Game Design Report

Nightmare Lord

JUNE 6

SIT254- Assignment 2

Submitted By:

Nina Rae Laminero
Ngo Cong Thanh Nguyen
Timothy Thian
Viet Nam Nguyen

Content

1 Project Vision, Scope and Goal

1.1 Summary

Nightmare Lord is an arcade game that will bring a real challenge and competition to mobile gaming. Players must survive throughout the level to advance to the next. The Game engages the players to be strategic to avoid contact with the enemy, which can result in losing. Players will have to think fast and be observant of every detail.

1.2 Vision

The projected vision for Nightmare Lord is that not only will it relieve the stress for people of all ages, but also, enhance people's eye reflexes and detail orientation when played frequently. In specific businesses, having these skills would help to reduce mistakes and, therefore, lesser customer complaints.

1.3 Scope

Nightmare Lord will excite and create a fun space for people who maybe need to kill time or relieve stress. This project will consist of creating a 2D and mobile arcade game. The project will be completed by the 6th of June, 2020. The success of the Game will be determined if the Game is completed and working and if it makes people play the Game again. A prototype is produced first to enable discussions before finalizing development. Below are the required features of a prototype.

- The user interface for the first and second screen
- Enemy/health random spawn
- Health bar
- Level
- Enemies movement
- Character movement
- The collision of enemy and character

Some features cannot be developed due to timeframe and therefore need to be left out. These features will not change the gameplay, and some added explicitly to sell the Game effectively. It is clarified that the Game will still be complete and working without these features.
Below is a list of features.

- Social media leaderboard
- Extra feature: Item (Bomb) to wipe up enemy

1.4 Objectives

With the given timeline, the goal is to have a complete working game that can be playtested, completed two worksheets, and a report.

2 Team Members

Nina Rae Laminero
ID: 219394956
Designer, Game Tester
GDD, Worksheet 1

Ngo Cong Thanh Nguyen
ID: 219394956
Scrum Master, Programmer
IMP, Code for Worksheet 2

Timothy Thian
ID: 218401053
Programmer

Viet Nam Nguyen
ID: 220381288
Sound Designer, Game Tester, GDD, Worksheet 2

GDD Writing Contributions:

Ngo Cong Thanh Nguyen: Code for Probability worksheet

Viet Nam Nguyen: 4,5,7,9,10, Probability Worksheet, Internal economy Worksheet = 50%

Nina Rae Laminero: 1,2,3,4,6,7,8,11, Internal Economy Worksheet = 50%

3 Key Requirements Summary

3.1 Design

Nightmare Lord will excite and create a fun space for people who maybe need to kill time or relieve stress. This project will consist of creating a 2D and mobile arcade game. The project will be completed by May 29, 2020. The success of the Game will be determined if the Game is completed and working, fun and if it makes people play the Game again. A prototype is produced first to enable discussions before finalizing development. Below are the required features of a prototype. Nightmare lord's theme is a horror to enhance the feel of the Game. The Game is designed to make players feel like the game is their Nightmare. Everything in the game is black and white to imitate a scary environment.

- The user interface for the first and second screen
- Enemy/health random spawn
- Health bar
- Timer
- Enemies movement

- Character movement
- The collision of enemy and character

3.2 Technology

Using Unreal Engine to develop a prototype of Nightmare lord.
Microsoft Word to develop GDD.
Trello for Project time management
Slack and Microsoft Teams to communicate.

3.3 People

Programmer, Designer, and Tester are the three roles required to complete the project expertly.

3.4 Timeframe

With the given timeframe of 4 weeks, considering students have other units and priorities, it is projected that the team will develop a working game before the due date with specific conditions. As the group only consists of 3 members, it would be impossible if leaderboard and extra power-ups are developed.

4 Concept Summary

4.1 Title

Nightmare Lord

4.2 Gameplay Hook

Arcade games are one of the prototypes of games to be introduced in the market. The game concept of staying alive while trying to achieve the highest score is one of the oldest mechanics used to this day in most modern games. The novelty of Nightmare Lord is the endless amount of levels if the player can survive and meet the required goals. Also, Nightmare Lord's leaderboard leverages the competitive nature of players, by providing a display of high scores to compare their performance.

4.3 Genre

Arcade game

4.4 Theme

Nightmare horror-themed

4.5 Platform

Nightmare Lord will be presented on Mobile devices such as smartphones and tablets, either Android or iOS. The Game will require touch-screen feature to play Nightmare Lord leaderboard

leverages the competitive nature of players, by providing a display of high scores to compare their performance.

5 Target Audience and Market Analysis

Nightmare Lord's arcade-like gameplay and design(graphics) allow the targeted audience to be vast, ranging from children to adults, but also parent/guardians of the children playing. Other popular games such as Fruit Ninja, Asteroids, and Ant Smash (which has similar characteristics) have a player base ranging from children to adults. DAR also shows evidence that 90% of parents will play with their children.

