

SP-Project

Game mechanics spec

Revision	Date	Author	Description
A1	10/05/2022	Jbristhuille	First draft
A2	06/06/2022	Jbristhuille	Rework player moves

Concept	3
Moves	4
Buttons	4
Buttons assignment	4
Momentum conservation	5
Conclusion	5
Level editor	6
Diegetic elements	7
Tutorials	8

Concept

SP-Project will take the form of a hard platform game based on speed (speedrun). There will also be an editor allowing the community to create and share levels.

The focus will be on the community aspect:

- Time displayed online with leaderboard
- Ability to see other player's replays (optional feature)
- Level creation and sharing with community rating system
- Integrated in-game level search, download and rating system
- Possibility to play a lobby with other player in real time (race mode) (optional feature)

Moves

Buttons

The player will have different movements, each having their use to overcome the difficulty:

- **Jump:** Jump and aerial jump allowing the player to gain height. Reset on ground touch (one jump on floor + one aerial jump).
- **Pole vault:** Only on the floor, Uniquement depuis le sol, allowing the player to perform an angle jump (from 0° to 45°) by accumulating speed (entry speed + speed gain).
- **Hook:** The grapple hook allows the player to hang in the air or on a wall/ceiling, and accumulate speed to overcome obstacles (entry speed + speed gain). Resets when the character hits the ground. (one aerial grappling hook)
- **Pogo:** The pogo is a downwards hitbox which will allow the player to bounce on elements and surfaces including normally lethal elements (ex: spikes, traps).
- **Slash:** As in the game *The messenger*, the slash performed on a defined element will restore the jump without having to touch the ground, it will extend an aerial movement.

Buttons assignment

B1: Jumps

- On floor: **Jump**
- Aerial: **Aerial jump**

B2: Angular moves

- On floor: **Pole vault**
- Aerial: **Hook**

B3: Attack

- Neutral or latééal: **Slash**
- Downward: **Pogo**

Momentum conservation

Speed can be built up through the **jump** and **hook** mechanics, and maintained through **pogo** bouncing.

An **air jump** or a **slash** with a direction opposite to the accumulated speed will slow down the character.

Conclusion

These elements put together they should allow the implementation of varied and technical levels putting the players to the test.

To work these mechanisms must be as precise and constant as possible, failing due to a floating, sliding or collision problem would be particularly frustrating for the player.

Level editor

The level editor will be present in-game and will allow the design and sharing of levels with a library of objects made available.

This editor must allow grid editing (tilemap), off-grid editing (free placement) and background editing (decorative objects without interaction).

It is important that the creator can keep/export a local version of the level (possibility for the community to create external creation tools).

Once completed, a level must be able to be published online and accessible to other players. For this, the creator must be able to add a screenshot(s) as well as a description and one or more tags intended to facilitate the search.

Diegetic elements

On the example of *Celeste* game, and in order to limit the information displayed via HUD, the information concerning the character (number of jumps remaining, hook available, etc.) must be present in a diegetic way (ex: in *Celeste* the dash is symbolized by changing the color of the character's hair).

The goal is to create an organic and readable experience by limiting the parasitic information of the interface but remaining perfectly understandable.

Tutorials

In order to introduce players to the different mechanisms of the game, a campaign designed as a tutorial must be present.