Lockdown Rumble

Technical Design Document

Dieter Shepherd & Andrew Paull

Bootleg Inc.

DESIGNERS

Charbel Ishak

Daniel Preece

PROGRAMMERS

Andrew Paull

Dieter Shepherd

ARTISTS

Alex Robert

Jerome Matawaran

Josh Yondjo

Contents

Game Mechanics

GAME ENGINE

For this project we are using Unity3D on version 2017.1, a newer version of the software than others at time of writing due to requirements put forth by myself. We are using this engine as it fits our needs specifically due to the implementation of Unity3D's in-house source control "Unity Collaborate" as it is incredibly easy to use, as well as other features that Unreal does not offer.

PLATFORM & OS

Initially we were going to target a multitude of console platforms (Xbox One and PS4) but due to myself and Andrew (another programmer) and our experience in working with other platforms and the issues that may occur, we decided at this time to only release for Windows/Mac/Linux, with the potential to branch out to other systems at a later time if we feel that it would be a possibility.

HARDWARE REQUIREMENTS

The minimum hardware requirements are as follows:

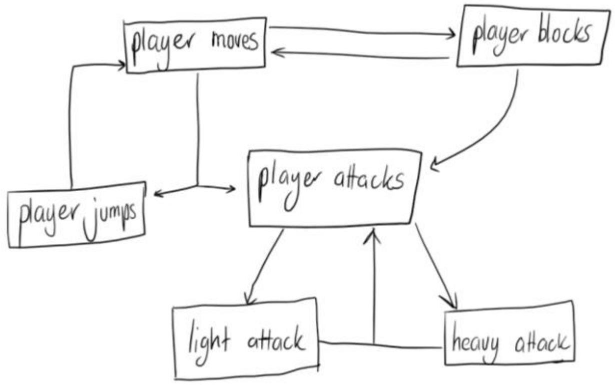
- OS: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+. - Graphics card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities. - CPU: SSE2 instruction set support.

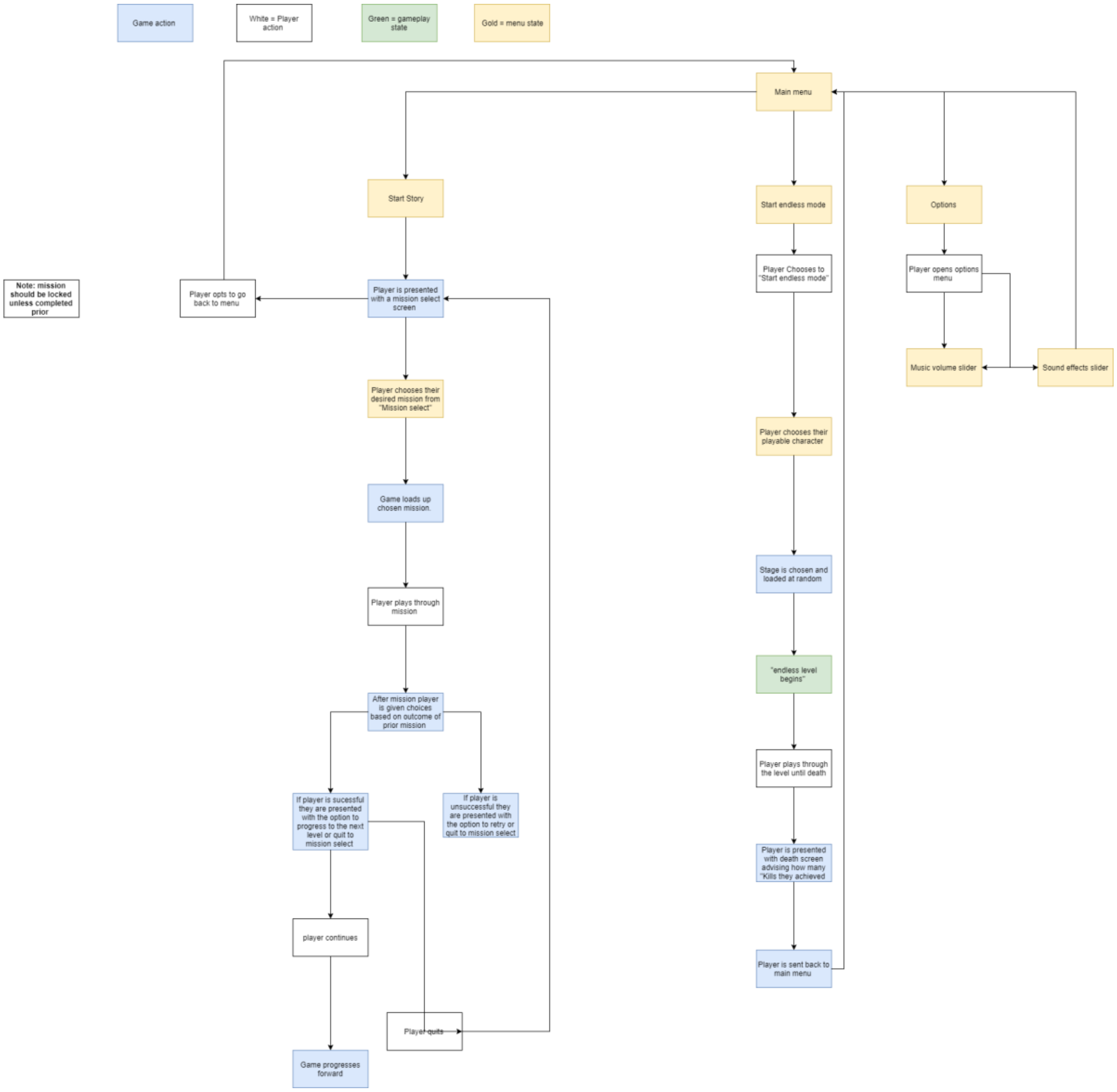
Naturally, the requirements can be higher depending on what is happening in screen.

