Game Design Document

Project Name: Bhaago

Made by: Rahul Bethi

1. Game Description

This is a 2.5D scroller game for Windows computer which can be played using Microsoft Kinect as an input device, where the player keeps running and tries to avoid obstacles by performing actions like jumping, ducking or moving sideways. This game is developed for the Kinesiology department at Texas A&M University Corpus Christi (TAMUCC), with an objective to encourage children with Intellectual Disability(ID) to perform basic locomotor skills, to improve their physical activities and self-confidence.

2. Design Goal:

The design goals of this game are as follows:

1. To serve as a prototype for the experiments conducted by Kinesiology department at TAMUCC on children with ID
2. To develop a 2.5D perspective game in which all the objects are 2D planes, but are projected in a 3D space
3. To develop an achievement system that rewards the player for not hitting obstacles on its path and height achieved while jumping during the gameplay
4. To develop a customizable level design by any tutor who can modify and rebuild the game as per individual's need
5. To develop automatically adjustable level of movements in the gameplay based on the capacity of the child

3. Influences & Sources

All the images and audio files used in the game were taken from free licensing websites. The game is also built on the free version of Unity game engine. Some research was done to look at some of the contemporary games for any relation to our goals, but they were not targeted at children with ID and the levels were not customizable as per the individual's need.

4. Target Users

Our target user has the following characteristics:

1. 5 to 12 years of age

2. Has a windows computer

3. Has Kinect device with an adapter to connect to a USB

4. Enjoys playing video games

Although children with ID are the primary target users, the game, at a fundamental level, has ~~a~~ universal appeal to any child.

5. Functional Specifications

5.1. Core Game Play

The game basically consists of a PC (Playable Character), a straight path with all obstacles on the path, and other environmental objects on the side of the path. Everything is setup in a 3D space, with the PC moving forward at a constant speed. The camera is placed at the back of the PC at a certain distance, pointing towards the PC's direction of movement. Similarly, there is a scenery at the far end of the game, which moves along with the player in order to create an illusion that it is far away from the PC.

Kinect captures the movement from the player and corresponding to that movement, the PC is moved up, down or sideways in its confined space as needed.

The player will encounter obstacles that need to be avoided by jumping, ducking or moving sideways. The game has a health system, a life system and a score system. When the PC hits an obstacle, its health and score drops, and on the other hand, the player is awarded extra score for avoiding the obstacles.

The levels are designed by reading characters from a text file, which contains a series of characters in rows and columns. The character and its position in the file decides the type of the tile and its placement in the gameplay. The difficulty of the level is increased by increasing the number of obstacles on the path, by placing them close to each other and by increasing the length of the gameplay of each level.

The game displays the details of the PC, such as health, number of lives and score. The game also displays whether the Kinect is tracking the player. Moreover, it shows the maximum height achieved by the player in that session.

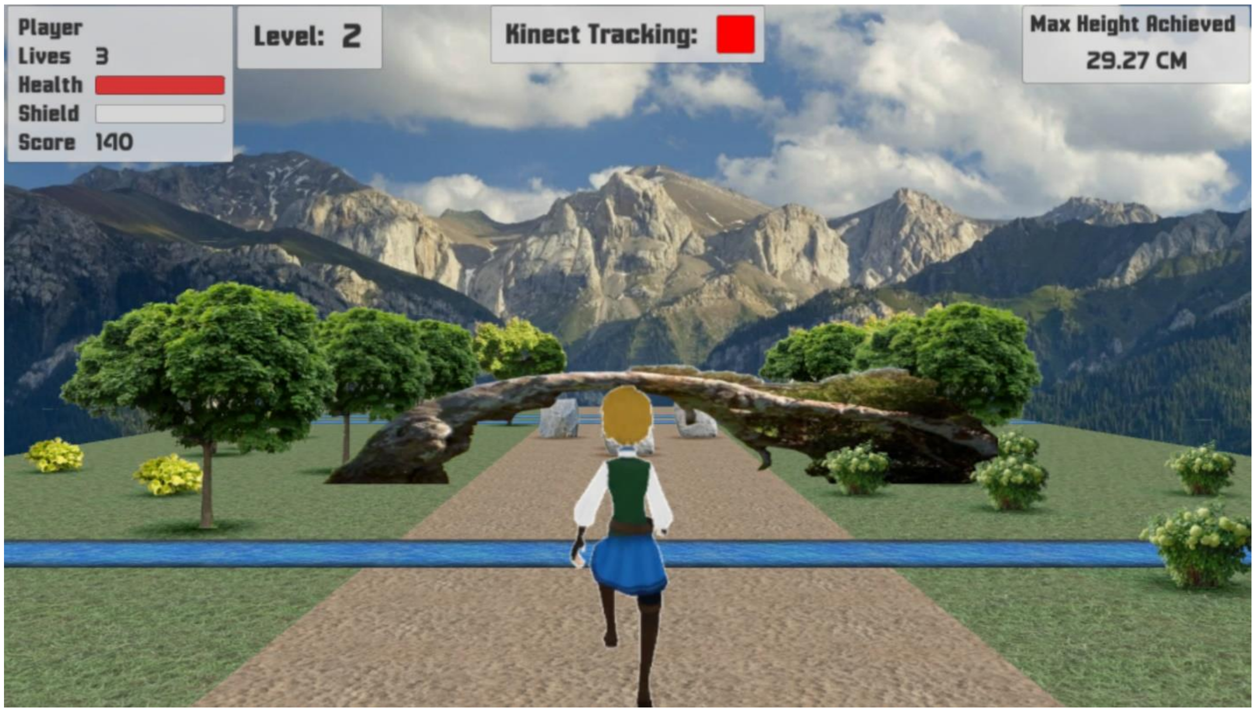


Figure 1. A screen shot of the Game play.

