# Competitor: Hyper Light Drifter (PEGI 12) (approx. £9 on steam)

# Target Audience Traits:

# Prototypes & Playtesting

We intend on making a small prototype for the game that we can playtest to test features and gain feedback from our target audience. This prototype will be created in the early stages of our game’s development and will steer the team in the correct direction for the remaining period of game development.

The prototype will consist of one core level of the game, and this will be the tutorial level. This is a suitable level to prototype because it will allow the play testers to come to grips with the game’s mechanics and controls in the short time frame they have access to the prototype (1 – 2 hours maximum).

This level will have the essential features implemented, however smaller, more unnecessary features (such as dialogue from non-important characters, certain pick up items which don’t cause progression within the prototyped level, non-interactable objects etc) will not be required for this early prototype.

Other than the playtest which will occur after the completion of the initial prototype, there will also be a second playtest when the game is mostly, or fully complete. This playtest will allow play testers to identify any faults or bugs within the game and this will be presented to the team via feedback forms. In relation to the release of the game, the final playtest should be set to occur a couple of weeks before intended release date.

The playtesting team will be comprised of the development team themselves, and any other volunteers who are interested in playing the game. The playtest will be advertised to mostly students; however, other groups of people are also desired.

# Technical Documentation

**Necessary Algorithms:**

* **A\* Search Algorithm:** This algorithm will be helpful whenever generating the game’s map, more specifically to find the most suitable, shortest path between two locations on the map to create a waypoint system. This feature will benefit the player as it will help them to navigate an open world environment at ease.
* **Sorting Algorithm**: This algorithm could be useful for things such as the shop that player can buy items from and could sort the items into the correct alphabetical order, high to low (price) and vice versa etc, to enhance the player’s experience.
* **Algorithms for Steering Behaviours:** Behaviours such as fleeing, following and arriving at waypoints will be useful for certain AI in our game to both make NPCs seem more human-like and to also contribute to the game world (i.e. Witch Doctor fleeing from player until he arrives at certain point on map and comes to a halt).
* **Finite State Machines:** A finite state machine is required for AI to enter different states (i.e. *Enemies:* Attacking the player, Idle, Wandering etc)

**Game Cheats for Testing:**

*Cheats should be used by the testers by entering a secret cheat code which will be supplied to them by the programmer.*

* Cheat which allows the user to skip to different levels in the game.
* Cheat for teleporting throughout the game’s map on the input of x and y values
* Cheat to skip to time of day by inputting the time in the 24-hour time format (i.e. 13:00 as opposed to 1:00pm)
* Cheat for increasing/decreasing the money player has
* Cheat for increasing/decreasing player reputation
* Cheat to increase/decrease health state
* Cheat to have unlimited money/health
* Cheat to add items to inventory

**Hardware Requirements:**

* PC equipped with mouse and keyboard
* Minimum system requirement of:

OS: Windows 10

Processor: Intel Core i5

Memory: 8 GB RAM

Graphics: GeForce GTX 660

Storage: 4 GB available space

Sound Card: DirectX compatible

# MVP Systems & Features

* The game’s general storyline should be in a way where it seems complete (i.e. beginning and end levels are most necessary)

# Localization

# Ideas & Questions