Don’t Die

Matt System Architecture and System Model

Hao User requirement definitions and system evolution

Carney requirement specification and Introduction

Tanaka Preface and Glossary

**Preface**

The expected readership of this documents expects people to know what a video game is and the basic mechanics that lie therein. The purpose of this document is to list and prioritize all requirements set for the game “Don't Die.”.

**Introduction**

This section describes why the system is valuable and why it's suitable for this semester project. It should briefly describe the system's functions and explain how it will work with other systems. What was the process you used to choose this particular project? Did you create prototypes or use other approaches to decide on the direction and scope of the project? Be sure to explain work that other people have done in this area and how your system is different. Give some explanation of why you think your goals can be accomplished in about a semester. How did you validate your requirements to make sure they were reasonable?

We are designing a mobile application that we call, “Don’t Die”. This is a “dodging” game. You will be a character that dodges bullets that the boss is shooting at you. You will also be able to pick up powerups that will help you survive. Our first thought when we got together as a team was to immediately start brainstorming ideas for games that we could program. We wanted something along the lines of Flappy Bird. However, we came up with something that is more complex.

For this project we will be using Android Studio to do all of our coding. The game will be played in landscape mode on your phone with a virtual joystick on the bottom left side of the screen to move. You will be able to change the character you are using by collecting coins that pop up randomly throughout the game. You will also be able to use these coins to unlock new backgrounds. In order to develop this game as quickly as possible, we are going to code a low grade version of the game that works. We will then start to tweak the graphics and add things that we think will enhance the game. We believe that we will have a working game by the end of the semester. Once we develop something that works we will be able to expand on that and shape the game into what we want it to be.

This project is going to give us a chance to expand our creativity with coding. We will be able to communicate well as a team and come up with new ideas on how to do things. Coding the AI for the “Boss” will give us a chance to dwell in a side of coding that we haven’t experienced before. Overall, we will be able to develop a fantastic mobile game though communication, teamwork and hard work.

**Glossary**

Android Studio - android studio is the official Integrated Development Environment (IDE) for Android app development

Flappy Bird - Popular mobile game from 2012 about a flying bird dodging pipes

Sprite - 2D image used as part of graphics display

Lag - where a game freezes for a duration of time

AI - artificial intelligence

Class - in java a class is a user defined blueprint or prototype from which objects are created

Java - Popular class based object oriented programming language

Android - android (google) is a mobile operating system that phones and tablets can run on.

Ios - Ios (Apple) is the mobile operating system that phones and tablets run on.

Joystick - mechanism for getting input from the user

**User requirements definition**

I will describe the services provided for the user. The non-functional system requirements should also be described in this section. This game should easy for people to play. You should at least run in Android 5.0 system called Lollipop with your phone. This is the list for the user to have. User can do these things in the list.

1. System should start up with a start menu that states, “Play”, “Unlockables”, ”Settings”.
2. When “Play” is tapped, the system should begin to run the game.
3. When “Unlockables” is tapped, the system should take you to a screen where you can unlock characters and backgrounds with the coins picked up in the game.
4. When “Settings” is tapped, the system should take you to a screen where you can change various settings such as music and sound effect volume.
5. While playing the game, Dragging the virtual joystick in the bottom right corner will cause the system to communicate with your character, making your character move.
6. When the character picks up a weapon, the new button will appear “shoot” and you can click on the button.

**System architecture**

There will be two main packages that Don’t Die will utilize, the state package and sprite package. The state package will be used to create different states and the sprite package will be used to create different characters in the game. The state package will consist of different classes that will be used to play the actual game, while the sprite package will consist of character interactions, items, and attributes.

