CSCI 205 Final Project: USER MANUAL

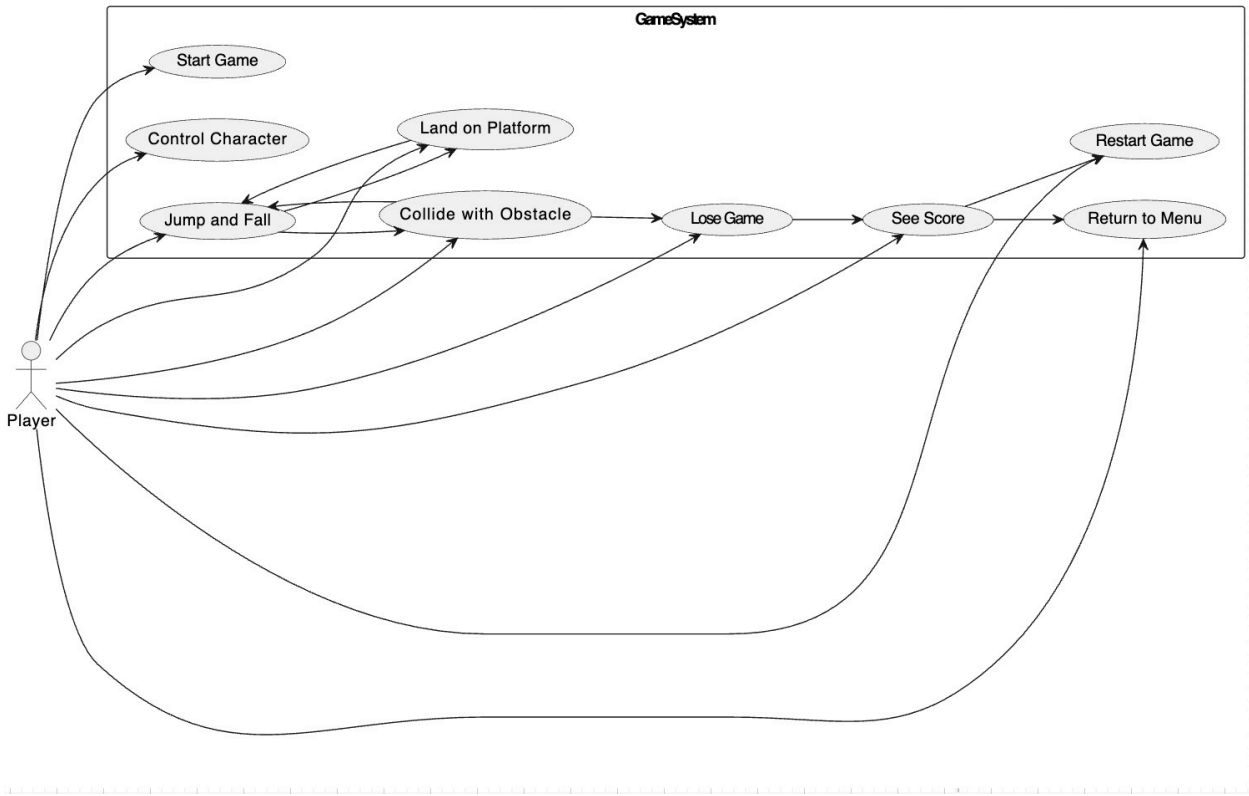
General problem statement:

Our scrum team developed a game inspired by the popular mobile game Doodle Jump. The game implements a vertical scrolling interface in which the player controls a character that jumps from platform to platform, trying to climb as high as possible without falling. The goal of our project was to create a game that has an engaging user experience, mirroring the game as closely as possible. We approached this project striving to create a game that is both cohesive from a technical design standpoint and fun from the user's perspective. Our game introduces the 'doodle character' to a continuously changing environment, giving the illusion of an infinite game world. Our game not only offers entertainment but also demonstrates how interactive graphics, animation, and user input can be integrated to create a smooth gaming experience. It features game design principles such as collision detection, character movement (according to user input), and increasing difficulty according to game levels.

Introduction, Background, and Motivation

We chose to replicate Doodle Jump specifically because we were intrigued by the seeming complexity and mathematical significance of vertical scrolling in an animated game world. Additionally, as the majority of our group are engineering students and quite busy, we don't have much time for breaks. Creating an 'infinite' game allows us to play it for a few minutes, and not linger on the idea that we are missing out on working towards a larger goal; it is a stress-free environment. We can take a break from our work, play a few rounds for five minutes, and then get right back to our work! We also wanted to understand and implement the physics involved in subjecting our doodle character to a gravitational force, and a bounce-like animation. While the rules of the game itself are simple, implementing them requires a clear understanding of object-oriented design principles. The core problem we addressed in building our game was creating a dynamic,

