

### Inleiding

Abunch is een verzameling van bestanden voor Pure Data (Pd), een real-time grafische programmertaal voor audio. Pd is een 'open source' programma dat gratis te installeren en te gebruiken is. Het werkt onder Linux, Windows en Mac OS besturingssystemen. Abunch is gemaakt in de basisversie van Pd ('vanilla' genaamd). Dit betekent dat er geen 'externals' vereist zijn en dat het probleemloos te gebruiken is in om het even welke versie van Pd vanaf 0.40.

Abunch levert een verzameling afgewerkte objecten die geluid kunnen opnemen, manipuleren, analyseren en controleren. Het bestaat uit synthesizers, samplers, sequencers, oscilloscopen, etc. Pd beschikt over een grote collectie van ingebouwde objecten om algemene programmeertaken uit te voeren en meer in het bijzonder om klankbewerkingen te programmeren. Abunch gebruikt deze ingebouwde Pd objecten om meer dan 50 'higher level' objecten met een grafische interface op te bouwen. Pd heeft bijvoorbeeld objecten als [tabread][tabwrite] en [array] om waarden in tabellen op te slaan en te lezen. Het [timeline] object is een deel van de Abunch bibliotheek en is gebaseerd op deze ingebouwde Pd objecten [tabread][tabwrite] en [array]. Het voorziet in een handig object om een tabel te lezen, om de leesnelheid of de grootte te veranderen, om een tekst bestand naar deze tabel te schrijven, etc. Op deze manier biedt Abunch een aantal extra mogelijkheden bovenop de bestaande in Pd.

### Uniforme opbouw

De Abunch objecten hebben een aantal gemeenschappelijke structurele kenmerken om hun gebruik meer intuïtief en hun combinatiemogelijkheden krachtiger te maken. Eerst en vooral werd er een preset system ontwikkeld waardoor preset bestanden in .txt formaat bewaard of gelezen kunnen worden. De positie van interface objecten als schuivers, knoppen en de naam en locatie van geopende geluidsbestanden kunnen opgeslagen worden dankzij het object [presets]. Behalve dit globale preset systeem kunnen sommige Abunch objecten ook lokaal een preset lezen of opslaan. Op deze manier kunnen bijvoorbeeld patronen in een sequencer of synthesizer object opgeslagen worden.

Een ander voorbeeld van de uniforme opbouw in Abunch is het help systeem. De normale procedure in Pd -rechtsklikken op een object opent een help bestand- werd gecopieerd naar Abunch en help bestanden die elk object in deze library verklaren zijn beschikbaar.

Nagenoeg alle Abunch objecten kunnen eenvoudig gecontroleerd worden door andere objecten, grafische interface objecten als knoppen of schuivers kunnen dus geautomatiseerd worden. De controle gebeurt via de ingangen van de objecten en de controle waarden zijn genormaliseerd in een bereik van 0 tot 127. Dit vereenvoudigt ook het gebruik van hardware MIDI controllers om Abunch objecten te controleren.

Een ander kenmerk van Abunch is dat tijdsgebonden objecten gecontroleerd kunnen worden door een externe klok (dankzij het object [clock]). Verschillende van de hierboven vermelde [timeline] objecten kunnen gesynchroniseerd worden door één gemeenschappelijke [clock] object. Er is ook een [multi-clock] object voor meer gesofistikeerde polyritmische of polymetrische muziek. Het stuurt 5 kloksignalen uit die in een aanpasbare verhouding tot één hoofdklok staan.

Abunch beschikt over verscheidene objecten om klankfragmenten -opgeslagen in een tabel- af te spelen en op te nemen. Deze kunnen niet enkel gebruikt worden met vooraf opgenomen samples of geluidsbestanden maar ook met fragmenten die live -tijdens de uitvoering- met het [record-sample] object opgenomen worden. Dit laatste object slaat geluid op in het geheugen van de computer en niet op de harde schijf. Het kan gecombineerd worden met andere objecten als [play-sample] voor basisbewerkingen van samples of [grain-sample] voor granulaire synthese.

Bovendien kan één van de externe klokken gebruikt worden om het start en einde van de opname te synchroniseren. Op deze manier kan een 'loopstation' samengesteld worden met deze [clock], [record-sample] en [play-sample] objecten.