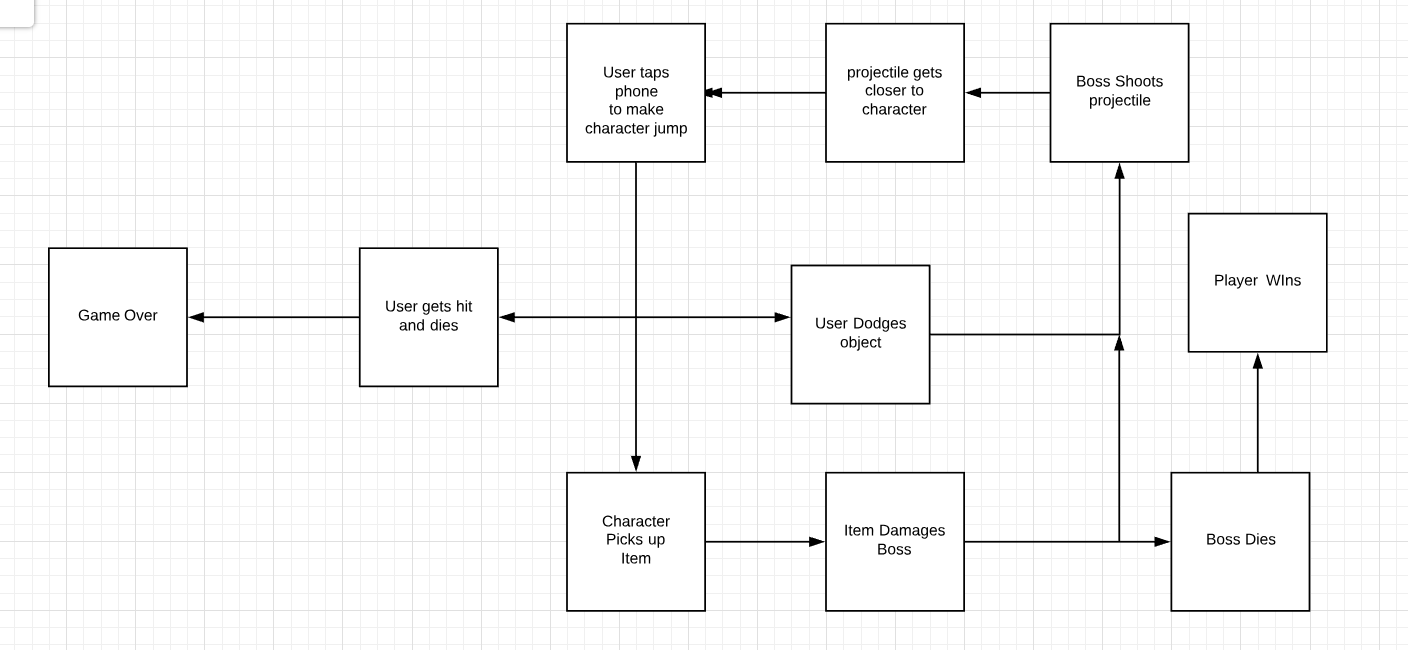
The system architecture of Don’t Die will be a stack of different states. We will have classes of different states that will be needed, such as a menu state, play state, pause state, ect. Each state will have different purposes; PlayState will be the actual game being played. MenuState will simply be the menu. GameStateManger is essential because this class is the controller of all the states. Whatever state is on top of the stack will be the state the game is in.

In the Don’t Die PlayState there will be our playable character, items. Coins, and a boss character, which all fall under the sprite package. Our main character class will have simple movement up and down and eternally progressing left to right, alongside health. The item class will contain a list of different possible items that the playable character can use to damage the boss character. The Boss class will contain a sprite object that will move up and down with AI, that will project objects at the playable character that will deal damage.

**System requirements specification**

* Should be able to run on Android and iOS.
* Must run on Android 5.0 and up.
* Must run on iOS 11 and up.
* Should start up relatively quick.
* Game should be developed using Android Studio.
* Game should never lag or freeze up.
* Game should be developed using Android Studio.
* System should not communicate with outside servers.
* User will not need to make an online account.
* 1 player game.
* Shall provide the ability to unlock characters and
* Must display warning that you must be 13 years or older to play.
* No in app purchases.
* Unlimited number of coins can be collected.
* Separate classes for Boss and Characters.

**System models**



**System evolution**

* Adding more characters and weapons
* Change the style of characters
* Can accept external controller instead of using phone only
* Add multiplayer model in the game
* Have a switch button to switch the weapon
* Make more detailed about game background

For the game system, we first make the structure about the game. We can create the single person game first. Then, we can think about the game character and weapon. In the whole project, we use Android studio to develop the game. In the end, we may put the game on the ios system.