EXTERNAL CODE

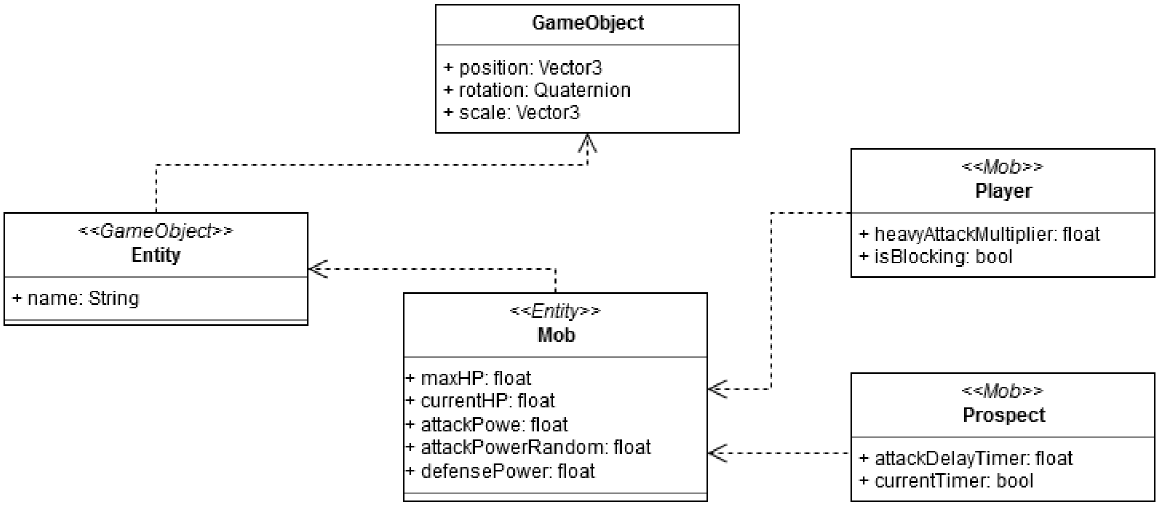
As of time of writing, all the programming will be done "in-house", requiring no external assets.

CONTROL LOOP




Game action
White = Player action
Green = gameplay state
Gold = menu state
Main menu
Start Story
Start endless mode
Options
Note: mission should be locked unless completed prior
Player opts to go back to menu
Player is presented with a mission select screen
Player chooses their desired mission from "Mission select"
Game loads up chosen mission.
Player plays through mission
After mission player is given choices based on outcome of prior mission
If player is sucessful they are presented with the option to progress to the next level or quit to mission select
If player is unsuccessful they are presented with the option to retry or quit to mission select
player continues
Player quits
Game progresses forward
Player Chooses to "Start endless mode"
Player chooses their playable character
Stage is chosen and loaded at random
"endless level begins"
Player plays through the level until death
Player is presented with death screen advising how many "Kills they achieved
Player is sent back to main menu
Player opens options menu
Music volume slider
Sound effects slider

GAME OBJECT DATA



DATA FLOW

For the proof of concept it is highly unlikely that we will implement saving, however Charbel (one of the designers) has expressed his interest in including this function further along in production. If so, we will likely save a list of statistics about the character as well as their progress in the game itself.

