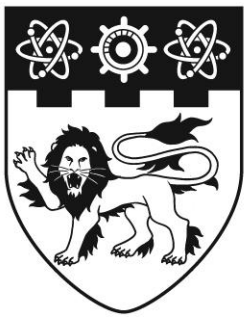


NANYANG TECHNOLOGICAL UNIVERSITY

GIRL CODE

Submitted by Navarro Jirah Marie Bathan
Matriculation Number: U1720524B

Supervisor: Kong Wai-Kin Adam

School of Computer Science and Engineering

A final year project report presented to the Nanyang Technological University
in partial fulfilment of the requirements of the degree of
Bachelor of Computer Science

2021

Table of Contents

1. Abstract

Engineering has been regarded as the most male-dominated position in the USA and even in Singapore. The culture that the Engineering field has established has to be masculine, manly, and male-centred. Since women place more emphasis on their personal lives and families, being an engineer is not for them (Hatmaker, 2013). When women participate in emotional and social work, women engineer's work efforts will disappear when they do not align with cultural norms (Fletcher, 1999). For example, in the fields of computer science and engineering, women are greatly underrepresented (CS&E, Ehrlinger, Ashby Plant, Hartwig, Vossen, Columb, Brewer, 2018).

As of 2019, according to NTU's (Nanyang Technological University) facts and figures, women still only represented one third of the Engineering cohort, Computer Science and Engineer in particular (Undergraduate Population by Gender AY2018-19 table by Nanyang Technological University). There needs to be a shift in perspective. Obviously, the required change of mentality must begin before they start choosing their courses in Polytechnics and even in University, such approach needs to be introduced in Primary and Secondary Schools. One way to support this kind of technical and heavy learning would be considering deploying digital games. According to research, this is what has been suggested to tailor the needs of society of the twenty-first century: entertainment titles that involves technical learning and practicality (Richard Sanford, 2014).

2. Introduction

This document is a game design report for an education game called Girl Code. It includes details of the design choices made for the game's key components. A working prototype of the game has been developed, even though it is not yet ready for release. The aim of this project is to empower women that they, too, can shine in this male domineering field. After all, we are entering the era of Artificial Intelligence where we place human intelligence in machines making them think like humans and imitate our actions, what better way to make use of a woman's complex mind.

3. Project Description

In this project, I will be developing a web-based educational game developed in Unity, Girl Code, that caters for young girls or even women who have no clue in programming. The game is a single player game that involves creativity and reading. The game includes a storyline of three women who are in different fields of engineering who face a dilemma on their program. Instead of typing the code manually, the players will only need to drop and drag the blocks to debug the whole program. The game is progressive and need not comprehensive tutorial for the player to get the idea. The concept of the game is to give women a glimpse of coding and what wonders it could do once it is compiled. Moreover, Girl Code also shows the reality of the profession of each engineer, by explaining what each field does, in a vibrant and user-friendly manner.

4. Target Audience

Girl Code's target audience would be typically females from the ages 7 to 16 years old. This is to raise awareness to girls about what an Engineering does in the means of coding and real-life situations.

5. Story

The story revolves around three women engineers that seems to be having problems in their fields. One of the engineers is a Robot Engineer. Her robot, Ginger, seems to be malfunctioning because she is not

programmed properly. The second engineer is a Sound Engineer. She produces music that sang by famous singers; however, her music seems to be mixed up with other instruments. Lastly, a Computer Engineer (specialized in Security) whose primary job is to protect the devices from physical to application attacks. The players will need to learn what viruses exists in the virtual world and how to protect their devices from such viruses.

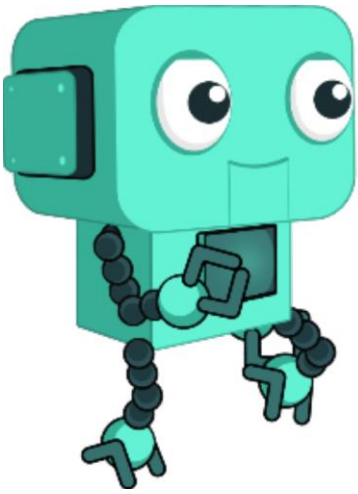
6. Aesthetics

The game's graphics are simple but vibrant. One of the earliest designs made for the game was, there would be no animation, it will stay as it is. However, to convey the message that Engineers make things work, the product must move. Hopefully, the visual style will appeal to young women and be encourage and learn at the same time what it's like to be an Engineer. Since the game is in 2D, the lack of image textures on most of them also reduces the minimum hardware requirements, which benefits potential players with older computers greatly. The game is made up of different Scenes where environments and obstacles will be placed.

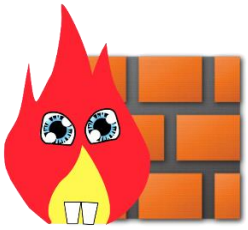
7. Characters

The Engineers and their assistants were all original and was made through Photoshop and Sprite2D. Girl Code includes Disney Characters to excite the players since Disney has a huge influence on children's imagination.

Character	Description
	Robot Engineer A Robot Engineer conduct research to find the most cost-efficient and safest process to manufacture their robotic systems. She will be the one instructing what to do in order to fix Tilly.