The game solely relies on touch-screen technology to enable play of the game, allows the targeted market to be iOS and Android devices. Currently standing, 67.95% of the people around the world uses a mobile device. These numbers indicate that the platform for the game markets best would be mobile devices.

6 Game Setting

Nightmare Lord is set in a dark place where scary creatures are present. This place represents the Nightmare of the player.

7 Gameplay

7.1 Rules and mechanics

7.1.1 Core Game Play

- The player must not allow the minions to collide with their character.
- The player is only allowed movement inside the game grid.
- The player must not lose all five hearts to progress to the next level.

7.1.2 Gameplay elements

- Player goals- To avoid contact with the enemy and lose all five hearts
- Rules- The player cannot go beyond the border
- Challenges- Get the highest score

7.1.3 Goal and constraints

The player's goal is to achieve the highest score. The player gains score by tapping on minions/enemies. Each enemy has specific points.

1. Skull -1 points
2. Ghost- 2 points
3. Clown- 3 points
4. Demon- 4 points

The enemy spawn is random and in higher levels, the probability of having demon or clown spawning in the game space is low.
Another constraint is having only 5 health slots. The player cannot claim more than 5 health.
Lastly, the player will not be able to drag their character out the boarder to logically avoid being hit by monsters.

7.1.4 Power-ups

The heart will increase the player's health by one. The probability of this object spawning in the game space is 25% every 15 seconds. Hence why it will increase the aspect of the game being challenging.

7.1.5 Game over

When the player loses all their health (displayed on the top left corner), or player was unable to eliminate enemies within the one-minute timer, the game will end. Their score will show to approve that they have achieved the specific rating. The Game will ask them to play again to encourage players to keep going back in the Game.

7.2 Worksheets

7.2.1 Feedback and Internal Economy Worksheet

7.2.2 Probability and Utility Worksheet

8 Art Design

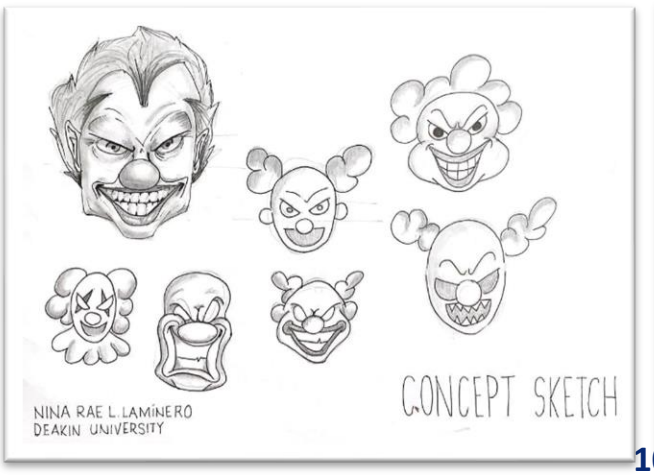
8.1 Visual Assets

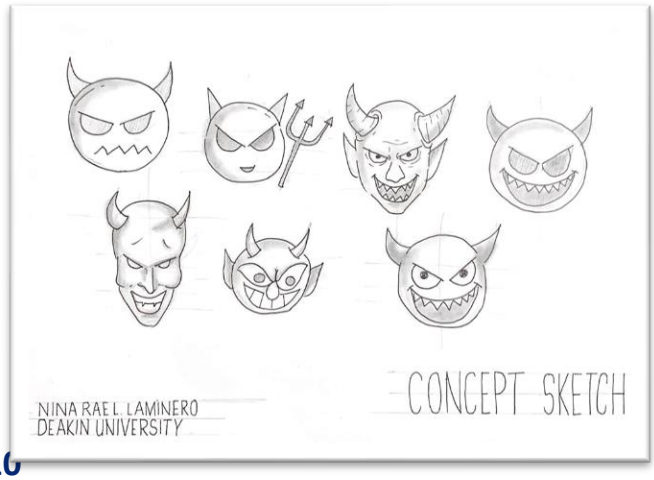
- Character- Players would be able to choose colors on the start screen. The character is used in the gameplay.
- Game enemies- The idea of the design is cute and scary to appeal to the players and make the game look aesthetic. They will spawn randomly throughout the Game.
- Heart asset- Designed like the health bar, to make it more understandable for young players.
- Start Button- Seen on the start screen. Clickable. (Text color- White, Font- Tekton Pro)
- Timer text- Displayed above the game grid at the center. The level changes every minute. (Text color- White, Font- Tekton Pro)
- Health bar- Displayed on the top left side above game space.
- Score text- Displayed on the top left side above game space. (Text color- White, Font- Tekton Pro)
- Level text- Displayed on the center of the screen after 1 minute if the player can proceed to the next level. (Text color- White, Font- Tekton Pro Bold)
- Game space background- #212626 (Midnight blue)
- Game background border- #ffffff (White)
- Start screen background- #212626 (Midnight blue)