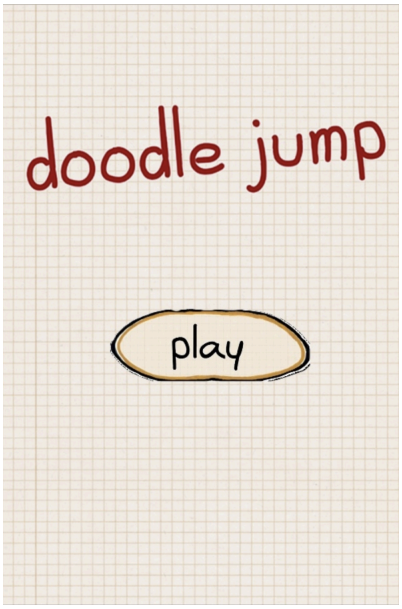
continuously expanding vertical world that responds to player input and maintains a smooth flow of gameplay. To solve this problem, we developed systems that can generate new platforms as the doodle character moves upwards, making chunks of the game no longer visible once the character jumps up, and handles collisions between the doodle character, platforms, and monsters in a way that feels intuitive for the user. We also focused on keeping gameplay clean and simple by making the doodle movements easy to operate, all while ensuring that there is enough variation and challenges to hold the player's interest. To help guide development, we created several user stories that represent common actions or experiences from the player's perspective. These include being able to control the doodle with arrow keys, the option to restart the game and navigate character movement using buttons. When designing and implementing them, we chose to keep the design simple, similar to the original game. The simplicity of the layout allows for casual gamers, as well as more competitive gamers to feel comfortable playing the game, and not too overwhelmed.

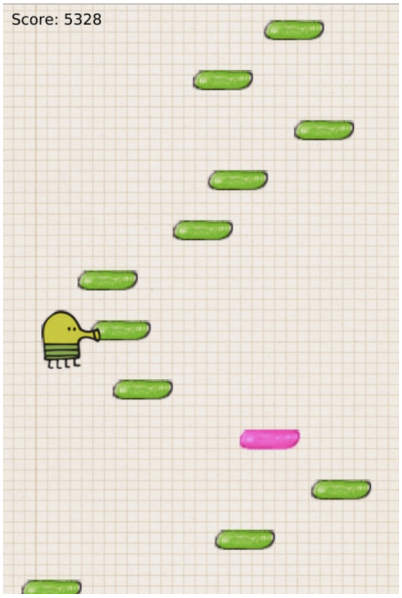


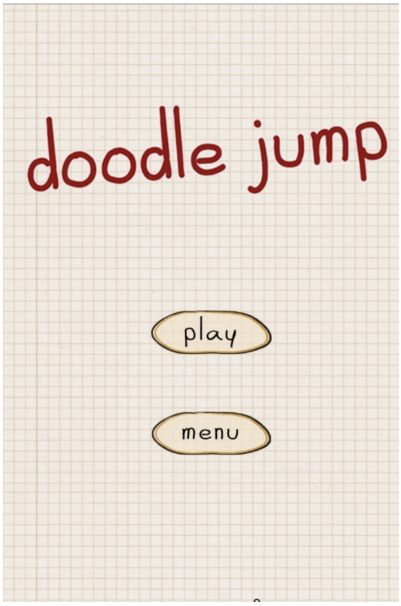
Our Case Diagram

Instructions for using the program:

In order to initiate the game, the user should run the MainApp method. The start screen will then appear with a play button to begin the game. After clicking the play button, the game screen will be called, immediately shooting the doodle character up from the bottom of the screen. The objective for the player is to make the doodle character travel as far upwards as possible by jumping on randomly generated platforms all while avoiding collisions with deadly obstacles or falling off the platforms. In order to control the direction of the doodle, the user should use the left and right arrow keys.







The green platforms give the doodle a standard bounce power, while the pink platforms give the doodle an extra boost upwards. The left and right sides of the screen act as a portal to the other side; if the doodle travels off the right side of the screen, it will immediately reemerge at the same latitude from the left side of the screen. The user

must keep the doodle character from touching the bottom of the screen; if this happens, the doodle dies and the game ends. When the game ends, the game screen switches to display the stat screen. Here, the most recent score will be displayed, as well as an option to play again and restart the game. As the game progresses and the score increases, the difficulty of the game increases as well, generating fewer platforms to jump on and making obstacles spawn in more frequently. The three types of obstacles that the user must overcome are black holes, stationary monsters, and moving monsters. The black holes must be avoided all together, both in the jumping state and the falling state; if the doodle character comes in contact with a black hole regardless of its state (jumping/falling/side-to-side), it dies immediately and the game is over. The doodle character can, however, collide with stationary and moving monsters while in the falling state. In other words, the doodle can land on the monsters, which then treats them like platforms. If the doodle character collides with a monster while in the jumping state (while it is moving upwards), the doodle will die, ending the game.