MEASURE UP: A SERIOUS GAME FOR TOPOGRAPHIC EDUCATION

Jelle Vermandere[1], Maarten Bassier[1], Maarten Vergauwen[1]

[1] KU Leuven, Department of Civil Engineering, Ghent, Belgium - (jelle.vermandere, maarten.bassier, maarten.vergauwen)@kuleuven.be

KEY WORDS: Geomatics, serious game, unity3D, surveying, education

ABSTRACT:

Topographic education requires the use of delicate and expensive equipment in a wide range of field exercises. To reduce the risk of improper handling by students, this work proposes the creation of a serious game to teach the basic principles and correct techniques to complete the field exercises successfully. Due to the constant evolution of the course material, there is also a need for a modular and expandable design to increase the relevance of the serious game over a longer period. Using a system of cloud-based hosting and progression, we can make sure the students are always using the most up-to-date version of the game. The game covers a wide variety of topics, ranging from levelling to triangulation. It is currently in use at the KU Leuven Geomatics group and has been evaluated positively by the bachelor students.

1. INTRODUCTION

The process of learning the profession of a surveyor involves familiarizing themselves with a wide range of equipment and techniques. While these techniques can be taught in the classroom, the more practical exercises require the student to go into the field and learn how to work with the devices. Providing these students with proper equipment is an expensive endeavour both, because of the high costs and the high risk of damage. Having a limited supply of devices leads to less hands-on time for the students, if a large portion of that time has to be dedicated to learning the basics, the actual exercise time is even more limited.

The goal of this work is to provide a safe virtual environment for the students, in the form of a serious game, where they can train and practice the skills needed to properly handle the expensive equipment. The gamified approach to the exercises can improve engagement and a better understanding of the skills required for the field exercises. Furthermore, due to the ever-changing nature of the curriculum, the game needs to be easily expandable and modular to allow for rapid iteration to stay true to the curriculum and state-of-the-art equipment.

The remainder of this work is structured as follows. The background and related work is presented in Section 2. Giving special attention to other serious games covering other topics. Following is the design methodology in Section 3. In Section 4, the different types of exercises and content are explained. Finally, the conclusions are presented in Section 5

2. BACKGROUND AND RELATED WORK

2.1 Serious Games

Serious gaming, or the use of games with a specific educational or training purpose, has been gaining popularity as a method for engaging and educating students in recent years. The use of serious games in education has been shown to have numerous advantages over traditional teaching methods, including increased motivation and engagement, improved learning outcomes, and the ability to simulate real-world scenarios. Providing a gamified version of a complex topic can help the students learn in a more practical and fun way (Robson et al., 2015).

While existing serious games, such as those developed by (Bontchev et al., 2022) and (Hart et al., 2020), provide significant value and a great learning platform in their respective fields, they are designed as static experiences and lack a clear and simple way to expand or adapt the software after release. This limitation is particularly critical in the ever-changing field of education. To address this challenge, (Dankov et al., 2021) proposes a framework that enables educators to easily expand the game beyond its release by storing large portions of the game's data in online databases, allowing the game to dynamically generate new content. However, while this method works well for simple games with consistent core gameplay, it may prove insufficient for more complex expansions required to accommodate the wide variety of surveying problems.

Meanwhile, existing topographic games are mainly focused on testing knowledge around the topic, paying less attention to the practical side of the field (Brovelli et al., 2018). This can be great preparation for a theoretical exam, but the field exercises require the students to apply their knowledge in the field.

2.2 Game design methodologies

The conceptual design methodology for serious games has been a field of much research like (Silva, 2020), the paper suggests using a non-related game genre as a starting point, where the learning material is appended as a reward or awareness during the gameplay. Other approaches suggest a better integration of content and gameplay (Caserman et al., 2020). This method is more practical for topics which invite the agency of the player such as learning a new skill. The concept of learning a new skill for a player is one of the most important parts of a game, even in the regular games industry. By using the same systems in a serious game, the student can be motivated to complete the exercises beyond the pressure of grading, something which is currently still the main driving force for a lot of students.

