



Platformer Game

Assignment 1

Multimedia & Applications
(MULT3118C)
Cohort: BSE19BFT

Submitted by:
Beebeejaun Moossabhye (1908_17812)
Moorar Hansley (1908_17876)

Acknowledgement

First and foremost, we would like to thank our instructor, Mr. Riyad Dhuny, for introducing us to this module and who provided us the opportunity to learn creative and logical ways on how to develop deployable ready to play platform games. We would like to offer our heartfelt gratitude to everyone who helped us complete this platformer game.

A special thanks to all the creators of the free components that were easily found online and made this project simple and affordable to construct.

Table of Contents

Acknowledgement	1
Abstract.....	3
1. Introduction	4
2. Resizing the assets in photoshop.....	5
3. Building the Game in Construct 3	7
3.1. Main characters	7
3.2. Levels.....	10
3.3. GUI	11
4. Codes implementation.....	12
4.1. main controls	12
4.2. Collectables.....	13
4.3. Enemies.....	15
4.4. End of levels	17
HUD	18
4.5. Powerups	19
4.6. Movable tiles.....	19
4.7. Particle effects	20
4.8. Boss Battle.....	22
4.9. Mobile touches	25
4.10. Monetization.....	26
4.11. Social media integration	28
4.12. Sounds & Audio.....	29
4.13. Main menu	30
5. Exports	31
5.1. Exporting to web version	31
5.2. Exporting to mobile version.....	33
6. Other.....	35
7. Bugs and fixes	36
8. Further work	37
9. SWOT analysis.....	38
10. Conclusion.....	38
References and credits.....	39

Abstract

Multilevel tiled-based game for both mobile and pc/laptop users by the use of Construct 3. The game is set in a medieval time and the story is created upon this. This game is a full fledged six level game where the user will have to do simple tasks compromising of: collecting coins, killing enemies, collecting all 3 clover leaves to complete a level, etc. It also contains option of Facebook login and monetization options using Google Admob. Players will face difficulties against tricky platforms that can be movable or even unreachable. It will be to the player wits, for some level, for the to deduce the correct way for the to reach to the final level (level 6). Battle your way and fight the boss, which can be quite hard if you don't know the right moves. Reach to the final gate and complete the game.

Appropriate assets, sounds and music have been imported from the websites mentioned in the references below.

1. Introduction

“By the sword”, is a 2D action-based game set in a Middle age era where the goal is to provide an overall enjoyable and challenging game for all users, with simple controls.

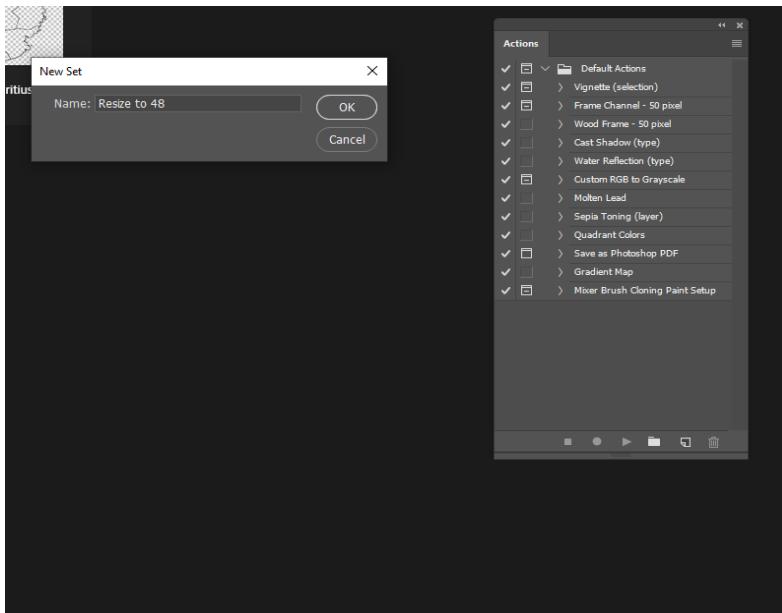
The game, which is set in a medieval setting, requires you to complete six stages by collecting all coins and health points while battling various spirits. After collecting all of the coins on the sixth level, the hero must confront the boss with a unique skill. Throughout all of the levels, the hero can gain skill boosts that will aid him in completing the level.

The game will require you to collect all “clover leaves” in order to complete a level else it will prompt you, at the end of that particular level, that the “clover leaves” are required.

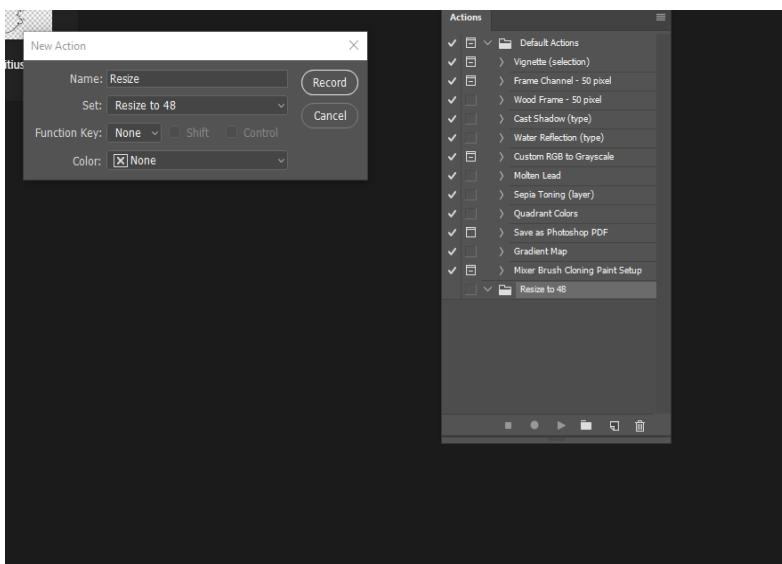
The idea is to produce and distribute a quality tiled-based platformer game online. The game must appear professional and complete.

2. Resizing the assets in photoshop

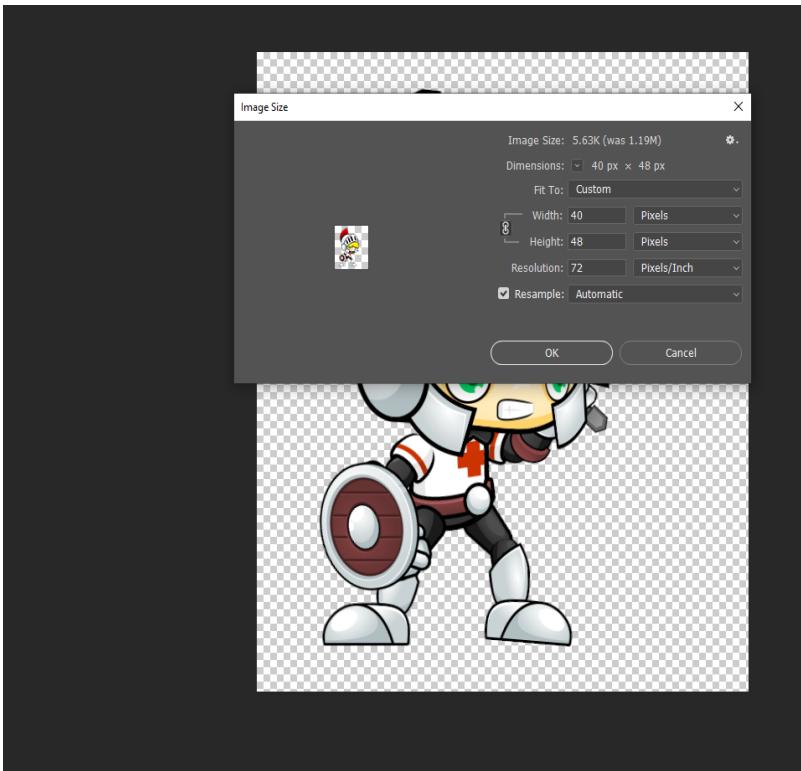
After picking all of the components for the game, it was discovered that none of the assets were of the required size of 48 x 48. To ensure that all assets were resized to the specifications, photoshop was used to automate asset resizing as follows:



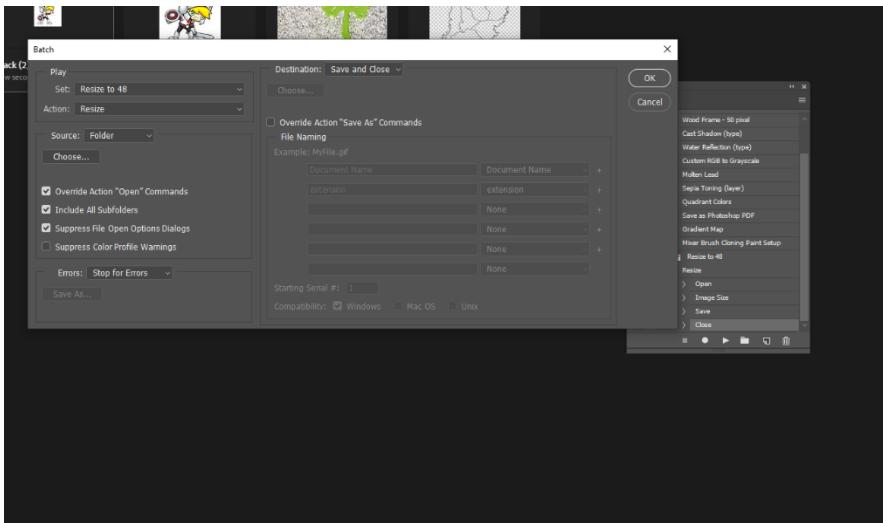
Creating a new in the action window



Creating Action Resize and Start recording actions.



Manually Import and Resize image size to 82 x 82

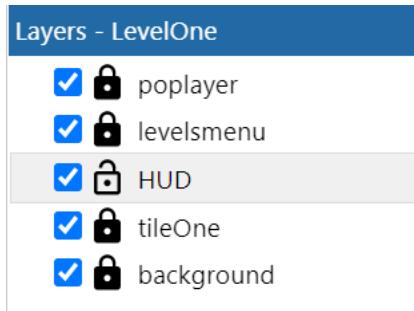


Automating the process using batch commands and setting the destination in same folder.

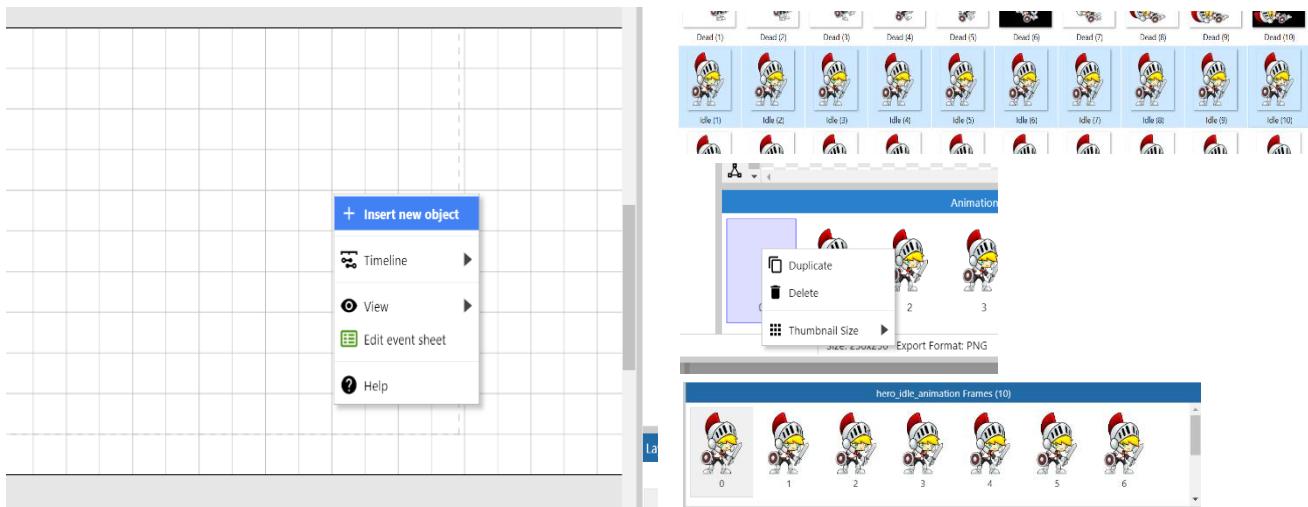
3. Building the Game in Construct 3

3.1. Main characters

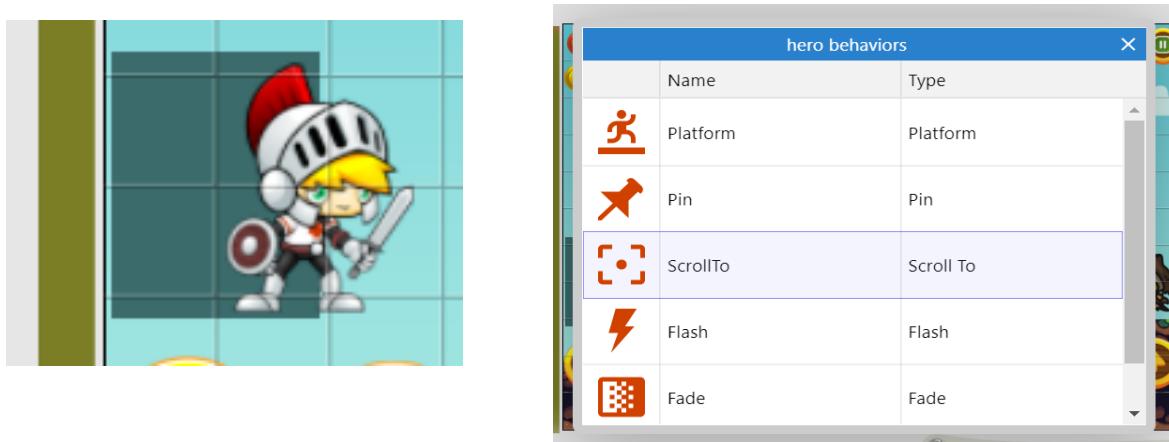
Step 1: the following layers are created. All backgrounds are placed in the “background” layer so as it can be locked later on to enable more controls on the tiles and characters that are in “tileOne” layer