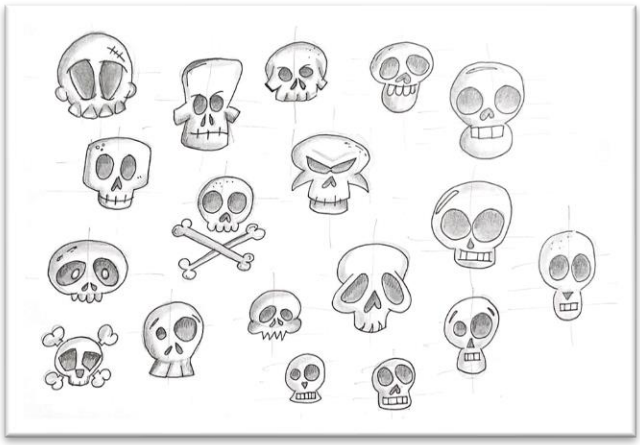
8.2 Visual Direction

The visual look of assets in the Game is horror-themed to intensify the nightmare feeling. Cartoon assets and texts will support the idea of having children play Nightmare Lord. With these features present in the Game, Nightmare Lord will greatly appeal to people of all ages.

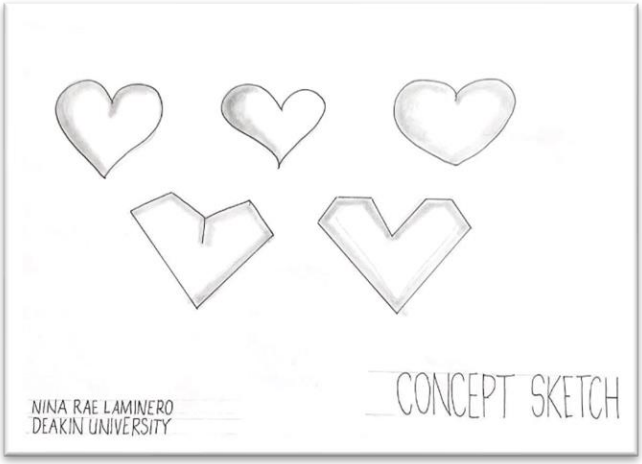
8.3 Concept Art

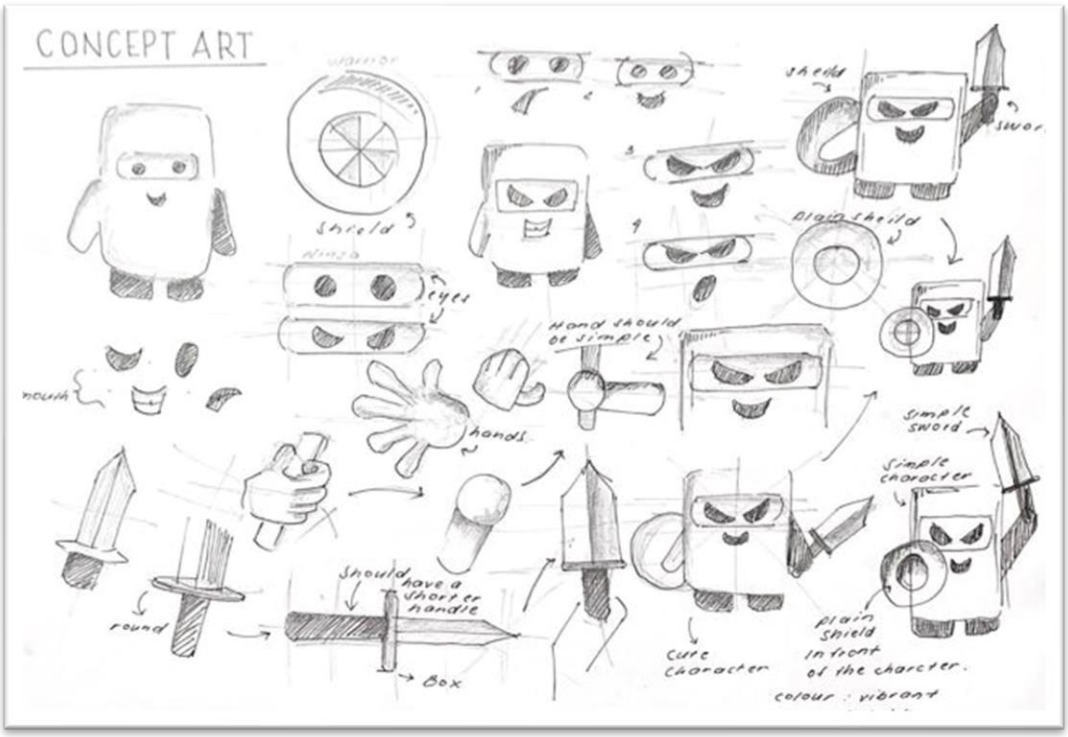







GHOST
CONCEPT SKETCH


NINA RAE LAMINERO
DEAKIN UNIVERSITY
CONCEPT SKETCH

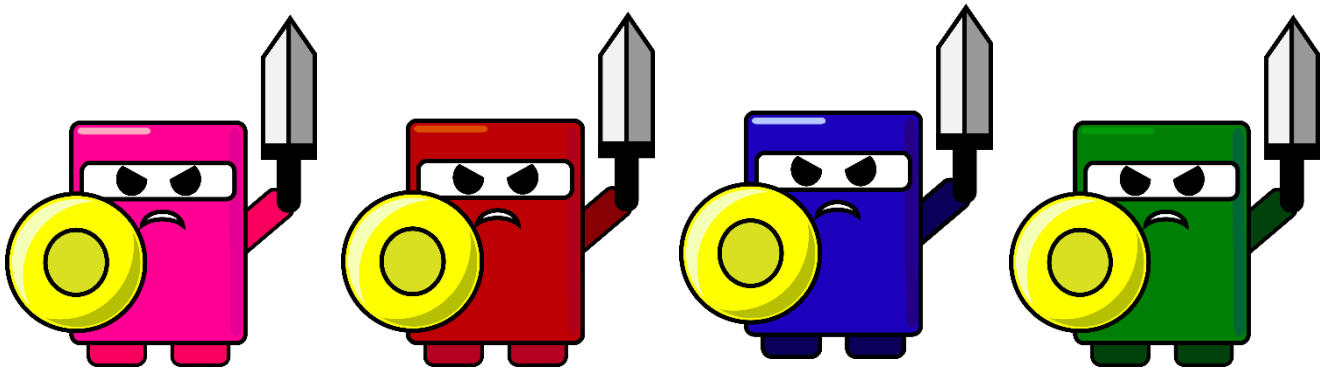

CONCEPT ART
shield
eyes
mouth
hands
Hand should be simple
Should have a shorter handle
round
Box
Cute character
Simple character
Simple sword
Plain shield in front of the character.
colour : vibrant

8.4 Style Guides

- All fonts are white and Tekton Pro Bold Style except for texts on images.
- All fonts should be readable and consistent.
- Start screen background- Plain Black.
- The abstract idea is that the Game is kept in simple colors (Black and white) except for characters to illustrate that the player overpowers their Nightmare.

8.5 Game Characters

The player would be able to choose game characters in the start screen. Ninjas and warriors inspire the design of the game character, hence why the character has an eye mask, shield, and sword. The player would be able to choose between 4 colors for their character. (Pink, Red, Blue, and Green) This game feature helps the player relate with the Game and creates freedom for players by choosing their character.