5.2. Game mechanics

Following are the actions that the PC can perform:

1. Jump (vertical movement)

   The player must jump in front of the Kinect. If the player crosses a minimum threshold height, the game calculates the speed at which he jumped and the PC jumps with that speed.

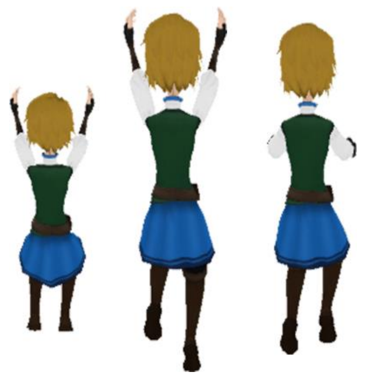


Figure 2. An example of Player jumping sprite.

2. Duck (vertical movement)

   The player must duck in front of the Kinect. If the player ducks below a minimum threshold, then the distance to which the player ducked is calculated and the PC moves down appropriately.

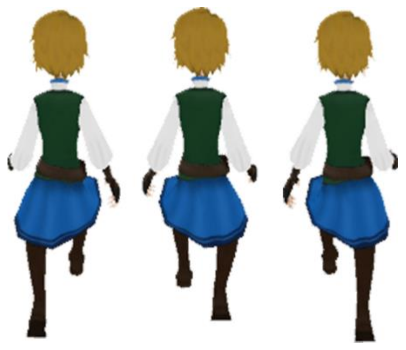


Figure 3. An example of player ducking and running animation sprite.

3. Move sideways (horizontal movement)

   Depending on the player's movement sideways from its start position, the PC moves accordingly.

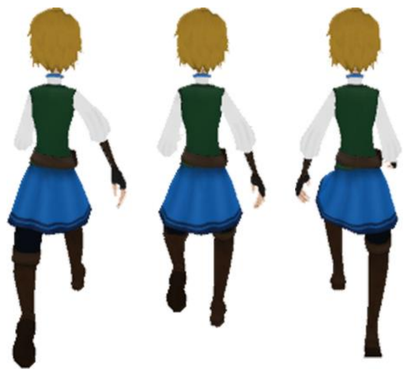


Figure 4: An example of the Player running sprite

5.3. Game Elements

***Camera*:** As mentioned in the core gameplay, camera is located at the back of the player, but only moves forward along with the player, and not upward, downward or sideways.

***Obstacles*:**

The main objective is to avoid these following obstacles and reach the final line:

Stones: These indicate a two kind of stones; the game chooses one of the images in random. PC must jump to half his height to avoid these obstacles. These are placed at the center of each tile.

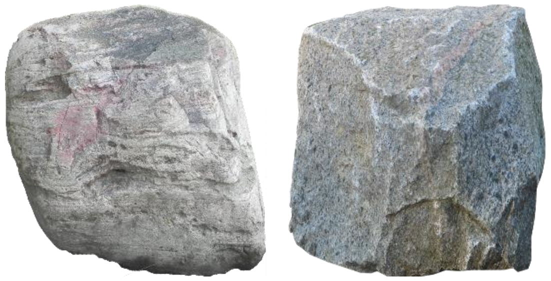


Figure 5. Texture of stone objects of Package 1.

Water: These are water patches which cover 1/3rd space on the path; the player must move sideways or jump to avoid this.

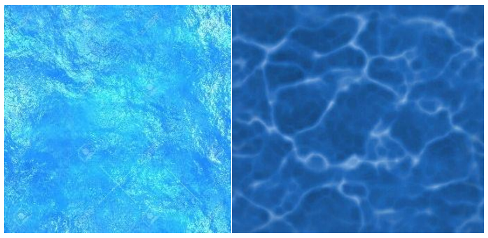


Figure 6. Images of tile textures – water, pond.

River: This resembles a small river stream. It also depicts a river flow animation, by moving the Image of the tile at a constant rate. It's horizontal to the path of the PC.

Log: This is a fallen tree, which has a gap underneath. PC must duck half of its own height to pass this object without colliding.

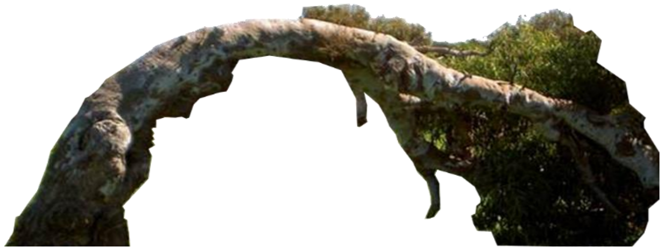


Figure 7 - showing a bent tree with a gap underneath it to pass by ducking.

Lava: These represent regions with hot molten lava. These cover 6 times that of water path and cover the whole path area. PC must jump higher than normal to avoid these obstacles.

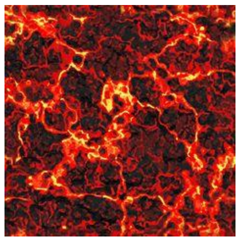


Figure 8 - showing lava texture of the tile.

***Power-ups*:**

These types of objects are used to give bonus to the player.

Health: PC*, on* hitting this power-up, replenishes its health to 100%

Invincible: This power-up makes the PC invincible for a few tiles, giving the player a bonus to rest.