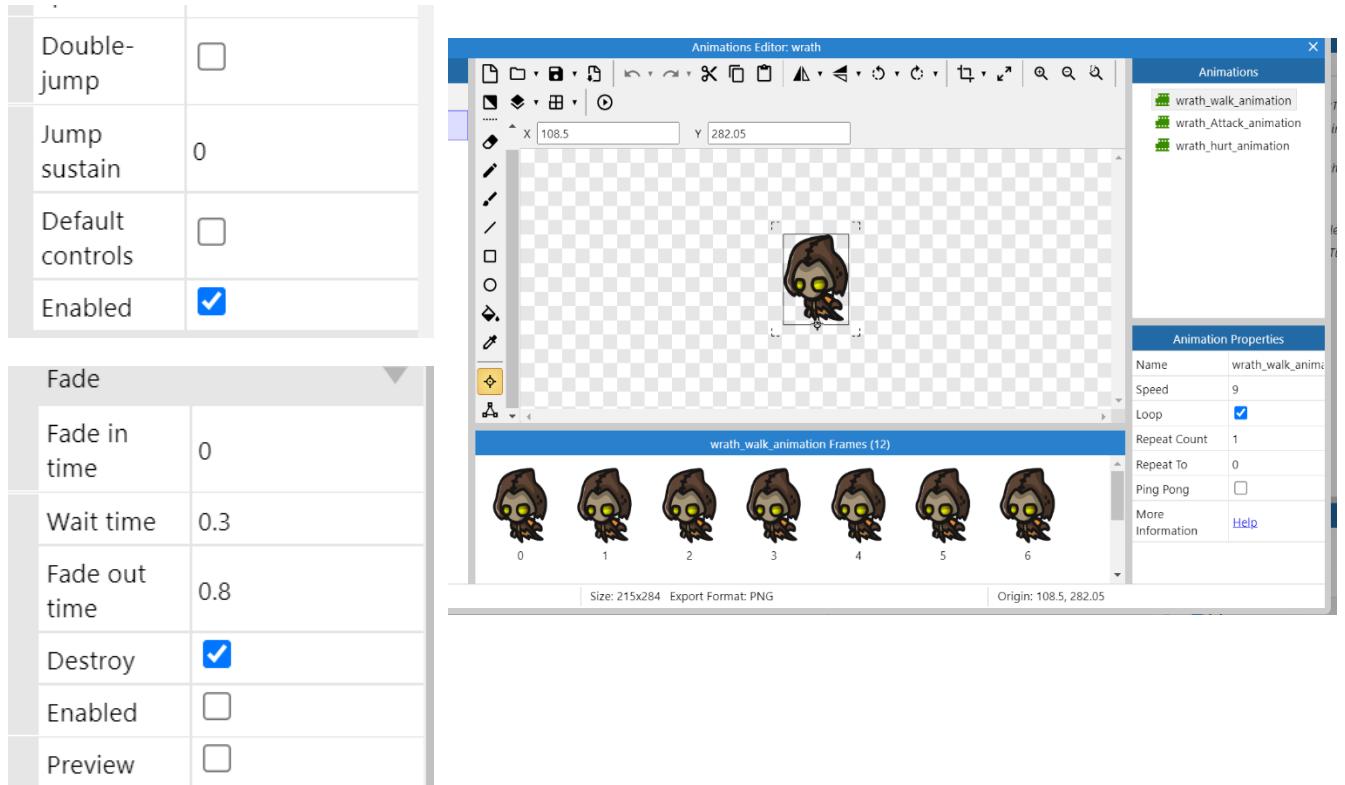
Step 2: here we will import the tiles and the characters. All the png needed for the frame are imported and the first frame “0” is deleted so as not to add a blank frame.



Step 3: the shadow object is then added and the hero is pinned to it. Note these following behaviors are also added to the hero.



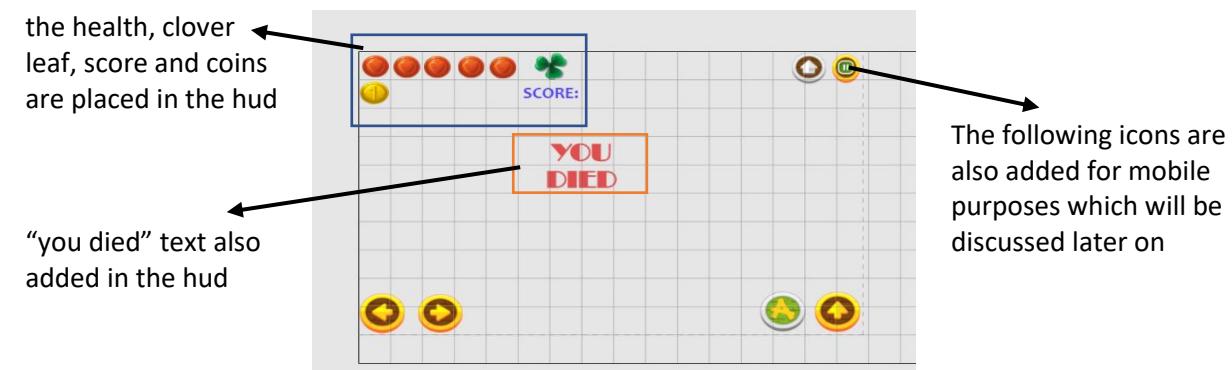
Step 4: enemies are added similarly as we imported the hero and a platformer and fade behaviors are also added. Note that the default controls are unchecked and some of the fade properties are modified.



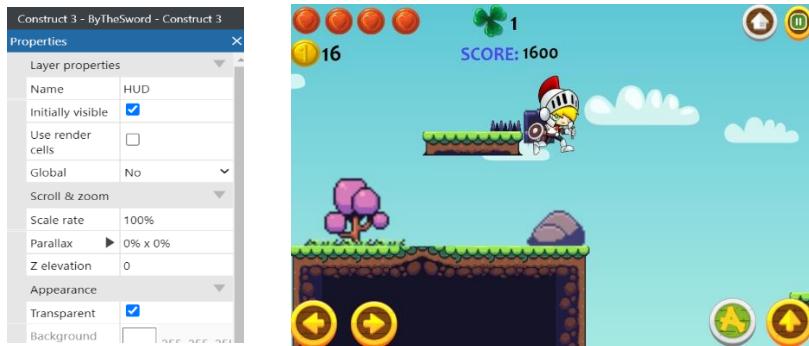
Step 5: next the coins are added to the layer at different intervals.



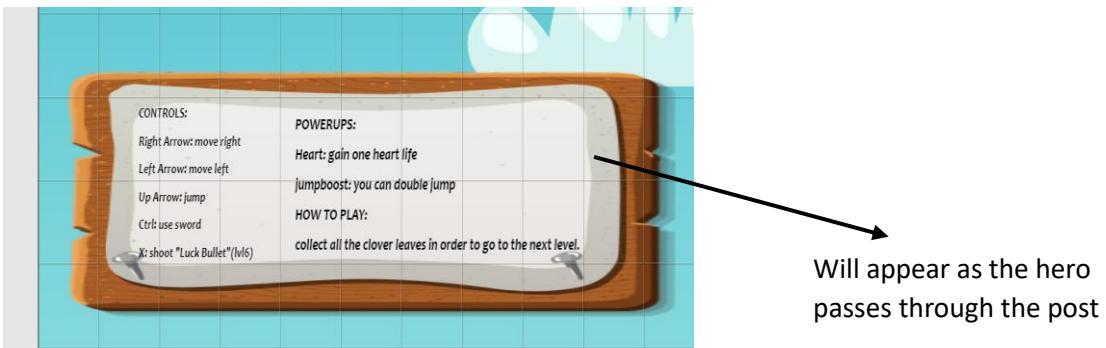
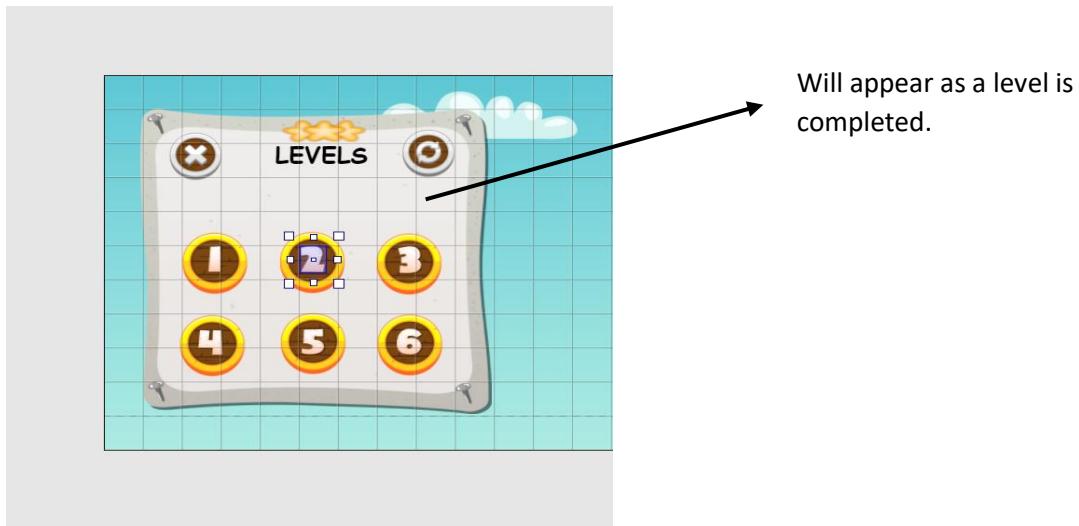
Step 6: all elements in the HUD are then added



the HUD scroll & zoom properties are modified. The parallax is set to 0 x 0 so that it stays fix to the screen as the player moves along.



Step 7: the level menu where the user will choose the level for them to move to the following level is designed on a new layer: “levelsmenu” and the controls pop up is designed on another layer called “poplayer”. This is done so that we can change the visibility of the levels easily as per the game requirements.



3.2. Levels

Level 1:



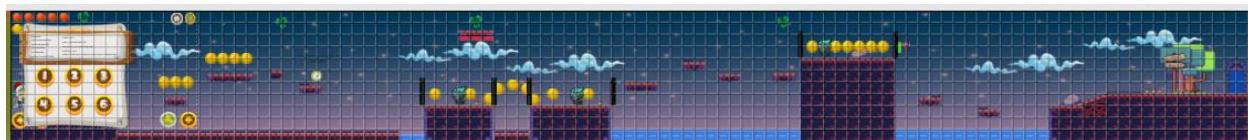
Level 2:



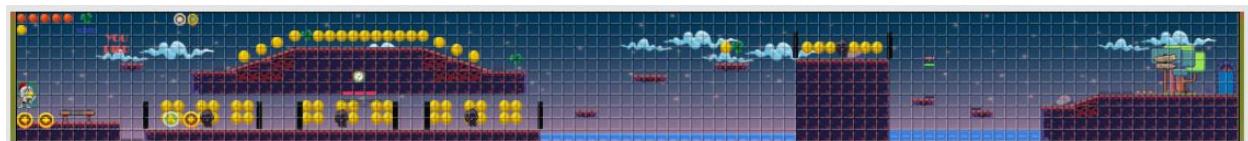
Level 3:



Level 4:



Level 5:

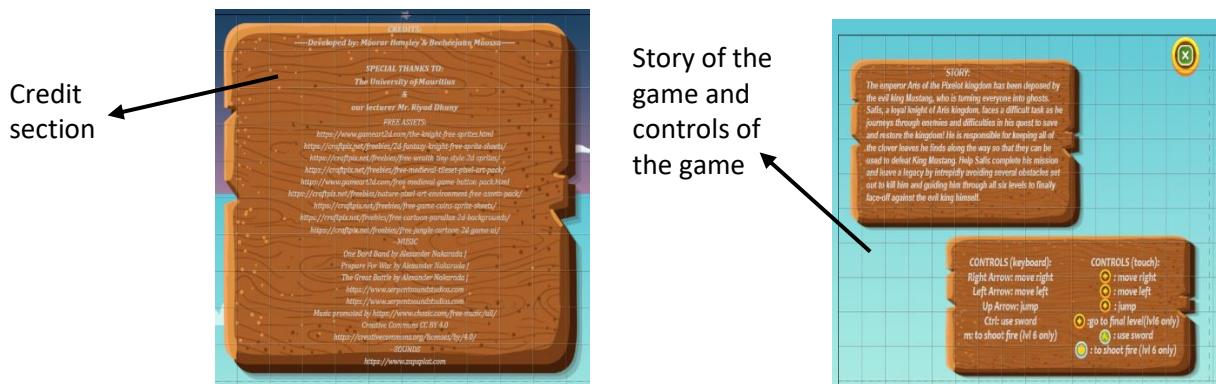
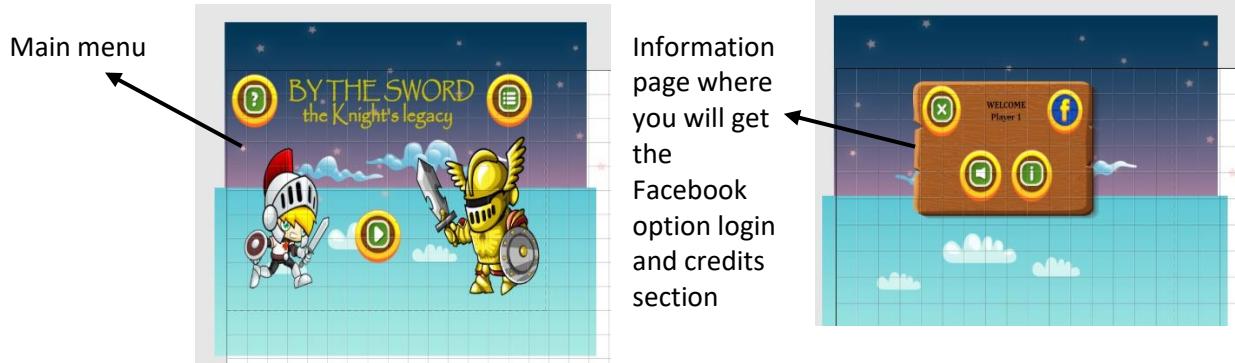


Level 6:



3.3. GUI

Main menu: the menu size is adjusted to 760 x 480. New layers are added so that the info section and the credit section can be implemented. The story and the controls of the game are added to a new layout.



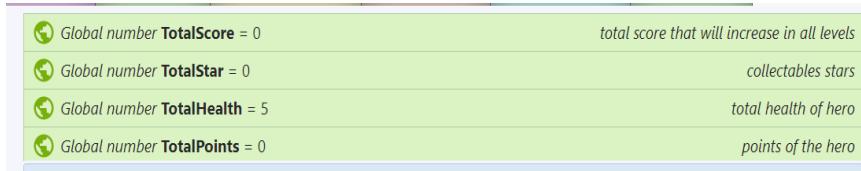
4. Codes implementation

4.1. main controls

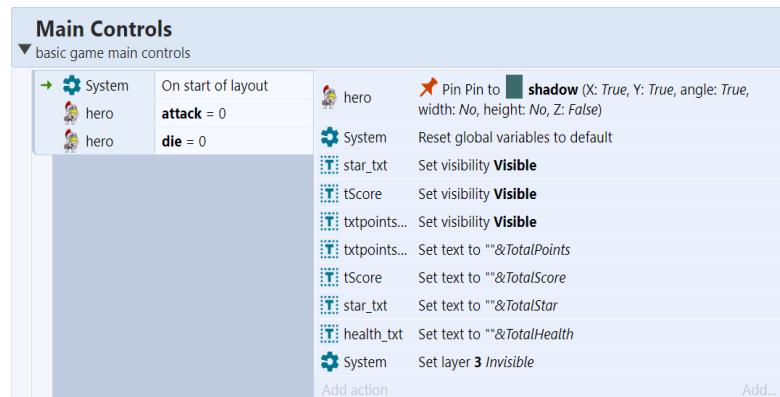
step 1: the basics events for staying idle, running etc. are created using the arrow keys for the movements. The hero idle animation is set back to normal if the user does not touch the arrow keys and the movements are mirrored if the user is going backwards.



Step 2: the following global variables are created. The “TotalScore” will save the score of the user as they progress, the “TotalStar” will keep track of the leaf collected, the “TotalHealth” is set to 5 as maximum life of the player and “TotalPoints” is to keep track of coins collected

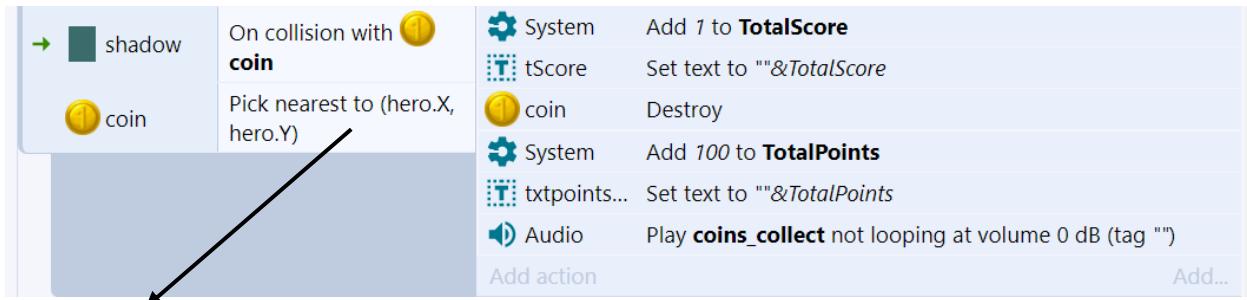


Step 3: next we have to enable so events as the level starts. We pin the hero to the shadow, we set some of the HUD visibility to false, set the text of the scores, set the layer 3 (levelsmenu) to be invisible and make sure to reset all the global variables as the game reset as to not be stuck in a loop.

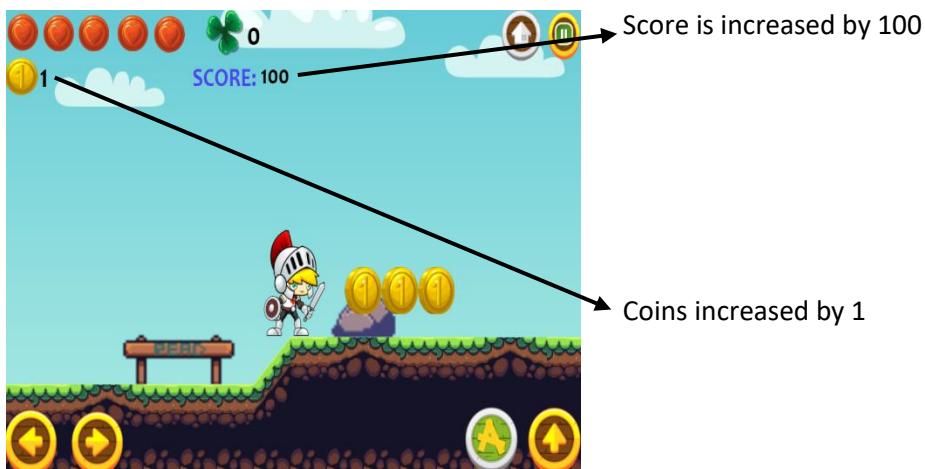


4.2. Collectables

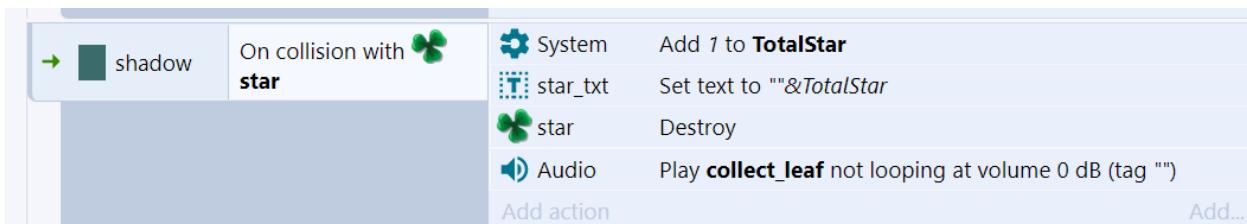
On collision with the “coin” object, + 1 will be added to the “TotalScore” and that score will be displayed in the text called “tScore”. The coin is then destroyed and the scored is also increased by a 100 for the “TotalPoints”. The Audio of coin_collect is then played once.



This is used to prevent all the coins in the level to be collected on the first touch of the first coin.



The same process is repeated for the “star” and a different audio is played.



Here a barrier has been set right under the treasure box. When the player hits that barrier the animation frame of the box changes to 1 which is where it opens. Then the “gain_health” object appears at the image point 1 of the box as shown below. The box starts to fade away afterwards and the barrier is destroyed.

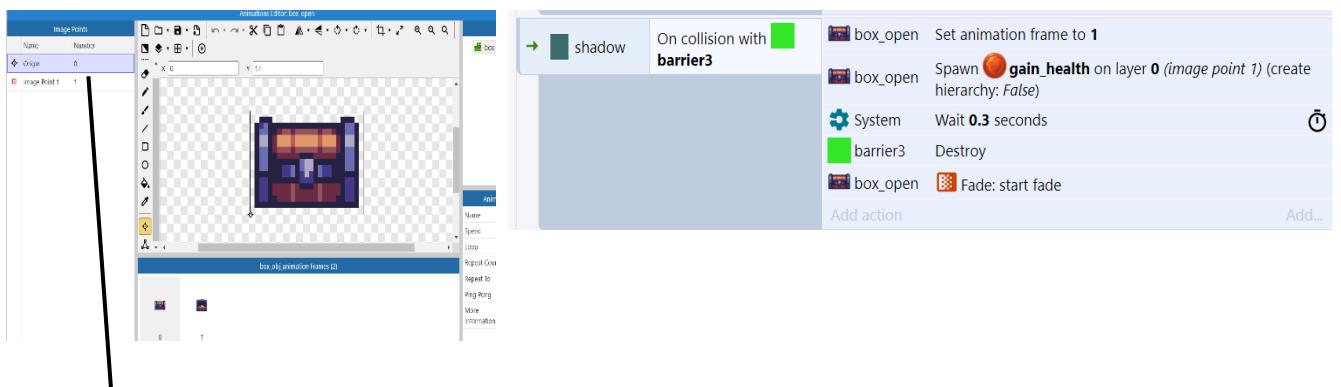
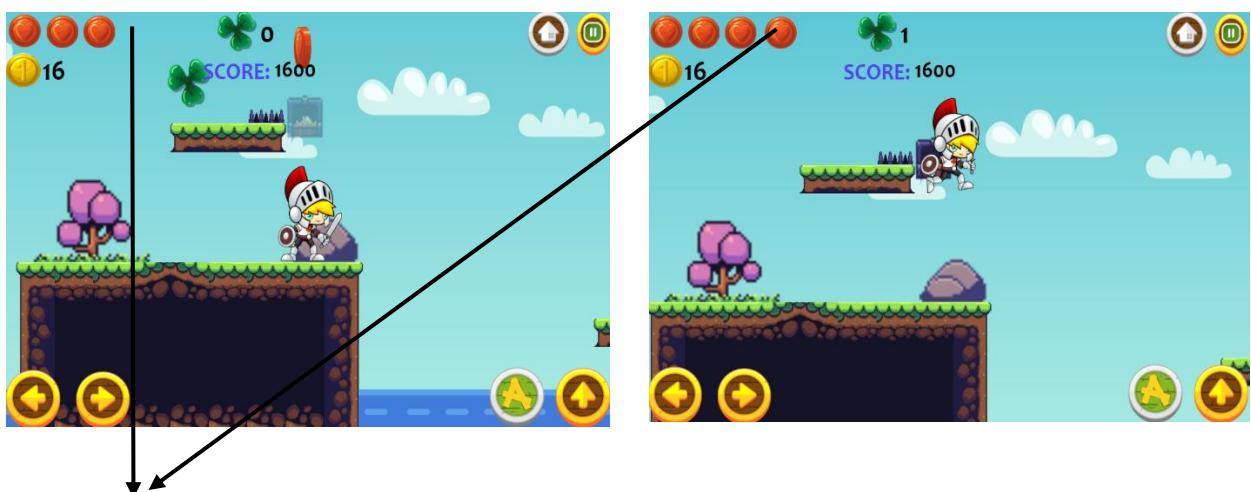
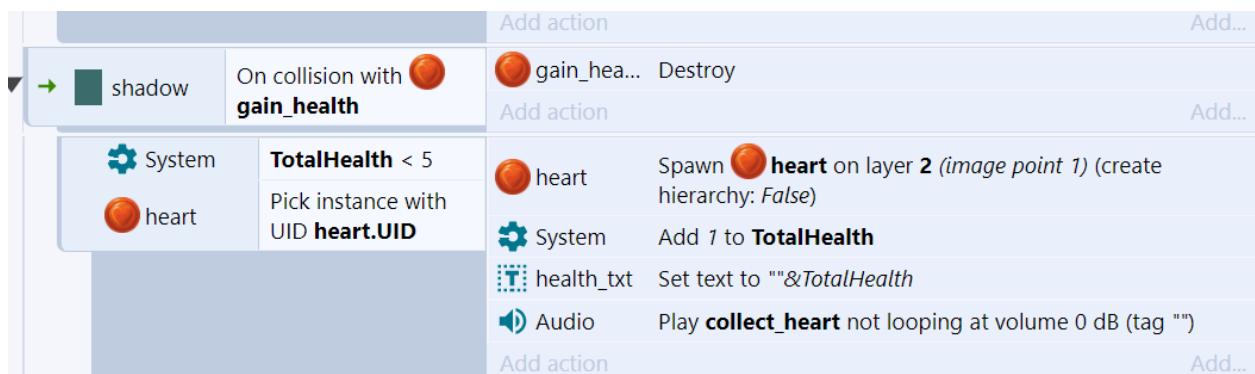


Image point 1 created found at the top of the box.