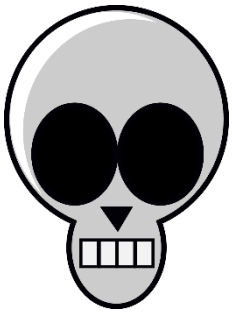


8.6 Game Enemies

EVIL CLOWN
D: 1220 x 1114
4 POINTS

DEMON
D: 1013 x 1192
3 POINTS

GHOST
D: 1011 x 1303
2 POINTS



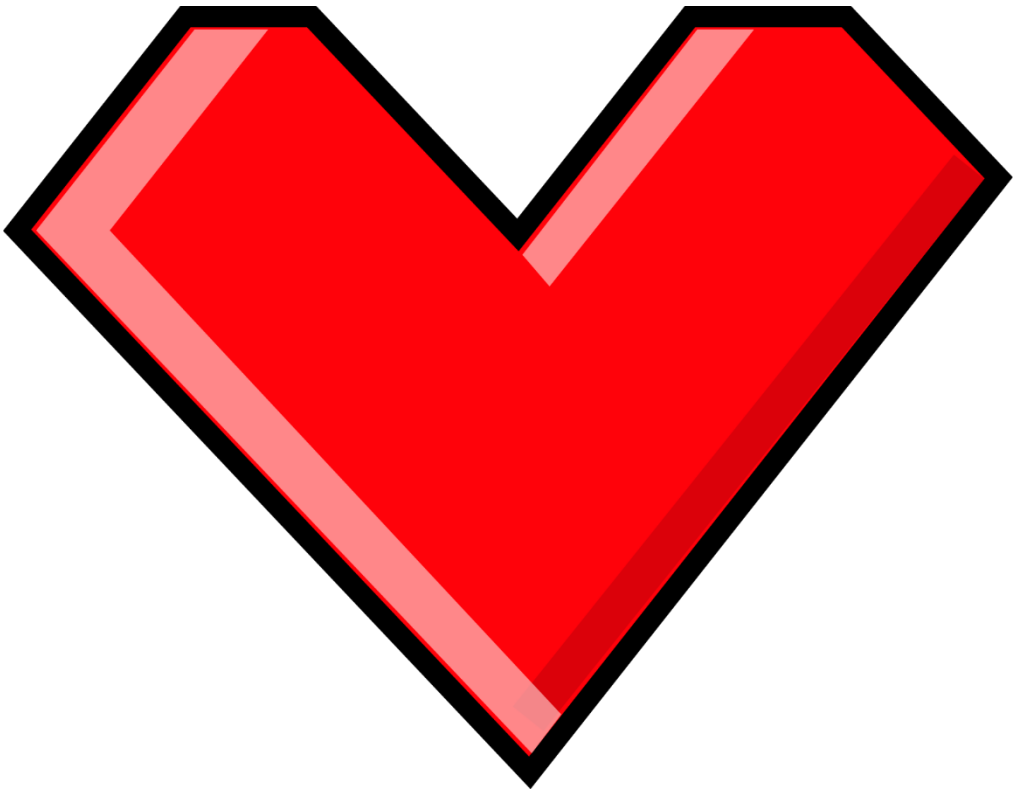
SKULL
D: 940 x 1270
1 POINT

These assets will spawn randomly throughout the Game. They are designed to look aesthetic, scary, and cute to appeal to the audience effectively. Like the character asset, enemy assets also followed the cartoon-based style and avoided the realistic and detailed look. The clown is worth 4 points, and demons are worth 3 points, hence why in the Game they are smaller in size than ghost (2 points) and skull (1 point). This feature intensifies the difficulty to get a higher score. Like any other game, Nightmare Lord implements the game mechanic of having the most top giving utility, the most difficult to catch.

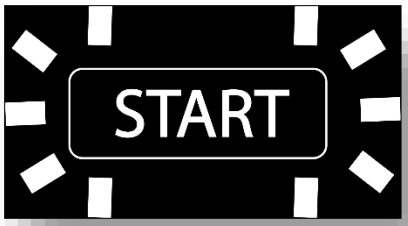
8.7 Heart Asset

The heart will increase the player's health by one and will spawn in the game space randomly. The idea is to keep assets consistent with the cartoon theme, hence why the heart asset is designed as it is.

Also, five hearts will represent the player's health. This will be displayed on the left side corner of the screen. Whenever the player loses health, it will deduct one heart on the screen.



8.8 Button Assets



The Start button is designed to animate when player hover or press. This feature will help visually approve the player's action.

8.9 Level Text Design

The Level text will pop up before a round starts.

8.10 Victory and Game Over Text Design

Victory image pops up when the player wins a round. Game Over image will pop up after the player loses the round. These images will be present at the center of the screen.

8.11 Audio Assets

- Background Music – A scary themed melody to enhance the theme of the Game.
- Enemy (Before Entry) – ‘Monsters are coming!’ in a demonic voice to indicate to the player to prepare.
- Enemy (Eliminating) – A evaporating sound to imitate the enemy being eliminated.
- Health (Gain) – Positive sound to demonstrate this is a rewarding item.
- Health (Loss) – A negative sound, which will indicate to the player this should be avoided.
- Level (Progression) – ‘Next level’ in a demonic voice is used for the indication of level progression.
- Selection (Text Buttons) – A clicking sound to indicate the player selecting the button.