2.3 Game Expandability

The increasingly online nature of games has allowed for a more continuous game as a service instead of the old ship-and-done

system. Games can now be continuously updated without the user having to manually reinstall or buy a new version. This is a very useful feature for serious games because exercises can evolve to better match the source material, or in other cases, the curriculum is adapted and the game needs to adapt with it. To achieve this, some games are designed with this principle in mind from the ground up (Carlos et al., 2019). With a modular approach, it becomes easy to add content at a later date. This can come in the form of large infrequent content updates, for example at the start of a new semester, or quick small iterations to single exercises when it becomes clear students are struggling with a very specific problem.

2.4 Game Localisation

A key component to making the game accessible to a lot of users is providing a multi-language implementation. This field has been researched thoroughly and there are a number of frameworks available to build on. The Unity3D game engine provides a localisation package (UnityTechnologies, 2020), but is missing some key features for our use case, which will be expanded upon in later sections.

3. DESIGN METHODOLOGY

The game is designed mostly around the principles outlined in (Silva, 2020), with some key differences. While the paper suggests using a non-related game genre as a starting point, we aim to make the gameplay the core learning experience itself. The design of the game started with a very clear goal: to teach the students the new concepts they have to master to complete the field exercises. This practical end goal lead to a progression-based system, where students could gain points by completing exercises. Each following exercise can build on concepts from previous exercises to gradually build up to the full field exercises. Due to the evolving nature of our education program, special care has been given to the expandability of the game. This is achieved on multiple levels, both content-wise and design-wise.

3.1 Content Structure

Due to the wide variety of topics, the game is divided into distinct chapters. Each chapter covers a certain topic and contains a list of exercises or levels. Each level in a chapter handles a specific problem from that topic's field as seen in figure 1. This separation ensures the player can progress both in parallel across topics and linearly within. By creating smaller sub-exercises, the player can both get faster feedback from intermediate results and not worry about making a mistake very early in the progress.

Players can play linearly through the exercises or replay already completed ones for extra points. They can also play a random level, allowing them to practice the recognition of different problems. Which each exercise building on the previous one, the last exercises of each chapter are generally more difficult and require knowledge gathered from all the previous exercises.

Since a lot of new material is being taught by the game, there is a robust helping system in place to aid the students during their exercises. Apart from the main question, each exercise also contains a general help guide to introduce the player to the new concepts as shown in figure 2. Furthermore, should a student

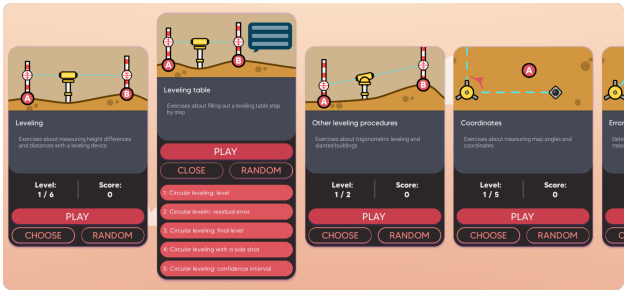

Figure 1: The main menu screen where the player can select the exercise, ordered by chapter. There are buttons in each chapter sub-menu to play the next level or choose a random level.

struggle with finding the correct value, the game can provide feedback on where the student could have made a mistake. To give a relevant help message, the game keeps track of the current state of the field. This is mostly achieved by tracking the set-up locations and actions by the player.

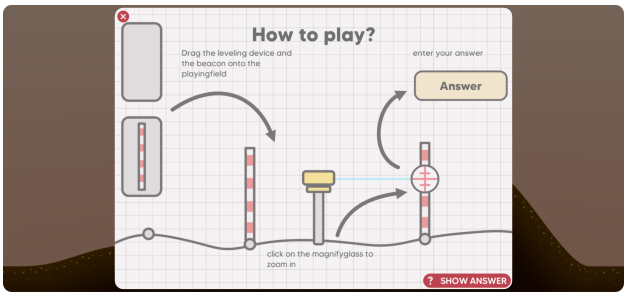

Figure 2: A screenshot of the help guide of a leveling exercise. Showing how to play the game and optionally show the correct answer.

3.2 Game design Structure

To facilitate a modular game, the code structure underneath must also be easily expandable. This is achieved with an inheritance-based workflow where each chapter only contains new functions to help with their specific problems. Everything that is consistent across the different chapters is defined at a lower level in the game structure. This allows new exercises to be created more rapidly and with less code and repetition. The same goes for all the supporting features like menus and save data storage.

