Game Core Document – Group 2 EWI3620TU

Team Members

Stein Stroobants

- Game Designer
- stein.stroobants@live.nl
- Skillset EWI3610TU: AI

Ivo van Kreveld

- Lead Programmer
- ivovankreveld@hotmail.com
- Skillset EWI3610TU: Programming

Lieke Michielsen

- Lead Artist
- L.C.M.Michielsen@student.tudelft.nl
- Skillset EWI3610TU: Blender

Daniel Brouwer

- World Builder
- danielbrouwer08@hotmail.com
- Skillset EWI3610TU: Programming, Procedural modeling

Matthijs Ottenhoff

- Producer
- mmottenhoff@gmail.com
- Skillset EWI3610TU: Blender, Photoshop

Theme & interpretation

Multiple game themes are combined in the game idea. Below is an overview of each theme and its interpretation (on a very high level) is outlined:

- 'You only get one **skill**': Each player will only get one specific chosen or assigned skill with which he or she can solve puzzles (for example swimming, shooting, flying, etc.)
- 'Unconventional weapon': Basically each item in the game scene can be picked up by the player and can be used to eliminate possible enemies or obstacles.
- '10 seconds': The 'puzzles' presented to the player will have to be solved in a limited amount of time.

Game concept

The game is about gnomes whose village is threatened by a flood caused by two unknowingly human beings who dropped their water bottle during a pick nick. The gnomes need to escape their village but this won't be an easy task. During their escape they are confronted with various obstacles (like water, leaves, tree sticks etc.) and various enemies (like mice, ants, bugs etc.). The game will consist of several levels and is played by two gnomes. You can either play alone (alongside an intelligent AI gnome) or with two players. The levels will be 'half' procedurally generated and will contain cut-scenes containing a storyline. The playing field has depth so movement in all directions is possible. The camera will only move in two dimensions and makes sure that every player is visible.

Game components

Computer graphics

- 3D animated models (★★★): Creating basic models and then adjusting them using a rather complicated sequence of shape keys (e.g. for different types of faces, plants, etc…) These models will then also be animated in game (think of player movement, environmental movement, etc..). Responsible: Matthijs en Lieke.
- Particle systems (★): Particle systems will be used to create various effects such as smoke, water falling down and splashing, etc.. Responsible: Matthijs.
- Animated Textures (★): Will be used to create water reservoirs, flames, etc.. Responsible: Lieke.
- Adaptive camera (★★): An adaptive camera will be used that always captures both players by adjusting its orientation to that of both players, making it more challenging to implement than a standard 3rd person camera. Responsible: Matthijs.
- Interactive camera (★): Camera shakes/alters when interacting with the environment (such as explosions, hits, etc..). Responsible: Matthijs.
- Start, pause, end screen (★): Relatively easy to add to your game, but necessary. Responsible: Lieke.
- High scores, Options & Credits (3x ★): Relatively easy to add to your game, but necessary. Responsible: Matthijs.

Total: 12 stars.

Artificial Intelligence

- Conscious enemies and level (★★): We plan on adding enemies that are conscious of the actions of the player and changes in the level made by the player (e.g. when the player comes running passed a sleeping enemy, it should wake up and chase). Responsible: Stein.
- Path finding (★) (at least): Enemies will follow the players and decide on a certain smartness level what the fastest route is. Responsible: Stein
- Huge set of smart and dumb enemies with different skills/weapons/targets. (★★★★★): We will create a huge amount of different enemies, to keep them unpredictable. Some will be dumb (e.g. Some kind of ranger enemy who only shoots arrows at the player) and others will be smart (e.g. Follow the player, try to cut a player of, or make sure the player can't proceed the level). Next to that we want to give every kind of enemy a different skill set (some can shoot an arrow, while others sleep and get angry when you wake them up). Responsible: Stein.
- Creating a companion that helps you when you play the game solo. It should learn based on a computational algorithm (★★★★★): We think this is worth 5 stars because it could be very hard to implement. We need to make a player that can learn from its mistakes and rethink its

decision based on what he has learned so far. We are not certain that we can implement this, but we think it would add a really cool feature to the game. Because we don't know for sure that we can fully implement this, we made sure that there is enough AI to fulfill the star requirement. Responsible: Stein.

Total: 13 stars.

Web & Database

- Customizable avatars (saved online) (★★): Used to store a personal character online and store scores under your avatar, players should have their own account and avatar and need to be identified every time they play the game. Responsible: Daniel.
- Social media connection (★): Used to share scores with friends, difficulty still unknown, but manual suggested 1 star.; Responsible: Matthijs.
- Saving playthrough data online (★): Used to store gameplay data to analyze later on, difficulty still unknown, but manual suggested 1 star.; Responsible: Daniel.
- Saving gamestate data online (★★): Used to store the gamestate data to analyze later on, difficulty still unknown, but manual suggested 1 star. Responsible: Daniel.