	Ginger Our Engineer's very own helpful robot who's in training. However, she's malfunctioning, and her program needs some fixing.
	Sound Engineer Sound engineers use machinery and equipment to record, synchronize, mix, or reproduce music, voices, or sound effects. In this case, she will instruct us what to do to produce music.
	Robo Band A virtual band created by our Sound Engineers. They played musical instruments skilfully. However, they can't seem to get the right tune to be able to sound good. The player needs to fix their program to be able to produce good music.

   	Guest Singers: Disney's Anna, Mogley, Rapunzel and Ariel Without the voice of these singers, our Sound Engineers will have difficulty in producing songs. And so, in this game we took advantage of their famous songs and remix and produce it.
	Computer Engineer Since there are a lot of fields in the computer engineering world, this game focuses in computer security. She will teach us about the protection of computer systems and information from harm, theft, and unauthorized use.

	Tilly She is our Computer Engineer's help. The player needs to program her for her to work. She will need to eat up the virus and at the same time, absorb defences in order to keep up.
  	Tilly's Defense 1.) Malwaria – also known as Malware helps to guard and look for any virus that's looking for trouble. 2.) Biometrix – also known as Biometrics Authentication who makes sure that only Tilly can use her device. 3.) Firewall-e – also known as the firewall, she protects Tilly in surfing the internet. The power of the three defences needs to be absorbed by Tilly to progress.
	Tilly's Enemies 1.) The Man – also known as Man In The Middle Attack whose skill is to snitch and steal significant things from Tilly 2.) Trojy – also known as Trojan Horse attack whose skill is to pretend to be Tilly's friend for Tilly to trust him but his true intent is to put Tilly in trouble 3.) Viria – also known as Generic Virus whose intent is to mess around with Tilly 4.) Wormie- also known as Computer Worm will replicate itself to shut Tilly's system down All these virus needs to be eaten by Tilly with the help of her defences.

8. Design and Implementation

i.) Design

Figure 1 Storyline of Robot Engineer

a.) Storylines

Girl Code has three stories that describe what usually happens in an Engineering life. This approach is important because it allows the player to be interested and involved in the game. It gives the game's events significance and aids the player in understanding what they must do.

Figure 2 Title Page of a Computer Engineer

b.) Point-And-Click

Girl Code uses Point-And-Click approach in reading storylines and answering quizzes. This approached is used because it does not require special execution skills, uses few system resources and requires no special devices.

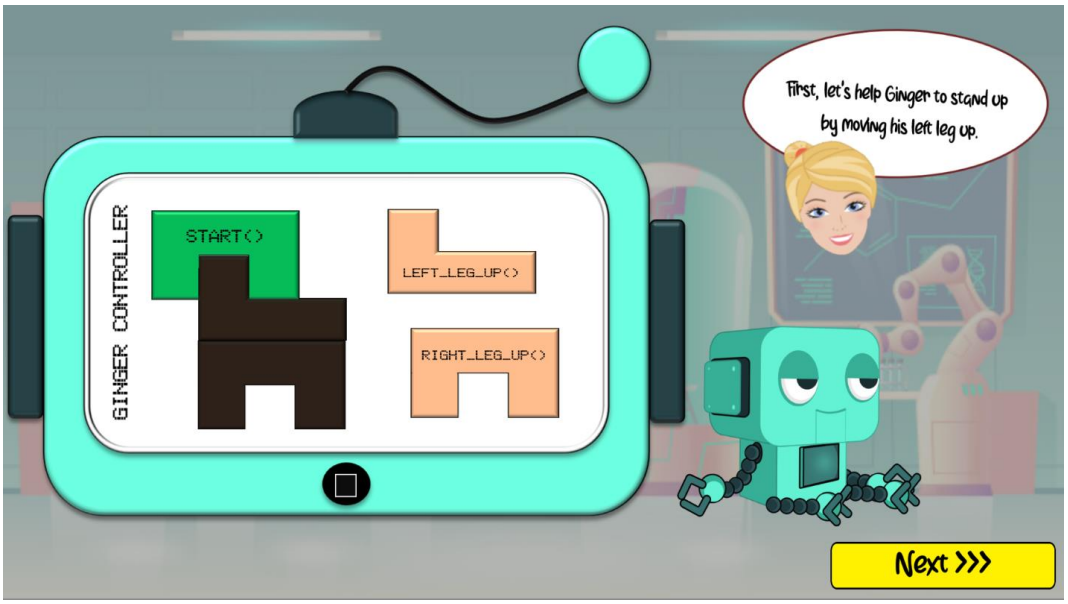


Figure 3 Demo of Drag and Drop in Robot Engineer Scene

c.) Drag and Drop

Girl Code uses Drag and Drop (DND) method whenever blocks need to be placed in order. The game has also adapted this interaction style to make the player feel like she is carrying the block itself. In addition, looking at the design of the user interface, children may have already expected that this is a DND interaction.