The two main scripts controlling the game are the "-Controller" and "-Questions". Each level contains a child script of the base controller- and questions-script, both specific to the relevant chapter as seen in figure 3. The controller script handles all the gameplay logic, meaning the actual placement of the objects, the interactivity on the field and the calculations of the corresponding measured values. The question script handles all the supporting logic around the exercise, like the question setup and the evaluation.

The modular nature of the game ensures all the other features are updated accordingly and automatically. By hosting the game on a website as a free software-as-a-service, the player is guaranteed to use the most recent version of the game. ensuring the ability to learn with up-to-date exercises.

3.3 Expanding the Game

The game logic is layered in multiple levels of inheritance, Making it easier to expand and create new content while still ensuring compatibility with existing game logic. It is possible to

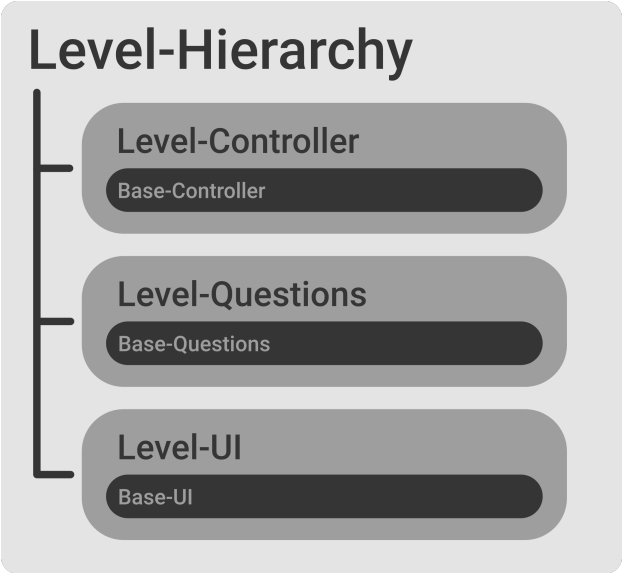

Figure 3: The inheritance structured, where the three main components are the ”Level-Controller”, ”Level-Questions” and ”Level-UI”. Each component inherits the base functions from its parent component

create a new level and, should the theme of the new level not match the available chapters, create a new chapter.

3.3.1 Adding a new Level As shown in figure 3, each level is contained in a Unity scene with a specific structure. The scene contains a number of predefined objects or ”prefabs” that contain the necessary game logic. Each type of exercise exposes several variables that can be altered to personalize the specific level as seen in figure 4. These range from changing the number of points to the question type. These variables are expanded on in the later chapters. It is possible to create a new level based on the chapter-specific template and only require the altering the variables.

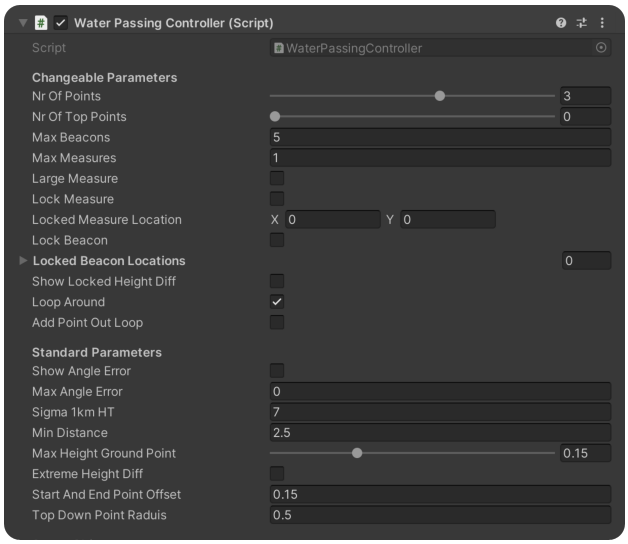

Figure 4: A screenshot of a controller script, with a number of variables exposed so the exercise can be altered.

3.3.2 Adding a New Chapter For bigger expansions where you cannot simply reuse the same base prefabs from an existing chapter, it is possible to define a new chapter. Each existing chapter is built on a set of interfaces which they inherit. New chapters can use the same structure to inherit all the important game logic, apart from the actual gameplay. The information about a level is stored in a Unity ”Scriptable Object” as seen in figure 5, a persistent storage medium to organize your data. The games uses the collection of chapters to build the UI during runtime, to ensure it always reflects the most recent content. More detailed information can be found in the full api documentation.

