extra toevoegingen

tijd loopt af.

er is voor gezorgd dat de tijd afloopt in plaats van oploopt. Om te bepalen hoeveel tijd er is wordt er gekeken naar het aantal targets op de map. De tijd is het aantal targets \* 30 seconden.

op moment dat de tijd om is wordt er een dialog aangeroepen die aangeeft dat de tijd om is.

tijdspakketjes

Als extra functie is ervoor gekozen om tijdpakketjes te maken. Deze tijdspakketjes zijn te herkennen aan hun afwijkende groene kleur met daarin een gele bliksemschicht. als je over dit vakje heen gaat krijg je 10 bonus seconden waarna dit vakje weer verdwijnt.

als het spel weer opnieuw begint worden de vakjes weer op hun oude plek terug gezet.

beep

bij het raken van een boom, oppakken van een (energie)pakketje en het landen op de thuisbasis op moment dat alle targets bezocht zijn geluidjes toegevoegd. dit omdat er dan duidelijk wordt dat er iets 'aparts' gebeurd. Zo gebeurd dit ook als er een target wordt opgepikt en als de gebruiker na het oppikken van alle targets weer op de thuisbasis terecht komt.

gamerules file aangemaakt.

Alle 'regels' staan nu in een aparte Gamerules file. dit om het overzicht in het programma te bewaren op moment dat de regels goed gescheiden zijn van de rest van de code wordt het overzichtelijker wat er wanneer gebeurd.

Zo kan bijvoorbeeld bij het raken van een boom een regel aangeroepen worden die 15 seconden tijdstraf aan de gebruiker toekent.

Highscore

Op moment dat de gebruiker een nieuwe highscore behaald heeft wordt er een window geopend waar de gebruiker zijn/haar naam in kan vullen. als hij/zij dit gedaan heeft wordt de highscore getoond in het scherm met de tijd en naam van de gebruiker.

Omdat de naam van de gebruiker kan verschillen in lengte wordt het naam venster voordat het wordt gevuld met de naam van de gebruiker eerst gevuld met alleen maar spaties. dit omdat er een nieuw dialog over het oude dialog heen getekend wordt.

als de naam dus korter is dan de eerder ingevoerde naam zal er in een ander geval nog een deel van de oude naam te zien zijn.

code verfraaid.

om het overzicht in de code te bewaren zijn functies verplaatst naar de headerfiles waar deze bij horen. dit om zo min mogelijk informatie tussen de bestanden uit te hoeven wisselen.

Macro's gedefinieerd om op die manier het overzicht te bewaren.

functie FORM\_2D\_TO\_1D gemaakt een tweedimensionaal veld wordt omgezet naar een eendimensionaal array. om de locatie hiervan te bepalen worden er een x en een y coördinaat gezocht.

om deze coördinaten te bepalen wordt er gebruikt gemaakt van de volgende berekening ((y \* nGameareaW) + x) x en y zijn dan de coördinaten en nGameareaW is de breedte van het veld.

Dialogs

Er zijn verschillende dialogs gemaakt dit omdat er zich aan het einde van het spel verschillende situaties voor kunnen doen.

De situaties kunnen als volgt zijn:

\* Geen tijd meer over

\* Wel tijd over en een highscore

\* wel tijd over en geen highscore

\* Crash

voor elk van de situaties is een apart venster gemaakt.

Ik heb er vooral voor gekozen om de code aan te passen en opnieuw in te delen dan dat ik tijd wilde besteden aan het toevoegen van extra toepassingen. Dit omdat een goede basis voor de code met daarin rekening houdend met een paar aspecten van grotere waarde kan zijn voor mijn verdere ontwikkeling. Er zijn een aantal principes waar ik rekening mee gehouden heb. Een daarvan is DRY (don’t repeat yourself) ik heb gemerkt dat er de berekening ((y \* nGameareaW) + x) meerdere keren voorkomt. Daarom heb ik bijvoorbeeld voor gekozen om dit in een macro te verwerken zodat er in een keer duidelijk is dat er telkens het zelfde gebeurd. Daarnaast heb ik ervoor gekozen om het bestand stdafx.h bijna helemaal leeg te maken en de defenitie’s en macro’s te veprlaatsten naar de header-files waar van de bestanden waar deze in gebruikt worden. Dit zorgt voor meer overzicht in de code.

Ik ben uitgegaan van een principe ‘don’t need to know’ hier ga ik uit van de informatie die gebruikt wordt in de .cpp bestanden. Ik probeer hier zo veel mogelijk informatie in de bestanden te houden en alleen de informatie die gebruikt wordt door andere bestanden deel ik in de header-files. Hiermee voorkom ik dat er onnodig informatie wordt blootgegeven.