www.panteon.games

PLAYABLE ADS DEVELOPER

DEMO PROJECT

HELLO!

At this stage of the interview, you are going to do a small project. The reason behind the project is to show a demo of daily tasks at Panteon to you. Also, it will let us observe and understand your competence and ability on developing games. You can find project details and instructions below.

GENERAL INFORMATION

There are few common rules that apply to all tasks:

1. Your game should be played in portrait mode and should support different aspect ratios.
2. You are expected to use **Unity 2021 LTS.**
3. A playable **windows build** should be included in the submission.
4. You can use third party assets for any requirement.
5. Use provided assets where applicable.
6. It is expected that you will create materials with given textures and apply wherever needed.
7. The main camera perspective is the third person. Feel free to change it whenever needed.
8. The core mechanics cannot be changed. But you can improve other aspects of the game as you like.
9. You should be able to answer all project related questions.
10. You must use all concepts properly which are listed in **“Design”** section white implementing the project.
11. If you can’t understand the tasks perfectly, feel free to ask.

SUMMARY

You are expected to develop a platform runner prototype. Player will race against AI controlled players while dodging various obstacles and collects coin. After passing finishing line, regardless of player’s rank, player will be able to paint a wall. Details of the features will be given below.

PLATFORM RUNNER

In this task, you are responsible for making a third person runner game. The goal is to reach the end of the platform without being hit by any obstacle. If our character hits any obstacle, the game will start from the beginning.

1. You can find the platform that will be used in game in the **“Platform”** folder.
2. The camera will follow the character from behind during the game. An example camera angle is included in the **“Screenshots”** folder.
3. A character will run on the platform trying to avoid obstacles. You can find the necessary assets in the **“PlayerCharacter”** folder. You are expected to find animations from third-party sites, preferably Mixamo.
4. There will be no lanes in the platform so the player will be able to move their character freely using joystick mechanics.
5. If a player hits an obstacle other than the rotator stick, it starts over.
6. The player will try to collect the coins where you will place on the platform. You can find the coin model to be used in "**Coin**" folder. There should be a coin currency UI element which is placed in the upper right corner of the screen. This element shows how much coin a player earns during the game. All collectable coins will be sent to the UI with animation using "**DoTween**" when you collide with them. The important thing is that when you collide a coin, it will send five objects to the UI. We should see how many coins the player has collected in the upper right corner of the screen.
7. Gameplay will be around 25 - 30 seconds. So use your time wisely and design your level accordingly.
8. We should see how many times we have failed the game in the upper right corner of the screen.

Obstacles:

There are stationary and dynamic obstacles in our game. You can find them in the **"Obstacles"** folder

Stationary Obstacle:

Stationary obstacle will be static. They will not have any kind movements.

Dynamic Obstacles :

There are 4 types of dynamic obstacle mechanics.

Half-Donut:

This obstacle should be placed on the sides of the platform. It’s stick moves in the x-axis in time intervals to trick the players. Stick and donut should be interactable with all players.

Rotator:

This obstacle contains two different meshes (Rotator and Stick). Stick should be combined with a rotator and rotated around the y-axis. When characters get hit by the stick, they should be pushed with force, but when they hit the rotator, they will start from the beginning. Stick and rotator should be interactable with all players. But only sticks should apply force.

Shining Obstacle:

This obstacle will rotate around y-axis as well as moving in the x axis on the platform. There will be a particle above our obstacle. When the characters hit the obstacle, the particle on them will change color.

Rotating Platform:

This obstacle is designed to rotate around the z-axis with fixed speed. All characters should be affected by it’s rotating speed when they are moving on the platform. For example, if the platform rotates left, characters should try to move right in order to stay in the middle. You can find an example of usage in **“Screenshots”** folder.

PLAYING AGAINST OPPONENTS

As stated before, there will be AI controlled players. How to implement AI is up to you. There are no restrictions about implementation. However, we expect to see “intelligent” opponents in game

1. There will be 10 opponents.
2. Every opponent will use the same model. You can find it in the **“OpponentCharacter”** folder.
3. All players should be able to collide with each other.
4. Objective for opponents is to finish the game by avoiding obstacles.
5. Opponents should move and avoid as perfect as possible.
6. If opponents hit any obstacle other than the rotator stick, they will start from the beginning.
7. At the top left corner of the screen, the player's current ranking should be shown in real time.

PAINTING THE WALL

In this task, you are responsible for painting an object with a desired color. You can find the reference in the “Screenshots” folder.

1. After passing finishing line, regardless of player’s rank, player will be able to paint a wall. The character will stop in front of the wall and the player won’t be able to control the character anymore. Instead, the player will be able to paint the wall with **yellow**, **red** or **blue** colors using swerve mechanics. We will be able to change the **brush size** from the UI screen. You can find the items you need to use in the UI in the **"UI"** folder.
2. You are expected to display the percentage of painted wall in real time.

DESIGN

1. OOP
2. S-O-L-I-D
3. Design Patterns
   - Singleton
   - Events
4. Clean Code

SAVE & SEND!

You are expected to share your project via BitBucket or Github (send us an email that contains the link to the project – hr@panteon.games).

GOOD LUCK AND HAVE FUN!

LEGAL NOTICE

We would like to inform you that you are entitled all legal usage rights of each product created by you and made available for trial purposes so that we may recognise your capabilities in accordance with the Law on Intellectual and Artistic Works. The work and/or code you have created will be reviewed by us exclusively to evaluate your skill sets and will not be used for any other purposes.

WE ARE PANTEON