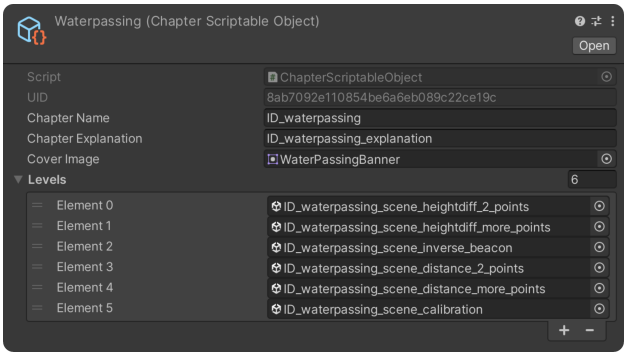

Figure 5: A screenshot of a chapter scriptable object, defining the name, explanation, cover-image and the levels that are part of this chapter.

3.4 Localisation

As stated in section 2, there are already a large number of localisation systems available. Each method, however, did lack some key features required for our needs. In particular, a system to inject more complex datatypes like arrays or vectors into the localised strings.

For our implementation, All the data is stored in a CSV file containing key-value pairs as seen in table 1. the first row is the key that is used to fetch the correct value of the specified language.

Key	EN	NL	...
ID_LANG	English	Nederlands	...
ID_HELLO	Hello world!	Hallo wereld!	...
ID_...	...	...	...

Table 1: An example of the Localisation CSV, using a key to match values from different languages.

When implemented, each textual element contains a script which automatically matches the keys to the correct language and displays the correct value in the scene.

3.4.1 Variable Injection Because our game heavily relies on solving problems, the questions need to include some numerical values from the scene environment. This can range from the position of a point to a precise angle measurment. While existing localisation solutions already allow developers to inject simple datatypes like floats or ints, They do not allow the injection of more complex data structures like arrays or lists.

It is possible to add values from public variables in the respective ”Chapter-Controller” to the question title and explanation by inserting {VARIABLE_NAME} or {VARIABLE_NAME[i]} in the localised values. adding a * in between the {} will multiply the value by the worldScale. For example: ”Place the beacons on the {nrOfPoints} points, with coordinates: {meetpunten[0]*}, {meetpunten[1]*}.” will become: ”Place the beacons on the 5 points, with coordinates: (1.35,2.65).”

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-5-W1-2023-69-2023 | © Author(s) 2023. CC BY 4.0 License.

3.5 Progression

Since the game is designed to allow the player to progress through the game along with the field exercises, there is a need to store their progression. To allow this, they can log in using their university email. Using the universal single sign-on method provided by the KU Leuven Shibboleth as seen in image 6.

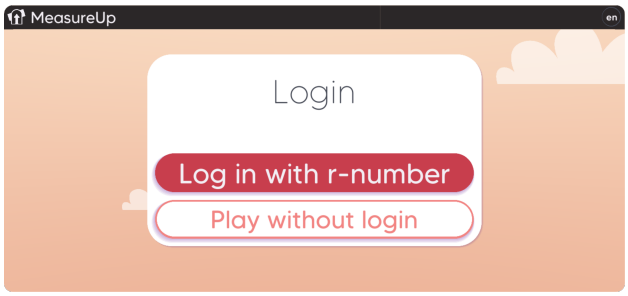

Figure 6: A screenshot of the login screen, where the player can use their university email to log in. It is also possible to change the language in the top right corner.

The progress data is then stored in a central database with a structure laid out in table 2. During the login process, the save data is retrieved and stored in a ”CustomInfo” class as shown below. Once the data is loaded in, The player can continue from where they left off last session.

```
public class AS_CustomInfo
{
    public int totalScore;
    public List<ChapterInfo> chapters
    {
        public string UID;
        public List<int> scores;
    }
}
```

The teacher is able to evaluate each student at the end of each course to see what parts were understood better or worse than others. The player data can be exported into a CSV file to easily evaluate each student’s performance in the tests. Points are awarded for each completed exercise, the amount can be adapted to the difficulty of the game, and the maximum awarded points can also be subtracted by the amount of tries a student needs to complete the exercise.

To further incentivise student participation, there is a global leaderboard which displays the overall scores of the top ten students as well as the student’s own score. Each semester, the scores are reset and the new students can compete for the top spot.

id	studentnr	custominfo	creationdate
The auto-generated identifier	The unique student number	The game data	The date at which the account was first created

Table 2: The player data is stored in a database according to this schema, where the main gameplay data is stored in the ”custominfo” field.

