández-Manjón, B.: e-adventure¿: Introducing educational games in the learning process. pp. 1121 – 1126 (05 2010). https://doi.org/10.1109/EDUCON.2010.5493056
50. Walton, E., Nel, N., Hugo, H., Muller, H.: The extent and practice of inclusion in independent schools in south africa. South African Journal of Education **29**(1), 105–126 (2009)
51. Willson, R.: Analysing qualitative data: You asked them, now what to do with what they said. In: Proceedings of the 2019 Conference on Human Information Interaction and Retrieval. pp. 385–387. CHIIR '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3295750.3298964, http://doi.acm.org/10.1145/3295750.3298964