Test task

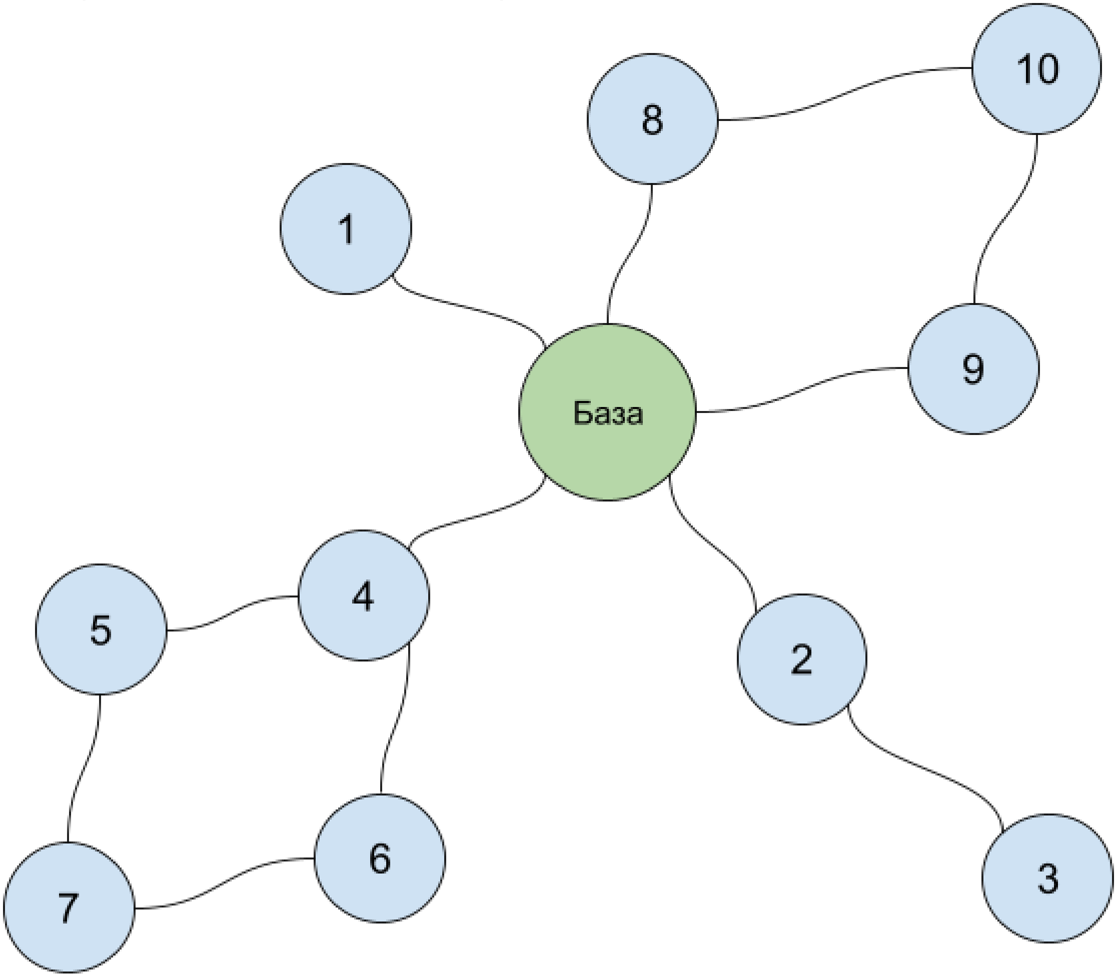
In the Unity environment, you need to develop a small application that implements the following functionality.

Let our RPG game allow the player to possess some **skills**. To learn a skill, the player spends **points**, which he earns during the game. Each skill costs the player one or more points to learn. The skills themselves are irrelevant (let it be something like jumping, running, flying), the only thing that matters for this quest is the fact of their presence.

There are non-directional **connections** between skills. Initially the player has only one **basic** skill. All other skills can only be **learned** if both conditions are met:

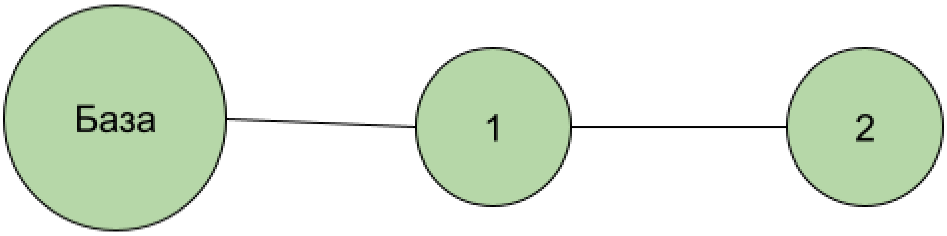
- the player has a sufficient number of points;
- the skill being studied is related to at least one skill already studied.

A sample skill scheme is shown in the picture.



Each skill is marked with a blue circle with a number, and the basic skill is marked with a green circle labeled Base. In this diagram skills 1, 2, 4, 8 and 9 can be learned at once (if you have enough points). Skill 10 requires skill 8 or 9 to be learnt first, and skill 7 requires either skills 4 and 5 or 4 and 6 respectively.