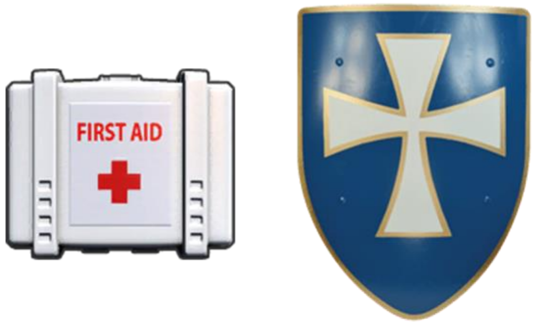


Figure 8. Health and invincible power-ups

Environmental objects:

These objects are used to decorate the game for more appeal.

Scenery: This is a big plane, and stays in front of the plane moving along with it, giving an illusion of a very long space. It has a scenery image of mountains and sky above it. As mentioned in the core gameplay, this object moves along with the player.

Figure 9. A scenery used in the game, package 0.

Trees: This object shows two kind of trees; the game chooses one of them in random.

Bushes: This object shows two kind of buses; the game chooses one of them in random.

Final: This object is used to end the game. PC touching this object will end the level.

Ground **-** *Tiles*:

These are the basic building blocks of the ground. They are square shaped planes. These are placed one after another, side by side to resemble a big plane. Properties and position of each tile are obtained by reading the level design (text) file for the corresponding level. These represent the obstacles, power-ups and environmental objects too. If the game reads a tree from the character in the level design file, then it will select an appropriate image of the ground at the base of the tree and places a tree object on the top of it, at its center. Similarly, all the objects are placed in the same way. However, objects like log and river which cover many tiles or entire row (horizontal to PC's path) are read based on the middle character of the row in the level text file. The path of the player takes 3 tiles of space.

Packages:

There are three set of packages to change the environment of the game and make it more appealing. They depict backgrounds such as desert, snow and monsoon. Objects such as trees, bushes, and stones, have three pair of images, with each pair belonging to each set and each image among the pair showing a different type of that background.

PC stats:

Lives: Number of chances left to play the game. A total of three lives are given at the start of the gameplay. If the PC's health goes to zero, then a life is lost and the level is repeated. If there are no lives left, then the game is lost.

Health Bar: A red bar, located below the number of lives, indicate the health of the PC. PC will lose 25% of its health when it hits an obstacle.

Invincible Bar: This indicates the amount of distance the PC can travel without getting affected by hitting the obstacles. This value resets to zero at the start of a level.

Score: Score of the game is displayed below the invincible bar. 10 points are awarded for not hitting an obstacle, these points are awarded as soon as the player crosses a row which has obstacles.

5.4. Game physics and Statistics

PC Movement: PC movement and game overall movement and progress in the core game play and individual object sections above. Additional to this, player has different sprites to run, jump and duck. These sprites are played 15 times a second, to get the animation.

Randomizing objects: Apart from these, there are randomizing mechanisms to make the game more appealing.

- Packages are selected in random, at the start of every level.
- Different images between stone, tree and bush are randomly selected.
- Type of obstacles are also randomized. Stone and water are randomly selected, which take up only one tile space. River and Log are randomly selected, which take up only one entire row of the PC's path.

Levels: This game contains three levels. If the PC reaches the finish line of a level, then the next level is loaded.

Game over: Game is lost when PC loses all his lives. Game is won when the player reaches the finish line of the last level.

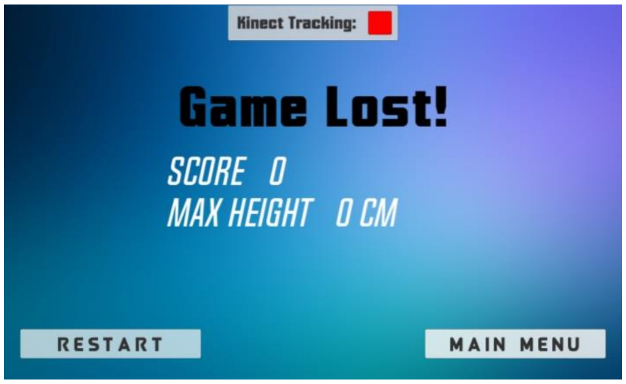


Figure 10. Screen shot of Game Over when player loses.

High score: There is a High score mechanism. The game stores the last 5 high scores of previously played game sessions. It also shows the maximum height achieved (in that session). When the game is won, the game checks for the existing high scores list. If it exceeds the last one, then this score is stored along with the others, in a descending order. The last score, which is lowest of all is then discarded.

Level design: As mentioned in the Core game play, the level is read from a text file and designed. The game starts reading from left to write and places objects from right to left with respect to PC's path in the game area. When the player reads a character, it will check if the character is available, if not it adds the default value type of grass or path, depending on the column. Middle three columns are reserved for path. If the game encounters a long obstacle like river or log in the center column, it then places the appropriate object in the center and puts the default value in the rest of the tiles. Objects if needed, like stone or tree, are placed and assigned to the tile position. Objects are disabled by default. Obstacles which needs to be randomized are also designated with a separate character. Each row of tiles corresponds to two lines of characters in the level text file:

- First row indicates the type of tile
- Second row indicates the type of power-up the tile can hold

This way, a tile can have obstacle and power-up at the same place.

***Recycling objects*:** At a point of time, only limited number of objects are displayed on the screen. This helps to build a very long level with less number of objects being loaded. When the PC crosses a row of tiles completely, those tiles are reassigned at the far end of the pack by reading the level text file of that new row. When the game reaches the finish line, the game reads the lines from first line of the level text file. Appropriate objects are unassigned and re-assigned to different tiles again, respective of the type of tiles.