8.12 Audio Direction

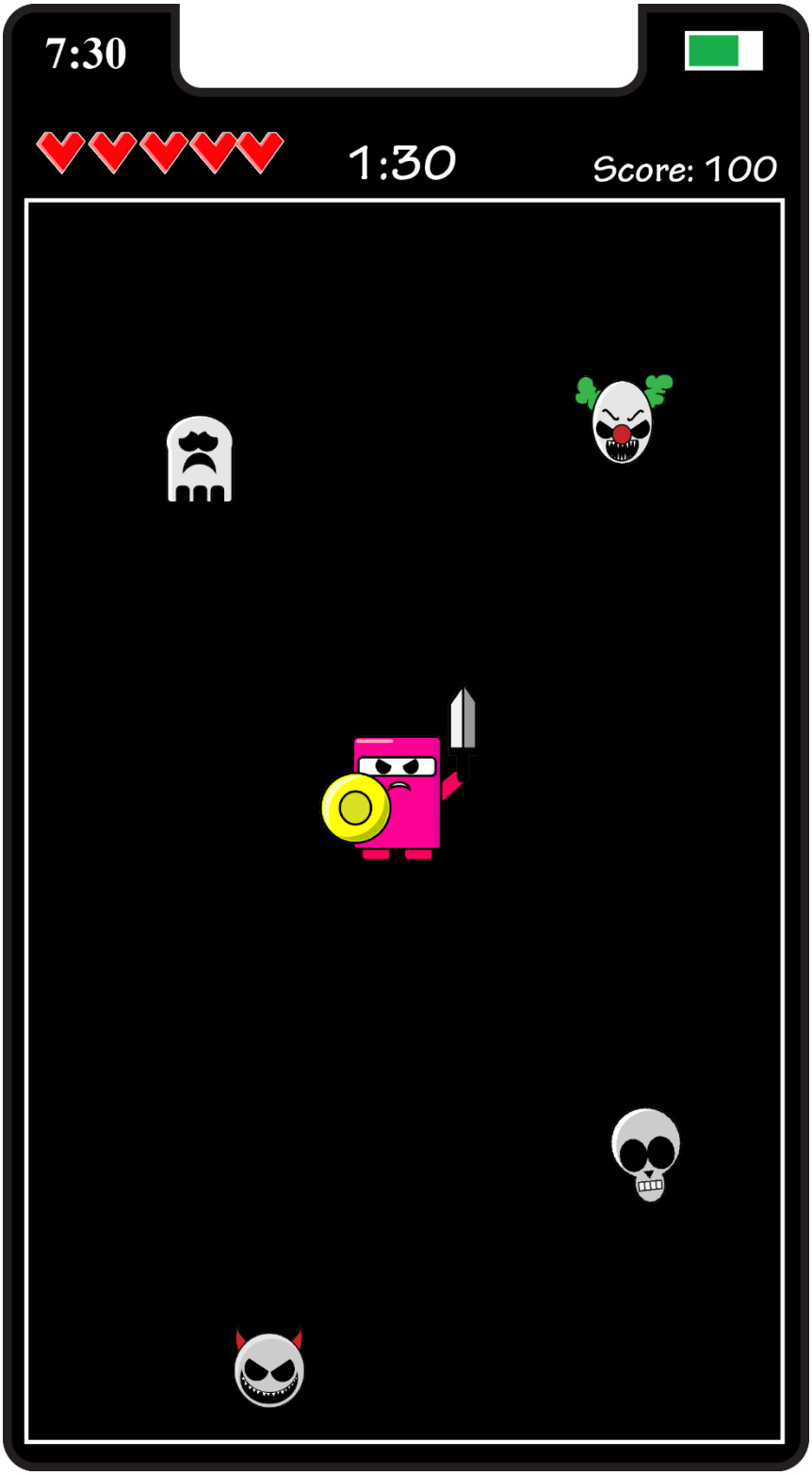
The name and art used in Nightmare Lord follow a horror theme, as intended the audio should give the player the same visualization through the sounds. Creating a slow and low toned melody achieves this aspect of being in a dark and weary place. Voice overs in a demonic voice will allow enhancing the horror theme even further. Settings feature could be implemented in future versions, allowing the player to decide the level of scariness in audio.

9 User Interface

9.1 In-Game UI Design

The border surrounding the screen is used to display the game grid that the player must interact with, thus also allowing the Game as much space of gameplay as possible. The other user-

interface elements such as five hearts, timer, and score will be displayed above game space to to avoid game interference.



9.2 Out of Game UI Design

The Game will consist of two easy to navigate UI screens for the convenience of young players. First, being the start screen displaying the title of the Game above the in-game character of the player. The arrow button emulates the swiping action on the mobile device for the selection of colors. The character in the center of the screen will cycle through the four colors as the user tap the arrow. Below the character, a 'START' button will be displayed, this will take the player to the second UI screen. A short instruction allowing the player to understand the gameplay will be displayed, along with a clickable 'PLAY' button in the bottom right-hand corner of the screen that will take the player to the gameplay.

9.3 Controller/Controls Design

The Game targeted market being mobile users, with the use of touch-screen technology. The player will only need to tap on UI buttons through the Game to select. Gameplay will use the

same concept of the player tapping to eliminate enemies or collect health, but also interact while in gameplay by dragging their character around the screen.

10 Level Design

10.1 Game Flow

Based on being an arcade-style game with relatively simple mechanics, making Nightmare Lord rely solely on the competitive aspect of the Game and its appeal in the replay value. The accumulation in difficulty as the player progress will allow the player to learn and have a better understanding of the different elements in the early few levels of gameplay.

10.2 Level, Mission, and Area Design

- The area of design is a grid with a dark background to simulate a dark-horror setting. Also, the border on the game space emulates entrapment in a nightmare.
- The only mission throughout the Game is to avoid contact with the enemy while eliminating them on each level.
- The increase in the number of enemies spawning and their speed with each level will keep the player challenged and preventing the Game from a stalemate.

11 Technical Design

11.1 Requirement Analysis

11.1.1 Technical Hardware

Nightmare Lord ‘s platform aspects are mobile devices, either Android or iOS. Anyone with a device that has a touch-screen feature will be able to play Nightmare Lord.

11.1.2 Technical Software: engine

The team used the Unity engine to develop Nightmare Lord.

11.1.3 System intelligence

Nightmare Lord is designed to be simple, and therefore, it does not have any system intelligence or artificial intelligence.

11.1.4 Data management/ support

The Game was initially designed to link with Facebook and battle friends by being the rank 1 in the leaderboard. The player's data will be recorded on the cloud, which will reflect on their social media leaderboards. Due to some unfortunate circumstances with having to switch from the first Game to Nightmare Lord, the team has not implemented this feature in the final prototype.

