**D-Sync Production Report**

**Report Summary:**

* **Game Engine**
* **D-Sync**
* **Week 04**
* **Jacob Lim**

**Game Summary:**

* **Dragon Attack**
* **A 2D action platformer set in a medieval fantasy.**

**Team Roster:**

* **D-Sync**
* **Jacob Lim, Producer, Graphics, Game Object, Dragon**
* **Andrew Chong, Lead Designer, Physics Engine**
* **Javon Lee, Technical Director, Collision Engine**
* **William Yoong, Product Manager, AI**

**Project Summary:**

**Project Status:**

* **Green**

**Upcoming Milestone:**

* **Engine Proof, Alpha, Beta, Final**
* **16 February 2018**

**Accomplishments:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Jacob Lim** | | | |
| **Graphics Engine**  **Game Object**  **Dragon** | **Estimated hours spent** | **Percentage completed** | **Remarks** |
| Graphics Engine:  Code the Graphics Engine for the Game Engine | 20 | 90% | Slight UV tearing issues, otherwise, completed.  Will consult TAs. |
| Game Object:  Code the Game Object base class for the Game Engine Architecture. | 6 | 50% | May need to add data members to it, depending on others’ code. |
| Dragon:  Code the player-controlled character. | 6 | 50% |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Andrew Chong** | | | |
| **Physics Engine** | **Estimated hours spent** | **Percentage completed** | **Remarks** |
| Physics Engine:  Code the Physics Engine for the game. | 7 | 100% | May have problems based on how it integrates with the collision. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Javon Lee** | | | |
| **Collision Engine** | **Estimated hours spent** | **Percentage completed** | **Remarks** |
| Collision Engine:  Code the Collision Engine for the game. | 7 | 80% | Need to figure out algorithm for Circle with Rectangle collision |

|  |  |  |  |
| --- | --- | --- | --- |
| **William Yoong** | | | |
| **Game State Manager**  **Input Handler**  **Basic AI** | **Estimated hours spent** | **Percentage completed** | **Remarks** |
| Game State Manager:  Code the Game State Manager for the game.  It manages the handling of each state of the game. | 2 | 100% |  |
| Input Handler:  Code the Input Handler.  It manages the game’s input. | 2 | 90% |  |
| Basic AI:  Code basic enemy AI, such as the common grunt or soldier. | 11 | 20% | Need to figure out how to link collision, physics and behavior |

**Objectives:**

|  |  |  |
| --- | --- | --- |
| **Jacob Lim** | | |
| **Game Object**  **Dragon**  **Graphics Engine** | **Estimated hours required** | **Remarks** |
| Game Object:  Finish up the Game Object base class | 5 | Will push for more information on what’s needed for the base class from my team mates |
| Dragon:  Complete a working player character which can attack. | 10 |  |
| Graphics Engine:  Fix the UV tearing issue. | 4 | This is slightly lower priority as the engine has a higher priority right now. |

|  |  |  |
| --- | --- | --- |
| **Andrew Chong** | | |
| **Physics Engine** | **Estimated hours required** | **Remarks** |
| Physics Engine:  Create a basic physics engine with basic functions for gravity. | 10 |  |

|  |  |  |
| --- | --- | --- |
| **Javon Lee** | | |
| **Collision Engine** | **Estimated hours required** | **Remarks** |
| Collision Engine:  Create a basic collision engine with basic functions for collision detection. | 10 |  |

|  |  |  |
| --- | --- | --- |
| **William Yoong** | | |
| **King Arthur AI** | **Estimated hours required** | **Remarks** |
| King Arthur AI:  Code the behavior of the King Arthur Boss. | 10 | Need to figure out how to code his behavior. |

**Highlights: *(Optional)***

* N/A

**Lowlights: *(Optional especially when the status is yellow or red)***

* N/A

**Risks & Mitigations: *(Optional)***

* N/A

**Code Review:**

* We spent a total of 4 hours reviewing each other’s code.

**Under Performing Teammates, for the past week: *(If any)***

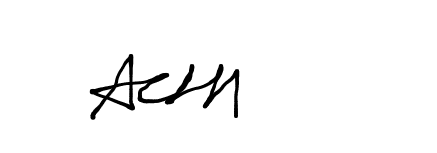
* None

**Signatures:**

**Jacob**

****

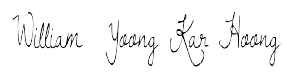
**Andrew**

****

**Javon**

****

**William**

****