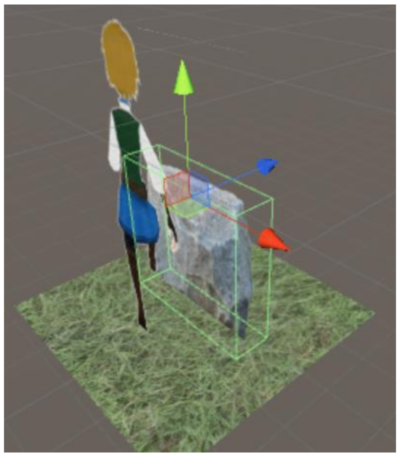


Figure 11. Example of the in-game character and its collision rules.

***Collision*:** Because of this recycling, PC is always in between the first and second row of tiles. This helps the game to calculate the collisions only between these three tiles (PC's path). Based on the type of tiles and position of the PC, the game calculates whether the PC is very near to the obstacle plane. If the object is very close for vertical objects (stone, log) and is on or below ground obstacles (river, water, lava), then PC is said to hit an obstacle. Collision is also calculated in the same way.

***Difficulty*:** Before the gameplay starts, the player is asked to jump as high as he can. This jump is compared the PC's maximum height it can reach in the game. A ratio is calculated between these two. This ratio is applied to the PC's jump speed, duck distance and side movements. This feature makes the game equally difficult to all the players irrespective of their capacity to jump. This makes the game equally challenging to kids with varying degrees of physical capability.

Multiple audio streams**:** This game has 5 audio sources. Only one audio clip can be played through each audio source. So, five audio clips can be played at any time in the game. The first audio source is reserved for PC's running sound clip. Rest of the audio clips are played by choosing if they are available to play. If an audio clip is being played, then the game chooses another audio source to play another clip.

5.5. User Interface

There are 10 menu screens. All the screens have a Kinect indicator. This indicator shows red when the Kinect is not tracking. When the Kinect is in

1. **Main menu**: The game starts with this menu screen. It has buttons to start the game, show high scores, mute the audio and to exit the application. It also introduces the name of the game, that is, "Bhaago!".

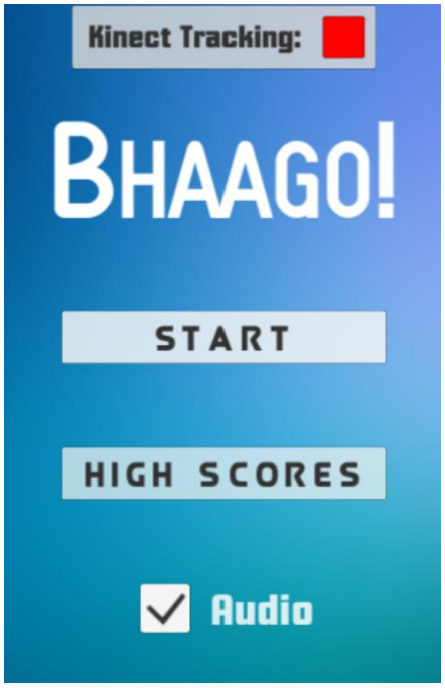
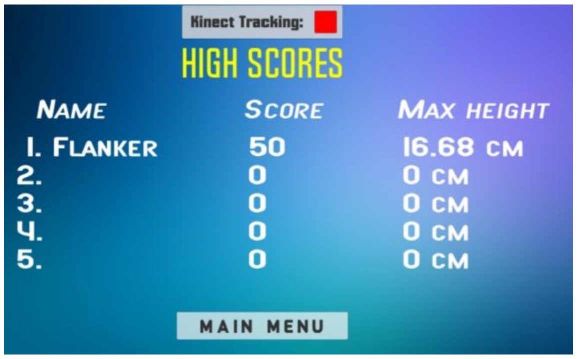


Figure 12: Main Menu screenshot.

2. High Scores menu shows the top 5 high scores stored in its memory. It has a main menu to go back to the main menu.

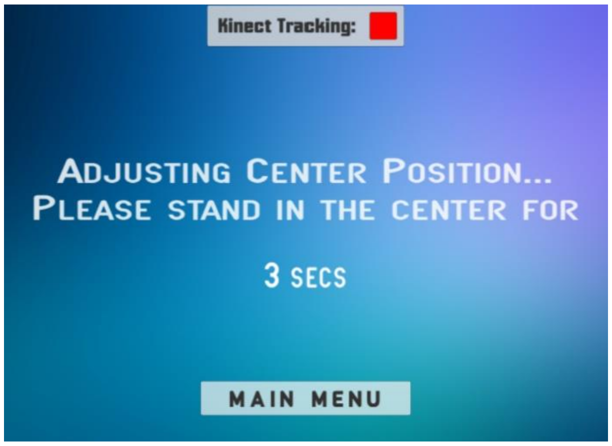

Kinect Tracking: 🟥

HIGH SCORES

NAME	SCORE	MAX HEIGHT
1. FLANKER	50	16.68 CM
2.	0	0 CM
3.	0	0 CM
4.	0	0 CM
5.	0	0 CM

Figure 13. High Scores screenshot.

3. Finding Center Screen: This is the first menu shown as soon as the start button is pressed at the main menu. It asks the player to stand in Kinect range to get tracked for 3 secs. This position, collected for 3 seconds, is calculated and is determined as the start position to calculate the movements and thresholds. This menu also has a button to go back to the main menu.


Kinect Tracking: 🟥

ADJUSTING CENTER POSITION...
PLEASE STAND IN THE CENTER FOR

3 SECS

Figure 14. Finding Center menu as soon as the game starts.

