|  |  |  |
| --- | --- | --- |
| Galton Master! | |  | | --- | | *01/30/24*  Nicolas G Tripp | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Concept | | | |
| 1 **Player Control** |  | You control a   |  | | --- | | *Galton machine* | | in this   |  |  | | --- | --- | | *2,5D* | game | |
|  | where   |  | | --- | | *keyboard* | | makes the player   |  | | --- | | *Displace several rows of obstacles* | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 **Basic**  **Gameplay** |  | During the game,   |  |  | | --- | --- | | *Small metal balls* | appear | | from   |  | | --- | | *the top* | |
|  | and the goal of the game is to   |  | | --- | | *get as many balls into the prize holes.* | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 **Sound**  **& Effects** |  | There will be sound effects   |  | | --- | | *winning/losing points* | | and particle effects   |  | | --- | | *winning points* | |
|  | [*optional*] There will also be   |  | | --- | |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 **Gameplay**  **Mechanics** |  | As the game progresses,   |  | | --- | | *Ball fall speed and ball population increases* | | making it   |  | | --- | | *harder* | |
|  | [*optional*] There will also be   |  | | --- | |  | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5 **User**  **Interface** |  | The   |  | | --- | | *Number of balls* | | will   |  | | --- | | *increase* | | whenever   |  | | --- | | *the level increases* | |
|  | At the start of the game, the title   |  |  | | --- | --- | | *Galton master* | will appear | | | and the game will end when   |  | | --- | | *Score becomes negative* | |

|  |  |  |  |
| --- | --- | --- | --- |
| 6 **Other Features** |  | |  | | --- | | *The highest score is recorded.*  *Original programming is refactored using OOP principles.* | |

# 

# Project Timeline

|  |  |  |
| --- | --- | --- |
| Milestone | Description | Due |
| **#1** | |  | | --- | | * *Refactor code applying OOP principles.* * *Decide which objects can be inherited.* * *Decide which special functions are needed for each child object.* * *Add getters and setters.* * *Add abstraction.* * *Check code performance using the profiler tool.* * *Optimize where needed.* | | |  | | --- | | *mm/dd* | |
| **#2** | |  | | --- | |  | | |  | | --- | | *mm/dd* | |
| **#3** | |  | | --- | |  | | |  | | --- | | *mm/dd* | |
| **#4** | |  | | --- | |  | | |  | | --- | | *mm/dd* | |
| **#5** | |  | | --- | |  | | |  | | --- | | *mm/dd* | |
| **Backlog** | |  | | --- | |  | | |  | | --- | | *mm/dd* | |

# Project Sketch



# Objects

Game scene

* Balls
* Paddles
* Obstacles
* Boundary box
* Status Canvas

Menu

* UI Canvas