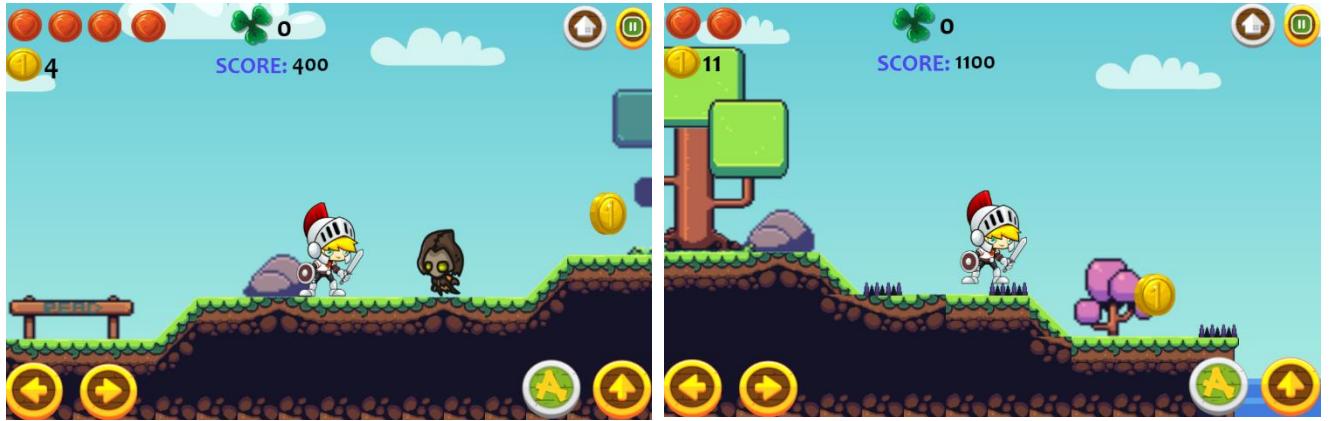
As the player collides with the “gain_health” object it will add 1 health to its health status. Note that the maximum of health allowed is 5. Hence the event is allowed only if the health is less than 5. An instance variable is also added to that object so as it decreases it does not do it all at once but rather one by one starting from the right.



Health collected.

4.3. Enemies

The first enemy are the wrath (left picture) and the second are the spikes (right picture).



The enemies' controls are similar to the hero. Note that the default settings in the properties of the enemy is unchecked. If the player hits the spikes or the wrath a point will be deducted from their health and the health status in the HUD will decrease, the flash behavior will be enabled for 1 sec and the "hero_hurt" sound will be played.

Wrath Enemy Events	Spikes Enemy Events
<ul style="list-style-type: none"> On collision with barrier1: Set direction to "Right" Else: Set direction to "Left" 	<ul style="list-style-type: none"> On collision with spikes: Pick nearest to (main_star.X, main_star.Y) heart: Destroy hero: Flash: Flash 0.1 on 0.1 off for 1.0 seconds System: Subtract 1 from TotalHealth health_txt: Set text to "&TotalHealth" Audio: Play hero_hurt not looping at volume 0 dB (tag "")

When the player falls and hit the sea the life will be set to 0, set the text of "you died" to visible. We make the system wait for 3 sec and then restart that layout.

Sea Collision Event
<ul style="list-style-type: none"> Audio: Play hero_hurt not looping at volume 0 dB (tag "") you_died: Set visibility Visible System: Wait 3.0 seconds System: Restart layout

Here we have set an event that whenever the health is equal to 0 then the “you_died” text will be visible. The hero will start to flash and the die animation will be played.

Note to set the die animation we have to create an instance variable called “die” and set it to 0 then whenever there is an animation the “die” variable must be set to 0 and only when the die animation will be played that we will set it to 1.

Enemies in level 1 to 3 are the same. 3 to 4 are different and 5 to 6 are also different. Thus, for each different enemy (wrath) the steps in this section have to be repeated.

The image shows the Construct 3 script editor on the left and a screenshot of the game on the right. The script editor displays three events:

- Event 55:** Triggers when TotalHealth = 0. It sets visibility to Visible for the "you_died" text, then sets die to 1, waits 3.0 seconds, and restarts the layout.
- Event 56:** Triggers when die = 1. It sets the hero's animation to "die_hero_animation".
- Event 57:** Triggers when the "die_hero_animation" is finished. It sets die to 0.

A black arrow points from the text "Die animation" to the third event (Event 57).

The game screenshot shows a character standing on a platform with a large red "YOU DIED" text overlaid. The score is 400. There are coins and a shield icon at the bottom.

You can attack the enemy (wrath) in 2 ways. When you hit the ctrl button, the attack animation will play. Or if you jump on their head. In the event whenever you collide and you're jumping the hurt animation for the enemy will be played and the enemy will start fading and the following audio will be played. Similarly for the attack with ctrl the same events will happen.

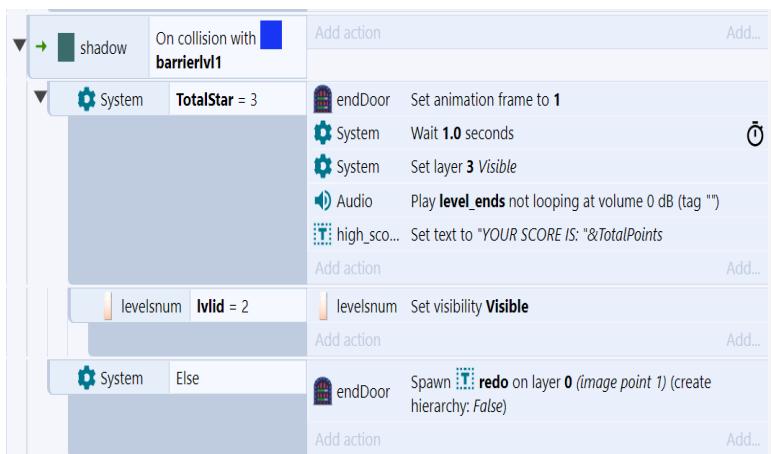
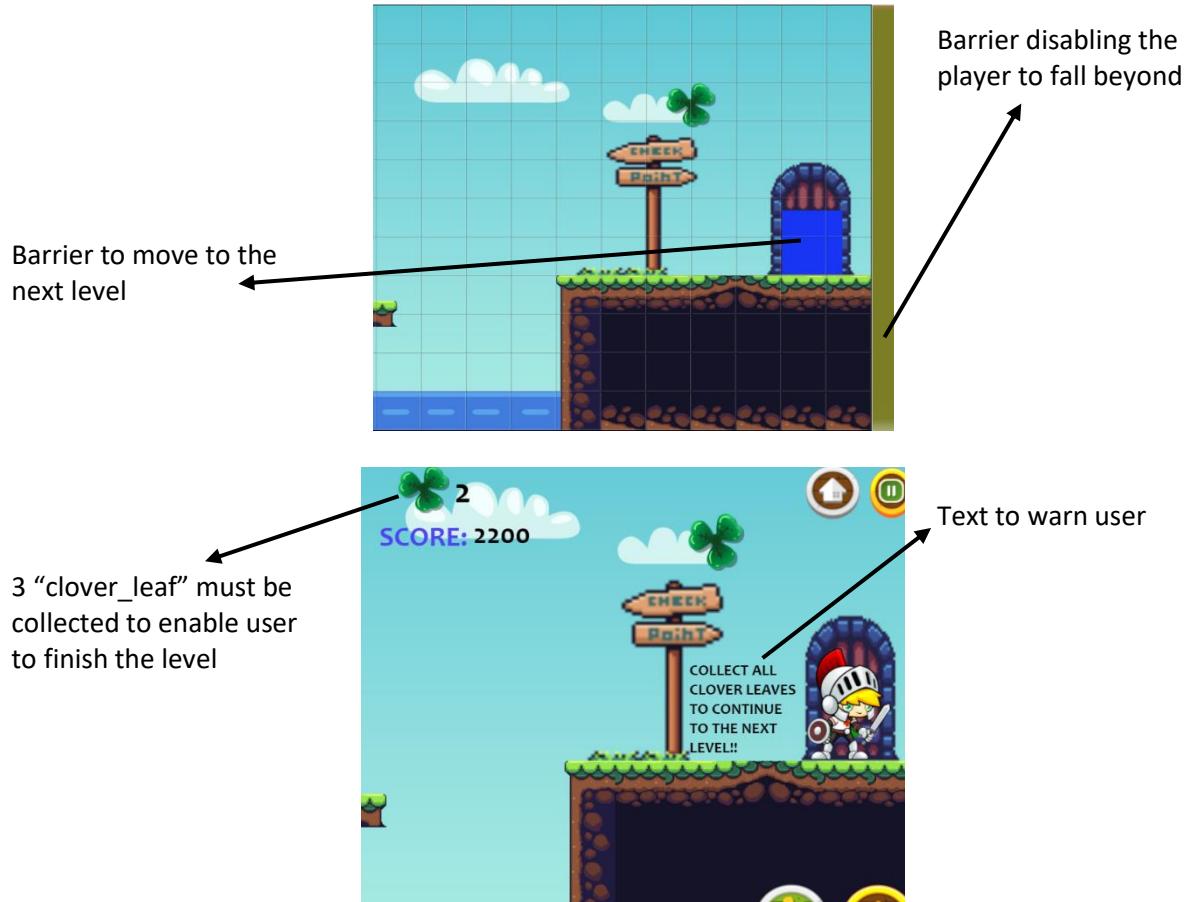
The image shows the Construct 3 script editor on the left and a screenshot of the game on the right. The script editor displays two events:

- Event 1:** Triggers on collision with the wrath enemy. It sets the shadow's platform to falling, then performs the following actions for the wrath enemy:
 - Set collisions Disabled
 - Set Platform Disabled
 - Set animation to "wrath_hurt_animation" (play from beginning)
 - Fade: start fade
 - Set Platform vector Y to -650
 - Play hit_enemy not looping at volume 0 dB (tag "")
- Event 2:** Triggers on collision with the wrath enemy while the hero's "attack_hero_animation" is playing. It performs the same actions for the wrath enemy as Event 1.

The game screenshot shows the hero standing above a wrath enemy, which is currently fading away. The score is 1300.

4.4. End of levels

First a barrier is placed to the door and is set to invisible in its properties. A new empty text field object is placed at the end as well to tell the player to collect all leaves before going to the next level.



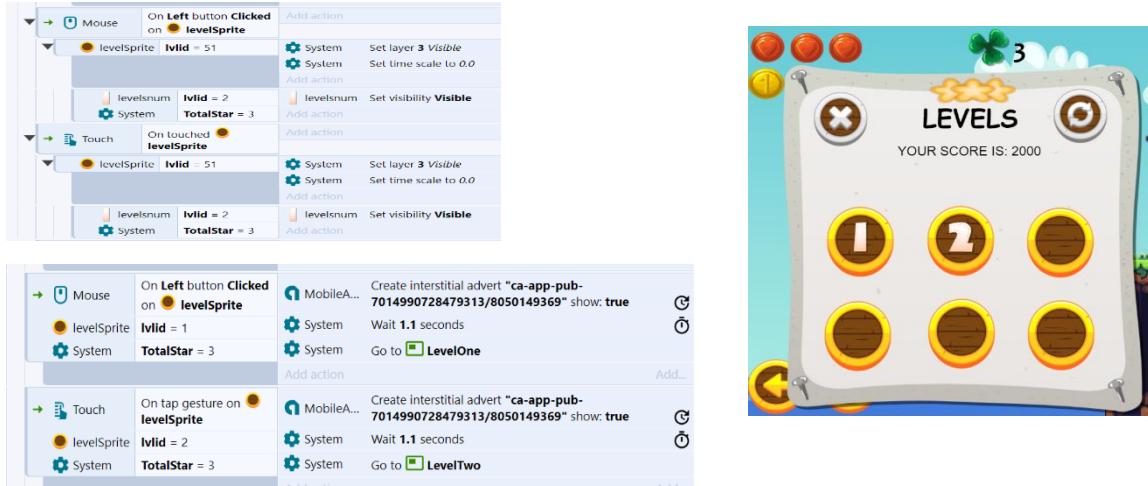
When the hero collides with the barrier, it will verify if the number of stars collected is 3.

If yes, the door animation will be set to 1, the sound "level_ends" will play and the "levelsmenu" layer will be visible with the total score and option to move to next level.

If not, the text warning the user to collect all stars will be spawned.

The levels menu on the layer 3 will be visible and the player will get to choose the next level only. So, if the level 1 ends the player can move to level 2 but not level 3.

These events are repeated throughout in all the layouts. The mouse click event is then used to enable the user to click to go to that level. Note also that the ads object will be explained later.