4. Setting Difficulty Screen: The Setting Center Screen takes the player to this screen after its timer runs out. In this screen, the player is required to jump as high as he can. This height is used to adjust the difficulty level of the game. This menu has two buttons, to go back to the main menu, or to start the game play.

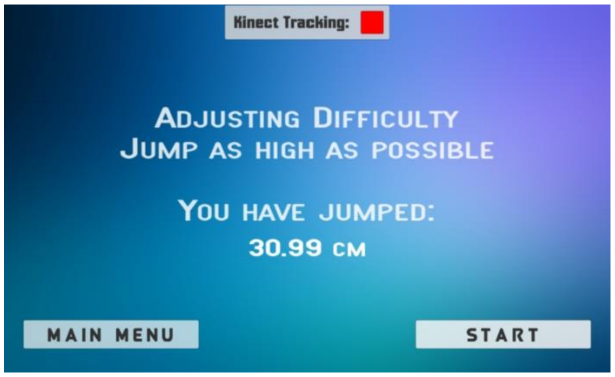


Figure 15: Setting difficulty screen.

5. Level start screen: This screen is shown for 2 seconds before every level starts, including the situation when the player loses his life and the level must restart.

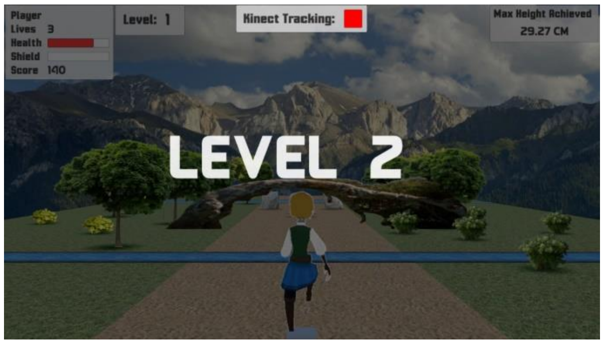


Figure 16: Level start screen, showing level 2.

6. Gameplay: This is the gameplay screen. All PC stats are shown here. When the player presses the escape key, the game shows the pause menu. And when the player touches the finish line, the game moves to the finish screen. Lost lives screen is shown when PC loses all his health.
7. Pause menu: This screen is shown when the game is paused (esc key). It has three buttons, namely, the resume button to resume the game, the restart button to restart the game from level 1, and the main menu to go back to the main menu (exiting gameplay).

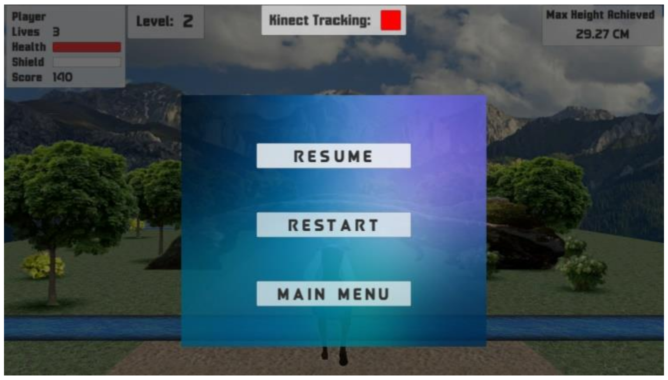


Figure 17: Pause menu.

8. Lost Life screen (transparent): The game pauses for 1 sec*ond* when PC loses all its health. After 1 sec*ond*, the level start screen of the same level or the Game Over screen (lost) is shown depending on the number of lives left.
9. Finish screen (transparent): The game pauses for 1 sec*ond* after the player hits the finish line. After 1 sec*ond*, the level start screen of the next level or the Game Over screen (won) is shown depending on the number of lives left.
10. Game Over: This screen has two buttons, Main menu button to go to the main menu and a restart button to restart the game. It also displays the score obtained and height achieved by the player in that session.

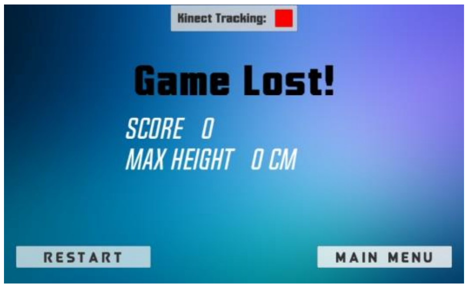


Figure 16. Screen shot of Game over screen, when game is lost.

   a. If the player wins, then it checks for the high score. If it is a high score, then it asks for the player's name. Pressing main menu will save this name, score and maximum height achieved in the list of top 5 high scores.

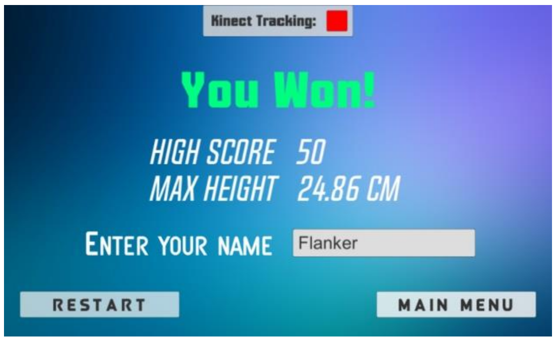


Figure 17. Screen shot of Game over screen, when game is won.