4. GAME CONTENT

The different chapters mostly cover variations of one of the 2 main topographic topics: Leveling and triangulation. They are distinctly separated because the levelling is viewed from the side, while the triangulation is displayed from a top-down perspective.

4.1 Leveling

The levelling is based on the placement of beacons and measurement devices. Using a click-and-drag interface, the student can intuitively choose the best placement of the beacons to ensure the device has a direct visual of both beacons. The angle of the measure can be changed, to allow for more extreme height difference calculations.

The main information is provided through the enlargeable looking glass which provides an accurate representation of the real view. allowing the students to practice the same method they need to use on the field as shown in figure 7.

The main parameters that can be changed to create unique exercises are: The number of points to spawn, the number of allowed devices, a toggle to let the points loop and more. All the Parameters are documented in the full documentation.

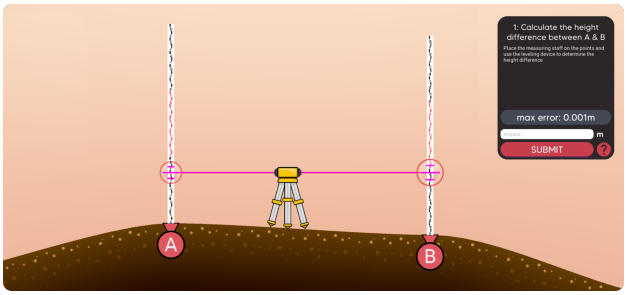

Figure 7: A screenshot of a levelling exercise. The player can read the values of the measurement at the enlarged circles near the beacon, after which they can enter the correct value in the answer field.

4.1.1 Height Differences To determine height differences students need to learn how to properly place and read the device and the beacons. It is possible to limit the number of objects a player can place to better simulate a real environment. By linking multiple points in one exercise, students also have to learn how to connect separate measurements to determine the final difference. As an added complexity, height differences can also be calculated with an angled theodolite as seen in figure 8.

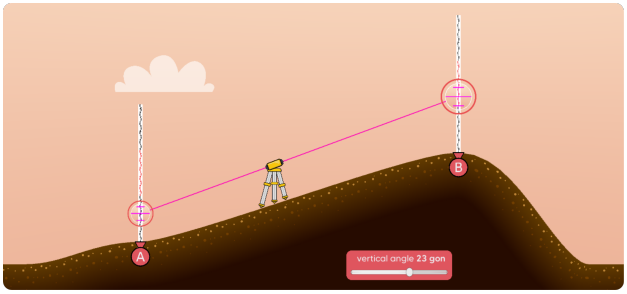

Figure 8: An example of an angled levelling exercise, where the player can choose the angle to reach both beacons and read the corresponding height differences.

4.1.2 Distance calculation Aside from height differences, students also need to learn how to calculate distances using a theodolite. The game provides a detailed view visible through the visor and the upper and lower wire are visible. To make better use of the screen size, a non-uniform scaling factor is applied to the world, to create a larger distance difference despite the limited width of the screen. This ensures the top and bottom lines of the measuring device are a sufficient distance away from each other to properly read the values as seen in figure 9.

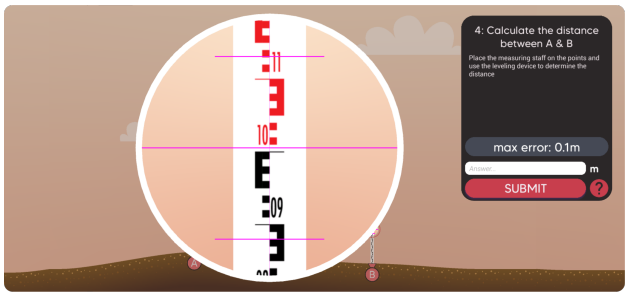

Figure 9: A screenshot of the zoomed-in lens player can see to read the correct values. Because the horizontal distance is scaled, the top and bottom lines are further apart, making it easier to read.

4.1.3 Circle Resection By using both the height differences and distances between a range of points, students are taught to perform a circle resection, including error correction. Since this can be a very long exercise and to prevent large progression losses during the whole process, the problem is divided into a number of sub-problems, each covering a certain step in the process as seen in 10. As an added variation, points can also be placed outside of the loop, meaning they are not considered in the error correction process.