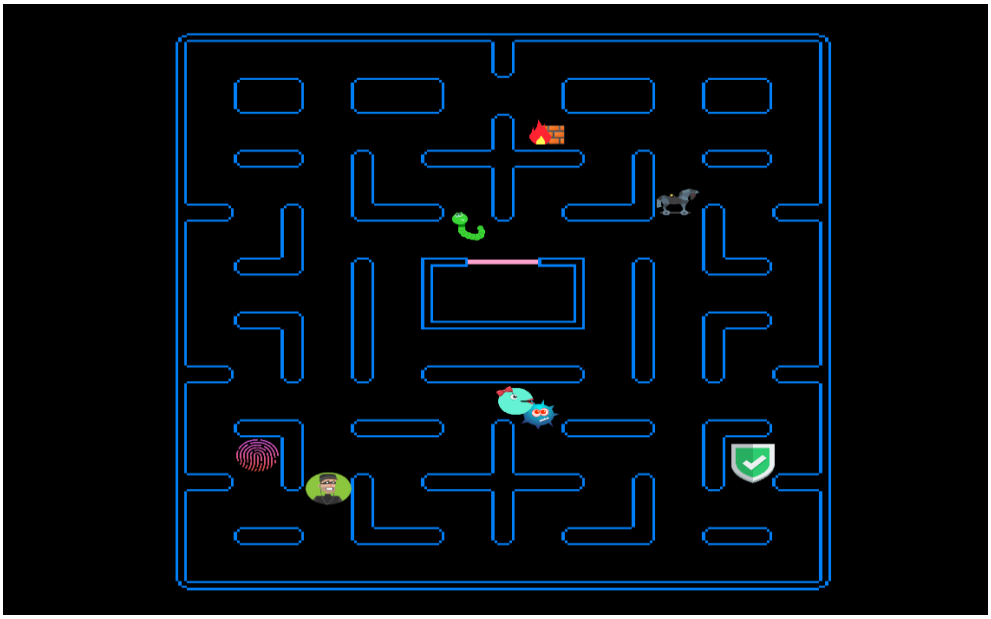


Figure 4 Scene when Tilly is at work

d.) Keyboard Controls

Girl Code's approach is to utilize all user interaction styles. This approach is used because the user controls should not just be limited using the mouse.

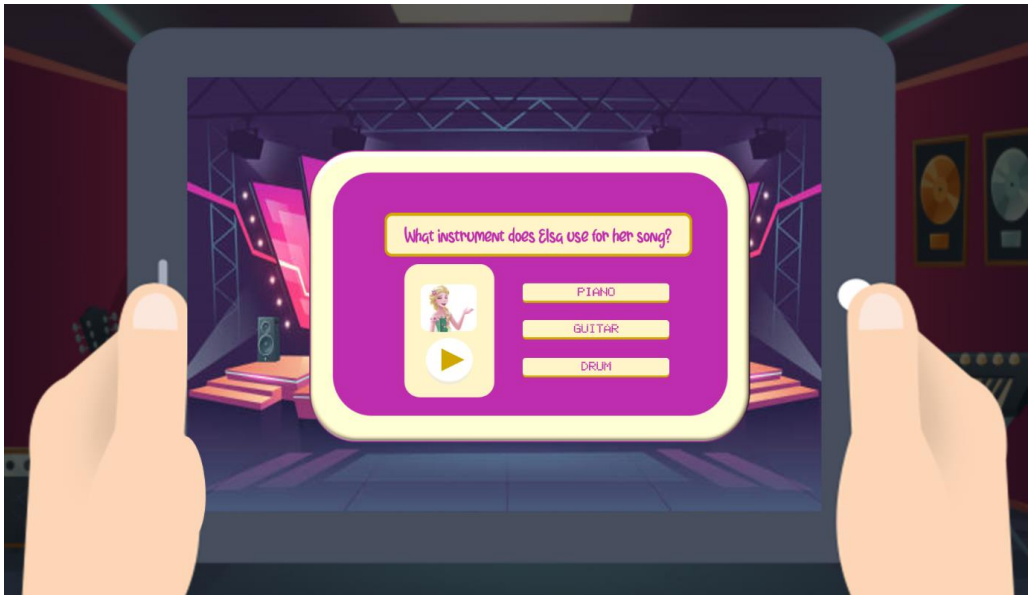


Figure 5 Quiz found in Sound Engineer's scene

e.) Quizzes

Only the Sound Engineer has this segment. This approach makes the player feel good and so , make them enthusiastic to play more. This scene tests the player's listening skill which is eseential in being a Sound Engineer.

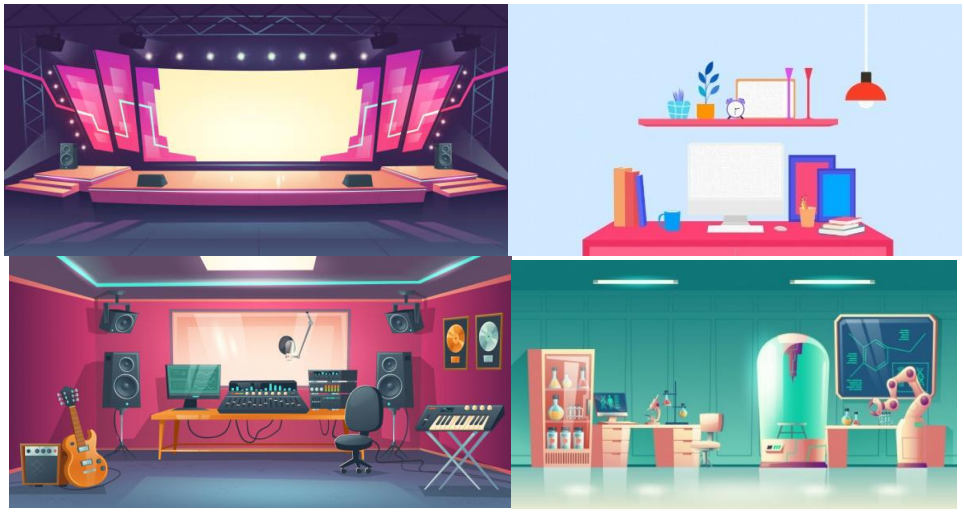


Figure 6 Environment Designs of the 3 Engineers

f.) Environment

The environment is based on the nature of the Engineers. Sound Engineer using a concert stage and music studio background. Robot Engineer used a laboratory background and a Computer Engineer uses a background of a desktop table.