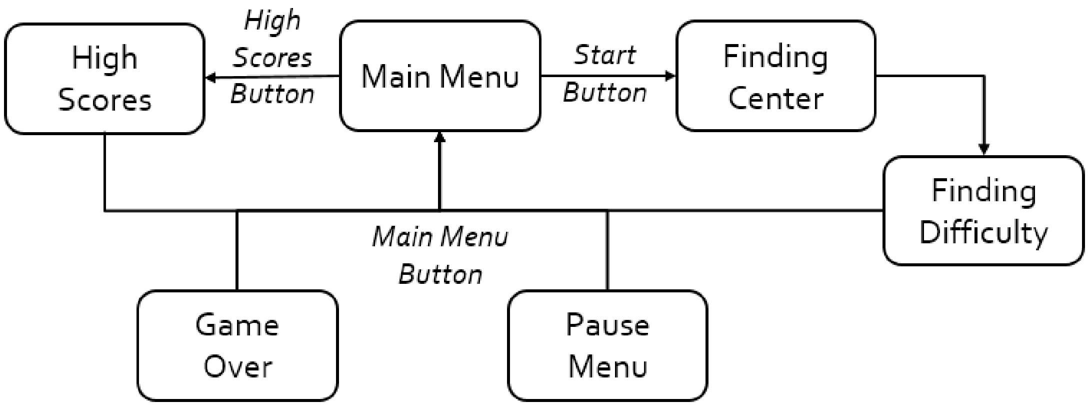


Figure 18. Flow chart for the Main menu

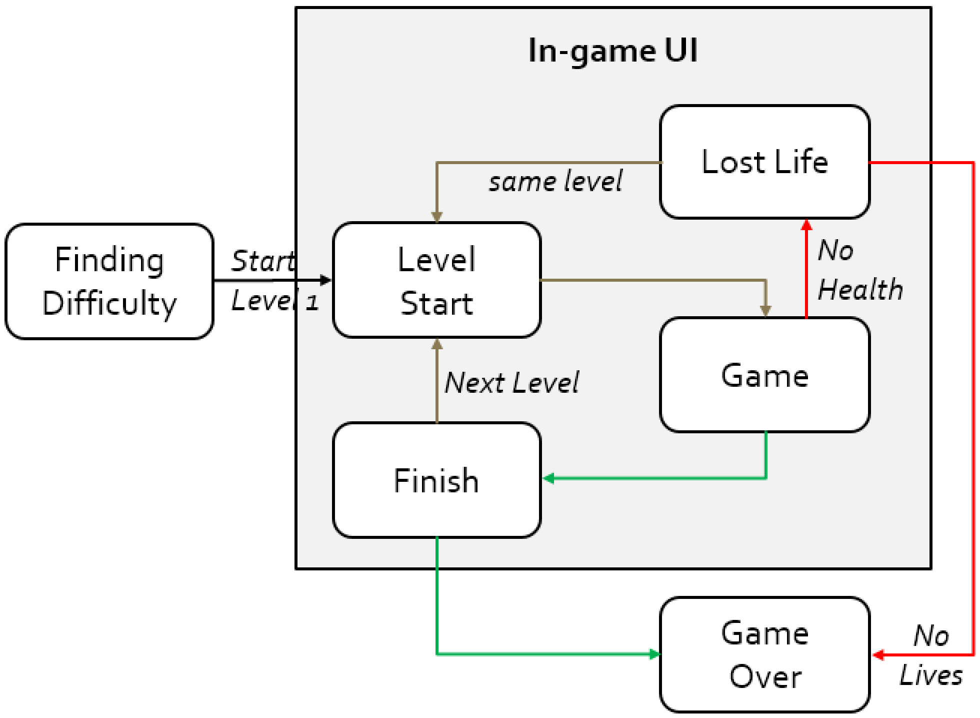


Figure 19. Flow chart for the in-game UI (game-play UI)