HUD

The HUD (heads-up display) is used so that it stays on the screen as the player moves throughout. This is done by setting the parallax setting to 0 x 0.



4.5. Powerups

The power ups are obtained in levels 3 to 6. As shown, the double jump power up will enable the user to jump twice higher as they can to collect unreachable objects.

The first image shows that the character cannot reach the leaf above. Once the power up is collected they then can jump higher and collect the leaf.



As the player collide with the power up it will enable double jump. A text will be displayed to the user to let them know that it has been enabled and the power up will be destroyed as it is collected.

14	shadow	On collision with sprint
		shadow Set Platform double-jump Enabled sprint Destroy power_t... Set visibility Visible power_t... Fade: start fade Audio Play powerup not looping at volume 0 dB (tag "") Add action Add...

4.6. Movable tiles

For the movable tiles, make sure to import these tiles differently from those already used. Once imported add the sine behavior to it and change it as needed, either horizontal or vertical.

horizontal_medieval_behaviors	
Name	Type
Solid	Solid
Sine	Sine

[Add new behavior](#)

Behaviors

Solid

Enabled

Tags

Sine

Movement Horizontal

Wave Sine

Period 1

Period random 0

Period offset 0

Period offset random 0

Magnitude 40

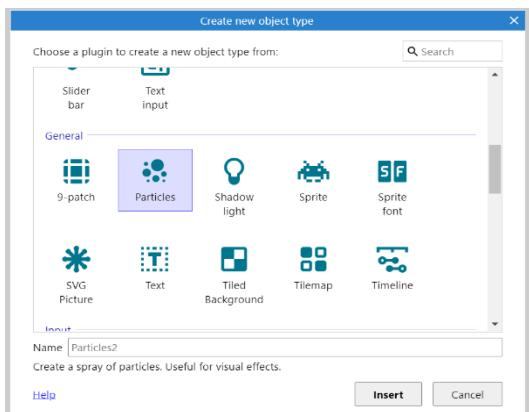
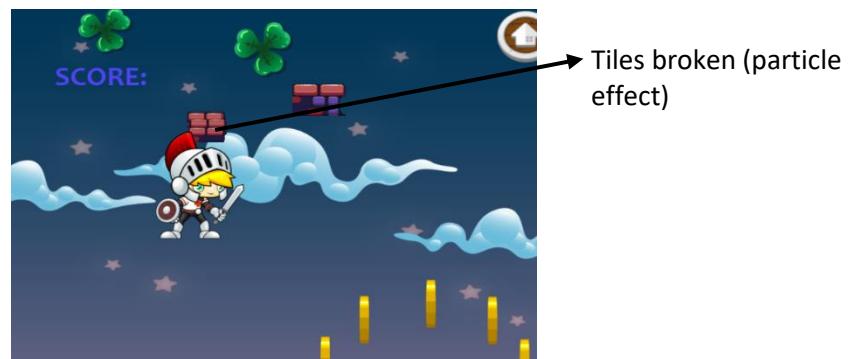
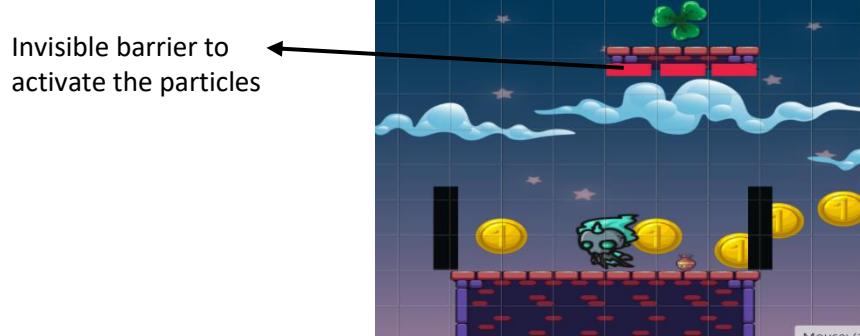
Magnitude random 0

Enabled

Preview

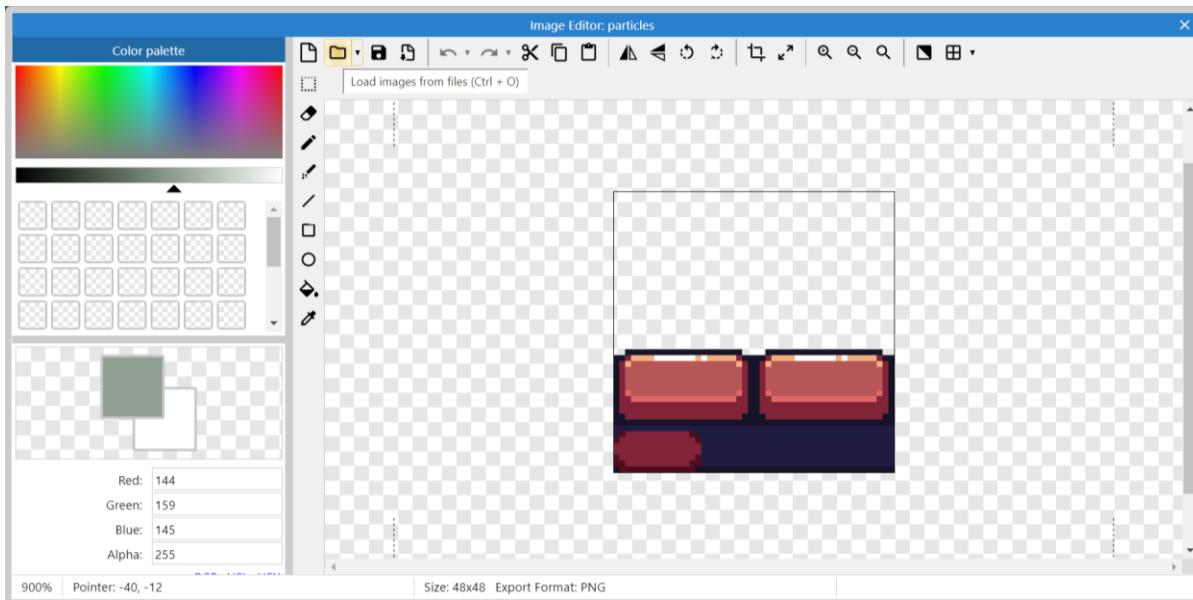
4.7. Particle effects

Some of the tiles are breakable. To reach some the player will first need to collect the power up in order to be able to break the tiles.



Step 1: First we create an object for the particles

Step 2: import the image that we intend to break



Step 3: inn the properties section of the particles these changes are made to make it fall with gravity and with the appropriate speed.

Two side-by-side screenshots of the Construct 3 Properties panel. The left panel shows the 'Effects' tab selected, with properties like Blend mode (Normal), Add / edit (Effects), Container (No container), Rate (20), Spray cone (60), Type (One-shot), Image (Edit), Object (checkbox), Preview (checkbox), and Initial particle properties (Speed: 200, Size: 32, Opacity: 100, etc.). The right panel shows the 'Initial particle properties' tab selected, with properties like Speed (200), Size (32), Opacity (100), Grow rate (0), X randomiser (0), Y randomiser (0), Speed randomiser (0), Size randomiser (0), Grow rate randomiser (0), Particle lifetime properties (Acceleration: -150, Gravity: 600, Angle randomiser: 0, Speed randomiser: 800, Opacity randomiser: 0, Destroy mode: Fade to invisible, Timeout: 1), and More information (Help).

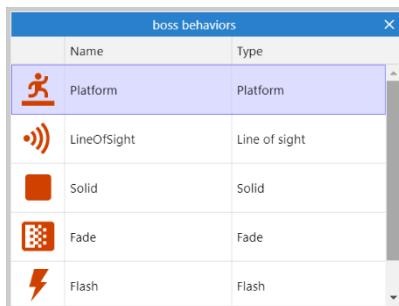
4.8. Boss Battle

Step 1: the enemy is imported and is given a bigger size compared to the wrath.

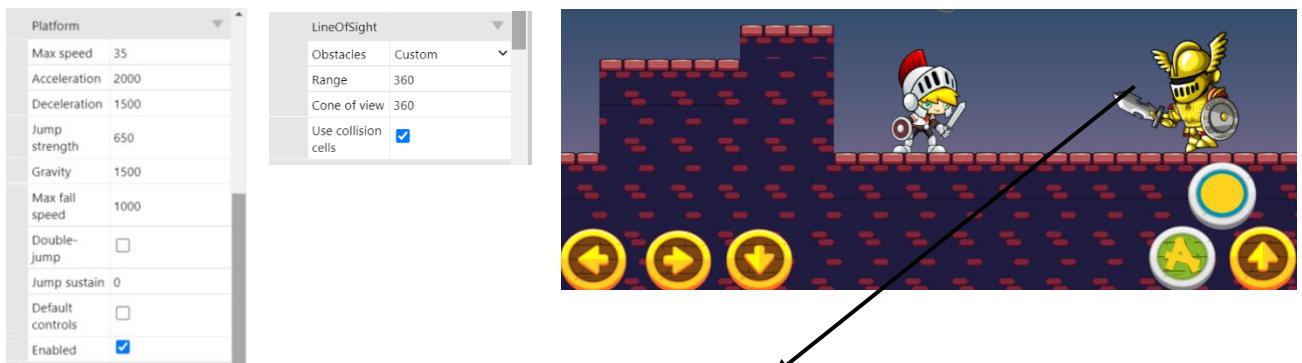


Step 2: the following behaviors are then added to it.

The line of sight will enable the enemy to see the hero within a certain range we set it to.



Step 3: the following changes are applied to its properties whereby the speed of movement of the enemy is slower compared to the other enemies and the line-of-sight range is set to 360 so that it sets an appropriate sight to the hero to it.



Moves as the hero gets to a certain distance

Step 4: the following events are added to the boss to make the line-of-sight work.

First the system with every tick is created and the distance is set for the hero and enemy X position.

```

graph TD
    S1[Set distance to distance(boss.X,boss.X,hero.X,hero.X)]
    S2[Simulate Platform pressing Left]
    S3[Set Mirrored]
    S4[Simulate Platform pressing Right]
    S5[Set not mirrored]
    S6[Set position to "run"]
    S7[Set animation to "boss_run_animation"]
    S8[Spawn fire on layer 1 (image point 1) (create hierarchy: False)]
    S9[Play fire_shoot not looping at volume 0 dB (tag "fire")]
    S10[Set visibility Visible]
    S11[Set angle toward (shadow.X, shadow.Y)]
    S12[Fade start fade]

    S1 --> S2
    S2 --> S3
    S3 --> S4
    S4 --> S5
    S5 --> S6
    S6 --> S7
    S7 --> S8
    S8 --> S9
    S9 --> S10
    S10 --> S11
    S11 --> S12
  
```

As soon as the hero get in the range of sight the enemy will move to the left (opposite) as the initial position of the enemy is to the right.



Here it will simulate the enemy controls if the distance at point X is greater to the distance to the hero shadow

For each 2 seconds, a small object will spawn and will be directed to the hero. The fade effect is also added to it so that it does not keep wandering.

```

graph TD
    C1[Has LineOfSight to shadow image point 0]
    C2[Set position to "idle"]
    C3[Set animation to "boss_idle_animation"]
    C4[Destroy]
    C5[Flash: Flash 0.1 on 0.1 off for 1.0 seconds]
    C6[Subtract 1 from TotalHealth]
    C7[Set text to "&TotalHealth"]
    C8[Play hero_hurt not looping at volume 0 dB (tag "")]

    C1 --> C2
    C2 --> C3
    C3 --> C4
    C4 --> C5
    C5 --> C6
    C6 --> C7
    C7 --> C8
  
```

The enemy goes back to idle animation if the line-of-sight is not in range anymore

1 health is removed with each collision with the "fire" spawned from the enemy

Step 5: the health of the boss is added. First, we set a global variable and set it to 20 (max). as the health decreases the health bar frames will also change and seen to be decreasing as well.



```

45 System boss_health = 20
46 System boss_health = 18
47 System boss_health = 15
48 System boss_health = 10
49 System boss_health = 8
50 System boss_health = 4
51 System boss_health = 0
52 System Trigger once
    Add action
        Set animation frame to 1
    Add action
        Set animation frame to 2
    Add action
        Set animation frame to 3
    Add action
        Set animation frame to 4
    Add action
        Set animation frame to 5
    Add action
        Set animation frame to 6
    Add action
        Set animation frame to 7
    Add action
        Set 'bottle' muted
    Add action
        Resume tag "One-Bard-Band"
    Add action
        Destroy
    Add action
        finalBarrier Destroy
    Add action
        Set visibility Invisible
    Add action
        Play boss_die not looping at volume 0 dB (tag 'dieBoss')
    Add action
        Set "dieBoss"-muted

```



Step 6: in order for the boos health to decrease the hero is given a special ability at the level 6. Once the hero presses X on the keyboard or the “lifer” button a “lifer_bullet” will be spawned and directed to the boss only and will decrease its health by 1.