ii.) Implementation

a.) Sample UI Mock up

Figure 7 Hi-fi designs of Girl Code

The initial design was to make the characters as Superheroes but after much evaluation, to maximise learning as well as to have fun, Girl Code's character turned to real-life occupations while maintaining vibrant colour.

b.) Sprites and Animation

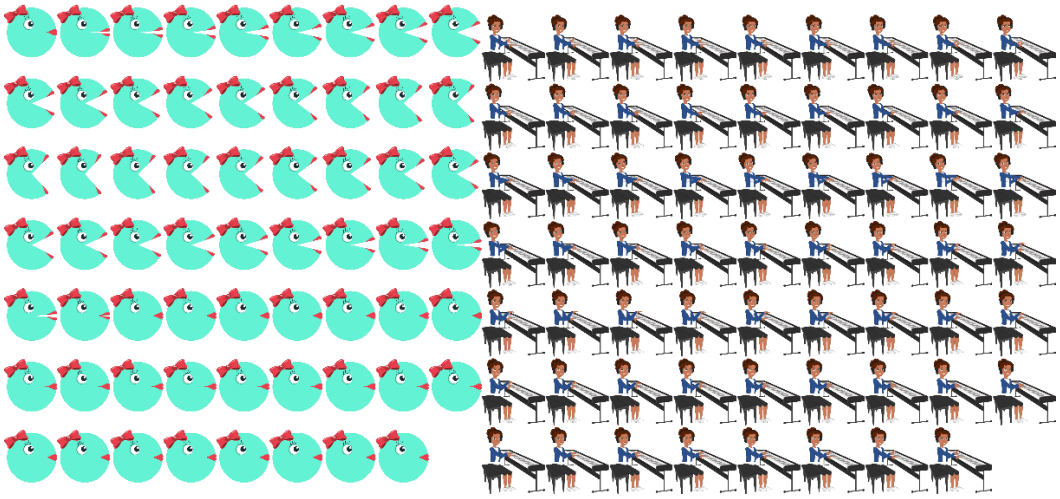


Figure 8 Sample of Sprites (Tilly and Robo Band's pianist)

Character sprites were made using photoshop and Sprite 2D. Each character that was made was very detailed as for animation to work smoothly.

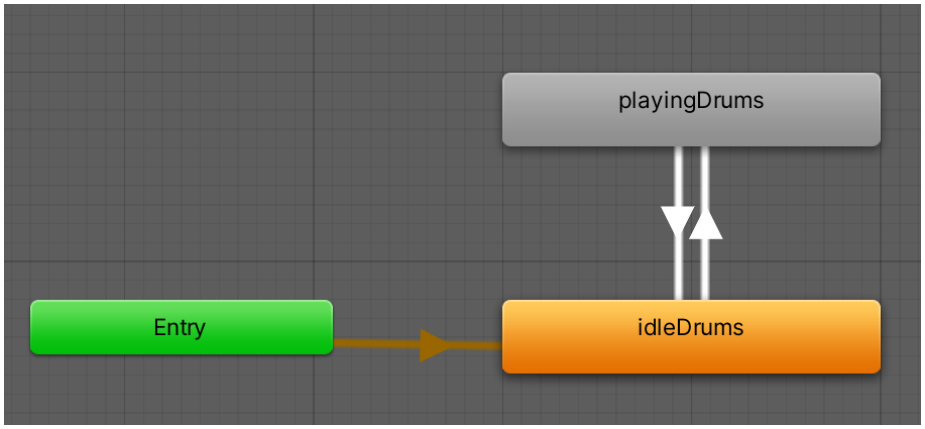


Figure 9 Animator for Robo Band's Drummer Girl

Each animation for individual character is carefully set so as to make the game more interactive and attractive. As seen in Figure 9, once the character is seen in the game, it will go to idleDrums which means it will not move until the player has enter a key. Once the player has entered the key, the animation for the drummer girl will start to play resulting to playingDrums to take into effect.

c.) Rigid Body and Box Collider

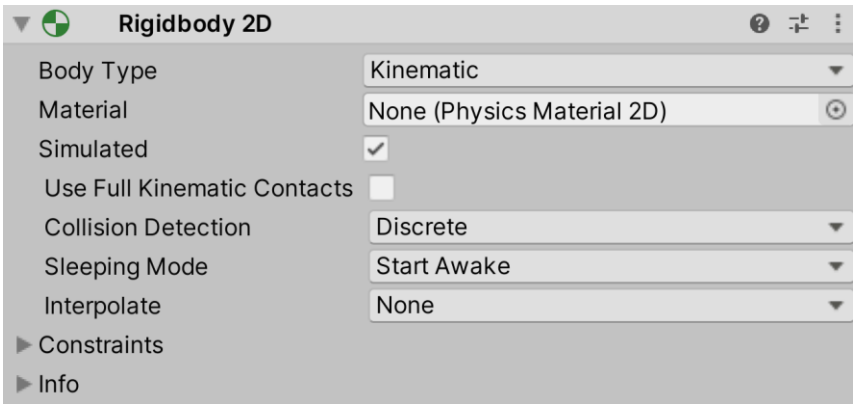


Figure 10 Rigidbody Setting for Demo Scene in Robot Engineer

Girl Code make use of the Rigidbody 2D component to make sure that the block stays in place and do not fall off the game screen. Not all objects need Rigidbody 2d, only those who needs to response in the drag-and-drop function.