Learned skills can be **forgotten**. A forgotten skill becomes unlearned again, and the player receives back the points spent on its learning. A skill may be forgotten if and only if all of the remaining learned skills have a connection (direct or chain) to the basic skill. The basic skill cannot be forgotten.



So, for example, in the illustration above, skill 2 can be forgotten, but skill 1 cannot, because in this case skill 2 will not be connected to the base.

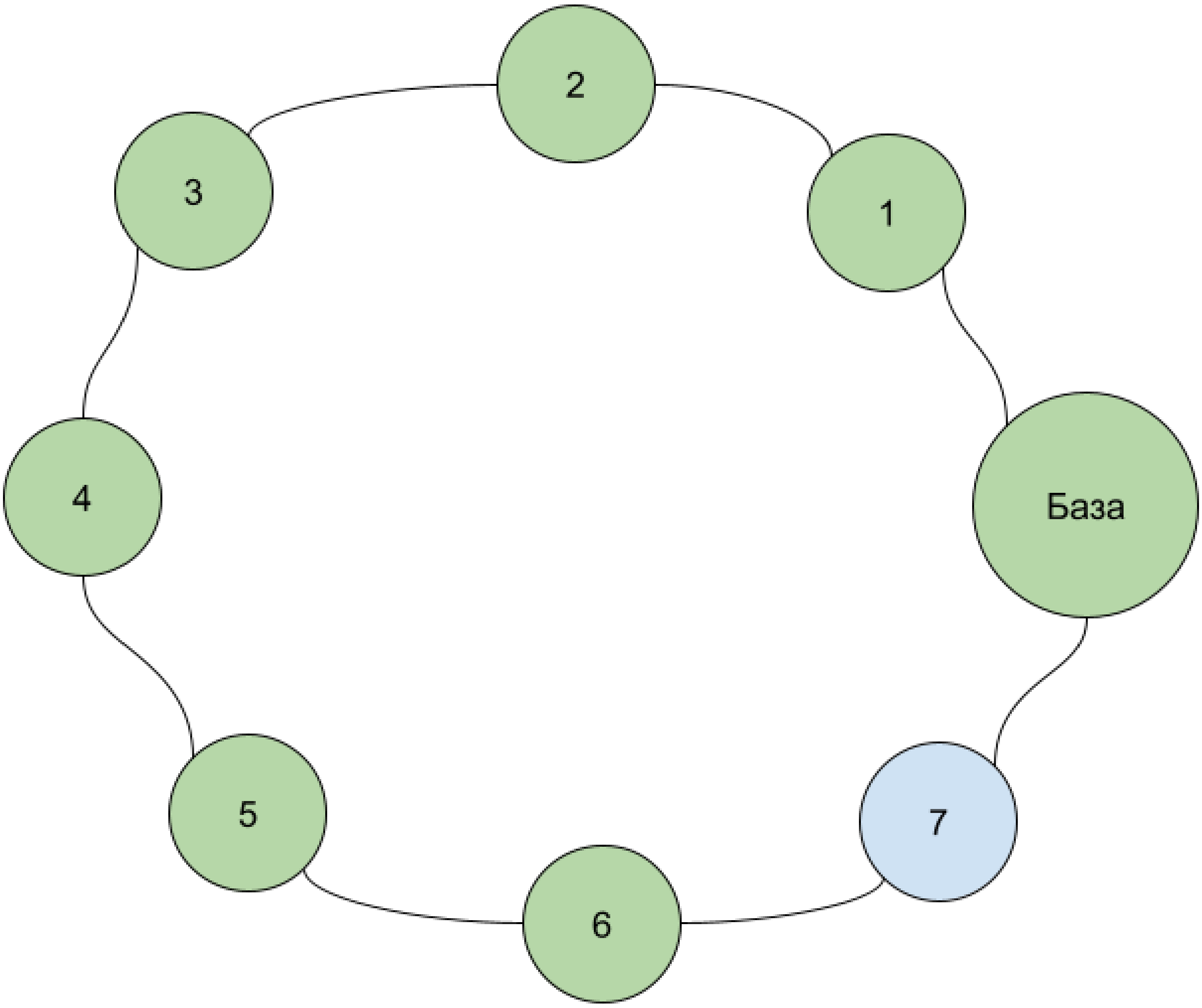
The following interface is supposed to interact with the game:

- Indicator of **the number of points** a player has.
- **Earn** Button. When you click it, the player gets one more point.
- A diagram with skills similar to the first figure. Skills are marked by circles with a number and have two states: learned/not learned.
- At any given moment, one (and only one) skill may be **highlighted**.
  All other interactions are done with the allocated skill.
- Indicator of **the cost** of the allocated skill.
- The **Explore** button. Active only if the selected ability can be studied. ● **Forget** button. Active only if the highlighted ability can be forgotten.
- The **Forget All** button. Always active, resets all skills and returns all points spent on skill training to the player.

The following aspects should be paid attention to when performing the task:

1. Separating game logic from interface logic, and interface logic from display details.
2. Cleanliness and quality of code.
3. Decision algorithms for learning and forgetting opportunities.

Formatting and other details for this task are unimportant, so are left to the discretion of the performer.


2
3
1
4
База
5
7
6