Rapid Prototype Planning

Each team member will build a *rapid prototype* to explore some part of your game design as quickly as possible. There are two types:

1) **Gameplay prototype**: the goal of this type of prototype is to build something **playable** in order to answer a specific question about the player experience and whether it is enjoyable or not.

2) **Technical prototype**: the goal of this type of prototype is to do a proof-of-concept for a piece of code functionality that the team is not entirely certain can be done or how to do it. It is intended to reduce project risk by quickly figuring out if a high-risk part of the design is worth pursuing or needs to be swapped out.

The gameplay prototypes that you select should be narrow in scope (don't try to integrate many systems), player-focused, interactive, and common (don't work on things that happen rarely in the game). You want the player to be able to perform actions and get feedback and outcomes to complete the loop. Keep it simple and focused, you only have around a week.

Poor choices are things like intro screens, menus, additional levels, graphics and sounds, things that only happened at the end of the game (e.g. boss fight), character upgrades, etc. Focus on the core interactions with the environment (moving, fighting, dialogue, crafting, dodging, abilities, etc) that are supposed to make your game fun. Don't sleep on feedback to let the player know what's happening (e.g. hit, miss), it's critical.

One of the challenges of starting a team project is that there are certain shared fundamentals (player movement is often one) that are a bottleneck for other features. Try to assign that to whoever you think will get it done fastest and feel free to share that code. Getting together in-person to hammer out that one part right away is also very effective.

Decide what prototype each team member is going to build and write a paragraph detailing each one in a shared document. Start with what aspect of gameplay it tests, or what technical concern it addresses. Be *very specific* about exactly what playable experience or proof-of-concept they are going to deliver. For example, don't say, "Make X". Detail how the entire planned interaction will go – player and entity actions, reactions, outcomes – and what constitutes successful delivery.