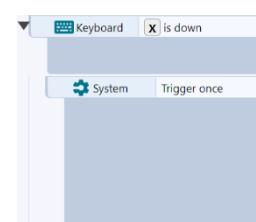


Lifer Button
(for mobile users)

Lifer bullet spawned
when X or the lifer
button is clicked



Barrier to
prevent the
hero to escape
without killing
the final boss.
Once the boss
eliminated the
barrier will be
destroyed.



```

Keyboard X is down
    hero Set animation to "lifer_hero_animation" (play from beginning)
    Add action
        System Trigger once
            Add action
                shadow Spawn lifer on layer 1 (image point 3) (create hierarchy: False)
                Add action
                    Play lifer_shoot not looping at volume 0 dB (tag "lifer")
                    Set visibility Visible
                Add action
                    Set position to (shadowX, shadowY)
                Add action
                    lifer Set angle toward (boss.X, boss.Y)
                Add action
                    lifer Fade: start fade

```

The user will get a text before
fighting the final boss whereby
they will have to press the down
key to get to the final battle.




```

shadow Is overlapping barriers goDown-- Set visibility Visible
    Add action
        hero Flash: Flash 0.1 on 0.1 off for 1.0 seconds
        Add action
            Systems Wait 1.3 seconds
            Add action
                shadow Set position to (2030, 367)
            Add action
                hero Set position to (shadowX, shadowY)
            Add action
                hero Flash: Flash 0.1 on 0.1 off for 1.0 seconds
    Add action
        hero On tap gesture on goDown
            Add action
                Systems Wait 1.3 seconds
                Add action
                    shadow Set position to (2030, 367)
                Add action
                    hero Set position to (shadowX, shadowY)
                Add action
                    hero Flash: Flash 0.1 on 0.1 off for 1.0 seconds

```

4.9. Mobile touches

The following images have been used for touch purposes for mobile devices.



The main controls such as running, attack, jump, etc. have been included as a touch event by using the touch object included with construct 3.

Mobile Controls

Touch	Is touching rightArrow	shadow	Simulate Platform pressing Right
hero	Platform is on floor	hero	Set animation to "run_hero_animation" (play from beginning)
hero	attack = 0	hero	Set Not mirrored
die = 0		hero	Set direction to "Right"
Add action			
Touch	Is touching leftArrow	shadow	Simulate Platform pressing Left
hero	Platform is on floor	hero	Set animation to "run_hero_animation" (play from beginning)
hero	attack = 0	hero	Set Mirrored
die = 0		hero	Set direction to "Left"
Add action			
Touch	Is touching jump_arrow	shadow	Simulate Platform pressing Jump
hero	attack = 0	hero	Set animation to "jump_hero_animation" (play from beginning)
die = 0		hero	Add action
Touch	Is touching rightArrow	shadow	Simulate Platform pressing Right
hero		hero	Add action
Touch	Is touching jump_arrow	shadow	Simulate Platform pressing Jump
hero	attack = 0	hero	Set animation to "jump_hero_animation" (play from beginning)
die = 0		hero	Add action
Touch	Is touching leftArrow	shadow	Simulate Platform pressing Left
hero		hero	Add action
Touch	On tap gesture on attack_arrow	hero	Set attack to 1
			Add action

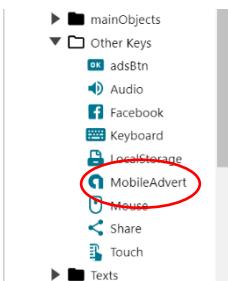
For touch movements we simply have to add the events of simulation (either right, left and jump) with the appropriate animation added.

Note that wherever the mouse event has been added the touch event should also be added.

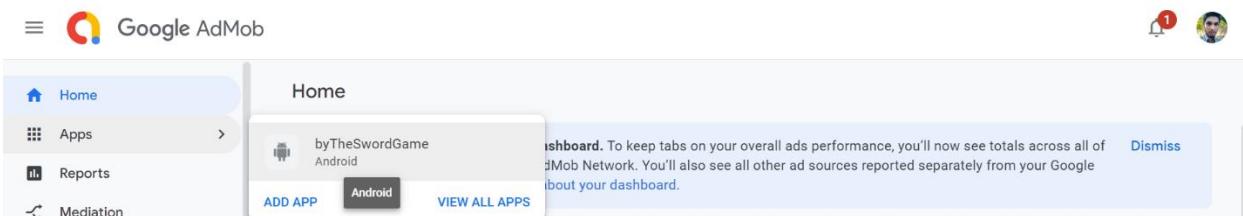
Touch	On tap gesture on levelSprite	MobileAd... Create interstitial advert "ca-app-pub-7014990728479313/8050149369" show: true
levelSprite	IvId = 2	System Wait 1.1 seconds
System	TotalStar = 3	System Go to LevelTwo

4.10. Monetization

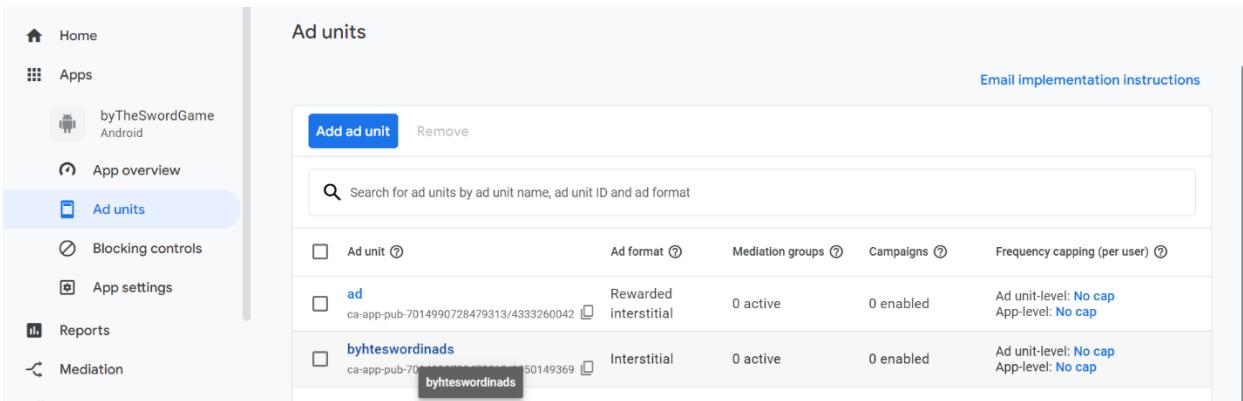
Step 1: first we import the advert object as follows



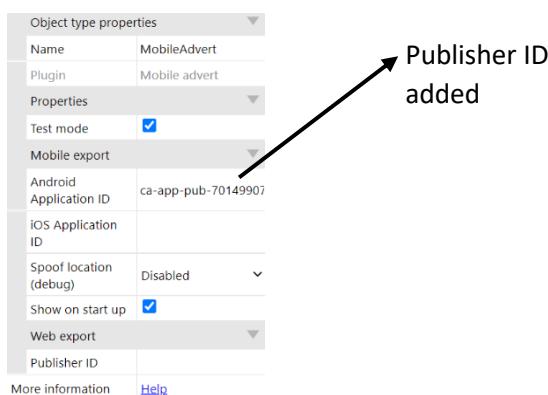
Step 2: an account is created on admob and a new app is added on the admob site



Step 3: for my game an interstitial ad is found to be more appropriate. In the ad units' section, an interstitial is created. Some forms need to be filled in order to get the android application ID for that ad will be provided.



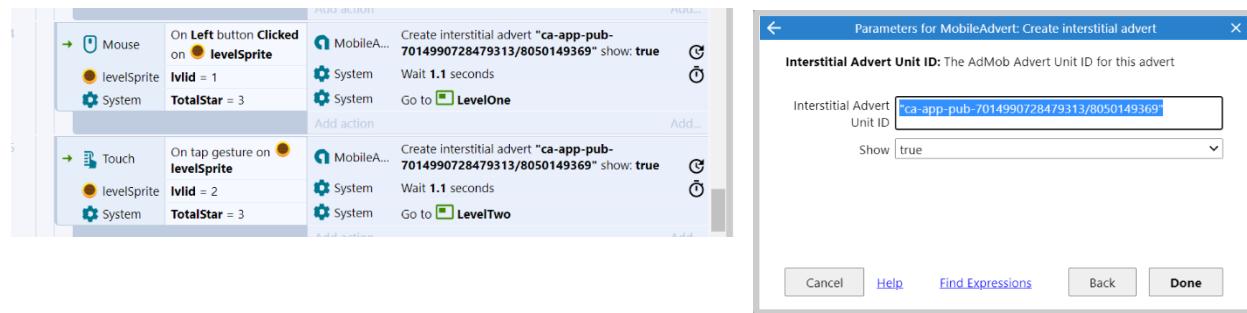
Step 4: the android application id for the publisher id are added to construct 3 mobile advert properties, as seen below.



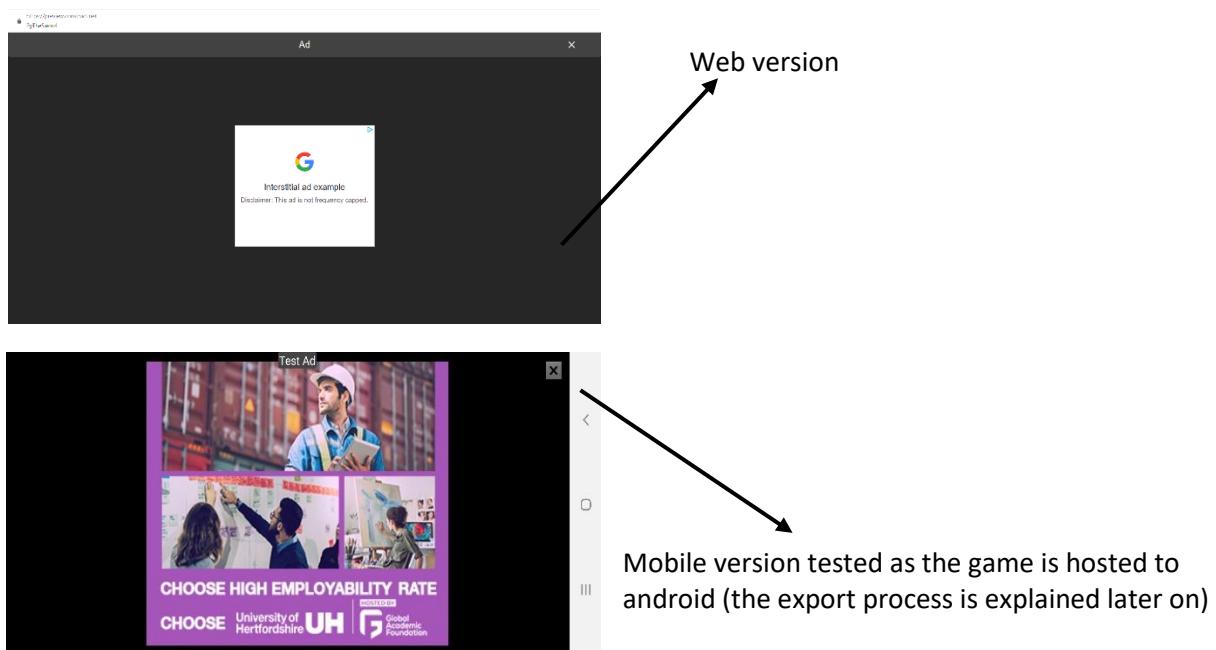
Step 5: for the game the idea is as follows

- The user finishes a level and the levels menu becomes visible
- The user clicks or touches on the level they need to go
- Before the level is shown an interstitial ad is shown