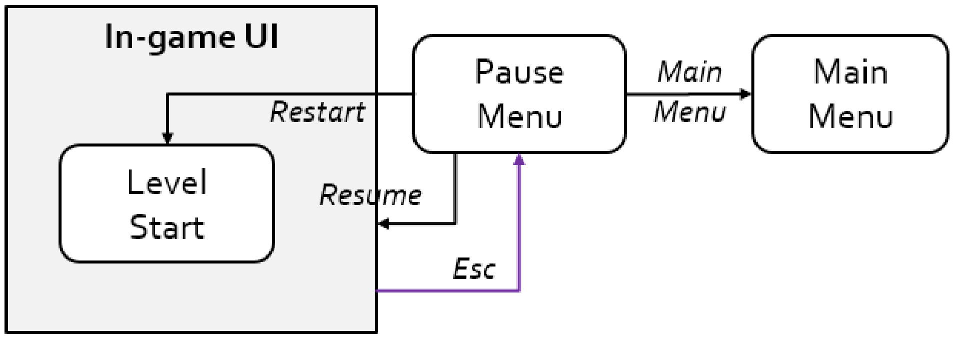


Figure 20. Flow chart for Pause menu

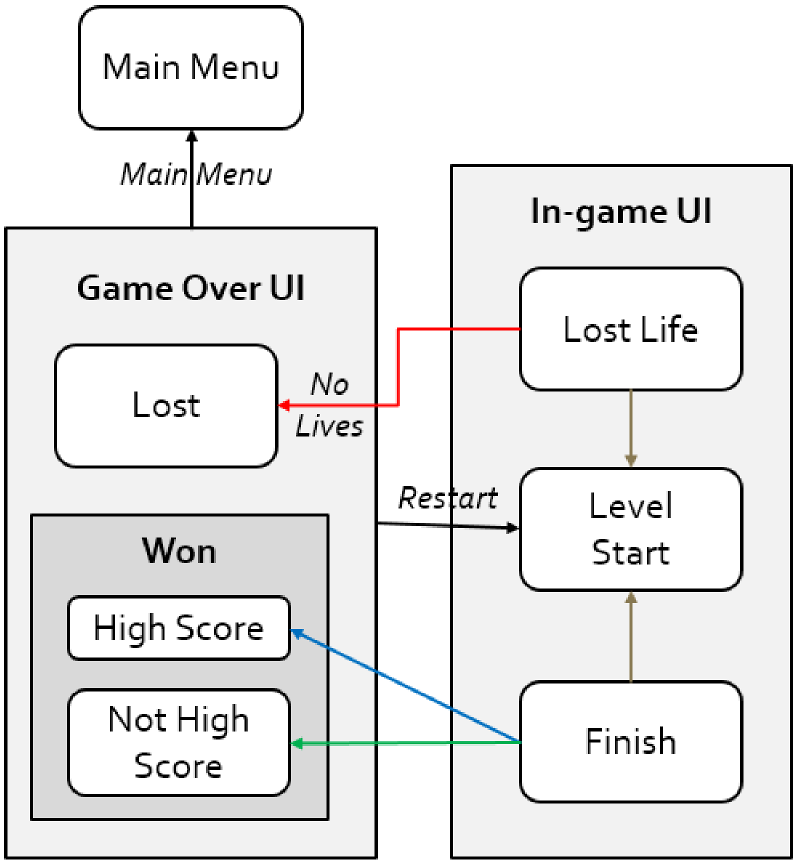


Figure 21. Flow chart of Game Over menu.

5.6. Audio

Audio of this game is straight forward, it has 11 files:

1. Button click sound – for all button clicks.
2. Level start sound – sound resembling counting ‘3 2 1 go’ before a race starts.
3. Level end sound – when PC collides the finish line.
4. PC running sound. This sound is played in a loop.
5. PC duck start sound – as soon as PC ducks
6. PC jumping start sound
7. PC jumping end sound – when PC lands on the ground after jumping.

8. PC hurt – when PC hits an obstacle
9. Power-up – when PC takes a power-up
10. Game Win sound
11. Game lost sound

Level map - designated characters:

g – grass

b – bush

t – tree

. – path

- - no power-up

i – invincible power-up

h – health power-up

s – stone

w – water

gggggg.l.gggggg – log (entire line)

rrrrrrrrrrrrrr – river (entire line)

gggggg.f.gggggg – finish (entire line)

p – lava

Random designations:

0 – tree or bush

1 – stone or water

gggggg.2.gggggg – log or river (entire line)

Level 1:

```
gggggg...ggbggg
---------------
ggtbgg...gggbgg
---------------
gggbgg...ggtggg
---------------
gggtgg...ggbtgg
---------------
gggtgg...gggbgg
---------------
gggggg...gggggg
---------------
ggg0gg...gggg0g
---------------
gg0ggg...ggg0gg
---------------
ggggg0.1.gggggg
---------------
gggggg...gggggg
---------------
gg0ggg..1ggg0gg
---------------
ggg00g...g0ggg0
---------------
gggggg1..gggggg
---------------
gg0g0g...gg0ggg
--------i------
gggggg..1gg00gg
---------------
ggg0gg...gggggg
---------------
ggg0gg...ggg0gg
---------------
gg0ggg1..g0gggg
---------------
gggggg...gggggg
---------------
gg0ggg...ggg0gg
---------------
gggg0g...gggg0g
---------------
gg0ggg...ggg0gg
---------------
ggggg01.1ggg0gg
---------------
gg0ggg...gg00gg
---------------
ggg0gg...ggg0gg
---------------
```

```
ggg0gg...gg00gg
---------------
gg0ggg...g00ggg
---------------
gggggg.2.gggggg
---------------
gggg0g...ggg0gg
---------------
gg0ggg...gg0ggg
---------------
gg0ggg...gg0ggg
---------------
gggggg...gggggg
---------------
gg0ggg...gg00gg
---------------
gg0ggg.2.gg0ggg
------h--------
ggg0gg...ggg0gg
---------------
ggg0gg...gg0ggg
---------------
ggg0gg...ggg00g
---------------
gggggg...gggggg
---------------
gg0g0g.1.gggggg
---------------
gg0ggg...gg0ggg
---------------
g0gggg...gg0ggg
---------------
ggg0gg...gg0ggg
---------------
gggggg..1gggggg
---------------
gg0ggg...gg0ggg
---------------
ggggggpppgg0ggg
---------------
ggggggpppgggggg
---------------
ggg0gg...ggg0gg
---------------
g00ggg...gg000g
---------------
ggg0gg...ggg0gg
---------------
gg0ggg...ggg00g
---------------
gggggg.2.gggggg
---------------
```

```
ggg0gg...gg0ggg
---------------
ggg00g...ggg0gg
---------------
gggggg.2.gggggg
---------------
gggggg...gggggg
---------------
gg0ggg...ggg0gg
---------------
ggg0gg...gg00gg
---------------
gggg00...gggggg
---------------
gg0ggg...gg00gg
---------------
gg0gggpppggg0gg
---------------
g0ggggpppggg0gg
---------------
ggg0gg...gggg0g
---------------
gg0ggg...ggg0gg
---------------
gggggg...gggggg
---------------
gg0ggg.1.ggg0gg
---------------
ggg0gg...ggg0gg
---------------
ggg0gg...gggggg
---------------
gggggg.f.ggg0gg
---------------
```