GAME PHYSICS & STATISTICS

We are using the standard physics engine as implemented by Unity.

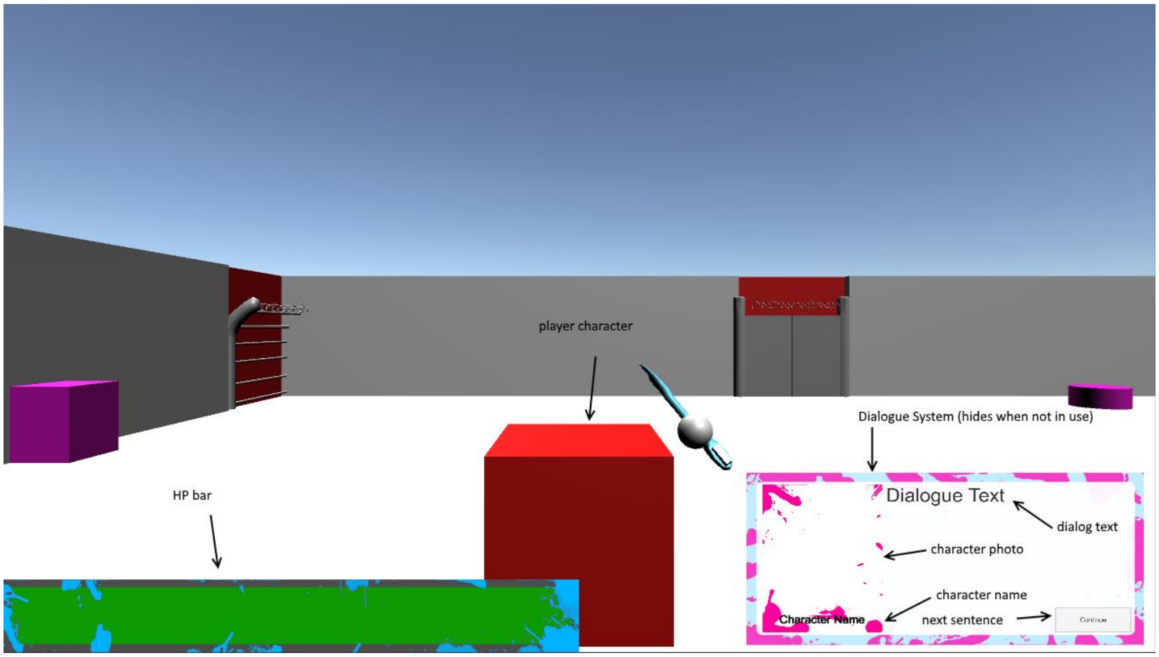
ARTIFCIAL INTELLIGENCE

Our game will rely on AI to run all humanoid models – enemies and allies, etc. This AI will be created in-house as to provide a more specific use, unlike using other AI plugins such as RAIN, which would not suit our needs at this time.

We will be employing flocking and crowding to create the feel of a dynasty warriors game, with the AI patrolling certain pre-designated points but engaging the player character upon sight.

User Interface

MAIN PLAY SCREEN



Art and Video

ARTIST INSTRUCTIONS

We have instructed our artists to create their assets in Maya as all computers in use on this project have Maya installed, as well as it being the standard. We have asked for generally lower poly looking models to fit a theme, and any and all videos that may likely be implemented in the project will be using one of the new functions of Unity 2017, a properly working video player – absent officially in other versions, however quite easily implemented if need be – said files will be created by the artists and Dieter.

Sound and Music

SOUND ENGINEERING INSTRUCTIONS

Dan will be handling the creation of sound assets at this time, and will be implemented using Unity’s standard audio features.

Level Specific Code

- Dialogue
- Nav meshes

Technical Risk

- Potentially high CPU cost due to AI
- High GPU cost due to number of entities in scene at a time.