Tower Sword  
Production Planning TDD

Team Name:

Contents

[About 3](#_Toc51679645)

[Change Log 4](#_Toc51679646)

[Team Members 4](#_Toc51679647)

[Development Environment 5](#_Toc51679648)

[Software Requirements 5](#_Toc51679649)

[Accounts 6](#_Toc51679650)

[Third Party Libraries 7](#_Toc51679651)

[Version Control 8](#_Toc51679652)

[Repository 8](#_Toc51679653)

[Contributors 8](#_Toc51679654)

[Commit Message Format: 8](#_Toc51679655)

[Target Platform 9](#_Toc51679656)

[<Platform> 9](#_Toc51679657)

[<Platform> Limitations 9](#_Toc51679658)

[Minimum <Platform> Specs 9](#_Toc51679659)

[Release Build Instructions 9](#_Toc51679660)

[Deliverables 9](#_Toc51679661)

[Controls 10](#_Toc51679662)

[Keyboard / Mouse 10](#_Toc51679663)

[Controller 10](#_Toc51679664)

[Mobile / Touch 10](#_Toc51679665)

[Custom Game Systems 10](#_Toc51679666)

[Coding Standards 10](#_Toc51679667)

[Coding Standards Enforcement 10](#_Toc51679668)

[Technical Goals and Challenges 11](#_Toc51679669)

[Technical Goals: 11](#_Toc51679670)

[Technical Risks: 11](#_Toc51679671)

[Risk Avoidance 11](#_Toc51679672)

# About

Describe the project / Game (1 paragraph)

Describe the purpose of this document (1 paragraph)

# Change Log

Updates made to the document should be described below.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date of change | Description |
| 0.0.1 | Duncan | 1/12/2020 | Initial edit |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Team Members

|  |  |
| --- | --- |
| Name | Role |
| Duncan Sykes | Programmer |
| Alex O’Dea | Artist |
| Caitlin O’Brien | Artist |
| S |  |
|  |  |
|  |  |

# Development Environment

This section outlines the required software and systems required for development of this project.

## Software Requirements

The below table outlines the software requirements for development of this project. Developers contributing to the project are required to use the approved software outlined below.

Any software that contributes to the direct development including planning and communication tools should be outlined below. A developer contributing to the project should have the below software available to them for use.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software | Version | License | Used By | Used For |
| Unity 3D | 2019.3.6f1 | Education | Programmers, Designers, Artists (On Campus) | Development of Game |
| Unity 3D | 2019.3.6f1 | Free | Programmers, Designers, Artists  (At Home) | Development of Game |
| Visual studio Code |  | Free | Programmer | Development of code for game |
| Maya |  | Education / Student | Artists | 3D modelling for game assets |
| Photoshop |  | Education / Student | Artists | 2D concepting and 2D asset creation |
| Github / Git |  | Free | Everyone | Version control |
| Gitkraken |  | Education / Student | Everyone | Version control and project management |
| HacknPlan |  | Free | Everyone | Project management and task tracking |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Accounts

The below table outlines any accounts that may be needed for the development of the project. An account is usually identified by 2 areas:

* **Individual**: Each developer of the project may need an individual user account, for various software or services. This includes software like Trello, HackNPlan or Git.
* **Organization**: A project or organization account is often developed for the software to integrate with other services, this includes things like Advertising / AdSense / git organizations / repos, Facebook developer account etc. An organization account is usually managed by 1 or more team members. Ownership of the account should be able to change between members.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Account/Service | License | Used By | Used For | Owner |
| Github | Free | Programmer, Art, Design | Contributing to projects hosted on github | NA |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Third Party Libraries

Unity/Unreal comes with a default collection of plugins, tools and assets. Its plausible, and often encouraged to pull in additional assets, tools, plugins or scripts etc. developed by a 3rd party. Any additional library or assets developed by the third party should be listed below.

|  |  |  |
| --- | --- | --- |
| Asset/Library/Package name | License | Used For |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Version Control

## Repository

https://github.com/bpgas01/towerSword

## Contributors

* Bpgas01: Duncan
* Zuezazon: Alex

## Commit Message Format:

**Format:**

|  |
| --- |
| Type (scope) : TaskId : Summary |

**Examples:**

|  |
| --- |
| Feature (menu) : 1302 : Added Exit button to main menu |
| Fix (menu) : 1395 : Updated button prefab with so that hover works on web builds |
| Feature (sandbox) : 1129 : Added rock asset to test scene, Created Rock prefab |

# Target Platform

This project will be deployed to the following platforms:

* Windows / PC
* Android
* iOS

## <Platform>

Duplicate this section for each desired platform

### <Platform> Limitations

Outline <platform> limitations, provide short description of the limitation. could include:

* Graphics capabilities, shaders, poly counts
* Available inputs (keyboard, mouse, touch, controllers etc)
* Performance constraints (max number of particles, game objects etc)

### Minimum <Platform> Specs

Outline the expected minimum system requirements required to run the project in release build.  
The minimum/maximum specs should consider target audience system specs and drive both technical and non-technical design decisions to ensure project runs on specified devices.

* System spec
* System spec

### Release Build Instructions

Detail the steps needed to build for the desired platform.