Figure 11 and Figure 12 Box Collider 2D component and a Demo Scene in Robot Engineer

Girl Code make use of the component Box Collider 2D to handle physical collision of objects. In drag-and-drop 2 objects will collide as seen in Figure 12. Left Leg Up() needs to be placed in the darker block and when the two objects collide, Ginger will response to the collision. In this case, she will lift her leg up.

d.) Mathematical Formula

```
private void OnMouseDown()
{
    if(Input.GetMouseButtonDown(0))
    {
        Vector3 mousePos;
        mousePos = Input.mousePosition;
        mousePos = Camera.main.ScreenToWorldPoint(mousePos);

        startPosX = mousePos.x - this.transform.position.x;
        startPosY = mousePos.y - this.transform.position.y;

        moving = true;
    }
}

private void OnMouseUp()
{
    moving = false;

    if (Mathf.Abs(this.transform.position.x - correctForm.transform.position.x) <= 0.5f &&
        Mathf.Abs(this.transform.position.y - correctForm.transform.position.y) <= 0.5f)
    {
        this.transform.position = new Vector3(correctForm.transform.position.x, correctForm.transform.position.y, correctForm.transform.position.z);
        finish = true;
    }
    else
    {
        this.transform.position = new Vector3(resetPosition.x, resetPosition.y, resetPosition.z);
    }
}
```

Figure 13 Sample programming when the mouse is clicked and when the mouse is not clicked

Objects which usually needs to be drag-and-dropped has this function. What this function does is that it calculates the current position of the object. If the object is placed in the variable correctform, the object stays in that position. Otherwise, the object has to go back to its original position. This approach allows players to place the block in proper position and not leave the block anywhere else. For example, as seen in Figure 12, Left Leg Up() block needs to be placed in the darker block that has the same shape, if not, it will go back to its original position.

e.) OnTriggerStay & OnTriggerExit

```
private void OnTriggerStay2D(Collider2D collision)
{
    if (collision.CompareTag("b2"))
    {
        robot_appear.SetActive(true);
        robot_disappear.SetActive(false);
    }
}

private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.CompareTag("b2"))
    {
        robot_appear.SetActive(false);
        robot_disappear.SetActive(true);
    }
}
```

Figure 14 Sample Code of Demo in Robot Engineer Scene

Similar to above's implementation, these functions also can be found in a lot of drag-and-drop objects. What this function does is that when the Left Leg Up() block in Figure 12 touches the darker block which is the corretform, certain objects will turn active and certain objects will turn inactive. Also, once the Left Leg Up() block exits the darker block, correctform, certain objects will turn active and certain objects will turn inactive.

f.) Tilly's Movement

```
bool Valid(Vector2 direction)
{
    // cast line from 'next to pacman' to pacman
    // not from directly the center of next tile but just a little further from center of next tile
    Vector2 pos = transform.position;
    direction += new Vector2(direction.x * 0.45f, direction.y * 0.45f);
    RaycastHit2D hit = Physics2D.Linecast(pos + direction, pos);
    return hit.collider.name == "energizer" || (hit.collider == GetComponent<Collider2D>());
}
```

Figure 15 Code of Tilly's Direction

```
void ReadInputAndMove()
{
    // move closer to destination
    Vector2 p = Vector2.MoveTowards(transform.position, _dest, speed);
    GetComponent<Rigidbody2D>().MovePosition(p);

    // get the next direction from keyboard
    if (Input.GetAxis("Horizontal") > 0) _nextDir = Vector2.right;
    if (Input.GetAxis("Horizontal") < 0) _nextDir = -Vector2.right;
    if (Input.GetAxis("Vertical") > 0) _nextDir = Vector2.up;
    if (Input.GetAxis("Vertical") < 0) _nextDir = -Vector2.up;

    // if pacman is in the center of a tile
    if (Vector2.Distance(_dest, transform.position) < 0.00001f)
    {
        if (Valid(_nextDir))
        {
            _dest = (Vector2)transform.position + _nextDir;
            _dir = _nextDir;
        }
        else   // if next direction is not valid
        {
            if (Valid(_dir))  // and the prev. direction is valid
                _dest = (Vector2)transform.position + _dir;   // continue on that direction

            // otherwise, do nothing
        }
    }
}
```

Figure 16 Code of Tilly's Movement

In this code, there is a need to find out if Tilly can move into a certain direction of if there is a wall. Using RaycCastHit2D, a line is cast to find out if there is a wall at the top of Tilly. If it hit Tilly then there was nothing in-between, otherwise there must have been a wall. If Tilly has collided with the energizer, with A key being pressed, the health point will increase 100 points.