## Eerste kennismaking met elektronische muziek

De aanvankelijke bedoeling van Abunch was om een praktisch 'open source' programma te bouwen waarmee beginners meer zouden kunnen leren over de muzikale mogelijkheden in real time van een computer. Gebruikers kunnen luisteren naar basistechnieken en gereedschappen en kunnen experimenteren met computer geluiden door deze technieken en objecten met elkaar te combineren om zo hun eigen 'klankmachine' te bouwen.

Wat is een sampler, een synthesizer, een ADSR-enveloppe, een sequencer of een delay effect? Hoe klinken deze technieken? Abunch is een doos vol klankobjecten en voorbeelden waarin het gereedschap en de kennis aangereikt wordt om deze vragen te beantwoorden. De objecten voor (audio) analyse en algoritmische compositie en improvisatie die ook in Abunch te vinden zijn verruimen de doelgroep tot uitvoerders, componisten en improvisatoren geïnteresseerd in computer muziek. Af en toe verkiezen deze mensen waarschijnlijk kant-en-klare objecten omdat ze geen tijd hoeven te verspillen aan het aanmaken van een specifiek object en zo onmiddellijk deze objecten kunnen beginnen combineren.

Met deze uitgebreide mogelijkheden is de 'open source' en multi-platform bibliotheek Abunch heel geschikt voor pedagogische toepassingen. In het verleden werden hiermee door de auteur zowel theoretische als praktische cursussen gegeven. Omdat Pd een grafische programmeertaal is en Abunch gemaakt is in Pd, kunnen studenten of geïnteresseerden in elektronische muziek de interne opbouw van elk Abunch object bekijken en meer in detail leren over dit onderwerp. Dit is een groot pedagogisch voordeel omdat gemotiveerde studenten zo autonoom kunnen beslissen om zich in hun eigen tempo verder te verdiepen in deze materie.

Abunch bevat meer dan 35 voorbeeldbestanden die niet enkel de algemene werking van Abunch verduidelijken en de interne structuur van audio apparaten als een synthesizer of loopstation maar ook algemene muzikale 'weetjes' als 'hoe produceer ik een drone?', 'welk geluid is effectief als input voor een phaser effect',... Sommige van deze demonstraties zijn gegroeid uit voorbeelden in het klaslokaal.

### Meer mogelijkheden

Alhoewel Abunch bedoeld was voor beginners, werden er al snel meer geavanceerde mogelijkheden aan toegevoegd. Op het eerste zicht lijkt Abunch beperkt door zijn uniforme structuur die op gebruiksvriendelijkheid mikt maar deze kan vrij eenvoudig aangepast en omzeild worden. Het normale controle bereik van 0-127 kan veranderd worden in om het even welk bereik en deze waarden kunnen naar elke knop, schuiver of andere parameter gestuurd worden. Het bereik van deze grafische interface elementen en hun interne ontvangstnaam (waarnaar de controle data gestuurd worden) kan uitgeprint worden op het scherm voor één of alle Abunch objecten. Noodzakelijkerwijs is de grafische gebruikersinterface beperkt tot een aantal aanpasbare parameters maar in de uitgeprinte lijst zijn ook meer gedetailleerde parameters zichtbaar die onder deze interface verborgen zitten en die ook gecontroleerd kunnen worden.

Het [text-score] object is een ander voorbeeld van de meer gesofistikeerde mogelijkheden.

Waarden binnen elk bereik kunnen van hieruit verstuurd worden naar elke parameter met een ontvangst naam. De partituur is een tekst bestand dat bewerkt kan worden in een tekstrogramma of software die met tabellen werkt (zoals Calc of Excel). Op deze manier kunnen alle mogelijke partituren voor elektronische muziek gecreëerd worden.

Abunch kan tenslotte ook een opstap zijn om meer te leren over de krachtige Pd programmeertaal. Wie de basis van Pd beheerst kan bestaande Abunch objecten aanpassen aan de eigen behoeften of deze compleet verbouwen. Daarom zitten in de voorbeeldbestanden van Abunch ook demonstraties waarin de werking van ingebouwde Pd objecten en de combinatie van Abunch en Pd objecten verduidelijkt wordt.