Junior Game Programmer

Job Application Evaluation Assignment

Kaan Karagülle

1. *In your opinion, what are the different development phases we should go through and what percentage of total development time should each phase take?*

1) Planning and Building a Game Design Document (%10):
   Planning out the general game features, budget, target audience, platforms etc.
   Building a game design document that will act as a roadmap for the game's development cycle. It would include things like; core game design, mechanics, gameplay loop, theme, art style, story, player experience, the game's name etc.

2) Building a prototype (15%)
   Developing a rough, playable prototype of the game

3) Development/Implementation (%70):
   Fully developing the a complete version of the game with all of its features. Including, art, music, story, mechanics etc.

4) QA/QC Testing and Debugging (%5):
   Doing rigorous testing to catch out bugs and fix them before the release of the final product

5) Marketing, Release and Maintenance:
   Since they either happen in tandem with the development or after it, percentage wasn't included

2. *We would like to get a glimpse of the gameplay of our new game at the earliest.*

   a. *What are the first 5 mechanics/functionalities you should develop for this specific game so that we have a playable build as soon as possible?*

1) Player Mobility: Walk, run, jump, crouch, roll, slide, dash, air dash, wall jumping etc.
2) Player Attacking: Melee, ranged attacks, combo attacks
3) Health system, healing, death state
4) Player Leveling/Abilities/Currency system
5) Enemy movement and attacking, spawning logic

b. *For each of the mechanics/functionalities you have listed in the previous section, what questions should the game designer have answered before you start coding?*

1) What is the aimed player speed, jump height? What are the main methods of movement, do they have internal cooldowns or reset when a condition is met (like being grounded for player jump specialties to reset)

2) What are different weapons' attack speed and spamability? What are ranged weapon projectile speeds or are they instant? Are there chained combos or jump attacks?

3) How does the health system function; does every hit do 1 damage or does damage vary depending on source? Are there things that instantly kill the player like spikes or pits? How does the player heal himself?

4) Is there a character leveling system, if so how are levels gained? How will new abilities be gained and existing ones upgraded? Is there a currency system, if so how is currency obtained and used?

5) What kinds of enemies are there? What are their movement and attacking logic? How are they spawned, in pre-determined or randomized spots?

4. *Describe your strategy to decouple UI/Input modules from the rest of the game so that platform specific code (input event detection, UI control etc) does not creep into gameplay modules (movement, jump, attack etc.). Hint: Try to isolate input/UI code from character prefabs.*

a. *Describe your strategy to decouple UI/Input modules from the rest of the game so that platform specific code (input event detection, UI control etc) does not creep into gameplay modules (movement, jump, attack etc.). Hint: Try to isolate input/UI code from character prefabs.*

I would create an InputManager class that handles the input and calls the appropriate functions in other classes according to the input. For platform dependent inputs I would use Unity's Input System package. I would also use Unity's built-in platform dependent compilation statements to enable/disable UI touch buttons and the Update function for input detection from non-touch controls (joystick, keyboard etc.)
(https://docs.unity3d.com/Manual/PlatformDependentCompilation.html)
The classes "InputManager" and "TouchButton" that I've added to the project make use these ideas. I didn't disable touch buttons in the Unity Editor for testing purposes but they would disable themselves in a non-mobile build of the game.