**DragonBall Switch**

**Introduction**

A single-player 2d side-scroller action adventure platformer game. The project is inspired by similar platformers such as Mario and the popular Japanese animated series “Dragonball Z”. The player will progress through the level, collecting dragonballs and defeating monsters. The player will be able to switch between two main characters, both with their own unique sounds and animations. Project is only localised with the English language.

The project is developed in Unity. Our reason for choosing this project, it is a change of pace from the other module projects that have been appointed so far. We both are fans of DragonBall Z and have not seen a similar character switching mechanic in the DragonBall Z games genre. Our target audience for this project was younger generation 35 and under.

**Minimum System Requirements**

* PC with Windows 7 or above
* 1gb of Ram
* 400mb of Hard Disk Space
* Unity 2018.3.7f1

**Recommended System Requirements**

* PC with Windows 7 or above
* 2gb of Ram
* 500mb of Hard Disk Space
* Unity 2018.3.7f1

**Technology**

For our main engine we used Unity engine. The game is developed in 2D. The main reason we chose the unity engine is because it is based upon the C# language, which we are familiar with from other modules and due to its beginner friendly nature. Unity is a common engine that is used in the production of today’s games. Games such as Shovel Knight, Hearthstone and Ori and the Blind Forest are manufactured using Unity and are very successful.

We used GitHub as our version control. GitHub is free software that can be accessed by anybody. We used GitHub Desktop to commit/push and pull our code into a main repository that can be found on GitHub’s online website. This was an easy collaboration tool to use between us.

We used Visual Studio 2017 as our main IDE.

We used Adobe Photoshop CC 2015 as our sprite editor. Manipulated free images found on the internet to fit into our game.

**References**

We examined games that are similar to our idea. This was mainly to examine games to see what makes 2-D platformers enjoyable and successful.

**Shovel Knights**



Shovel Knights core features are it is a single player game. It is 2D a platformer. Contains numerous enemies and bosses that will thrive the user to proceed in the game upon receiving loot from these harder difficulty enemies. It will put the users developed skills from playing to use.

The game was published in 2014 on multiple platforms upon successful funding through KickStarter. It has won multiple awards due to its engaging gameplay.  
  
From this game, we took inspiration from the beautiful art style, level design and objective design such as killing a difficult enemy at the end of the level.

**Castlevania**



Castlevania’s core features are it is a single player game. It was developed in 1987 as a 2D action platformer. The Castlevania genre was so successful that is published dozens of games after their first. It lured its audience with its engaging worlds and hard difficult gameplay. The game rewarded the player while he battled through the levels.

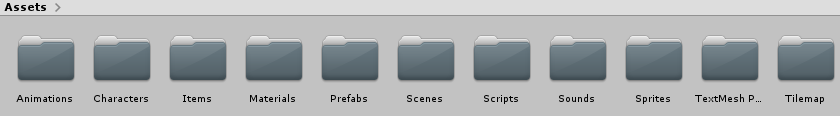
From this game, we took inspiration from the UI design and level design.

**Design Methodology**

For this project we used a Bottum up Design. We began with creating a basic level and then populating it with our initial player sprite. Then adding scripts to this sprite to allow for movement etc. We incrementally tacked on all the necessary features, e.g. Movement, Attack, Animation, Second character, Enemies etc.

For organization we used a Tree like storage method in which there is one main folder called “DragonBall Switch” that contains all the sub folders that make up the solution.

E.g. Assets Folder



Forward into the “Assets” folder contains all the material is used inside the solution, this includes; Sprites, Animations, Animation controllers, Music, Sound Effects, Prefabs and Scripts.