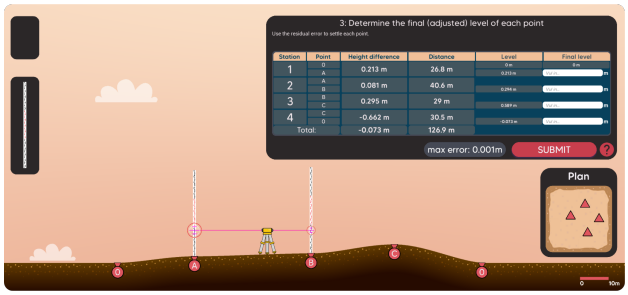

Figure 10: A screenshot of a levelling table. The player must fill in the correct values

4.2 Triangulation

The triangulation is approached from a top-down perspective and its main goal is teaching the students about angle and distance measurements and how they can be used to determine positions as shown in figure 11.

Another important aspect is the error propagation and how the positions of beacons can affect it. Having an interactive real-time view of the potential error has proven very valuable in the understanding of the problem.

The player can place the total station and beacons by clicking on the screen at the desired position, and remove them with a right-click. Depending on the exercise, the beacons will either connect in series, or in parallel, this ensures a correct measurement can be performed without the need to move the station multiple times.

4.2.1 Coordinate Calculation The main way the students learn how to determine a coordinate with a total station is through the calculation of the map angle and distance from a reference point. Students have to learn to determine these values, even in more complex situations like transformed reference systems as seen in figure 12 or multi-step calculations. By placing obstacles in between the start and end points, the player is forced to find an optimal position to place extra beacons to be able to calculate the final position.

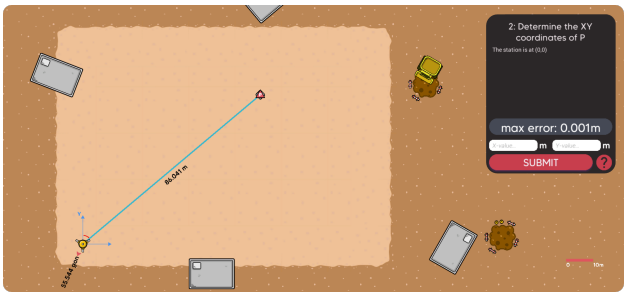

Figure 11: A screenshot of a top-down triangulation exercise. The player can place beacons on the field and calculate the coordinate based on the angle and distance measurement.

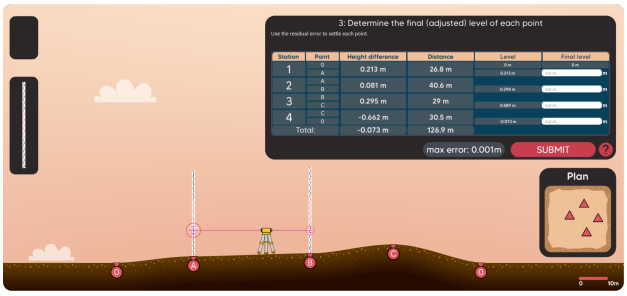

Figure 12: A rotated axis exercises where the player must first determine the transformation from the red to the blue axis, and then determine the position of the point in the new axis system.

4.2.2 Error Propagation A big part of performing accurate measurements is the ability to identify and correct measurement errors. Because a digital simulation is inherently perfect, the game can apply slight reading errors to the displayed values. To visualise these potential errors, each measurement point can display its error ellipse, the area of uncertainty where the point could be. Because it is common to perform multiple measurements in series, these errors get propagated over the length of the measurement as seen in figure 13. The focus of these exercises lies in determining or minimizing the error ellipse size. The real-time feedback also allows the player to explore the impact of the different error types.

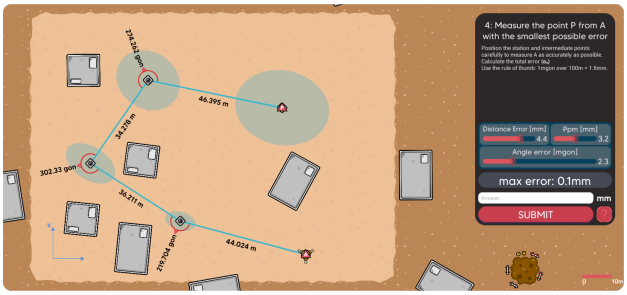

Figure 13: A screenshot of the error propagation exercise. The blue ellipses show the expected error at each point, becoming bigger with each subsequent measurement.