The function ReadInputAndMove() found in Figure 16 initialize Tilly's movement. Vector2.right means Tilly will move right while -Vector2.right means Tilly will move left. Similarly, for the up and down keys.

g.) AI of Tilly's Enemies

```
public void AILogic()
{
    // get current tile
    Vector3 currentPos = new Vector3(transform.position.x + 0.499f, transform.position.y + 0.499f);
    currentTile = tiles[manager.Index ((int)currentPos.x, (int)currentPos.y)];

    targetTile = GetTargetTilePerGhost();

    // get the next tile according to direction
    if(ghost.direction.x > 0)   nextTile = tiles[manager.Index ((int)(currentPos.x+1), (int)currentPos.y)];
    if(ghost.direction.x < 0)   nextTile = tiles[manager.Index ((int)(currentPos.x-1), (int)currentPos.y)];
    if(ghost.direction.y > 0)   nextTile = tiles[manager.Index ((int)currentPos.x, (int)(currentPos.y+1))];
    if(ghost.direction.y < 0)   nextTile = tiles[manager.Index ((int)currentPos.x, (int)(currentPos.y-1))];

    if(nextTile.occupied || currentTile.isIntersection)
    {
        //---------------------
        // IF WE BUMP INTO WALL
        if(nextTile.occupied && !currentTile.isIntersection)
        {
            // if ghost moves to right or left and there is wall next tile
            if(ghost.direction.x != 0)
            {
                if(currentTile.down == null)    ghost.direction = Vector3.up;
                else                            ghost.direction = Vector3.down;

            }

            // if ghost moves to up or down and there is wall next tile
            else if(ghost.direction.y != 0)
            {
                if(currentTile.left == null)    ghost.direction = Vector3.right;
                else                            ghost.direction = Vector3.left;

            }

        }
    // IF WE ARE AT INTERSECTION
    // calculate the distance to target from each available tile and choose the shortest one
    if(currentTile.isIntersection)
    {

        float dist1, dist2, dist3, dist4;
        dist1 = dist2 = dist3 = dist4 = 999999f;
        if(currentTile.up != null && !currentTile.up.occupied && !(ghost.direction.y < 0))      dist1 = manager.distance(currentTile.up, targetTile);
        if(currentTile.down != null && !currentTile.down.occupied &&  !(ghost.direction.y > 0))     dist2 = manager.distance(currentTile.down, targetTile);
        if(currentTile.left != null && !currentTile.left.occupied && !(ghost.direction.x > 0))  dist3 = manager.distance(currentTile.left, targetTile);
        if(currentTile.right != null && !currentTile.right.occupied && !(ghost.direction.x < 0))    dist4 = manager.distance(currentTile.right, targetTile);

        float min = Mathf.Min(dist1, dist2, dist3, dist4);
        if(min == dist1) ghost.direction = Vector3.up;
        if(min == dist2) ghost.direction = Vector3.down;
        if(min == dist3) ghost.direction = Vector3.left;
        if(min == dist4) ghost.direction = Vector3.right;

    }
}

// if there is no decision to be made, designate next waypoint for the ghost
else
{
    ghost.direction = ghost.direction;  // setter updates the waypoint
}
}
```

Figure 17 Algorithm for Tilly's Enemies

Tilly's enemies move at a random direction. However, the game has to make sure that that random direction make sense. For example, Figure 17 shows the code when Tilly's enemies bump the wall. The game has to consider the placement of the walls and the enemies should know how to react to it. Similarly, for intersections, the enemies should be able to choose a tile that is available and shortest to achieve.

9. Gameplay Mechanics

i.) Robot Engineer

Progression

Progression takes place when player reads the storyline about the dilemma of Girl Code's Robot Engineer. After that, she will start explaining what needs to be done to solve her dilemma. In her explanation, she stated that her dilemma is that her robot, Ginger, is malfunctioning and the player needs to fix her by 'reprogramming' her.

The scene progressed to an explanation of functions and then a demo on how we can reprogram her. The player needs to drag and drop the code blocks in sequence to be able to fix her. Robot Engineer also explained the use of while function in terms of eating analogy that while the food is not finished, Ginger will not stop eating. Lastly, the objective of the last scene is to teach the player of if-else function. It uses the analogy that if Ginger has reached the flagpole, she should stop running. Otherwise, she keeps running.

Rewards

Finishing each coding scene will take you to another scene with different challenge until you have succeeded in helping Girl Code's Robot Engineer.

ii.) Sound Engineer

Progression

Progression takes place when player reads the storyline about the dilemma of Girl Code's Sound Engineer. After that, she will start explaining what needs to be done to solve her dilemma. In her explanation, she stated that her dilemma is that her virtual band, Robo Band, is malfunctioning and the player needs to fix them by 'reprogramming' her.

The scene progressed by showing the player how messed up the program of the Robo Band. She then led the player with the explanation of what is a function in computer programming and how will programming would help the Robo Band.

Next, Girl Code's Sound Engineer would introduce singers that needs Robo Band's talents. The game progressed in a quiz segment to test the listening skill of the player. A music will be played, and the player must guess what instrument was used in the song. There will be right and wrong answers. The IF function will then be introduced in an analogy of when a particular singer sang, this instrument will be played. The player needs to drag and drop the code blocks in sequence to move to the next scene.

Lastly, the objective of the last scene is to teach while function using the analogy that while Ariel is singing, the music, which the player remixed, will play. This scene uses a drag-and-drop interaction.