Level 2:

```
gggggg...gggggg
---------------
gggggg...gggggg
---------------
ggtbgg...grgbgg
---------------
gggggg...gbgggg
---------------
ggtggg...gtbggg
---------------
rrrrrrrrrrrrrrr
---------------
ggggtg...gggggg
---------------
```

```
gggggg...gbgggg
---------------
gggbgg...bggbgg
---------------
gggggg...gggggg
---------------
gggggg.l.gtgggg
---------------
gbggtg...ggbggg
---------------
ggtggg...ggtggg
---------------
ggggbg...ggbggg
---------------
ggbggg...bggbgg
---------------
ggtggg.s.ggbggg
---------------
gggggg...gggggg
---------------
ggbggg.s.ggtggg
---------------
ggggtg...ggbggg
---------------
ggtbgg...gggtgg
---------------
ggbggb...gggbgg
---------------
ggtgggs.sbggggg
---------------
gbgggg...gtgggg
---------------
ggggbg...ggtggg
---------------
gbgggt...bggbgg
---------------
rrrrrrrrrrrrrrr
---------------
gggtgg...ggbggg
---------------
ggggtg...ggggtg
---------------
rrrrrrrrrrrrrrr
---------------
gbgggg...ggtggg
---------------
ggggtg...ggbggg
---------------
gggbgg...gggtgg
---------------
gggtgg...ggbggg
---------------
```

```
gbbggb...gggbgg
---------------
ggbgtg.l.gggtgg
---------------
ggbggg...tggbgg
---------------
gbggbg...btgggg
---------------
gggtgg...ggtggg
---------------
gggbgg...gggbgg
---------------
gtggbg.ssgbggtg
---------------
ggbggg...tggbgg
---------------
ggtggg...gggbtg
---------------
ggbgtg...gtgbgg
---------------
ggggggpppggtggg
---------------
ggbgggpppgggggg
---------------
ggbgtg...gbgggg
--------i------
gggbgg...gggtgg
---------------
gggggg...gggggg
---------------
rrrrrrrrrrrrrrr
---------------
ggbggg...gtgggg
---------------
ggggtg...ggggbg
---------------
gbgggg...gggtgg
---------------
ggbggg...gggtgg
---------------
gggggg.l.gggggg
---------------
ggbgtg...tbgtgg
------h--------
gggggg...gggggg
---------------
ggtggg.l.gbgggg
---------------
gggbgg...gbggbg
---------------
gtgggb...gbgggg
---------------
```

```
rrrrrrrrrrrrrr
---------------
gbggbg...tgggbg
---------------
gggggg...gggggg
---------------
rrrrrrrrrrrrrr
---------------
gggtgg...gggbgg
------h--------
ggbbgg...ggtggg
---------------
gggtgg...ggbtgg
---------------
gggggg...gggggg
---------------
gtgggg.f.ggtggg
---------------
```

Level 3

```
gggggg...gggggg
---------------
gtbggg...gggggg
---------------
gggggg...gggtbg
---------------
gggbgg...ggbtgg
---------------
gggggg...ggbtgg
---------------
ggggggpppgtgggg
---------------
gggtggpppgggggg
---------------
ggggtg...ggbggg
---------------
ggbbgg...ggtggg
---------------
gggggg...gggggg
---------------
gggbgg...gggggg
---------------
gggtgg.l.gbtggg
---------------
gggggt...ggbggg
---------------
ggtbgg...gggbtg
---------------
gggggg...gggggg
```

```
---------------
gbgtgg...bgtggg
---------------
bggggb.l.ggbggg
---------------
gbgggt...gggbgg
---------------
ggggbg...gtgggg
---------------
gggggg...gggggg
---------------
rrrrrrrrrrrrrrr
---------------
gggtgg...gbggtg
---------------
gggggg...gggggg
---------------
rrrrrrrrrrrrrrr
--------h------
gggbgg...gggtgg
---------------
gggggg...gggggg
---------------
gggbgg.wsgggggt
---------------
gggggg...gggggg
---------------
ggggggw..gggggg
---------------
ggggtg...ggggbg
---------------
gggggg...gggggg
---------------
ggbgggswsgggtgg
---------------
gggbgg...gggtgg
---------------
gggggg...gggggg
---------------
gggtgg...bggtgg
---------------
ggggbg.l.gggtgg
---------------
gggggg...gggggg
---------------
ggggbg...gbgtgg
---------------
gggbgg...gggtgg
---------------
rrrrrrrrrrrrrrr
---------------
ggbtgg...ggtbgg
```

```
---------------
gggbbg...gttbgg
---------------
rrrrrrrrrrrrrrr
---------------
ggggbg...ggggtg
---------------
ggtggg...gggbgg
---------------
rrrrrrrrrrrrrrr
---------------
gggtgg...bgbggg
---------------
gggggg...ggbtgg
---------------
gtgbgg...ggggtg
---------------
gggbgg.s.gggtgg
---------------
gggbgg...gggbgg
---------------
gtgggg.s.ggtggg
---------------
gggbgg...gggtgg
-------i-------
ggggggw.sgggggg
---------------
ggbggg...ggggtg
---------------
ggtggg...ggtggg
---------------
gggbgg...ggtggg
---------------
gggbgg.l.gggtgg
---------------
gggbgg...gggtgg
---------------
gggggg...gggggg
---------------
gggtgg...gggtgg
---------------
ggtgggsswgggtgg
---------------
ggggtg...gggbgg
---------------
gggggg...gggggg
---------------
gggtgg...gggbgg
---------------
gtggggpppgtggtg
---------------
gggtggpppggbggg
```

```
-------h-------
ggbtgg...ggggtg
---------------
gggggg...gggggg
---------------
gggtgg...gggbgg
---------------
rrrrrrrrrrrrrrr
---------------
ggbggg...ggtbgg
---------------
ggbggg...gggbgg
---------------
ggbtgg...gggtgg
---------------
ggtggg.f.ggbggg
---------------
```