E.g. Subfolder



**Features of the Implementation**

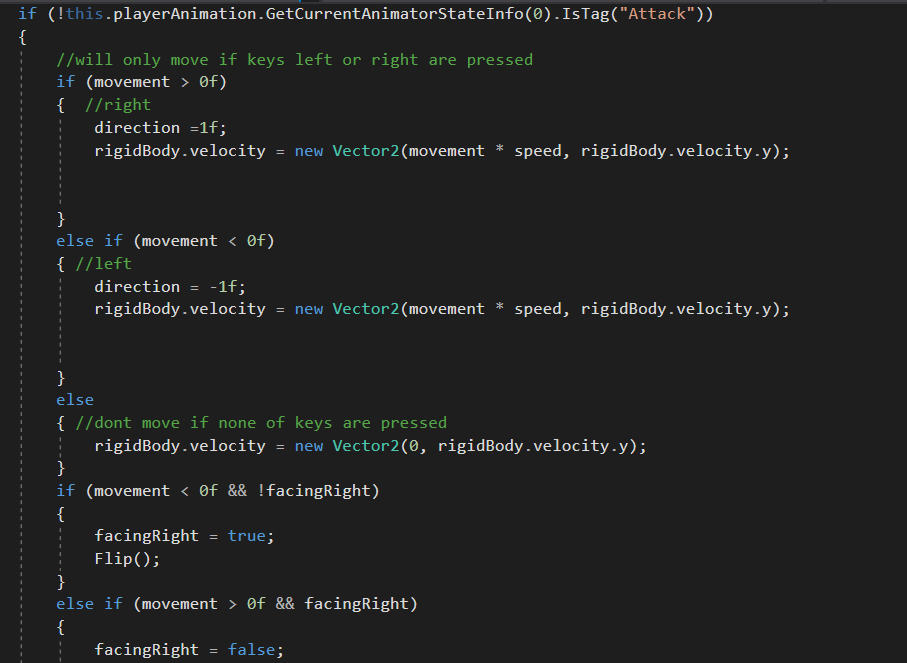
* High quality player animations

The player character is represented by fluid, aesthetically pleasing and stylistically accurate sprites which are displayed and controlled by an animation controller, this controller contains all the relevant sprites and animations for the player to function, this controller is called in the games scripts and then plays the correct animation.



* Responsive player movement

The player object is controlled by a Player Move script. This script contains all of the player actions, it uses a multitude of different functions, IEnumators and variables to translate the players intentions to the unity engine and in turn provide a satisfactory playing experience, For example if the user presses to move to the right, e.g.(getKey(..)) the player character will begin to move and slowly build momentum, upon releasing this key the player character will quickly lose momentum and come to a stop.



* Enemy Artificial Intelligence

The enemy will automatically begin to follow the player upon entering view, once it gets to a certain distance from the player it will begin firing a projectile attack, if the player closes the gap to melee attack the enemy then the AI will either counter attack or begin to retreat. Upon leaving the view the enemy’s script is disabled, this is for optimization purposes as it prevents an endless barrage of projectiles coming at the player and saves memory. The enemy AI is automatically reenabled upon being seen again.

* Collectables & Score System

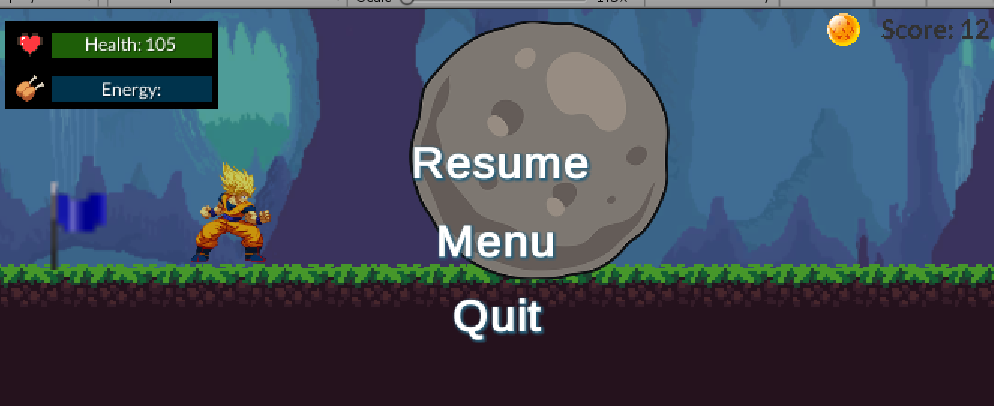
One of the secondary goals of the game is to increase the players score. This is done by picking up the many collectables, in this case the titular ‘Dragon Ball’. These are scattered throughout the map, some in unconventional places in order to encourage player exploration. Upon picking up an item the players score is updated and the collectable is destroyed.