Once the event is added the interstitial id has to be added as follows

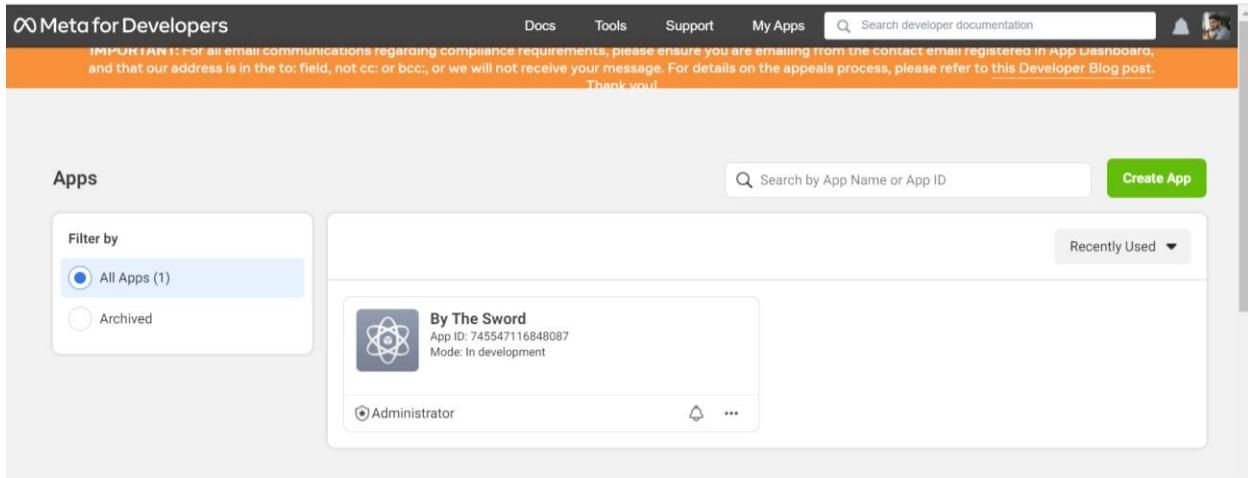


The following result has been generated from web and mobile



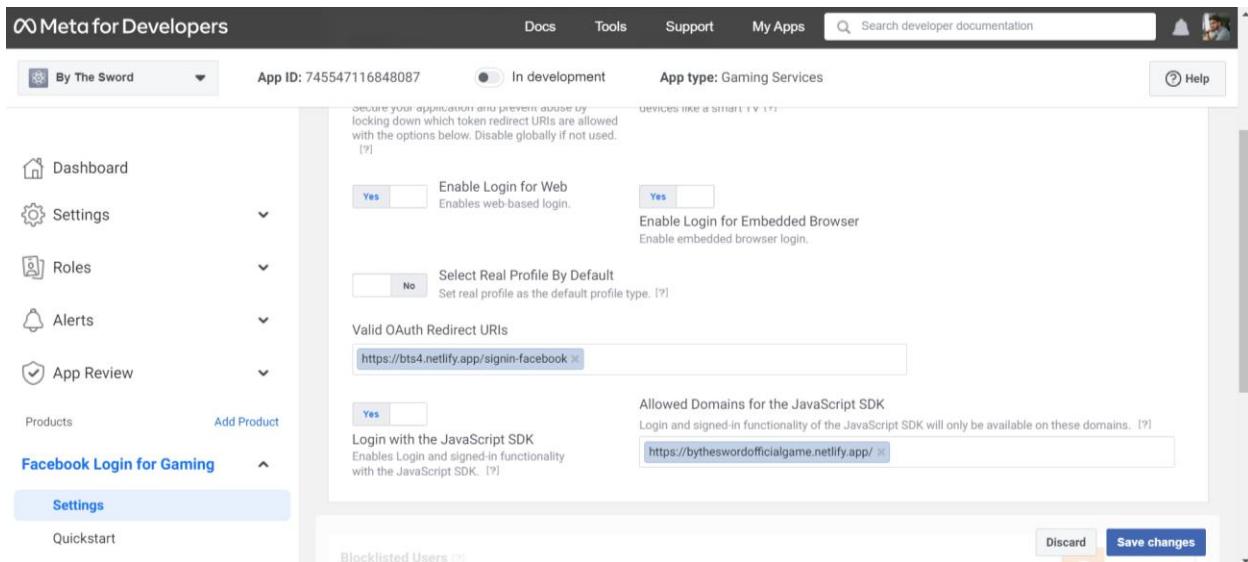
4.11. Social media integration

Step 1: An account is created on <https://developers.facebook.com/> and an app is created.



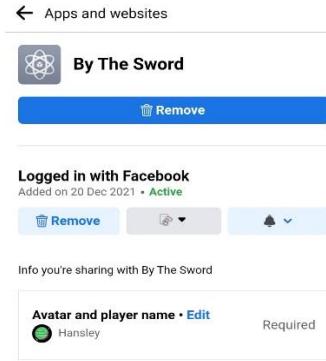
The screenshot shows the Facebook Developers App dashboard. At the top, there's a navigation bar with links for Docs, Tools, Support, and My Apps, along with a search bar. Below the navigation is an orange banner with a message about email communication compliance. The main area is titled 'Apps' and contains a search bar and a 'Create App' button. On the left, there's a filter sidebar with 'All Apps (1)' selected and an 'Archived' option. The main content area displays a single app entry: 'By The Sword' (App ID: 745547116848087), which is in development mode and set to Administrator. There are also three dots for more options.

Step 2: add the option of Facebook for gaming and enable the JavaScript SDK. The “Valid OAuth Redirect URIs” and the “Allowed Domains for the JavaScript SDK” are mandatory to be filled with valid hosting links else the Facebook integration will not work. The changes are then saved.



The screenshot shows the 'By The Sword' app settings page. The left sidebar lists various sections like Dashboard, Settings, Roles, Alerts, App Review, Products, and Facebook Login for Gaming. Under 'Facebook Login for Gaming', the 'Settings' tab is selected. The main content area has tabs for 'In development' and 'App type: Gaming Services'. It includes sections for enabling login methods (Web and Embedded Browser), selecting a default profile, setting valid OAuth redirect URIs (with a field containing 'https://bts4.netlify.app/signin-facebook'), and specifying allowed domains for the JavaScript SDK (with a field containing 'https://bytheswordofficialgame.netlify.app/'). At the bottom right, there are 'Discard' and 'Save changes' buttons.

Step 3: the Facebook object is added and placed in the event. As the user clicks or touches the fb button, it will trigger the login option. A text field has been created to get the full name of the user. After successful login we get the full name of the user and set that text to the text field.



4.12. Sounds & Audio

The following sounds and audio are imported to their respective folder. The format will be automatically changed to .webm

A name is given to each sound and audio so as to easily manage each of them for their particular usage throughout the game. One example is given below.

The screenshot shows the Scratch workspace with the "Sounds" and "Music" folders open. The "Sounds" folder contains files like "boss_die.webm", "break.webm", etc. The "Music" folder contains files like "battle.webm", "One-Bard-Band.webm", etc. To the right, a "Parameters for Audio" dialog is open, showing a "Tag" input field with "One-Bard-Band" and a "State" dropdown set to "Resume".

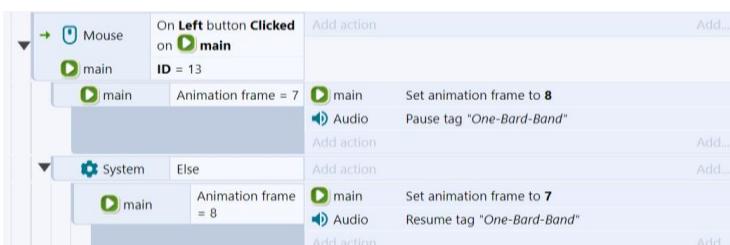
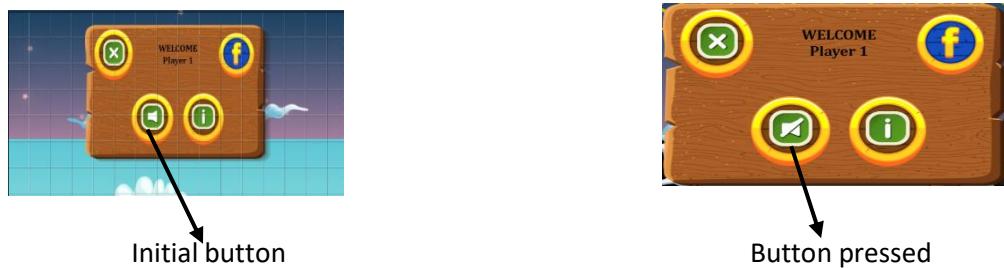
4.13. Main menu

The buttons are from the same imported frames, hence rather than importing each one separately, we simply add an instance variable to each of them called “ID” and set each one differently. Then we trigger that event based on that ID.

To go to the first level the ID = 1 is used and the event will simply be the system go to that certain layout.



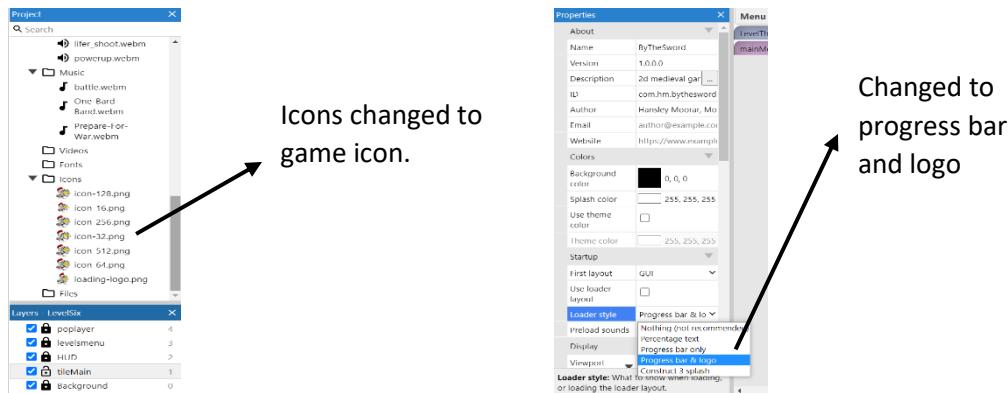
For the sound, the frame is changed when the button is clicked and the sound is either paused or resumed depending on that particular frame.



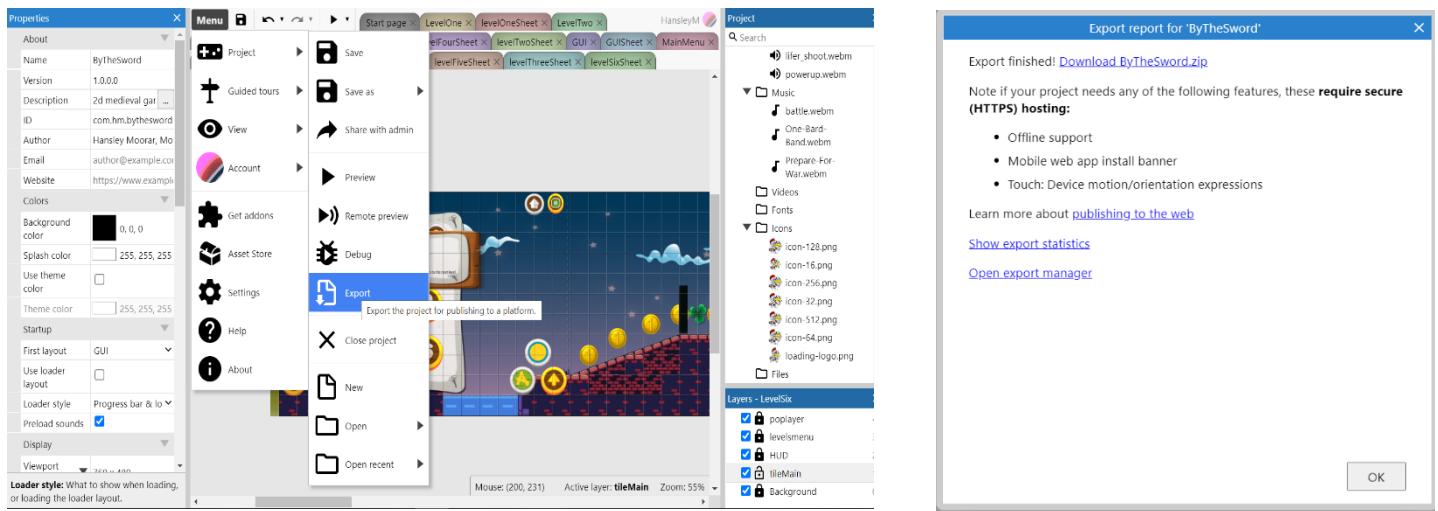
5. Exports

5.1. Exporting to web version

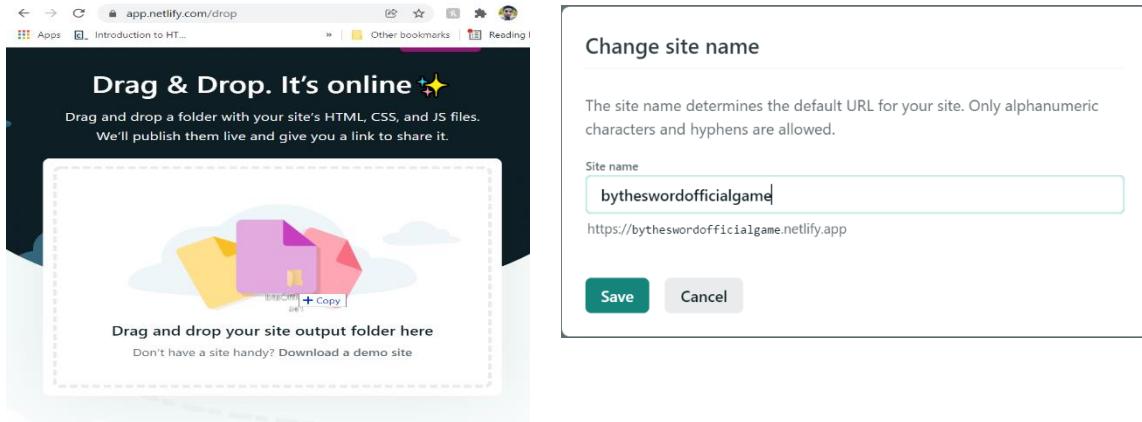
Step 1: first of all, make sure you have changed the icons preloaded to construct 3 icons to the icon you intend to use. Make sure to respect all the sizes else the game will not work. Also make sure that you have selected the loader style in the project properties.