11.2 2D/3D Graphics

Nightmare Lord is a 2D game, hence why every element in the game will be 2D. This includes User Interface, buttons, assets, and texts. The Game does not need any graphics card to perform best. Hence why the Game will have a small graphics intake since it is projected for mobile phones.

11.3 Animations

Having animations effectively makes the Game looks aesthetic, but it is not an essential factor to consider when developing the Nightmare Lord. With the timeline, the designer has thoroughly thought about the implications that will come when animations are developed throughout the Implementation process. Therefore, the only animation designed was the blinking start button. When the player hovers or presses the button, the blinking design will show. This feature will contribute to the aesthetic factor of the overall Game.

12 Appendices

12.1 Probability and Utility Worksheet

12.1.1 Probability distribution.

Enemy spawn – there are four enemies within the Game, the outcome of the next monster to spawn is random.
Health – Hearts are spawned throughout the Game will be random on a % and timer
Spawn Location – The outcome of the random spawn location of the enemy

Outcome	Probability
Enemies will have a different value in score, which will reflect on their value of spawn rate. Assume value: Skull – 1 point, ghost – 2 points, Demon – 3 points and clown – 4 points Before testing	Skull: 0.25 Ghost: 0.25 Demon: 0.25 Clown: 0.25 Total: 1
Enemies will have a different value in score, which will reflect on their value of spawn rate. Assume value: Skull – 1 point, ghost – 2 points, Demon – 3 points and clown – 4 points Final rate	Skull: 0.50 Ghost: 0.25 Demon: 0.15 Clown: 0.10 Total: 1
Chance of spawning power-up at every quarter of the game time(60 seconds)	Heart: 0.25 x 15 secs Heart: 0.25 x 15 secs Heart: 0.25 x 15 secs Heart: 0.25 x 15 secs Total: 1
Spawn Locations would use either 4 sides of the grid. Probability would be equal since no side would change the difficulty	Top: 0.25 Left: 0.25 Right: 0.25 Bottom: 0.25 Total: 1

12.1.1 Game Code/ Algorithm

```
        void Update()

        {

            transform.position =
        Vector2.MoveTowards(transform.position, moveSpot.position, speed
        * Time.deltaTime);



            if (Vector2.Distance(transform.position, moveSpot.position) < 0.2f)

            {

                if (waitTime <= 0)

                {

                    waitTime = startWaitTime;

                    if (GameObject.FindGameObjectWithTag("Player"))

                    {

                        int trace = Random.Range(0, 2);

                        if (trace == 1)

                        {

                            Debug.Log($"enemy {name} is tracing player");

                            moveSpot.position = GameObject.FindGameObjectWithTag("P
        layer").GetComponent<Transform>().position;

                        }

                        else

                        {

                            Debug.Log($"enemy {name} is ignoring player");

                            moveSpot.position = new Vector2(Random.Range(min_X, max
        _X), Random.Range(min_Y, max_Y));
```

```
                            }

                        }

                        else

                        {

                            moveSpot.position = new Vector2(Random.Range(min_X, max_X),
        Random.Range(min_Y, max_Y));

                        }

                    }

                    else

                    {

                        waitTime -= Time.deltaTime;

                    }

                }



            }
```

Enemy movement behavior:
The enemy has 2 phases: move and stop
Move phase:
When there is a player exist in the playground, one enemy has 50% chance (D2) of targeting the player, or just move to a random location within the border, then take a rest for an amount of time.
If there is a situation that there is no player, for example, the player is dead, enemies then just moving around randomly.
Stop phase:
Count down its timer, then move again.

```
void Update()
{
    if (isBeingAttacked)
    {
        if (name == "Skull")
        {
            ScoreSpript.scoreValue += 1;

        }
        else if (name == "Ghost")
        {
            ScoreSpript.scoreValue += 2;
        }
        else if (name == "Demon")
        {
            ScoreSpript.scoreValue += 3;
        }
        else if (name == "Clown")
        {
            ScoreSpript.scoreValue += 4;
        }
        else if (name == "Health")
        {

            if (Player.currentHealth < 5)
            {
```

```
                    Player.currentHealth += 1;
                }
                else
                    Player.currentHealth = 5;
            }
            Destroy(gameObject);
        }
    }
```

Enemies give score based on their name when they are being attacked
“Health” is an enemy object but it heal player instead of dealing damage

```
    // Update is called once per frame
    void Update()
    {
        heartsCoolDown += Time.deltaTime;
        if (heartsCoolDown >= timeBetweenHearts)
        {
            heartsCoolDown = 0;
            if (GameObject.FindGameObjectWithTag("Heart") == null)
            {
                int chance = Random.Range(0, 3);
                Debug.Log(chance);
                if (chance == 0)
                {
                    SpawningHeart();
                }
```

```
        }

    }
```

