**Developers Manual**

*Kirklind Signature*

Unity Version and download instructions:

Unity Version: **6** more specifically (**6000.0.34f1**)

Steps to download:

1. Visit the unity website <https://unity.com/download> .
2. Click the “download” button this will prompt the download of Unity Engine and Unity Hub.
3. Once this has been completed depending on the platform follow the same steps you would when downloading any other application from the web.
4. You can now open Unity hub and start installing unity engine (In top right corner select installs).
5. Once done you now can open and build unity projects.
   1. You also should have the most updated version of unity which means it should support building and running of games on older versions.

High Level View of Existing Code based on Class Diagrams



A class diagram for the SFX of the Warehouse Warriors game. Includes:

* AudioManager
  + MusicManager
    - EndScreenMusic
    - GameMusic
    - MainMenuMusic
  + SoundManager
    - DeathSound
    - PickupSound
    - ShootSound
    - Etc.

Summary: This code will focus on the SFX portion of the game making it more entertaining and enjoyable for the user to interact with.

A screenshot of a diagram

Description automatically generated

A class diagram for the Animations of the Warehouse Warriors game. Includes:

* Renderer
* AssetLoader
* Character
  + AnimationController
    - AnimationState
      * AttackAnimation
      * IdleAnimation
      * WalkingAnimation

Summary: This code will focus on the Animation portion of the game making it more visually appealing and also more sophisticated.



A class diagram for the Enemies of the Warehouse Warriors game. Includes:

* RangedEnemy
* MeeleEnemy
  + Enemy
    - Vector
    - PlayerProjectile
    - EnemyProjectile
  + Player

Summary: This code will focus on the Enemy portion of the game dealing with their movement and damage mechanics. This will make the game into a shooter and complete the MVP of the project.



A class diagram for the Menus of the Warehouse Warriors game. Includes:

* Menu
  + HelpMenu
  + PauseMenu
  + HomeMenu
  + I\_Menu
* GameManger

Summary: This code will focus on the Menu portion of the game making it so the user can navigate the game easily and intuitively.



A class diagram for the SFX of the Warehouse Warriors game. Includes:

* GameObject Maner
  + PlayerController
    - Health
  + Shoot
    - PlayerShoot
  + Bullet
  + EnemyController
    - Health

Summary: This code will focus on the Player combat portion of the game making it so the user can actually attack and deal damage to the enemy objects



A class diagram for the Boss Fight of the Warehouse Warriors game. Includes:

* Enemy
  + UniqueBoss
  + Boss
    - BossAnimation
  + EnemyWeapon
    - BossWeapon
      * WeaponAnimation
* LoseScreen
* WinScreen

Summary: This code will focus on the Boss Fight portion of the game making it more of a complete storyline upon completion of the subsequent initial levels.

A diagram of a computer

Description automatically generated

A class diagram for the HUD/UI of the Warehouse Warriors game. Includes:

* HUDSystem
  + InventorySystem
    - Item
  + ObjectManager
    - Objective
  + PlayerInteraction

Summary: This code will focus on the HUD/UI portion of the game making it so the user is conscious of the objects they possess and the objectives of each level

In summary each of these elements will work together using cohesion and other factors to interact to serve the user a fully built and smooth working top down clear the room shooter. This is just an overall summary of each of the classes and the scripts each person will focus on implementing. More details can be found in individuals champion docs which outline the uses cases and other necessary details to understand the granular details of the classes and scripts.

Oral Exam Components

How to Create a Prefab

1. In your unity project create a game object in the hierarchy of your needs.
2. Apply all the correct components to the game object (Scripts, colliders, rigid bodies, etc.)
3. Once all components have been added then open your personal folder in the prefab directory.
4. Drag your game object from the hierarchy into your specific prefab folder.
5. You now have made a specific prefab for this game object.

Prefab Notes

* Prefabs are good when you need to create a game object during run time. Use prefabs on objects that are not in the scene upon start.
  + Ex: A bullet is not in the scene until it is called by the shoot function so a bullet should be a prefab.
* Document your prefabs. Make sure there is enough comments in the code attached to prefabs or a document in your folder that defines what the prefab does.
* Try to keep prefabs general and vague so they can be used in a variety of situations.
* Prefabs can be modified when in the hierarchy, so they are good when you want a baseline game object.