## Deliverables

A Build of the project should be generated every week and placed in the following location:

|  |
| --- |
| https://github.com/bpgas01/towerSword/builds |

# Controls

## Keyboard / Mouse

* Spacebar (emulates touch input for PC build and testing
* Mouse input for menu navigation

## Mobile / Touch

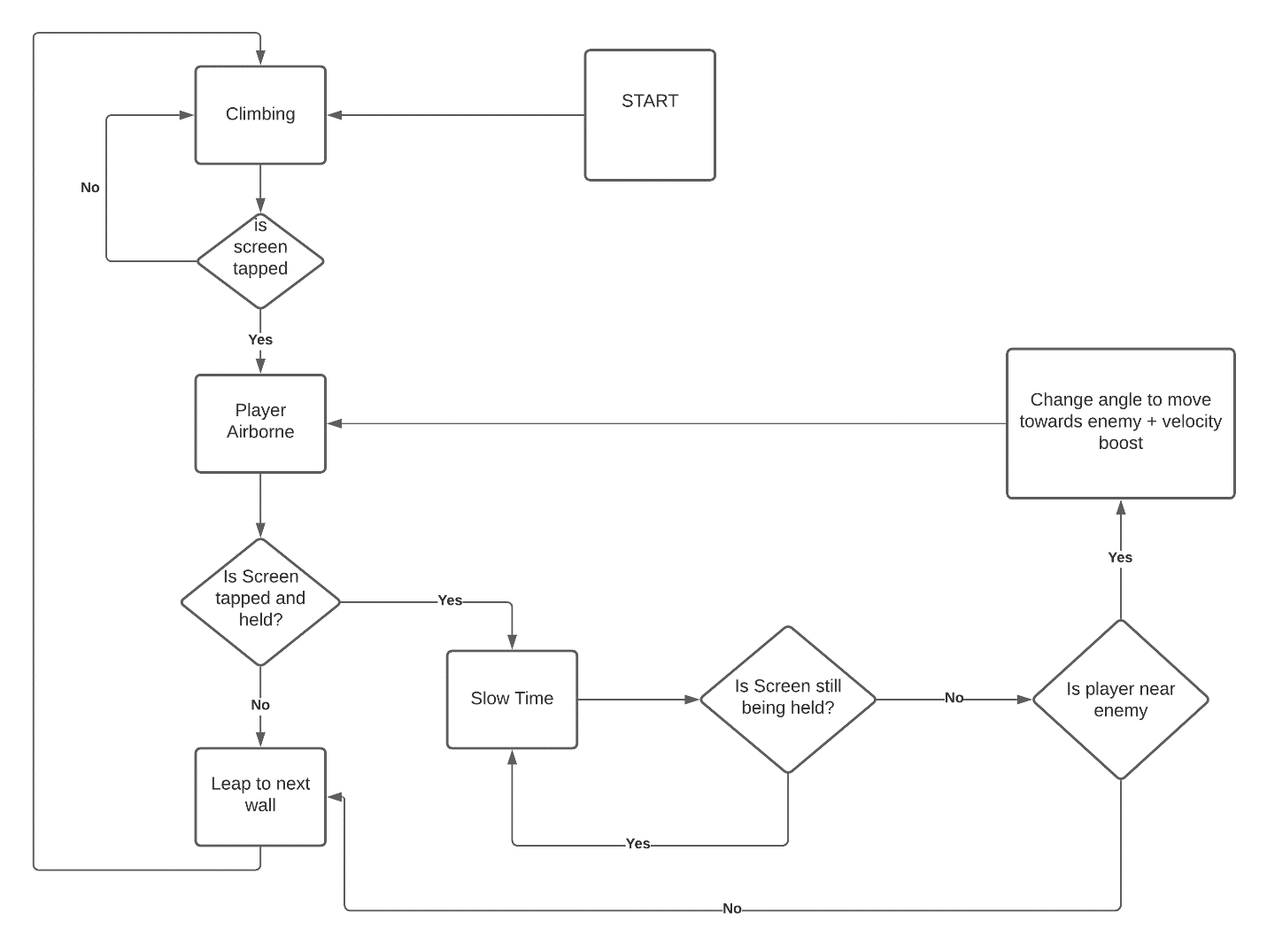
* Touch input for menu navigation
* Gameplay (Touch screen at given area):
  + Tap while climbing: Leap
  + Hold while airborne: Time Slow
  + Release near target: Sword Dash

# Custom Game Systems

* Wall climbing system (passive / automatic): How character / player moves forward. Updated constantly and doesn’t need manual control.
* Tap to control system:
  + Based on tap length and duration change the state of the player.
  + As outlined above, style of “interaction” changes the overall outcome of the players state.
  + EG:
    - Quick tap results in leap to other wall
    - Long hold slows “time” (character and enemy movement speed)
    - Release near enemy (detect using sphere overlap + tag check) results in momentum boost in that direction
* Attacking / Dealing damage  
  How would objects be identified as “Attackable”

Provide UML inheritance diagrams, flow charts or other to help communicate the design (if applicable)

Tap Input control system (FlowChart)



# Technical Goals and Challenges

## Technical Goals:

* Smooth character controller using 3D vector math
* Custom tap input system
* Modular code
* Random level generation – but still as different stages. Example: Hades level generation

## Technical Risks:

* Smoothing and design of vector math system