There are 25% chance to spawn a heart for every “heartCoolDown” amount of time, one heart can be existed at a time.

```
void SpawningEnemy()

{

    //D20 for enemies

    int chance = Random.Range(0, 19);

    Debug.Log(chance);

    Transform _sp = spawnPoints[Random.Range(0, spawnPoints.Length)];




    int arrayIndex;

    if (chance <2)

    {

        arrayIndex = 0;

    }

    else if (chance >= 2 && chance<6)

    {

        arrayIndex = 1;

    }

    else if (chance >=6 && chance <11)

    {

        arrayIndex = 2;

    }
```

```
            else

            {

                arrayIndex = 3;

            }

              Sprite enemySprite = enemySprites[arrayIndex];

            string enemyName = enemyS

        prite.name;



            GameObject newEnemy = Instantiate(enemyPrefab, _sp.position,
        _sp.rotation);

            newEnemy.name = enemyName;

            newEnemy.GetComponent<SpriteRenderer>().sprite = enemySprite;

            newEnemy.GetComponent<EnemyPatrol>().speed = this.enemySpeed;

        }
```

Make a D20 for spawning enemies
Make a D4 for spawn point
Assign the random number from D20 to a specific Sprite
Make an enemy with the assigned sprite, sprite name
Spawn an enemy in the location from a randomly chosen spawn point

12.1.2 Utility

Outcome:	**Value:**
Each monster is assigned a point value associated with their difficulty aspect	Skull: 1 Ghost: 2 Demon: 3 Clown: 4
A power-up with hearts will add health to the player	Health: +1 heart/hp
The enemy coming into contact with character, reduction in health Before testing	Health: -1 heart for all enemies

The enemy coming into contact with character, reduction in health Final	Skull: -1 heart Ghost: -1 heart Demon: -2 hearts Clown: -3 hearts
Increasement in monsters spawning within each level	Level 1: 15 Level 2: 20 Level 3: 25 Level 4: 30

*Expected utility of monsters in points: 0.50 * 1 + 0.25 * 2 + 0.15 * 3 + 0.10 * 4 = 1.85*
Expected utility of monsters (spawn each level): 15 + 5 + 5 + 5 to infinite
*Health: 0.25 * 15 = 3.75*

This suggests that if the player can score above 2 points per second, they are on track to beat the current level. We can add to the difficulty with each level by increasing the speed of the enemy. The level design is informed by adjusting the utility of various sources in the Game, which includes monsters per level, speed accumulating, and power-up. The availability of a power-up throughout the Game, as well as the different value of deduction each monster, will ultimately come down to the player's skill and speed.

12.2 Feedback and Internal Economy Worksheet

12.2.1 Resources used in the Internal Economy

Name of resource	***How it is used in the Game***
Score points	*Players receive points for eliminating the enemies spawned in-game grid.* *Evil clown- 4 points* *Demon- 3 points* *Ghost- 2 points* *Skull- 1 point*
Health (HP)	*Players can gain additional health by collecting love hearts spawned throughout the Game. The player can only store five maximum hearts.*
Count down timer (60 seconds)	*The player must eliminate enemies within the 60-second time frame.*
Bombs	*The player can collect these throughout the Game to clear(eliminate) all enemies in the current screen. The player can store a maximum of 3 bombs.*

12.2.1 Feedback mechanism

Type of feedback (positive/negative/random)	***Input resources (amount of feedback depends on the quantity of these resources)***	***Output resources (resource whose level is changed as a result of this feedback)***	***The goal of this feedback mechanism***
Positive	*Enemy*	*Score points*	*The player will earn points for each enemy the player eliminates. Points are ranging from one to four.*
Negative	*Enemy*	*Health*	*Allowing the enemy to touch the player's avatar will decrease*

			their health by a certain amount.
Negative	Timer	Gameplay time	As the timer counts down, the player has less time to clear the stage.
Positive	Bomb	Points/Time	While the bomb clears all enemies on the screen, it will add their value to the score in points. It is positively reflecting on the time by clearing out the multiple enemies if used at a particular time.
Random	Health	Gameplay	The health will spawn randomly in the game space.

12.3 Economy Part 2

The enemy will grant value in points, positive feedback allowing the player to accumulate a score.
Health and bombs are spawned at a random rate, on a timer. This will balance the Game as the difficulty increases, while bombs will allow the player to use them through their gameplay tactically.
The timer feature in the Game will ensure the need for the player to continuously eliminate the enemy to further advance to the next level.
The number of enemy spawns at an increasing rate as the level progresses, a mechanism to challenge the players as they progress.
The enemy will target the player's avatar/character to challenge players throughout the Game effectively. When the level increases, the speed of the enemy increase as well.

Nightmare Lord is an arcade-based game, hence why the game structure's pattern is quite simple. The further the player advances through the levels with a full bar of health, and bombs collected gives the player the highest possible point score. The game design can be manipulated by incorporating more internal economy with items spawning, adding more enemy types with different movement patterns, and a mechanic that would positively add to the time.