Total: 7 stars

Programming

- Procedurally generated levels (★★): planned to used partly when making levels, We plan to use procedurally generated content as much as possible to obtain a varied game-play therefore we assigned 2 stars.; Responsible: Daniël.
- Moving platforms (★): Used as part of a level to make it more difficult, It should not be that hard because Unity already has a physics engine.; Responsible: Daniël.
- Race against the clock (★): A timer to use as score, it should not be that hard because Unity has a build in time function.; Responsible: Ivo.
- Local multiplayer (★): Two players can play at the same time, adding an extra character should not be that hard.; Responsible: Ivo.
- Implement mobile controls (★): Can be playable on Android, should not be hard to implement; Responsible: Daniël.
- Checkpoints (★★★): If player loses life, it should be redirected to the last checkpoint reached. Checkpoints are also used as point where game data is saved; Responsible: Ivo.
- Story dialogs and cut-scenes (★★★): The game should have an introduction cut-scene where a movie file needs to be made and imported into Unity. Dialogs should contain the story and

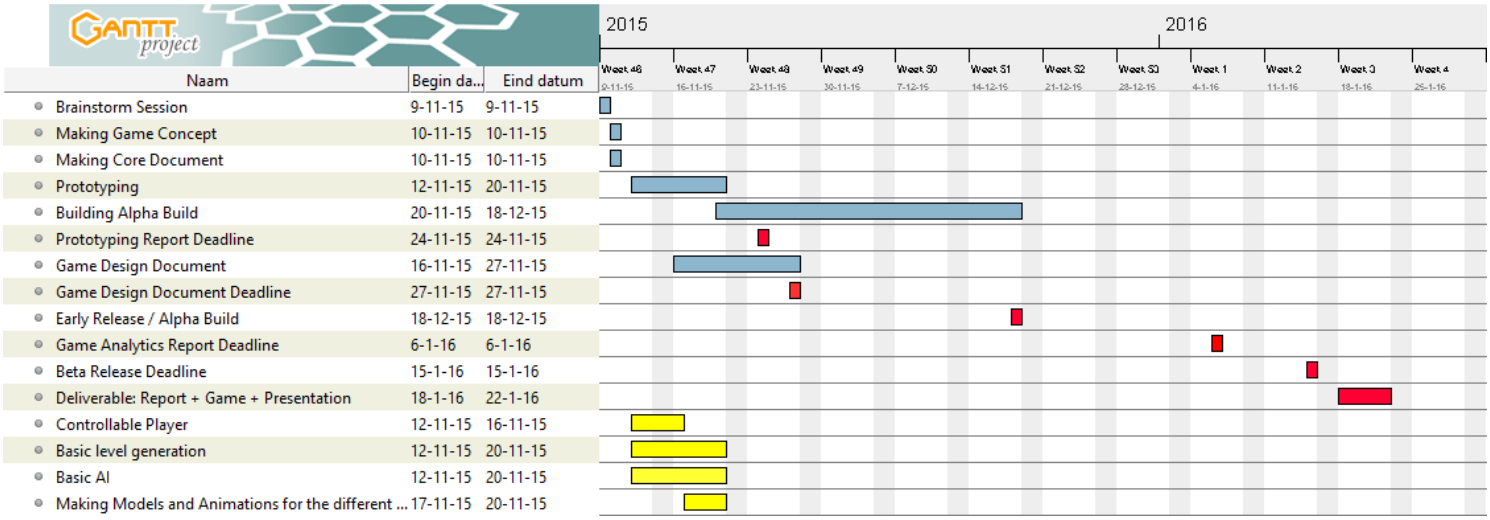
appear in-game. The story makes the game more attractive but this will take much time therefore 3 stars are assigned to this part. Responsible: Lieke.

- Triggers and collision (★★): Used to make puzzles (bridges, buttons, keys etc.). Because the game is based on puzzles, this should have lots attention therefore two stars are assigned; Responsible: Ivo.

Total: 14 stars

This leads to a total amount of 12+13+7+14 = 46 stars.

Schedule



Naam	Begin da...	Eind datum
Brainstorm Session	9-11-15	9-11-15
Making Game Concept	10-11-15	10-11-15
Making Core Document	10-11-15	10-11-15
Prototyping	12-11-15	20-11-15
Building Alpha Build	20-11-15	18-12-15
Prototyping Report Deadline	24-11-15	24-11-15
Game Design Document	16-11-15	27-11-15
Game Design Document Deadline	27-11-15	27-11-15
Early Release / Alpha Build	18-12-15	18-12-15
Game Analytics Report Deadline	6-1-16	6-1-16
Beta Release Deadline	15-1-16	15-1-16
Deliverable: Report + Game + Presentation	18-1-16	22-1-16
Controllable Player	12-11-15	16-11-15
Basic level generation	12-11-15	20-11-15
Basic AI	12-11-15	20-11-15
Making Models and Animations for the different ...	17-11-15	20-11-15

Github link

https://github.com/danielbrouwer08/EWI3620TU_Group_2