* Menus

The game is designed with different menus to aid the gameplay and navigation throughout the game. A series of menus are presented to the user once the game is launched and while user navigates the user interface. A Pause menu is present once game has begun by pressing down on the “esc” key. The





**Gameplay and Setting**

The game is centered around two characters who can switch in a tag team manner with a press of a button to fight their way through the levels and boss encounters.

The world revolves around two characters called “Goku” and “Vegeta” who are both “sayians”, a type of superhumans with special abilities (Vegeta left, Goku right). 

The game consists of one level which is populated and infested with saibamen who are green, mindless zombie like creatures in the Dragonball Z universe. At the end, a powerful battle awaits between an arch nemesis called “Broly”.

The player will have to dodge incoming projectiles from the enemies while also   
trying to fire back towards them. The player will have to manage his health and energy   
when using abilities. Picking up spawned food throughout the level replenishes energy and health.

The controls of the game are designed to be compatible with a keyboard and a mouse. They are ergonomically designed. Special character abilities and combos are triggered by Z,X,C,V,F keys while assisted with character specific animations and sounds. The player can test out all of these abilities in the Tutorial area of the game.



**Limitations and Known Bugs**

The game currently is in a playable state, known bugs are:

* Enemies linger for a second too long when killed.
* Boss AI does not function 100%.
* Special attack does not damage enemies.
* Player respawn bugs out.
* Spamming Jump can climb up any wall.

**Testing**

The game has been tested and played by family and friends. Each had their own feedback, which consisted of further development ideas and positive feedback. The game has been thoroughly tested by us through various interactions with the enemies and the environment.

**Recommendations for Future Development**

* Implement an inventory system, so that the player can store pickups instead of having to use them upon pickup.
* A shop system so that the player can use the points acquired from picking up dragonballs to purchase either Health items or possibly, cosmetic items.
* Additional states, so the player can possibly upgrade the character, making them more powerful in certain aspects (e.g. Upgrade char 1’s melee power, Upgrade char 2’s fireball power.)
* Destructible environments, possibly using sprite masks allowing for more complex level design such as hidden pathways, hidden items and such.
* More enemy variation.
* Addition of a timer, on which expiring the game ends.
* Additional Boss AI states.
* Difficulty settings (e.g. Easy: Enemy damage is reduced 50%, Hard: Enemy damage and health increased.)
* More sound variation, (e.g. If attack animation 1 plays, play a random sound an array rather than a static sound each time.)
* More levels
* An offline multiplayer mode in which each player picks one of the two main characters and can play in a versus battle.

**Conclusions**

**Jake:** Upon completion of this group project I must say that I am happy with the work we have produced. The project proved to be challenging both in terms of coding as well as collaboration however it was a challenge I enjoyed, and I feel am much better off for completing. The coding aspect of the project was the most challenging part, having so many scripts running side by side, figuring new functions of the engine from unity was hard but ultimately rewarding as it was a different style of coding than we are used to from the labs/projects. Gathering the resources such as sprites, backgrounds and creating the sound assets were the enjoyable as it provided some respite from the coding. I feel that we managed to combine the two quiet well, resulting in a fluid responsive game that can be built upon. I feel our collaboration was great, myself and Tomas worked together quite well and were often on the same wave length when it came to the project and the direction it was heading. The only taxing part of the collaboration was when either of us didn’t comment code adequately or didn’t pull the repository often enough or notify the other that we had pushed etc. These however were no real issue as they were very easily resolved. This module has improved my ability to work as part of a team and taught me more about using GitHub as part of a group.

**Tomas:**  This project has been a learning experience with the use of Unity engine. I’ve learnt numerous unity functions and my understanding of C# programming language has increased. I am now able to produce a 2D unity game and have set up my knowledge for any future 2D unity projects. This is a field I am interested in and may adventure into in the future as a career or a hobby at the least. My scheduling and teamwork skills have improved. As I am a person who procrastinates a lot with projects and assignments, the project has taught me to take a manageable workload and tackle it week by week. Collaborating with another fellow course student has thought me to work as a team in order to tackle problems. We communicated often and had different ideas to tackle any problems/bugs that have risen. I have also learnt how to use photoshop to mask and render images with transparent backgrounds. Overall, I am pleased with the work we have accomplished as a team, we came in with little to no Unity knowledge and produced a playable game within the given time.