

Core Project Document

Groep 6 Escape

November 2014

1 Group Roles

Game Designer	Xander de Ronde	4170334	X.E.J.deRonde@student.tudelft.nl
Lead Programmer	Tim de Boer	4007352	T.deBoer-2@student.tudelft.nl
Lead Artist	Jordi Hessels	4155343	J.F.B.Hessels@student.tudelft.nl
World Builder	Arnoud Jonker	4170431	A.Y.A.Jonker@student.tudelft.nl
Producer	Mike Koch	4232313	M.R.Koch@student.tudelft.nl

2 Theme and interpretation

You only get one lifetime, don't spend it locked up.

3 Game idea

The basic idea of our game is to escape from a multi-level building. The game will be styled like animation movies. The game will be build like a puzzle where interaction with game objects can help solving the puzzle. Solving puzzles will make challenges with intelligent enemies easier. The room layout, for instance the placing of the game objects, will be procedurally generated. The game will have multiple levels, which will be different rooms. Traveling between this rooms will be possible by using doors, stairs, elevators etc. The goal of the game is to find the exit of the building. An extension of the game could be to add a condition of your player which you have to refill constantly during your game. For example oxygen or water.

4 Key components

Computer Graphics:

- 3D models
 - 3D-models (*) The game will contain 3D-models for most objects, except for flat surfaces like the floor. - Jordi
 - Animated 3D-models (**) Some 3D-models will be animated to make the game look more alive. - Jordi
- Textures
 - Procedurally generated textures (****) By making textures through algorithms, we can prevent repetitive looking textures. - Arnoud
- Special Effects & Juiciness

- Sound effects (*) Sound effects can be downloaded online, but we will also record some sounds ourselves. - Jordi
- Camera shakes (*) The camera shake effect can be used when a bomb explodes or the player gets hit by an enemy. - Jordi
- Particle Systems (*) To make the game look better aesthetically and bring the attention of the player to the right elements, particles are very useful. - Jordi
- Rendering
 - Play with lights and shadows (*) The player may be able to hide in the shadows, while light may help for overview of the room. Light can also be used to emphasize important objects in the room. - Arnoud
- User Interface (with new Unity3D UI tools)
 - Start, pause, end screen (*) It is almost impossible to find a modern game without these elements. - Xander
 - High scores (*) The high scores are saved from the top x players. Playing through the game faster and solving more puzzles results in a higher score. - Xander
 - Options (*) The options menu will contain elements like adjusting volume and sound. - Xander
 - Credits (*) A separate button on the menu leads to a screen where the team Escape is introduced. - Xander

Artificial Intelligence:

- Pathfinding using own algorithms (***) We will use an algorithm so enemies can increase their intelligence while searching for the player. This will probably be one algorithm for all of the enemies. We believe this is worth three stars, because writing a good, working algorithm takes quite some time. - Arnoud
- Use genetic algorithms (****) During the game we want the enemies to become smarter, so we want to use the previous good working enemies to create new ones. This will also be one algorithm for all the enemies and this will be hard to implement in the game. - Mike
- Enemy you always lose from (***) By using an enemy the player always loses from we want to oblige the player to come with a solution to solve this problem. For instance by distracting the enemy. We probably use a couple of these enemies. Making it impossible to defeat the enemy will require some time, so three stars are suitable for this task. - Arnoud

Web & Databases:

- Collect playthrough data (**) During the game we will collect some interesting data of the playthrough. One could think about scores, times, collectables, lives, etc. - Xander
- Store data on web server (**) All collected data must be centralized saved on a web server with a database. - Tim
- Visualize data on webserver (**) All collected data could be visualized. For example trend-lines and graphs. - Mike
- Collect and show highscores from web server (**) The game must be able to retrieve the current high scores and present them in a highscores menu. As extra feature, the highscores could also be presented on we webpage, for live statistics. - Tim
- Online gamer accounts with avatars (***) Based on unique identifiers (computer id?) or username and password. In game the gamer could chose default avatars and/or upload custom avatars. As extra feature, this could also be done via a website. - Tim
- Save and share states with others through social media (***) A "share with friends" and/or "tweet" button in game for sharing statistics. As an extra feature, the server could "share" and "tweet" daily the top 3 gamers. - Tim
- (extra) mobile device as second screen?? (***) - Tim

Programming:

- Game Mechanics
 - Procedurally generated levels (***) The layout of the levels will be procedurally generated. We intend to write one algorithm for procedurally generating the layout. So this will take some time in the beginning an when good writed will not take any extra time later. So three stars are assigned. - Arnoud
 - Race against clock (*) Escaping is ideal with a clock, this can be implemented in multiple ways. This is quite easy so one star is suitable. - Xander
 - Online multiplayer (****) We want to implement a multiplayer mode where the players have to work together to escape the building and win the game. We prefer to do this online, but this will be quite difficult. So this is a four star component. - Tim
- Game Loop
 - FPS independent (use Time.deltaTime) (**) We are going to make the game fps independent this is worth 2 stars. - Jordi
- Physics
 - Use Unity's triggers only for collision checks (*) We are going to use this function of unity, for instance to pick up objects. This is worth one star. - Jordi
 - Use Unity's full physics simulation for all movement, collisions etc (*) For walking around and not walking through wall, for example we will use Unity's physics simulation. This is worth one star. - Jordi

5 A rough schedule

- Week 2.1: Initial exploration of possible game concepts and further development of said game concepts.
After that, initial prototyping of game play elements will start.
- Week 2.2: Further prototyping of game play elements and game mechanics.
- Week 2.3: Further prototyping and finishing prototyping, adjusting Core Project Document.
- Week 2.4: Start development of the game, using the prototypes created before.
Creating a working prototype of the actual game.
- Week 2.5: Further development of the game,
implementing new ideas into the base game if any were found and expanded on.
- Week 2.6: Further development, updating documents accordingly.
Early access version will have to be finished by now.
- Week 2.7: Further development.
- Week 2.8: Further development, game entering Beta stage.
- Week 2.9: Further development and release of the game.

6 Link to the GitHub project page

<https://github.com/tim427/minorproject/>