**Game Development Documentation.**

**Release Title:** The Black Samurai

**Differences between Final rendition and the initial Game Concept Outline:**

Dual Axis Directional attacking control scheme was not implemented.

This was to be one of the games major features and was only left out of the final rendition due to a shortage of time available to combat the learning curve required to correctly implement this functionality. Because I had tried to maintain the cross platform input capabilities and also due to the complexity of the animations this function of the game proved too difficult to implement within the timeframe. Therefore only basic attacking system was used.

Additional Physics effects with player movement (picking up speed down hills to add to jump velocity)

Again due to time constraints and the learning curve I was unable to implement this feature. Player cannot pick up any speed to add to jump velocity. However the player can gain additional jump velocity running up hills to attain access to different areas. As a result this also affected the level design.

Level design changed with player physics.

The player physics used in the final edit resulted in a change of the level design from. Instead of holding a resemblance to Sonic with many sloping up and down platforms to add to speed and affect trajectory, the player now uses angled platforms to add vertical force to jump higher. The levels are constructed in more of a Mario style but with some added challenges in platform placement requiring use of the jump physics to progress.

**Feedback gathered from testing of early model:**

Overall the graphic and animations were extremely popular and added a lot of enjoyment in the game. Players really like the jump mechanics with the vertical velocity gain only. I made special not of the problem solving aspect with the level interaction as a couple of players found it impossible to continue. I believe this is due to the demographic they belonged to though. The testers that fall more into the target audience range found the problem solving aspect quite enjoyable and some of them even figured out what to do within seconds.

As the directional attacking was not fully functional it was disabled for the testing session. I instead opted to explain the idea of the original dual axis control scheme and they all said that would be a wonderful idea and agreed that it would port over to touch screen controls very fluidly. Unfortunately this was not implemented in the final rendition of the game due to dime constraints.

During Testing Air control for the player was disabled as I considered the problem solving aspect of the levels to be too easy with it on. Each player got to test the game again with air control turned on and they seems to enjoy it more whilst still finding the puzzle parts challenging. Air control is enabled in the final rendition.

Some bugs were noticed during testing where the one way box collider scripts (to allow player to jump through platforms from underneath) were also being triggered by the enemies. This means that the player was unable to stand on a platform while an enemy was walking underneath it.

**Asset List**

**Major starting components used:**

“Standard Assets for Unity 4.6” from the Unity Asset store to begin with a working character acquired from, <<https://www.assetstore.unity3d.com/en/#!/content/21064>>

From this assets pack I used;

* Camera follow script (Camera2DFollow.cs)
  + Modified scripts behaviour and added a Y offset feature
* Player prefab (including scripts: PlatformerCharacter2D.cs, Platformer2DUserControl.cs)
  + Removed original art and added custom art.
  + Added custom animations.
  + Modified Scripts to handle my custom graphics and animations.
* KIllBox Prefab (Including Restarter.cs)

**All artwork completely created by myself from scratch:**

* UI Buttons on opening screen
* All Scenery
  + Grass
  + Mountains
  + The sun and clouds background image.
* Platforms (including all prefabs)
  + Houses
  + Walls
  + Huts
* Enemy Characters

**Artwork that initially came from other sources that were heavily modified and customised:**

* The background image on the opening game screen was acquired from, <<http://artemisdefenseinstitute.com/Blog/3261659>>
  + Heavily customised this image to word with wider screen resolutions and look more dramatic
* The Samurai Player Character chraphics was constructed from an image acquired from, <<http://www.gaksdesigns.com/The-Headless-Samurai>>
  + Dissected the image into pieces
  + Heavily customised from the original (lighting effect etc)
  + Drew in the gaps to fill out the body parts
  + Drew additional parts for extra limbs etc.

**Custom Scripts written completely by myself:**

* Enemy AI (EnemyBehavour.cs)
* UI Scrips from Opening Scene (OpeningScreenButtons.cs)
* Players Sword Damage (PlayerSword.cs)
* Live Debugging Information (Statistics.cs)
* One way platforms that player can jump through. (onewayPlatformTriger.cs)

**Custom Scripts written with the help on online tutorials viewd at <** [**http://brackeys.com/**](http://brackeys.com/)**>**

* Parallax effect with backgrounds (Parallaxing.cs)
* Tiling of backgrounds with camera movement (Tiling.cs)
* Arm rotations. *Note: wasn’t used in the end* (ArmRotation.cs)

**Music in opening screen (12. Rise Inside.mp3)**

Artist: Killswitch Engage

Album: Alive Or Just Breathing

Title: Rise Inside

**The Black Samurai User Guide. (PC Version)**

****

**You play as the Black Samurai.**

Fight to reclaim your honour and free Japan from the rule of a tyrant ruler. There sill be some problems on your path trying to get to the emperor. You must use your problem solving and jumping skills to figure out how to progress.

You can gain additional height from running up an incline and jumping at the end.

**Controls.**

The Samurai is controlled with either AWSD or arrow keys for movement. Space for jump. You can still change direction of movement while in the air. Use your mouse to attack by clicking.