Rewards

The player must get the quiz scene correct to progress to the next level. Also, the player needs to remember what the singer's instruments in the IF-ELSE scene were because at the end of the quiz, the singers will have a concert and if the player coded the singer with a wrong instrument block, the concert is ruined, and the player needs to reprogram the Robo Band.

iii.) Computer Engineer

Progression

Progression takes place when player reads the introduction of what a Computer Engineer does. She then introduced her friend named Tilly who will help to protect our devices. Next, she explained what a function does and how we will use it to instruct Tilly.

Girl Code's Engineer started introducing viruses that should be avoided and defences that needs to be kept. After that, using the drag-and-drop approach, code blocks need to be dragged and dropped to program Tilly. We programmed Tilly by using the IF-ELSE function that if up key is pressed, Tilly will be moved up. If the down key is pressed, Tilly will move down. If left key is pressed, Tilly will move left and lastly, if right key is pressed, Tilly will move right. In addition, we also teach the player how the WHILE function work. While the A key is pressed hold, Tilly will absorb her defences to protect her from the virus and while the E key is hold, she will then eat the viruses that tried to attack her.

Rewards

Every time Tilly absorbs her defences 100 points is earned and every time Tilly eats the virus, 100 points is taken away. Tilly needs to a minimum of 100 points to eat the virus. After eating all the virus, the player has successfully helped Girl Code's Computer Engineer.

10. Testing, Results and Evaluation

Functionality	**Result**
Check whether the game immediately starts up when the Girl Code icon is clicked	PASS
Check the game size corresponds to the screen size	PASS
Background	
Check whether the background is not covering the characters	PASS
Scene Progression	
Check whether the sequence of the scenes are correct and in order	PASS
Point and Click	
Check whether the buttons can be clicked in all the scenes	PASS
Drag and Drop	
Check whether the blocks can be drag and dropped in the proper place	PASS
Check whether the blocks can not be dragged out of the game	PASS
Keyboard Controls	
Check whether only keys P, D, G are working for the Sound Engineers scene	PASS
Check whether only UP arrow, DOWN arrow, LEFT arrow, RIGHT arrow, key "A" and key "E" are working for the Computer Engineer scene	PASS
Robot Eating Sequence	
Check whether the correct sequence is the following: 1.) Sequence 1 a.) Open Mouth, b.) Get Food, c.) Put Food In The Mouth, d.) Chew Food 2.) Sequence 2 a.) Get Food, b.) Open Mouth,	PASS

c.) Put Food In The Mouth, d.) Chew Food	
Check whether if first input block of "Chew Food" is placed, an error message of "You cannot chew the food without opening the mouth and without food" is shown	PASS
Check whether if first input block of "Put Food In Mouth" is placed, an error message of "You cannot put food in your mouth without opening it and without getting food" is shown	PASS
Check whether if first input block is "Open Mouth" and the second block is "Chew Food" is placed, an error message of "You cannot chew the food without getting and putting food in your mouth" is shown	PASS
Check whether if first input block is "Get Food" and the second block is "Chew Food" is placed, an error message of "You cannot chew the food without opening and putting food in your mouth" is shown	PASS
Check whether if first input block is "Open Mouth", the second block is "Get Food" and the third block is "Chew Food" is placed, an error message of "You cannot chew the food without putting food in your mouth" is shown	PASS
Check whether if first input block is "Get Food", the second block is "Open Mouth" and the third block is "Chew Food" is placed, an error message of "You cannot chew the food without putting food in your mouth" is shown	PASS
Quiz Check	
Check whether the player is alerted if the answer is wrong	PASS
Check whether the scene does not progress if the answer is wrong	PASS
Check whether a sound is produced for right and wrong answer	PASS
Music Check	
Check whether if Elsa is singing, piano must be played.	PASS
If the above condition passed, audio elsaQuizSing must be played	PASS
Check whether if Rapunzel is singing, guitar must be played.	PASS
If the above condition passed, audio rapunzelQuizSing must be played	PASS
Check whether if Mogley is singing, drum must be played.	PASS
If the above condition passed, audio mogleyQuizSing must be played	PASS
If Elsa is singing but the user put guitar block as her instrument, EI audio is played	PASS
If Elsa is singing but the user put drum block as her instrument, EW audio is played	PASS
If Rapunzel is singing but the user put piano block as her instrument, RL audio is played	PASS
If Rapunzel is singing but the user put drum block as her instrument, RW audio is played	PASS
If Mogley is singing but the user put guitar block as her instrument, MI audio is played	PASS
If Mogley is singing but the user put piano block as her instrument, ML audio is played	PASS
Check whether if the player chooses EURO sound first and EURO sound for the second, audio V2C23 is played	PASS
Check whether if the player chooses EURO sound first and POP sound for the second, audio V2C21 is played	PASS
Check whether if the player chooses POP sound first and POP sound for the second, audio V2C11 is played	PASS
Check whether if the player chooses POP sound first and EURO sound for the second, audio V2C12 is played	PASS
Check whether if the player chooses TRAP sound first and EURO sound for the second, audio V2C42 is played	PASS
Check whether if the player chooses TRAP sound first and EURO sound for the second, audio V2C43 is played	PASS
Tilly's Gameplay	
Check whether UP arrow is pressed only once for Tilly to continuously move up	PASS
Check whether DOWN arrow is pressed only once for Tilly to continuously move down	PASS