4.2.3 Point Finding Determining the position of a point with the aid of reference points can be achieved in a number of different ways, depending on the number of points and equipment there are three main ways: Forward propagation, backward propagation or bilateration. Determining the location of a point using forward propagation requires two separate angle measurements from known points. To prevent the player from setting up at the asked point, it is possible to hide the angle display depending on which point it is currently stationed at. Because some methods require distance, a clear visual distinction between the types of measurement is created by using different

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-5-W1-2023-69-2023 | © Author(s) 2023. CC BY 4.0 License.

line types as seen in figure 14. Dotted lines indicate an angle measurement without a distance component, while the distance measurement is indicated with a full line. As a final exercise, the player is asked to complete a traverse table. It is here that the small visual errors are necessary, otherwise, all the measurements would be correct and there would be no need for an error correction.

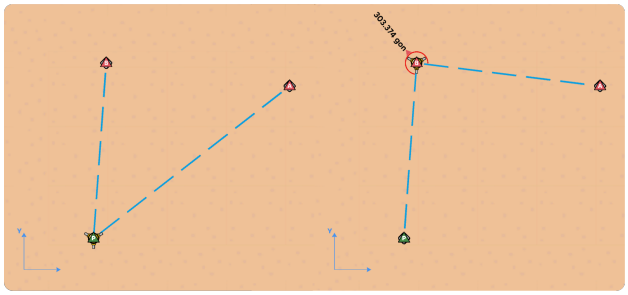

Figure 14: A comparison of the different set-up points to force the students to use a certain method to calculate the point. The left shows the point being set up at the unknown point which is not allowed to perform an intersection, so the angle display is not shown.

5. CONCLUSIONS

In this work, we provided a method and content to create a serious game in the field of Geomatics. Having created an expandable and accessible game will ensure the game can evolve with the content of the course and prevent it from becoming obsolete in the near future. The content is structured in a way to encourage both linear progression throughout a topic and parallel progression across the different topics. The online progression storage makes it easy for educators to evaluate the students on a personal level and the class as a whole. The serious game has already provided great value to our students and has been part of the curriculum for a number of years now. By allowing the students to practice these principles at home before they go out on the field, they have a better understanding and can complete the field exercise more accurately and faster.

ACKNOWLEDGEMENTS

This project has received funding from the FWO-SB grant (grant agreement 1S16923N) and the Geomatics research group of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium.

REFERENCES

Bontchev, B., Antonova, A., Terzieva, V., Dankov, Y., 2022. “Let Us Save Venice”—An Educational Online Maze Game for Climate Resilience. *Sustainability (Switzerland)*, 14.

Brovelli, M. A., Celino, I., Fiano, A., Molinari, M. E., Venkatachalam, V., 2018. A crowdsourcing-based game for land cover validation. *Applied Geomatics*, 10, 1-11.

Carlos, J., Delgado, S., Bazán, P., 2019. Educational Serious Games as a Service: Challenges and Solutions Juegos Serios Educativos Como Servicio: Retos y Desafíos.

Caserman, P., Hoffmann, K., Müller, P., Schaub, M., Straßburg, K., Wiemeyer, J., Bruder, R., Göbel, S., 2020. Quality criteria for serious games: Serious part, game part, and balance. *JMIR Serious Games*, 8.

Dankov, Y., Bontchev, B., Terzieva, V., 2021. Design and creation of educational video games using assistive software instruments. 271, Springer Science and Business Media Deutschland GmbH, 341–349.

Hart, S., Margheri, A., Paci, F., Sassone, V., 2020. Riskio: A Serious Game for Cyber Security Awareness and Education. *Computers and Security*, 95.

Robson, K., Plangger, K., Kietzmann, J. H., McCarthy, I., Pitt, L., 2015. Is it all a game? Understanding the principles of gamification. *Business Horizons*, 58, 411-420.

Silva, F. G., 2020. Practical methodology for the design of educational serious games. *Information (Switzerland)*, 11.

UnityTechnologies, 2020. Localisation (1.0.5). retrieved from https://docs.unity3d.com/packages/com.unity.localization.