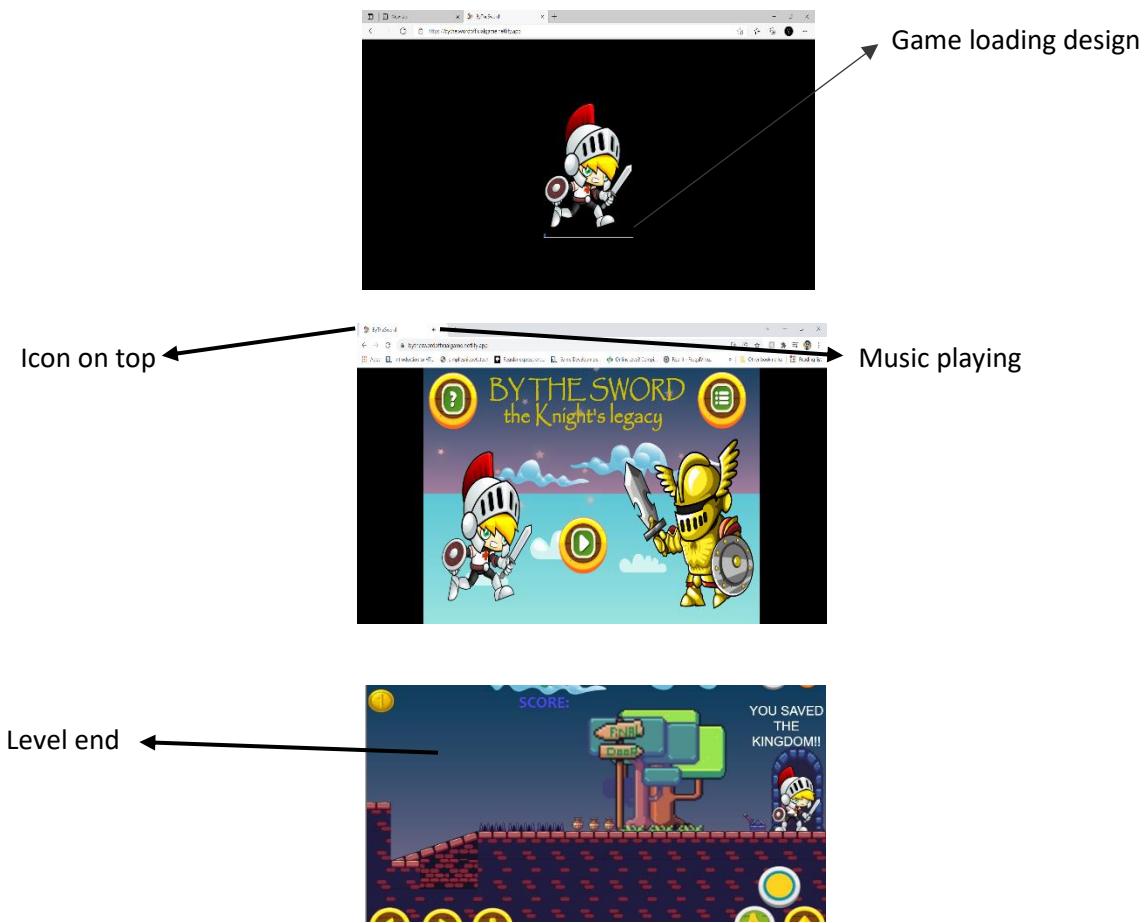
Step 2: to export to the web first we go our menu, project and export. From there you need to pick web version. You will get a zip file containing your index, script, images etc. Download the zip file and extract the contents in a normal folder.



Step 3: to host the game now, browse to <https://app.netlify.com/drop>. To use the website, you will need to register first. Next is to simply drag and drop the unextracted folder to it. Then go to the site settings and change the name of your website. Make sure that the name is the same as the one given to the Facebook URL in the Facebook developer page. The link can now be opened.

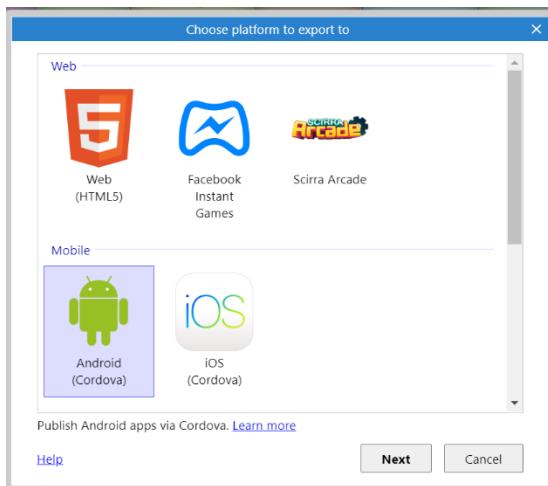


Screenshots of the game when hosted



5.2. Exporting to mobile version

Step 1: similarly to the web version, the mobile version is exported. Make sure the publisher ID and interstitial ID have been correctly added else the game may crash. Go to your project, then export and choose android from there.



Step 2: here however I have chosen the signed release apk. By choosing this you will have to create a keystore. Once created, download the jks file and add it to the signing options and click next. Your file will then be exported to android and an apk file will be generated.

The three screenshots illustrate the process of creating a signed release APK:

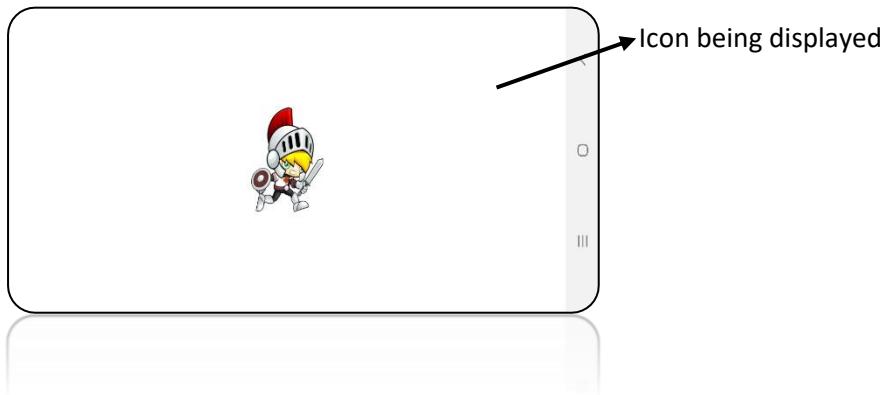
- Cordova options (Left):** Shows the "General" tab with "Min. version" set to "5.1+ (Lollipop)" and "Android build" set to "Signed release APK". The "Properties" tab includes a URL whitelist and permission checkboxes. A red circle highlights the "Signed release APK" dropdown.
- Create a new keystore (Middle):** A dialog for generating a keystore. It asks for First and Last Name (John Doe), Organizational Unit (Mobile Game Development), Organization (My Game Company), City or Locality (San Francisco), State or Province (California), Country Code (US), Alias (production), Validity (years) (25), and Keystore Password/Confirm Password. It also has checkboxes for Hide status bar, Require Vibrate permission, Require Camera permission, Require Microphone permission, and Require External storage write permission.
- Cordova options (Right):** Shows the "Signing" tab with a "keystore.jks" entry, key alias "hm", and key password "*****". It also includes checkboxes for Hide status bar, Require Vibrate permission, Require Camera permission, Require Microphone permission, and Require External storage write permission.

Step 3: download the apk file and test it either on android studio or on a mobile phone.

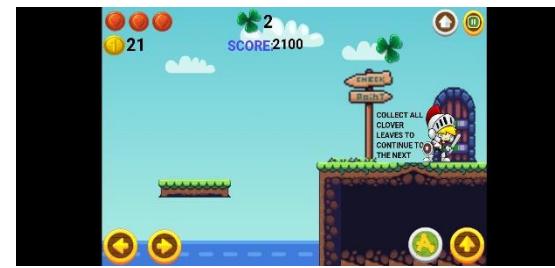
Mobile application logo after installation on mobile phone



Icon being displayed

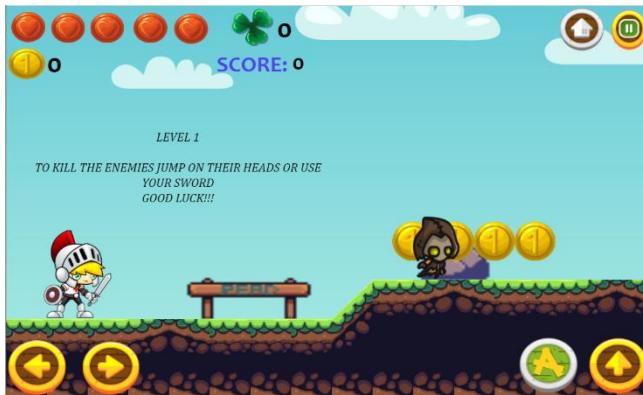


Screenshots of the game on android phone

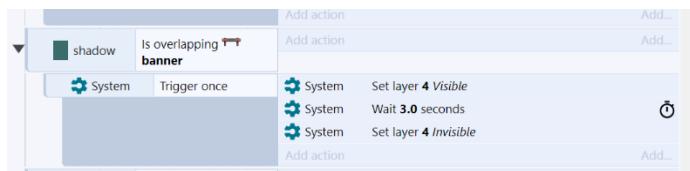


6. Other

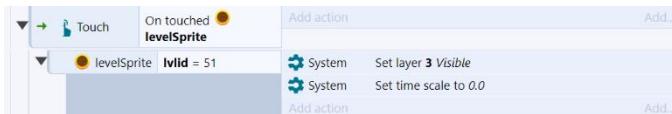
- A text will be seen at each level to let the player know certain tips.



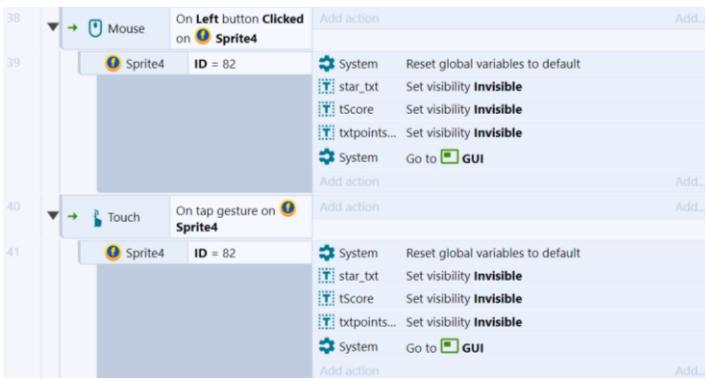
- As the player moves forward a little bit an overlap with the barrier the controls of the game will be displayed



- Pause menu, the time scale is set to 0 so that the background is frozen at that time.



- The move menu will trigger the user back to the splash screen.



7. Bugs and fixes

Bug

- At the level 4 and above, the user may accidentally touch the screen anywhere in the middle that may restart or trigger another level. This is due because the “levelsmenu” is only set to invisible which means that if a user accidentally clicks on that level trigger they will be automatically be sent to that level.

Fixes

- To fix this the “levelsmenu” is set to disable as the levels are in progress and set back to enable is the user pauses the game or finishes a level.

Bug

- At some of the slope tiles in the game the user seems to be stuck on something. This is due because of the collision polygon that is colliding with the player’s collision polygon.

Fixes

- We simply need to fix the collision polygon of the slope to be forming the shape of the slope itself.

8. Further work

1. Characters

The next step for the game is maybe more playable characters rather than only one.

2. Paid items

Users will be able to collect a certain amount of money or items in the levels and with these, be able to buy powerups, new clothes, items, boosts, etc.

3. More levels

An increase in levels will be nice so that the game does not feel too easy or stays idle once the player has completed all the levels.

4. Google Playstore

The next step of the game will be to publish the version on playstore for user to be able to download and play the game both nationally and internationally.

9. SWOT analysis

Strength	Weaknesses
<ul style="list-style-type: none">• Appealing design• Easy to understand• Addictive• Easy mobile touches• Users are not very loyal due to many similar games	<ul style="list-style-type: none">• It's difficult to maintain the game new & fresh and keep people coming back• High on graphics• Some minor lag only at the start of the 1st level

10. Conclusion

To summarize, this initiative assisted us in quickly tapping into undiscovered technologies and discovering a simpler and more fascinating approach to create these exceptional games. We simply handled our work as a team and had a lot of fun working on build 3. As a result of our examination of this assignment, we can conclude that we have acquired the essential abilities to quickly develop a platformer game and that we may now explore the unlimited possibilities that can occur from creating different games.

References and credits

Other game references obtained on:

<https://www.miniclip.com/games/en/>

Free assets obtained on these websites:

<https://www.gameart2d.com/the-knight-free-sprites.html>

<https://craftpix.net/freebies/2d-fantasy-knight-free-sprite-sheets/>

<https://craftpix.net/freebies/free-wraith-tiny-style-2d-sprites/>

<https://craftpix.net/freebies/free-medieval-tileset-pixel-art-pack/>

<https://www.gameart2d.com/free-medieval-game-button-pack.html>

<https://craftpix.net/freebies/nature-pixel-art-environment-free-assets-pack/>

<https://craftpix.net/freebies/free-game-coins-sprite-sheets/>

<https://craftpix.net/freebies/free-cartoon-parallax-2d-backgrounds/>

<https://craftpix.net/freebies/free-jungle-cartoon-2d-game-ui/>

Free music obtained on these websites

One Bard Band by Alexander Nakarada |

Prepare For War by Alexander Nakarada |

The Great Battle by Alexander Nakarada | <https://www.serpentssoundstudios.com>

<https://www.serpentssoundstudios.com>

Music promoted by <https://www.chosic.com/free-music/all/>

Creative Commons CC BY 4.0

<https://creativecommons.org/licenses/by/4.0/>

Free sounds obtained on these websites

<https://www.zapsplat.com>