Check whether LEFT arrow is pressed only once for Tilly to continuously move left	PASS
Check whether RIGHT arrow is pressed only once for Tilly to continuously move right	PASS
Check whether "A" key must be pressed and hold for Tilly to absorb the defences	PASS
Check whether "E" key must be pressed and hold for Tilly to eat the viruses	PASS
Check whether the health points increase by 100 if Tilly absorbs the defences	PASS
Check whether the health points decrease by 100 if Tilly eats the virus	PASS
Block Naming	
Check whether all the blocks are named correctly according to their functions: 1.) Robot Engineer a.) Demo – Start, Left Leg Up, Right Leg Up b.) While – While, Food Is Not Finished, Open Mouth, Get Food, Put Food In Mouth, Chew Food c.) If/Else – If/Else-If, Ginger Have Not Reach The Flag, Keep Running, Ginger Have Reach The Flag, Stop Running 2.) Sound Engineer a.) Elsa Quiz – Start, If, Elsa Singing, Play Piano b.) Rapunzel Quiz- Start, If, Elsa Singing, Play Piano, Else-If, Rapunzel Singing, Play Guitar c.) Mogley Quiz - Start, If, Elsa Singing, Play Piano, Else-If, Rapunzel Singing, Play Guitar, Else-If, Mogley Singing, Play Drum d.) Ariel Remix- Start, While, Ariel Is Singing, Play Trap, Play Euro, Play Pop, Play Euro, Play Pop 3.) Computer Engineer a.) Tilly's Programming – Start, If Left Arrow Pressed, Move Left, Else-If Right Arrow Pressed, Move Right, Else-If Up Arrow Pressed, Move Up, Else-If Down Arrow Pressed, Move Down, While, 'A' Pressed, Absorb Defences, While, 'E' Pressed, Eat Virus	PASS

11. Conclusion

Women are underrepresented in the technology sector in general. To close the gender gap, it's important to get girls interested in these fields and aware of their potential possibilities. Ultimately, women of all ages feel most inspired when they see other women who are like them and easy to connect to and Girl Code make sure of that. The aim of this project is not just to empower girls but also to introduce to them the power of coding using blocks and real-life situations of the three Engineers. Going forward, we will continue to build and scale the game with Visual Reality (VR). In addition to enhance the game, we will make it available from children, no matter the gender, ages 2-5 years old.

12. Reference

[1] R. Sandford, "Teachers as game designers: using a game in formal learning in a Singapore primary school," Educational media international, vol. 51, no. 1, pp. 66-78, 2014, doi: 10.1080/09523987.2014.889410.

[2] D. M. Hatmaker, "Engineering Identity: Gender and Professional Identity Negotiation among Women Engineers," Gender, work, and organization, vol. 20, no. 4, pp. 382-396, 2013, doi: 10.1111/j.1468-0432.2012.00589.x.

[3] J. Ehrlinger et al., "Do Gender Differences in Perceived Prototypical Computer Scientists and Engineers Contribute to Gender Gaps in Computer Science and Engineering?," Sex roles, vol. 78, no. 1, pp. 40-51, 2018, doi: 10.1007/s11199-017-0763-x.

[4] R. Cantuni, Designing Digital Products for Kids: Deliver User Experiences That Delight Kids, Parents, and Teachers. Berkeley, CA: Apress L. P, 2020.

[5] F. S. Becker, "Why don't young people want to become engineers? Rational reasons for disappointing decisions," European journal of engineering education, vol. 35, no. 4, pp. 349-366, 2010, doi: 10.1080/03043797.2010.489941.

[6] M. C. Solal, "Change The Game," in Celebrating and empowering women in mobile gaming with CHANGE THE GAME, ed. The Keyword: Google, 2017.

[7] R. C. J. Musker, "The Little Mermaid," ed. United States: Buena Vista Pictures, 1989, p. 83 minutes.

[8] A. Menken, "Part Of Your World," in The Little Mermaid: An Original Walt Disney Records Soundtrack, ed. United Sates: Walt Disney, 1989.

[9] Guto Collector, "Ariel," c. 2018, Italy [Online]. Available https://twitter.com/gutocollector/status/1051929412248186881 on: October 16, 2018.

[10] N. G. B. Howard, "Tangled," ed. United States: Walt Disney Studios Motion Pictures, 2010, p. 100 minutes.

[11] A. M. G. Slater, "I See The Light," in Tangled: Original Soundtrack, ed. United States: Wonderland Music Company/ Walt Disney Music Company, 2010.

[12] W. Reitherman, "The Jungle Book," ed. United States: Buena Vusta Distribution, 1967, p. 78 minutes.

[13] R. M. S. R. B. Sherman, "I Wanna Be Like You," in The Jungle Book, ed. United States, 1967.

[14] C. B. J. Lee, "Frozen," ed. United States: Walt Disney Studios Motion Pictures, 2013, p. 102 minutes.

[15] K. A.-L. R. Lopez, "Let It Go," in Frozen: Original Motion Picture Soundtrack, ed. United States: Wonderland Music Company, 2013.

[16] Pacman. (1980). JP: Namco & NA: Midway Games, Japan.

[17] L. Faust, "DC Super Hero Girls," L. Faust, Ed., ed. United States: Warner Bros. Television Distribution, 2019, pp. 11-15 minutes.